Probabilistic Numerics: Bayesian Quadrature and Human-AI Collaboration

Masaki Adachi

St Catherine's College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Hilary 2025

Acknowledgements

Completing my DPhil at my dream place, working with wonderful people, has been a life-changing experience. I am deeply grateful to everyone who helped my research, social activities, and the funding that made this journey possible.

Thank you to my wife, Aki, for her unwavering support and understanding, even in the face of my sudden decision to return to school on the other side of the world and live apart for three years. I am also deeply grateful to my family—Toshifumi, Kaori, and Akane—and to all my friends, particularly to Yuki and Juliusz for their steadfast encouragement and support.

I am profoundly grateful to my advisers, Mike Osborne and Dave Howey, for their unwavering support, guidance, and mentorship throughout my DPhil journey. Working with you both has been an incredible privilege and joy, and it has truly been one of the greatest assets of my life. Mike evangelised me to embrace Bayesian thinking and consistently supported me in pursuing my interests, while Dave instilled in me the importance of simplicity and focusing on the problem itself rather than the method, as an engineer should. I look forward to continuing our collaboration and hope to offer my support in the future, just as you both supported me during my DPhil.

I have been blessed with many talented co-authors and friends, including Satoshi Hayakawa, Siu Lun (Alan) Chau, Wenjie Xu, Juliusz Ziomek, Csaba Tóth, Joachim Schaeffer, Yuxin Lin, Philipp Dechent, Martin Jørgensen, Vu Nyguen, Pierre Osselin, Xingchen Wan, Masahiro Fujisawa, Yannick Kuhn, Colin N. Jones, Birger Horstmann, and Harald Oberhauser, to name a few. Thanks to all my coauthors, lab colleagues, and friends. A special note of gratitude goes to Satoshi and Alan, whose inspiration and support made my DPhil journey an incredibly joyful experience.

I am deeply grateful to the Clarendon Scholarship, Oxford-Kobe Scholarship, Watanabe Scholarship, and British Council Japan Association Scholarship for their financial support, which enabled me to pursue my studies in Oxford and engage in academic activities.

I would like to express my sincere thanks to my colleagues at Toyota Motor Corporation for supporting my return to academia, especially Kunihiro Nobuhara, Hirohito Hirata, and Hiroaki Okuchi for their continuous encouragement and guidance.

I am profoundly thankful to my examiners, Prof. Philipp Hennig and Prof. Xiaowen Dong, for their thoughtful and insightful evaluation of my thesis. I also extend my gratitude to Prof. Jacob Foerster, Prof. Tom Rainforth, Prof. Stephen Roberts, and Prof. Charles Monroe for their valuable feedback during my transfer and confirmation stages. Special thanks to the anonymous reviewers of my papers; your constructive critiques significantly improved the quality of this thesis.

I am thankful to Dr. Krikamol Muandet for hosting and mentoring me during my two-month visit at CISPA in Saarbrücken, Germany. It was an unforgettable experience where I gained valuable insights into identifying impactful research topics, fostering effective collaborations, and exploring concepts in economics and imprecise probability. I would also like to extend my heartfelt thanks to Alan, Anurag, Kiet, Rattaya, Swathi, and Saptarshi for making my time in Saarbrücken both enjoyable and memorable.

I would like to extend my heartfelt gratitude to Dr. Emtiyaz Khan for hosting me during my three-month internship at RIKEN AIP in Tokyo, Japan. Special thanks to a member of Approximate Bayesian Inference Team-Thomas, Yohan, Christopher, Kego, Hugo, Sin-Han, Anita, Bai, Eiki, Marco, Hyungi, Rin, Alex, Adrian– and particularly for Yohan for being a mentor.

A special thank you to the BatteryDEV team—Joachim, Simon, Raymond, and Anoushka—for the unforgettable experience of organizing a hackathon event together in München, Germany.

Abstract

While machine learning for science has garnered significant attention, existing approaches often require well-defined and error-free expert inputs or reduce experts to mere data providers for the machine. However, at the forefront of scientific advancement, even human experts face uncertainties in their processes, necessitating a balanced collaboration between humans and algorithms. In this thesis, we view science as a human endeavor to update scientists' beliefs based on objective evidence, while algorithms represent encoded opinions.

This thesis investigates aligning algorithms with human beliefs and desiderata through Probabilistic Numerics, a principled framework for tasks such as black-box optimization, integration, and inference. It employs computational agents that address these tasks as machine learning problems using diverse policies. Within this framework, the alignment challenge translates into the efficient synchronization of computational and human agents at both the policy and modelling levels.

From a policy perspective, we focus on Bayesian quadrature as a unified solver. We conceptualize this solver as *Bayesian data compression*, which compresses datasets into smaller, representative points while propagating their (un)certainty in the distributional estimate. This perspective unifies diverse tasks and policies by framing them as differences in target (belief) distributions. This unification simplifies policy alignment while enhancing flexibility and adaptability across various tasks, including approximate Bayesian inference (Chapter 3), Bayesian optimization and active learning (Chapter 4), applications in battery control problems (Chapter 5), time-series forecasting, and Bayesian continual learning.

On the modelling side, we developed more efficient communication between computational agents and human users to align their beliefs. This involves aligning algorithms with humans while also helping humans align with algorithmic beliefs. The former requires algorithmic methods to elicit human beliefs faithfully, including their uncertainties, while the latter involves algorithmic explanations to convey the algorithm's current knowledge to humans. We addressed these challenges using economic approaches, particularly expected utility theory, encompassing prior elicitation and algorithmic explanation (Chapter 6), as well as connections to information-theoretic approaches (Chapter 7).

Publications

This thesis is based on the following original papers (1)-(5) [9–12, 239]. The list follows the order of appearance in the manuscript (Chapters 2, 3, 4, 5, and 6, respectively), and * denotes the equal-contribution.

(1) **Fast Bayesian inference with batch Bayesian quadrature via kernel recombination**
Masaki Adachi*, Satoshi Hayakawa*, Martin Jørgensen, Harald Oberhauser, and Michael A. Osborne. *Advances in Neural Information Processing Systems (NeurIPS)* 35, 16533–16547, 2022.

(2) **Adaptive batch sizes for active learning: A probabilistic numerics approach**
Masaki Adachi, Satoshi Hayakawa, Martin Jørgensen, Xingchen Wan, Vu Nguyen, Harald Oberhauser, and Michael A. Osborne. *International Conference on Artificial Intelligence and Statistics (AISTATS)* 238, 496-504, 2024.

(3) **Bayesian model selection of lithium-ion battery models via Bayesian quadrature**
Masaki Adachi, Yannick Kuhn, Birger Horstmann, Arnulf Latz, Michael A. Osborne, and David A. Howey. *IFAC-PapersOnLine* 56(2), 10521–10526, 2023.

(4) **Looping in the human: collaborative and explainable Bayesian optimization**
Masaki Adachi, Brady Planden, David A. Howey, Michael A. Osborne, Sebastian Orbell, Natalia Ares, Krikamol Muandet, and Siu Lun Chau. *International Conference on Artificial Intelligence and Statistics (AISTATS)* 238, 505-513, 2024.

(5) **Principled Bayesian optimisation in collaboration with human experts**
Wenjie Xu*, **Masaki Adachi***, Colin N. Jones, and Michael A. Osborne. *Advances in Neural Information Processing Systems (NeurIPS)* **Spotlight** 37, 104091-104137, 2024.

For all the above publications (1)-(5) [5, 8, 9, 12, 239] I am the primary author. For the equal-contributed papers (1),(5) [9, 239], I contributed the core concepts, algorithm, coding, experiments, and writing while Hayakawa and Xu contributed the proofs, core concepts, algorithm, and writing equally. For (1) [9], Osborne contributed to the idea of online learning, supervision, and editing, Oberhauser and Jørgensen contributed to supervision and editing. For (2) [10], I contributed the core concepts, algorithm, coding, experiments, writing, and editing. Hayakawa contributed the proof and algorithm, Jørgensen, Wan, Ngyuen, Oberhauser contributed the editing, Osborne contributed to supervision and editing. For (3) [11], I contributed the core concepts, algorithm, coding, experiments, writing, and editing. Kuhn, Horstmann, and Latz contributed the editing, and Osborne and Howey contributed the supervision and editing. For (4) [12], Chau contributed to the idea of the explanation of acquisition function, the pairwise comparison, supervision and editing, Planden contributed the experiment as human experts and editing, Orbell and Ares contributed editing, Howey, Osborne, Muandet contributed supervision and editing. For (5) [239], Jones and Osborne contributed the supervision and editing.

Although the main content of the following papers is not included in this thesis, I have had the privilege of contributing to several other projects. Additionally, I have several papers currently under review, all of which are listed below. In the conclusion section of Chapter 8, we briefly discuss their connections and broader impact rather than presenting their full content.

First-Authored Papers that Are Not Included

(6) **Fixing the Pitfalls of Probabilistic Time-Series Forecasting Evaluation by Kernel Quadrature**
Masaki Adachi*, Masahiro Fujisawa*, Michael A. Osborne, *Accepted for International Conference on Probabilistic Numerics (ProbNum)*, 2025

(7) **A quadrature approach for general-purpose batch Bayesian optimization via probabilistic lifting**
Masaki Adachi, Satoshi Hayakawa, Martin Jørgensen, Saad Hamid, Harald Oberhauser, and Michael A. Osborne. *Under review for Journal of Machine Learning Research (JMLR).*

(8) **Bayesian optimization for building social-influence-free consensus**
Masaki Adachi, Siu Lun Chau, Wenjie Xu, Anurag Singh, Michael A. Osborne, and Krikamol Muandet. *Under review for Advances in Neural Information Processing Systems (NeurIPS) 2025.*

(9) **Scalable Valuation of Human Feedback through Provably Robust Model Alignment**
Masahiro Fujisawa*, **Masaki Adachi***, and Michael A. Osborne. *Under review for Advances in Neural Information Processing Systems (NeurIPS) 2025*

(10) **High-dimensional discrete Bayesian optimization with self-supervised representation learning for data-efficient materials exploration**
Masaki Adachi. *NeurIPS AI for Science Workshop*, 2021

(11) **Mixture-of-Experts Ensemble with Hierarchical Deep Metric Learning for Spectroscopic Identification**
Masaki Adachi. *NeurIPS Machine Learning and the Physical Sciences Workshop*, 2021

(12) **SOBER: Highly parallel Bayesian optimization and Bayesian quadrature over discrete and mixed spaces**
Masaki Adachi, Satoshi Hayakawa, Saad Hamid, Martin Jørgensen, Harald Oberhauser, Michael A. Osborne, *ICML 2023 Workshop: Sampling and Optimization in Discrete Space*, 2023

For (6)-(12) [3–8, 76], I am the primary author contributing the core concepts, algorithm, coding, experiments, writing, and editing.

Co-Authored Published Papers and Submissions Under Review

(13) **Bayesian optimisation with unknown hyperparameters: regret bounds logarithmically closer to optimal**
Juliusz Ziomek, **Masaki Adachi**, and Michael A. Osborne. *Advances in Neural Information Processing Systems (NeurIPS)*, 37, 86346–86374, 2024

(14) **Learning to Forget: Bayesian Time Series Forecasting using Recurrent Sparse Spectrum Signature Gaussian Processes**
Csaba Tóth, **Masaki Adachi**, Michael A. Osborne, and Harald Oberhauser. *International Conference on Artificial Intelligence and Statistics (AISTATS)* 258, 4654-4662, 2025.

(15) **Time-Varying Gaussian Process Bandits with Unknown Prior**
Juliusz Ziomek, **Masaki Adachi**, and Michael A. Osborne. *International Conference on Artificial Intelligence and Statistics (AISTATS)* 258, 4294-4302, 2025.

(16) **Natural Evolutionary Search meets Probabilistic Numerics**
Pierre Osselin, **Masaki Adachi**, Xiaowen Dong, and Michael A. Osborne. *Accepted for International Conference on Probabilistic Numerics (ProbNum) 2025*

(17) **Machine learning benchmarks for the classification of equivalent circuit models from electrochemical impedance spectra**
Joachim Schaeffer, Paul Gasper, Esteban Garcia-Tamayo, Raymond Gasper, **Masaki Adachi**, Juan Pablo Gaviria-Cardona, Simon Montoya-Bedoya, Anoushka Bhutani, Andrew Schiek, Rhys Goodall, Rolf Findeisen, Richard D. Braatz, and Simon Engelke. *Journal of The Electrochemical Society* 170(6), 060512, 2023.

(18) **BASiL: Fast Broad-band Line-rich Spectralcube Fitting and Image Visualization via Bayesian Quadrature**
Yuxin Lin, **Masaki Adachi**, Silvia Spezzano, Gordian Edenhofer, Vincent Eberle, Michael A. Osborne, Paola Caselli, *Accepted for Astronomy & Astrophysics*

(19) **Demonstrating Linked Battery Data To Accelerate Knowledge Flow in Battery Science**
Philipp Dechent, Elias Barbers, Simon Clark, Susanne Lehner, Brady Planden, **Masaki Adachi**, David Howey, and Sabine Paarmann. *Under review for Batteries & Supercaps*

(20) **A Primer on Bayesian Parameter Estimation and Model Selection for Battery Simulators**
Yannick Kuhn, **Masaki Adachi**, Michael A. Osborne, David A. Howey, Arnulf Latz, Birger Horstmann. *In submission to Journal of Electrochemical Society*

For (12),(14),(15) [217, 246, 247], I contributed the writing, coding, experiments, and editing. For (13) [193], I contributed the dataset preparation, coding, supervision, writing, and editing. For (16) [175], I contributed writing and editing. For (17) [61], I contributed editing. For (18),(19) [138, 148], I contributed the core concept, algorithm, writing, coding, and editing.

Contents

1 Introduction

This thesis is presented in an integrated format, consisting of 8 chapters that together provide a cohesive narrative. The Chapter 1-2 serves as the introduction and background for the thesis's topic, and the following five intermediate chapters are based on the papers I have published during my DPhil studies. Each chapter is self-contained, offering a comprehensive introduction, literature review, and its unique contribution, contextualizing the research within its specific topic. The contributions of this thesis are divided into two main parts, corresponding to the two major challenges faced by PN for scientific experts, as addressed in Chapters 3–7:

1. **Bayesian quadrature**: A unified solver for Probabilistic Numerics
2. **Human-AI collaboration**: knowledge elicitation and integration

To tackle these challenges, we proposed two principled methodologies: *Bayesian quadrature* and *preference learning*. We will provide a detailed explanation of our contributions alongside an outline of the main body of the thesis (Chapters 3–7) in the following Chapter 2.

Compression is comprehension [242].

— Gregory Chaitin, Mathematician and Computer Scientist

Basically every problem can be formulated as compression by trying to compress (task, solution) pairs. [208]

— Christian Szegedy, Computer Scientist

2 Background

This thesis aims to enhance the capabilities of the Probabilistic Numerics framework for scientific experts through Bayesian quadrature and preference learning. In this chapter, we briefly review these three key concepts.

2.1 Probabilistic Numerics

Probabilistic Numerics (PN; Cockayne et al. [52] and Hennig et al. [101]) is a field that seeks to augment classical numerical algorithms—such as optimization, quadrature, differential equation solvers, and linear algebra—with probabilistic counterparts. PN introduce a novel perspective by framing numerical algorithms as machine learning tasks. Specifically, it aims to infer black-box functions for optimization, integration, or simulation using queried data, often within a Bayesian framework.

At the heart of PN is the use of Bayesian surrogate models, which not only predict the target function but also quantify predictive uncertainty. This approach recasts numerical tasks as sequential decision-making problems under uncertainty, treating computational agents as decision-makers. By doing so, PN mimic the way humans solve numerical problems carefully and efficiently when operating

under resource constraints. This paradigm offers several unique advantages over classical numerical methods:

1. **Adaptivity**: PN are well-suited for handling black-box functions in tasks such as optimization, integration, and inference. They adapt to the problem by leveraging information from queried data.
2. **Flexibility**: PN can tailor their models to specific contexts by adjusting computational agents or modifying policies, extending beyond merely processing data.
3. **Reliability**: By explicitly modeling computational uncertainty, PN provide insights into their limitations. This feature is especially valuable when the outcome of a numerical task influences subsequent decision-making.

PN approaches have been successfully applied across a wide range of impactful fields, particularly in scientific and engineering domains. For example:

1. **Black-box optimization**, also known as Bayesian optimization (Garnett [81], Mockus [162], and Osborne et al. [174]), Gaussian process bandit (Srinivas et al. [204]), kernelized bandit (Chowdhury et al. [50]) has been widely used in science and engineering such as materials discovery (Adachi [3] and Gómez-Bombarelli et al. [84]), product design (Adachi et al. [12] and Grosnit et al. [90]).
2. **Black-box integration**, also known as Bayesian quadrature (Diaconis [62], Larkin [144], and Osborne et al. [173]), Bayes-Hermite quadrature (O'Hagan [170]), Bayesian Monte Carlo (Acerbi [1, 2] and Rasmussen et al. [186]) has been applied to control (Adachi et al. [11] and Prüher et al. [183]) and finance (Lehdili et al. [147]).
3. **Probabilistic linear algebra** methods have been employed for scalable Gaussian Process models (Wenger et al. [228] and Wu et al. [237]).
4. **Probabilistic ODE/PDE solvers**, also known as Bayesian filtering and smoothing (Särkkä et al. [192]) enable efficient and scalable computation (Bosch et al. [29] and Krämer et al. [135]).

The viewpoint of PN naturally blends well with tasks involving human experts. Scientific tasks such as experiments or design are essentially decision-making under uncertainty, with human experts acting as real-world agents who make decisions

based on their policies. Most work focuses on replacing the human agent's role with computational agents, often praised for 'taking humans out of the loop.' However, computational agents' policies do not necessarily match those of human experts. Human experts might use knowledge or conditions that computational agents cannot access, leading experts to choose option A instead of the computational agent's suggestion B. In real scientific applications, human experts are the final decision-makers, and aligning with them is essential. This alignment has been under-explored. We classify the alignment challenge into two main areas: creating more unified and generic algorithms and improving knowledge elicitation.

The first challenge is to establish a cognitively simple method to meet the diverse needs of human experts. Bayesian methods require careful modeling to match specific conditions and goals, leading to a proliferation of acquisition functions and models in the literature. Selecting the appropriate algorithms can be challenging for domain experts who are not familiar with statistics. A simpler and more unified approach to reflect their needs is essential.

Another significant challenge is integrating human expertise into algorithms. Experts gain valuable knowledge through experience, which should accelerate the decision-making process. However, at the forefront of science, experts are also engaged in trial and error, and their knowledge can be incomplete. Treating their input as error-free and well-defined can be counterproductive. Additionally, how we query experts is crucial for effectively eliciting their knowledge. The interaction method should be cognitively simple to elicit more accurate information. This becomes more complicated when involving multiple experts. While group intelligence should, in theory, be more reliable than individual input, consensus is often influenced by interpersonal dynamics among experts. Debiasing these effects to uncover the true underlying knowledge is another challenge in multi-expert situations.

In the next section, we introduce the concept of Bayesian data compression, a promising approach that aims to unify these diverse PN tasks under a single cohesive framework.

2.2 Bayesian Quadrature as Data Compression

In the Bayesian compression setting, we are particularly interested in cases where the target function is unknown, and our access to its information is limited. In this context, we review how Bayesian quadrature can flexibly compress both known and unknown information. To explain the compression task systematically, we proceed step by step, starting with the simplest i.i.d. compression, moving to kernel quadrature, and finally to Bayesian quadrature.

2.2.1 i.i.d. compression (Monte Carlo integration)

Let μ be a Borel probability measure on the domain $\mathcal{X}$, and let $x \sim \mu(x)$ be an i.i.d. sample drawn from μ. Consider an integrable function $f : \mathcal{X} \to \mathbb{R}$. The i.i.d. compression task is defined as minimizing the following error:

$$\mathrm{err}(N) := \left| \int_{\mathcal{X}} f(x) \mathrm{d}\mu(x) - \sum_{i=1}^{N} f(x_i) \right|, \tag{2.1}$$

where $x_i \sim \mu(x)$ are i.i.d. samples drawn from the measure μ. The i.i.d. compression task can thus be understood as minimizing the error with respect to its expectation. In other words, i.i.d. samples represent a compressed form of the data distribution μ. The convergence rate of the compression error is well-known to be $\mathrm{err}(N) = \mathcal{O}(1/\sqrt{N})$ (Hennig et al. [see 101, Lemma 9.2]), which is relatively slow. In real-world applications, we often require a better compression rate with fewer samples, N.

To address this, we relax the following two conditions: (i) instead of relying on random samples, we can carefully select $x_i \in \mathcal{X}$, and (ii) we can assign additional information in the form of weights w_i to each sample x_i. In this setting, x_i and w_i are often referred to as quadrature nodes and quadrature weights, respectively. Notably, when the target measure μ is a discrete probability distribution, the following result holds:

Theorem 1 (**Tchakaloff's theorem**). *Let $x_1, \cdots, x_n$ be n quadrature nodes, $w_1, \cdots, w_n \geq 0$ be (positive) quadrature weights such that $\sum_{i=1}^n w_i = 1$, $\{x_j\}_{j=1}^N = \mu(x)$ be a discrete measure with $N > n$, $\boldsymbol{\varphi} := (\varphi_1, \cdots, \varphi_M)^\top$ be a M-dimensional, integrable, and vector-valued function with $M \leq N + 1$, and*

$$\int_{\mathcal{X}} \boldsymbol{\varphi}(x) d\mu(x) = \sum_{i=1}^{n} w_i \boldsymbol{\varphi}(x_i). \tag{2.2}$$

This is the so-called Tchakaloff's theorem (Tchakaloff [209]). The quadrature nodes and weights in this theorem are often referred to as *cubature* (Stroud [206]). The proof of Tchakaloff's theorem is essentially based on the classical Carathéodory's theorem (Carathéodory [39]). This theorem states that a larger cardinality of the discrete measure N can be compressed into n quadrature nodes and weights without incurring any additional error. This *lossless compression* implies that by carefully selecting quadrature nodes and weights, we can achieve a more sample-efficient compression than i.i.d. compression. The flexibility and efficiency of this approach make it an appealing alternative for applications requiring a higher compression rate.

2.2.2 Kernel quadrature

To connect the classical cubature approach to modern machine learning, we consider the case where the function f belongs to a reproducing kernel Hilbert space (RKHS), denoted as $f \in \mathcal{H}_k$. Specifically, we consider a function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ that satisfies the following properties: it is symmetric ($k(x, x') = k(x', x)$ for all $x, x' \in \mathcal{X}$) and positive definite (for any $t \in \mathbb{N}$, $\{a_i\}_{i=1}^t \subset \mathbb{R}$, $\{x_i\}_{i=1}^t \subset \mathcal{X}$, it holds that $\sum_{i,j=1}^t a_i a_j k(x_i, x_j) \geq 0$). Functions meeting these criteria are referred to as *kernels*.

Each kernel k has an associated Hilbert space, the RKHS $\mathcal{H}_k$, with the following two properties: (A) $\mathrm{span}\{k(\cdot, x) \mid x \in \mathcal{X}\}$ is a dense subspace of $\mathcal{H}_k$, and (B) $\langle f, k(\cdot, x)\rangle_{\mathcal{H}_k} = f(x)$ holds for each $x \in \mathcal{X}$ and $f \in \mathcal{H}_k$, where $\langle \cdot, \cdot \rangle_{\mathcal{H}_k}$ denotes the inner product in $\mathcal{H}_k$. According to the Moore–Aronszajn theorem (Aronszajn [15]), there is a one-to-one correspondence between the kernel k and its RKHS $\mathcal{H}_k$. See Berlinet et al. [25] for a formal introduction to RKHS.

Using kernels, we define *kernel quadrature*. Consider an n-point quadrature $Q_n(x) := \sum_{i=1}^n w_i \delta_i(x)$, where δ_i is the Dirac measure located at x_i, consisting of points $\{x_i\}_{i=1}^n \subset \mathcal{X}$ and weights $w_{i i=1}^n \subset \mathbb{R}$. This can be interpreted as a weighted discrete distribution: $\int_{\mathcal{X}} f(x) \mathrm{d}Q_n(x) = \sum_{i=1}^n w_i f(x_i)$. A good quadrature Q_n minimizes the *worst-case error (wce)*:

$$\mathrm{wce}(Q_n; \mathcal{H}_k, \mu) := \sup_{\|f\|_{\mathcal{H}_k} \leq 1} \left| \int_{\mathcal{X}} f(x) \mathrm{d}\mu(x) - \int_{\mathcal{X}} f(x) \mathrm{d}Q_n(x) \right|. \tag{2.3}$$

Eq. (2.3) can be viewed as a natural extension of the simple error in Eq. (2.1): the cubature approach compresses the distribution μ into a weighted sum of quadrature nodes, and the integrable function f now belongs to the RKHS $\mathcal{H}_k$, where the worst-case error corresponds to its supremum.

The worst-case error in Eq. (2.3) is equivalent to the *maximum mean discrepancy* (MMD; Gretton et al. [88, 89] and Muandet et al. [165]), also known as the integral probability metric (Müller [166]). Given two Borel probability measures μ and ν, the MMD is defined as:

$$\begin{aligned} \mathrm{MMD}(\mu, \nu)_k &:= \sup_{\|f\|_{\mathcal{H}_k} \leq 1} \left| \int_{\mathcal{X}} f(x) \mathrm{d}\mu(x) - \int_{\mathcal{X}} f(x) \mathrm{d}\nu(x) \right|. \\ \mathrm{MMD}(\mu, \nu)_k^2 &= \iint_{\mathcal{X} \times \mathcal{X}} k(x, y) \mathrm{d}\mu(x) \mathrm{d}\mu(y) - 2 \iint_{\mathcal{X} \times \mathcal{X}} k(x, y) \mathrm{d}\mu(x) \mathrm{d}\nu(y) \\ &\quad + \iint_{\mathcal{X} \times \mathcal{X}} k(x, y) \mathrm{d}\nu(x) \mathrm{d}\nu(y). \end{aligned} \tag{2.4}$$

This squared formula is commonly used to compute the MMD. When $\nu(x) = Q_n(x)$, the MMD and the worst-case error become equivalent. Thus, kernel quadrature can be understood as finding the discrete distribution Q_n that best approximates the original measure μ in terms of the MMD distance.

Kernel quadrature is also referred to as MMD coresets (Williamson et al. [232]) or MMD quantization (Graf et al. [87] and Teymur et al. [212]). Intuitively, minimizing the MMD naturally leads to sparse quadrature nodes, which is closely related to notions such as sparse sampling, diversified sampling, and repulsion sampling. In particular, determinantal point processes (DPP; Kulesza et al. [139]) are known to have a strong connection to kernel quadrature (Belhadji [21] and Belhadji et al. [23]).

2.2.3 Bayesian quadrature

We further extend the kernel quadrature framework to the Bayesian setting. While kernel quadrature assumes that the true kernel function f is known, the Bayesian approach considers the more general case where the true function f is unknown. This setting is often referred to as *black-box integration*. Under the black-box assumption, we cannot directly access the function's analytical form or gradient information; instead, we can query noisy, pointwise observations. Using these observations, we construct a *surrogate model* to infer the true function f. A Gaussian process (GP; Rasmussen et al. [187] and Stein [205]) is a widely used choice for the functional prior in Bayesian quadrature.

In the Probabilistic Numerics book [101, 171], the Monte Carlo approach is noted as a Frequentist method, as opposed to the common belief that Markov Chain Monte Carlo (MCMC; Geyer [83]) is a Bayesian tool. Indeed, Monte Carlo methods do not impose a prior over the function space, requiring only that the integrand is (Lebesgue) integrable. In contrast, a GP prior makes a more restrictive assumption by assigning non-zero probability only to continuous functions. While this constraint aligns with most black-box functions of interest, the choice of kernel provides additional flexibility, enabling the selection of specific function properties. This flexibility offers an opportunity to incorporate domain-specific context into solving PN tasks.

Let $(\Omega, \mathcal{F}, \mathbb{P})$ represent a probability space. A GP is a stochastic process $g : \mathcal{X} \times \Omega \to \mathbb{R}$ characterized by its mean function $m_0 : \mathcal{X} \to \mathbb{R}$, defined as $m_0(x) = \mathbb{E}[g(x, \cdot)]$, and its covariance function, defined using a kernel: $k(x, x') = \mathbb{E}[(g(x, \cdot) - m_0(x))(g(x', \cdot) - m_0(x'))]$. A GP induces a probability measure over functions and allows conditioning on observed data in closed-form under conjugate likelihood assumptions. In the regression setting, we assume the observations follow the model $y = f(x) + \epsilon$, where f is the true function to estimate, $\epsilon \sim \mathcal{N}(0, \sigma^2)$ is i.i.d. Gaussian noise, and σ^2 is the noise variance. Given a dataset $\mathcal{D}_t = \{x_i, y_i\}_{i=1}^t := (\mathbf{X}_t, \mathbf{Y}_t)$ and the covariance matrix $\mathbf{K}_{X_t X_t} = (k(x_i, x'_j))_{1 \le i,j \le t} \in \mathbb{R}^{t \times t}$,

the GP posterior predictive distribution is given by $f_t \mid \mathcal{D}_t \sim \mathcal{GP}(m_t, C_t)$, where:

$$m_t(x) = m(x) + k(x, \mathbf{X}_t)(\mathbf{K}_{X_t X_t} + \lambda \mathbf{I}_{t\times t})^{-1}(\mathbf{Y}_t - m_0(\mathbf{X}_t)),$$
$$C_t(x, x') = k(x, x') - k(x, \mathbf{X}_t)(\mathbf{K}_{X_t X_t} + \lambda \mathbf{I}_{t\times t})^{-1}k(\mathbf{X}_t, x').$$

Here, $m_t(\cdot)$ and $C_t(\cdot, \cdot)$ are the mean and covariance functions of the posterior predictive distribution, $\mathbf{I}_{t\times t}$ is the identity matrix, and $\lambda = \lambda_0 + \sigma^2 \geq 0$ is a regularization constant that ensures the invertibility of $(\mathbf{K}_{X_t X_t} + \lambda \mathbf{I}_{t\times t})$. While λ is often set equal to σ^2 in Bayesian settings, i.e., $\lambda_0 = 0$, we treat λ_0 as a positive constant to ensure invertibility even in noiseless cases.

Assuming $f \in \mathcal{H}_k$, the RKHS induced by the kernel, a GP is a proper prior distribution for the true function f. Chowdhury et al. [50] showed that the worst-case error of the GP estimate is bounded by the kernel-dependent maximum information gain γ_t:

Theorem 2 (**Information-theoretic bound**). *Let $f \in \mathcal{H}_k$, $\|f\|_{\mathcal{H}_k} \leq B$, $\mathcal{X}$ be compact and non-empty. For any $\delta \in (0, 1)$, with probability at least $1 - \delta/2$, the following holds for all $x \in \mathcal{X}$ and $1 \leq t \leq T$, $T \in \mathbb{N}$,*

$$|\mu_t(x) - f(x)| \leq \beta_t \sqrt{C_t(x, x)}, \qquad \beta_t := \left(B + \sigma\sqrt{2\left(\gamma_t + 1 + \ln(2/\delta)\right)}\right),$$

where $\gamma_t := \max_{X\subset\mathcal{X};|X|=t} \frac{1}{2}\log|\boldsymbol{I}_{t\times t} + \lambda^{-1}\boldsymbol{K}_{X_t X_t}|$ is the maximum information gain, and $\gamma_0 = 0$.

Theorem 2 guarantees that with probability at least $1 - \delta/2$, the true function $f(x)$ lies within the predictive covariance $C_t(x, x)$. Additionally, β_t and $C_t(x, x)$ are known to decrease monotonically with t, or *submodularity* (Nemhauser et al. [168] and Seeger et al. [195]), ensuring that the GP surrogate model asymptotically converges the true function $f(x)$ within the inherent limit of observation noise σ. This makes the GP-based approach to black-box integration principled.

Furthermore, the GP induces the jointly Gaussian integral estimate $\hat{Z} :=$

$\int_{\mathcal{X}} f_t(x)\mathrm{d}\mu(x)$ with a univariate Gaussian distribution:

$$\mathbb{E}_{f_t\sim\mathcal{GP}(m_t,C_t)}[\hat{Z}] = \int_{\mathcal{X}} m_t(x)\mathrm{d}\mu(x) = z_0(\mathbf{X}_t)^\top(\mathbf{K}_{X_tX_t} + \sigma^2\mathbf{I}_{t\times t})^{-1}(\mathbf{Y}_t - m_0(\mathbf{X}_t)) \tag{2.5a}$$

$$\mathbb{V}_{f_t\sim\mathcal{GP}(m_t,C_t)}[\hat{Z}] = \iint_{\mathcal{X}\times\mathcal{X}} C_t(x,x')\mathrm{d}\mu(x)\mathrm{d}\mu(x') = z_0' - z_0(\mathbf{X}_t)^\top(\mathbf{K}_{X_tX_t} + \sigma^2\mathbf{I}_{t\times t})^{-1}z_0(\mathbf{X}_t), \tag{2.5b}$$

where $z_0(x) := \int_{\mathcal{X}} k(x',x)\mathrm{d}\mu(x')$ and $z_0' := \iint_{\mathcal{X}\times\mathcal{X}} k(x,x')\mathrm{d}\mu(x)\mathrm{d}\mu(x')$ represent the kernel mean and variance, respectively. For certain kernel-measure combinations, these values have closed-form solutions. For example, the combination of an RBF kernel and Gaussian measure is a popular choice in Bayesian quadrature (O'Hagan [170] and Rasmussen et al. [186]) and is widely used. We name such kernel-measure pairs as 'conjugate', and refer to Briol et al. [see 33, Table 1] for a comprehensive list of conjugate pairs.

2.2.4 Connecting it all together

Huszár et al. [111] and Ritter [189] demonstrated that the worst-case error, MMD, and integral variance are *equivalent*. The Bayesian quadrature (BQ) expectation in Eq. (2.5a) can be expressed as a weighted sum: $z_0(\mathbf{X}_t)\boldsymbol{K}_t^{-1}(\boldsymbol{Y}_t - m_0(\mathbf{X}_t)) = \sum_{i=1}^n w_{t,i}(y_i - m_0(X_i))$, where $w_{\mathrm{t,\,j}} := \sum_{i=1}^n z_0(\mathbf{X}_i)\boldsymbol{K}_{i,j}^{-1}$ and $\boldsymbol{K}_t^{-1} := (\mathbf{K}_{X_tX_t} + \lambda\mathbf{I}_{t\times t})^{-1}$. These weights can be interpreted as forming a discrete distribution: $\nu_t := \sum_{i=1}^n w_{t,i}\delta_{x_i}$. This interpretation allows the integral variance estimation to be expressed in terms of the MMD as follows:

$$\underbrace{\mathbb{V}_{f_t\sim\mathcal{GP}(m_t,C_t)}[\hat{Z}]}_{\text{integral variance}} = \underbrace{\mathrm{MMD}^2_{\mathcal{H}_{C_t}}(\nu_t,\mu)}_{\text{MMD distance}} = \underbrace{\inf_{\mathbf{w}_t\subset\mathbb{R}^n}\mathrm{wce}(\nu_t;\mathcal{H}_{C_t},\mu)^2}_{\text{minimum possible worst-case error}} \tag{2.6}$$

for a fixed set of nodes X, where $\mathcal{H}_{C_t}$ denotes the RKHS associated with the GP predictive covariance kernel $C_t(\cdot,\cdot)$. This identity has been known in the literature (e.g., Briol et al. [32] and Xi et al. [238]). Kanagawa et al. [118] showed that the predictive mean function m_t lies in the RKHS of the GP predictive covariance kernel $C_t(\cdot,\cdot)$[1]. Consequently, $\mathcal{H}_{C_t}$ can be viewed as the posterior RKHS, in contrast to

[1] A sample path drawn from a GP lies within the power of the RKHS $\mathcal{H}_{C_t'}$, rather than $\mathcal{H}_{C_t}$. Since $\mathcal{H}_{C_t} \subset \mathcal{H}_{C_t'}$, sample paths lie in $\mathcal{H}_{C_t}$ *almost surely*. For an explanation of this subtle distinction, we refer to Kanagawa et al. [see 118, Section 4].

$\mathcal{H}_k$, which represents the prior RKHS. Following Bayes rule, the posterior RKHS $\mathcal{H}_{C_t}$ monotonically contracts as the number of observed data points t grows.

Interestingly, the worst-case error of kernel quadrature is equivalent to the *average-case* error of Bayesian quadrature. The infimum of worst-case error shows the Bayesian quadrature gives an optimally weighted quadrature rule for a given set of points $\mathbf{X}_t$.

2.3 Bayesian Quadrature for Probabilistic Numerics

We propose a new principle for solving PN tasks based on Bayesian quadrature. The key idea is to view all PN tasks as sample-efficient approximations of certain expectations of desired quantities. Once a PN task is translated into the form of an expectation, the sample selection task can be reframed as minimizing the variance of the expectation estimate.

For example, black-box inference involves a sample-efficient approximation of the likelihood over the prior belief $\mathbb{P}(x)$ (i.e., evidence). Black-box optimization involves a sample-efficient approximation of the global maximum over the belief of global maxima $\mathbb{P}(x^\star \mid \mathbf{D}_t)$. Active learning involves a sample-efficient approximation of the black-box function over the data distribution $\mathbb{P}(\mathcal{S})$. As such, PN tasks can be interpreted as differing primarily in the choice of measure μ, allowing all tasks to be solved using a unified Bayesian quadrature approach. In the following sections, we elaborate on this interpretation to facilitate a smooth transition to Chapters 3-5, including some unpublished theoretical results with proofs.

2.3.1 Active sampling for the quadrature node

While the equivalence between kernel quadrature and Bayesian quadrature bridges different fields, it does not inherently offer a method for selecting quadrature nodes x_t to minimize variance, distance, or error in a sample-efficient manner. Osborne et al. [173] introduced the concept of selecting quadrature nodes x_t as an online learning task, where query points are chosen sequentially through decision-making.

Building on this, Gessner et al. [82] proposed using integral variance reduction (IVR) as the acquisition function to guide the selection of the next query point x_t:

$$\Delta C_t(x_{t+1}, \mu) := \underbrace{\iint_{\mathcal{X}\times\mathcal{X}} C_t(x, x')\mathrm{d}\mu(x)\mathrm{d}\mu(x')}_{\text{old integral variance}} - \underbrace{\iint_{\mathcal{X}\times\mathcal{X}} C_{t+1}(x, x' \mid x_{t+1})\mathrm{d}\mu(x)\mathrm{d}\mu(x')}_{\text{new integral variance}},$$

The IVR can be expressed independently of C_{t+1}, allowing for a one-step-ahead optimal decision without requiring the unseen new data y_{t+1}.

Proposition 1. *Let $f_{t+1} \sim \mathcal{GP}(m_{t+1}, C_{t+1})$ be the updated GP from the previous iteration f_t, with the updated dataset $\boldsymbol{D}_{t+1} = \boldsymbol{D}_t \cup (x_{t+1}, y_{t+1})$, where $y_{t+1}$is a noisy output observation at x_{t+1}. For conjugate kernel-measure combinations with a closed-form $z_0(x)$, the acquisition function also simplifies to the following closed-form:*

$$\Delta C_t(x_{t+1}, \mu) = z_t(x_{t+1})(C_t(x_{t+1}, x_{t+1}) + \lambda)^{-1} z_t(x_{t+1}), \tag{2.7}$$

where $z_t(x_t) = \int_{\mathcal{X}} C(x, x_t)\, d\mu(x)$ is the posterior kernel mean embedding of μ.

Proof. Under the GP prior $f_t \sim \mathcal{GP}(m_t, C_t)$, the joint distribution of the random variables $\mathcal{N}(f_t(x), y_{t+1})$ is a joint Gaussian. Hence,

$$\begin{bmatrix} f_t(x) \\ y_{t+1} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m_t(x) \\ m_t(x_{t+1}) \end{bmatrix}, \begin{bmatrix} C_t(x, x) & C_t(x, x_{t+1}) \\ C_t(x_{t+1}, x) & C_t(x_{t+1}, x_{t+1}) + \lambda \end{bmatrix} \right), \tag{2.8}$$

Therefore, conditional covariance can be derived from the known identity of conditional Gaussian distribution:

$$\mathrm{Cov}[f_t(x) \mid y_{t+1}] = C_t(x, x) - C_t(x, x_t)(C_t(x_t, x_t) + \lambda)^{-1} C_t(x_t, x), \tag{2.9}$$

The posterior GP covariance, C_{t+1} needs the covariance for $(f_t(x), f_t(x'), y_{t+1})$. Still, by extending the same argument to this joint distribution, one obtains exactly the same conditional covariance structure by just treating $\boldsymbol{u} = (f_t(x), f_t(x'))$ in the block partition. Therefore, we have

$$C_{t+1}(x, x' \mid x_{t+1}) = C_t(x, x') - C_t(x, x_t)(C_t(x_t, x_t) + \lambda)^{-1} C_t(x_t, x'), \tag{2.10}$$

and the covariance reduction and its integral become

$$\begin{aligned} \Delta C_t(x_t, x_t) :=& C_t(x, x') - C_{t+1}(x, x' \mid x_{t+1}), \\ =& C_t(x, x_t)(C_t(x_t, x_t) + \lambda)^{-1} C_t(x_t, x'), \end{aligned} \tag{2.11}$$

$$
\begin{aligned}
&\Delta C_t(x_{t+1}, \mu) \\
=&\iint_{\mathcal{X}\times\mathcal{X}} C_t(x, x') - C_{t+1}(x, x' \mid x_{t+1}) \mathrm{d}\mu(x) \mathrm{d}\mu(x') \\
=&\iint_{\mathcal{X}\times\mathcal{X}} C_t(x, x_{t+1})(C_t(x_{t+1}, x_{t+1}) + \lambda)^{-1} C_t(x_{t+1}, x') \mathrm{d}\mu(x) \mathrm{d}\mu(x'), \\
=&\int_{\mathcal{X}} C_t(x, x_{t+1}) d\mu(x) (C_t(x_{t+1}, x_{t+1}) + \lambda)^{-1} \int_{\mathcal{X}} C_t(x_{t+1}, x') \mathrm{d}\mu(x'), \\
=&z_t(x_{t+1})(C_t(x_{t+1}, x_{t+1}) + \lambda)^{-1} z_t(x_{t+1}).
\end{aligned}
$$

□

As such, the next query x_t can be optimally selected by $\max_{x\in\mathcal{X}} \Delta C_t(x_{t+1}, \mu)$.

2.3.2 Batch active sampling

Proposition 1 naturally extends to multiple node selection, or *batch active sampling*. Formally, we consider selecting n points for the next batch, $\mathbf{X}_t^n = \{x_\tau\}_{\tau=1}^n$, instead of a single point x_t. The (myopic)[2] IVR is then expressed as:

$$
\Delta C_t(\mathbf{X}_t^n, \mu) = z_t(\mathbf{X}_t^n)^\top (C_t(\mathbf{X}_t^n, \mathbf{X}_t^n) + \lambda \mathbf{I}_{t\times t})^{-1} z_t(\mathbf{X}_t^n), \tag{2.12}
$$

Thus, $\max_{\mathbf{X}_t^n \subset \mathbb{R}^{n\times d}} \Delta(\mathbf{X}_t^n, \mu, C_t)$ can yield the one-step optimal batch nodes. However, each node must be jointly optimized and the computational complexity scales with both the dimensionality d and the batch size n, making it non-trivial. Furthermore, z_t must be closed-form for deterministic optimization, which restricts the kernel-measure combinations to conjugate pairs.

Remarkably, this maximization task can be reduced to a kernel quadrature task:

Proposition 2. *Let $f_{t+1} \sim \mathcal{GP}(m_{t+1}, C_{t+1})$ be the updated GP from the previous iteration f_t, with the updated dataset $\boldsymbol{D}_{t+1} = \boldsymbol{D}_t \cup (\boldsymbol{X}_t^n, \boldsymbol{Y}_t^n)$, where $\boldsymbol{Y}_t^n = f(\boldsymbol{X}_t^n) + \epsilon_t^n$ are n noisy output observations, $\epsilon_t^n \sim \mathcal{N}(0, \sigma^2)$ is an independent noise vector, and ν_t^n is the weighted discrete distribution with nodes $\boldsymbol{X}_t^n$ and weights $\boldsymbol{w}_t^n$. We have*

$$
\underbrace{\max_{\boldsymbol{X}_t^n \in \mathbb{R}^{n\times d}} \Delta C_t(\boldsymbol{X}_t^n, \mu)}_{\text{batch integral variance reduction}} = \underbrace{\min_{\boldsymbol{X}_t^n, \boldsymbol{w}_t^n} \mathrm{MMD}^2_{\mathcal{H}_{C_{t+1}}}(\mu, \nu_t^n)}_{\text{sparse sampling}} = \underbrace{\inf_{\boldsymbol{X}_t^n, \boldsymbol{w}_t^n} \mathrm{wce}(\nu_t^n; \mathcal{H}_{C_{t+1}}, \mu)^2}_{\text{optimal kernel quadrature}},
$$

[2]We adopt a myopic approach instead of a roll-out strategy (sequentially optimizing and conditioning on the point), as it is challenging to address scenarios where the measure μ also depends on the iteration t, as observed in the Bayesian optimization case.

Proof. Using the definition of IVR and the identity in Eq. (2.6), we have

$$\Delta(\mathbf{X}_t^n, \mu, C_t) = \underbrace{\mathrm{MMD}^2_{\mathcal{H}_{C_t}}(\mu, \nu_t)}_{\text{old MMD: constant with fixed } \mathbf{X}_t} - \underbrace{\mathrm{MMD}^2_{\mathcal{H}_{C_{t+1}}}(\mu, \nu_{t+1})}_{\text{new MMD: } \mathbf{X}_{t+1}=\mathbf{X}_t\cup\mathbf{X}_t^n}, \tag{2.13}$$

$$\max_{\mathbf{X}_t^n} \Delta(\mathbf{X}_t^n, \mu, C_t) = \max_{\mathbf{X}_t^n} \underbrace{\cancel{\mathrm{MMD}^2_{\mathcal{H}_{C_t}}(\mu, \nu_t)}}_{\text{independent from } \mathbf{X}_t^n} - \mathrm{MMD}^2_{\mathcal{H}_{C_{t+1}}}(\mu, \nu_{t+1}), \tag{2.14}$$

$$= \min_{\mathbf{X}_t^n} \mathrm{MMD}^2_{\mathcal{H}_{C_{t+1}}}(\mu, \nu_{t+1}). \tag{2.15}$$

Let $\phi : \mathcal{X} \mapsto \mathcal{H}_{C_t}$ be the feature map associated with the kernel C_t. Then, the posterior kernel mean embedding

$$z_t(x) = \int_{\mathcal{X}} C_t(x, x')\mathrm{d}\mu(x') = \langle \phi(x'), \phi(\cdot)\rangle_{\mathcal{H}_{C_t}}, \tag{2.16}$$

can be expressed as the RKHS norm. The mean embedding is denoted as $z_t(\mu) = \int_{\mathcal{X}} \phi(x)\mathrm{d}\mu(x)$. Using this notation, we have

$$\begin{aligned}
&\min_{\mathbf{X}_t^n, \mathbf{w}_t^n} \mathrm{MMD}^2_{\mathcal{H}_{C_{t+1}}}(\mu, \nu_{t+1}) \\
&= \min_{\mathbf{X}_t^n, \mathbf{w}_t^n} \|z_t(\mu) - z_t(\nu_{t+1})\|^2_{\mathcal{H}_{C_{t+1}}}, && \text{(definition of MMD)} \\
&= \min_{\mathbf{X}_t^n, \mathbf{w}_t^n} \|z_t(\mu) - [z_t(\nu_t) + z_t(\nu_t^n)]\|^2_{\mathcal{H}_{C_{t+1}}}, && \text{(Discrete measure } \mathbf{X}_{t+1} = \mathbf{X}_t \cup \mathbf{X}_t^n) \\
&= \min_{\mathbf{X}_t^n, \mathbf{w}_t^n} \| \underbrace{[z_t(\mu) - \cancel{z_t(\nu_t)}]}_{\text{independent from } \nu_t^n} - z_t(\nu_t^n)\|^2_{\mathcal{H}_{C_{t+1}}}, \\
&= \min_{\mathbf{X}_t^n, \mathbf{w}_t^n} \|z_t(\mu) - z_t(\nu_t^n)\|^2_{\mathcal{H}_{C_{t+1}}}, \\
&= \min_{\mathbf{X}_t^n, \mathbf{w}_t^n} \mathrm{MMD}^2_{\mathcal{H}_{C_{t+1}}}(\mu, \nu_t^n). && \text{(MMD definition)}
\end{aligned}$$

By using the known equivalence between MMD and worst-case error in Eq. (2.6), the last equality is trivially proven. □

Thus, selecting $\mathbf{X}_t^n$ to minimize $\mathrm{MMD}^2_{\mathcal{H}_{C_t}}(\nu_t^n, \mu)$ corresponds to the optimal node selection for reducing the integral variance across multiple points.

Proposition 2.3.2 can also be interpreted inversely: the kernel quadrature algorithm provides both quadrature nodes and weights. Specifically:

$$\inf_{\mathbf{X}_t^n \subset \mathcal{X}, \mathbf{w}_t^n \subset \mathcal{R}^n} \mathrm{wce}(\nu_t^n; \mathcal{H}_{C_{t+1}}, \mu)^2 = \max_{\mathbf{X}_t^n \subset \mathcal{X}} \Delta C_t(\mathbf{X}_t^n, \mu). \tag{2.17}$$

Therefore, the nodes selected by the kernel quadrature algorithm $\mathbf{X}_t^n$ are optimal for minimizing integral variance. Iteratively applying this batch selection process

as Bayesian compression achieves one-step optimality. This concept forms the core idea of the first half of this thesis, as introduced in Chapter 3.

The kernel quadrature community has developed numerous algorithms that achieve a fast convergence rate for the worst-case error (Bach [19], Belhadji et al. [22, 23], Chen et al. [47], and Dwivedi et al. [67]). By leveraging the state-of-the-art kernel quadrature algorithm (e.g., Epperly et al. [70] and Hayakawa et al. [95]), we can solve the n-point selection task $\mathbf{X}_t^n$ to minimize[3] the kernel quadrature task.

Here, $\mathcal{H}_{C_{t+1}}$ needs to be conditioned on the unseen batch nodes $\mathbf{X}_t^n$. Therefore, we use the prior' kernel C_t as an approximation. The approximation error is precisely the difference kernel: $C_t(x,x') - C_{t+1}(x,x') = C_t(x,\mathbf{X}_t^n)(C_t(\mathbf{X}_t^n,\mathbf{X}_t^n)+\lambda\mathbf{I})^{-1}C_t(\mathbf{X}_t^n,x)$. This approximation error exhibits better convergence properties compared to the total Bayesian quadrature error (typically $\mathcal{O}(t^{-\alpha})$, e.g., $\alpha = \nu/d$ for the Matérn kernel (Xi et al. [238])). Moreover, its one-step (incremental) difference convergence has $\mathcal{O}(t^{-\alpha-1})$ in general. Thus, this 'prior' kernel approximation is an asymptotically accurate method and is at least superior to the final goal's convergence properties.

This framework elegantly unifies several key concepts: constructing an optimal quadrature rule with the worst-case error, minimizing MMD distance for sparse sampling, and integral variance reduction for average-case predictive uncertainty reduction. Remarkably, all these objectives can be jointly optimized using a kernel single quadrature solver. To apply this approach, we simply set the 'integral objective' as $\hat{Z} = \int_{\mathcal{X}} f_t(x)\mathrm{d}\mu(x)$. The batch selection objective is then formulated as: $\max_{\mathbf{X}_t^n \subset \mathcal{X}} \Delta_t(\mathbf{X}_t^n, \mu, \mathcal{H}_{C_t})$, then kernel quadrature solver returns optimal $\mathbf{X}_t^n$.

2.3.3 Additional benefits of the compression approach

This approach aligns naturally with information-theoretic bounds. The convergence bounds from an information-theoretic perspective fundamentally depend on the kernel-specific maximum information gain, as established in Theorem 2. This information gain is influenced by the eigenvalue decay of the Gram matrix, a

[3] In a rigorous sense, this is not true minimization but rather an effort to make it as small as possible. However, for the sake of intuition, we use the term 'minimize' here. A similar argument applies to the use of 'optimal' in the following sentences.

well-documented property for widely used kernels such as RBF and Matérn kernels (Kandasamy et al. [122] and Vakili et al. [219]). Consequently, this connection provides a foundation for analyzing the theoretical convergence rates across various subfields of PN. Notably, our objective in Eq. (2.6) is formulated on distributions rather than point estimates. This distinction enables the derivation of fully Bayesian convergence rates, offering a more robust framework for downstream tasks.

Furthermore, the compressed distribution ν_t^n can be viewed as the *representation* of the model's uncertainty over the measure μ, and their corresponding quadrature weights $\mathbf{w}_t^n$ indicate the relative importance of each point. Khanna et al. [128] and Williamson et al. [232] highlighted that compressed samples can serve as useful explanation tools. Consequently, this compression-based approach not only provides efficient expectation estimates and uncertainty quantification but also offers interpretable insights into the model as an explanability tool.

This compression also plays a central role in *memory*. By compressing the desired measure with respect to the model—such as the posterior distribution or data distribution—we enable more efficient and scalable computations. Posterior compression, also referred to as Bayesian coreset (Campbell et al. [38]) or thinning (Dwivedi et al. [67]), is crucial for scalable Bayesian inference when dealing with millions of data points. It is also vital for continual learning (Chang et al. [44], Khan et al. [127], and Pan et al. [177]), allowing models to retain knowledge over time while efficiently managing memory. This approach facilitates the sample-efficient approximation of non-closed-form posterior distributions, which is highly valuable for downstream tasks.

2.3.4 Summary

In summary, this new approach offers several unique advantages over existing methods.

1. **Scalable active learning solver**: The distributional formulation naturally extends to batch settings, enabling massive parallelization. For large-scale datasets and high-dimensional spaces, more data points are typically required, regardless of convergence rate advantages. In such cases, a cheap and scalable active learning solver becomes crucial. The complexity of our approach scales linearly with the number of data points, N, making it efficient for large-scale applications.

2. **General domain support**: This approach inherently supports general domains, including discrete, ordinal, or even Wiener spaces (Lyons et al. [154]), depending on how the measure μ and domain $\mathcal{X}$ are defined. Unlike optimization-based approaches, which require compact domains, kernel quadrature can handle non-compact domains.

3. **Robustness against misspecified RKHS**: Karvonen et al. [124] demonstrated that maximum likelihood estimation for GP hyperparameters can be ill-posed. Our approach ensures that the worst-case error of the integral estimate remains bounded even in misspecified cases—robustness not supported by existing methods (Bogunovic et al. [27] and Huszár et al. [111]).

4. **Explainability**: The compressed quadrature nodes are interpretable representation points of the current belief and weights represents importance (Khanna et al. [128] and Williamson et al. [232])

5. **Compressed posterior**: The compressed posterior distribution can make the downstream tasks scalable, such as simulation-based inference, continual learning, or thinning.

6. **Alignment with information-theoretic bounds and Bayesian principles**: The use of the MMD metric in the distributional formulation provides a natural way to analyze convergence rates for downstream tasks (Khribch et al. [129]), fully aligning with Bayesian principles.

7. **Integration of numerical precision and uncertainty**: The precision requirements of the task influence the selection of points, and vice versa. This dynamic interaction allows the algorithm to adaptively adjust its precision based on computational uncertainty.

8. **Separation of exploitation and exploration roles**: The measure μ encapsulates current domain knowledge, while the predictive covariance of f represents exploration. This eliminates the need to manually design acquisition functions to balance exploitation and exploration.

9. **Stopping criterion**: The uncertainty in the integral estimate, $\mathbb{V}_f[\hat{Z}]$, reflects the model's uncertainty and can serve as a stopping criterion, signaling when sufficient accuracy has been achieved.

The details of the above benefits are described in the following sections with concrete tasks.

2.4 Bayesian quadrature for Black-box inference

2.4.1 Task: Black-box inference.

In Chapter 3, we applied our approach to solve a black-box inference task, expressed as:

$$\mathbb{P}(\theta \mid \mathbf{D}, \mathcal{M}) = \frac{\mathbb{P}(\mathbf{D} \mid \theta, \mathcal{M})\mathbb{P}(\theta)}{\mathbb{P}(\mathbf{D} \mid \mathcal{M})} = \frac{\ell(\theta)\mu(\theta)}{\int_{\mathcal{X}} \ell(\theta)\mathrm{d}\mu(\theta)}, \tag{2.18}$$

where

$\ell(\theta) := \mathbb{P}(\mathbf{D} \mid \theta, \mathcal{M})$ is the likelihood function,

$\mu(\theta) := \mathbb{P}(\theta)$ is the prior distribution,

$Z = \int_{\mathcal{X}} \ell(\theta)\mathrm{d}\mu(\theta)$ is the marginal likelihood (or evidence),

$q(\theta) = \mathbb{P}(\theta \mid \mathbf{D}, \mathcal{M})$ is the posterior distribution,

and $\mathcal{M}$ represents the model, which often refers to a simulator, $\theta \in \mathcal{X}$ represents the parameters to infer, and $\mathbf{D}$ represents the observed dataset. While this follows the standard Bayes theorem, we consider the case where the likelihood function ℓ is black-box. This situation arises frequently in scenarios where the model is a simulator: we can query pointwise likelihood values by running the simulator, but the closed-form likelihood function is unavailable. Such problems are also known as simulation-based inference (SBI; Cranmer et al. [56]), likelihood-free inference (LFI; Gutmann et al. [93], Hinton [105], Huang et al. [107], and Hyvärinen [113]), approximate Bayesian inference (ABC; Csilléry et al. [57] and Fujisawa et al. [77]), amortized inference, indirect inference Gourieroux et al. [86], or synthetic likelihood Price et al. [182] and Wood [235].

2.4.2 Solution: Batch Bayesian quadrature.

We propose solving this problem using Bayesian quadrature. By placing a surrogate model on the likelihood function ℓ, the estimation of the evidence becomes equivalent

to a Bayesian quadrature task. Specifically, we aim to minimize the variance of the integral estimate:

$$\hat{Z} = \int_{\mathcal{X}} f_t(\theta)\mathrm{d}\mu(\theta) \overset{\text{IVR}}{\Longrightarrow} \max_{\mathbf{X}_t^n \subset \mathcal{X}} \Delta C_t(\mathbf{X}_t^n, \mu) \overset{\text{BQ}}{\Longrightarrow} \inf_{\nu_t^n} \mathrm{wce}(\nu_t^n; \mathcal{H}_{C_t}, \mu)^2, \tag{2.19}$$

subject to a given budget T. We solve this objective using an online algorithm, selecting n points at each iteration and repeating this batch selection process until the budget is exhausted.

Simultaneously, the estimation of ℓ allows for the computation of the posterior distribution: $q(\theta) = {}^{\mathbb{E}_\ell[\ell(\theta)]\mu(\theta)}/_{\mathbb{E}_\ell[Z]}$. Thus, the Bayesian quadrature approach enables the inference of both the evidence and the posterior distribution simultaneously, analogous to variational inference. However, unlike variational inference, Bayesian quadrature provides an exact evidence estimate rather than a lower bound, while also allowing for efficient querying strategies, resulting in sample-efficient approximations. In the MCMC-based approach, only Nested Sampling (Buchner [34, 35], Feroz et al. [71], Higson et al. [104], and Skilling [203]) can estimate both posterior and evidence at the same time. However, Bayesian quadrature is only directly applicable when the kernel mean and variance have closed-form expressions. To generalize to arbitrary combinations, a secondary estimation is required to approximate these intractable integrals.

2.4.3 Kernel quadrature via Nyström method

Using the equivalence between Bayesian quadrature and kernel quadrature in Eq.(2.6), we solve the quadrature node selection problem using the kernel quadrature algorithm (Hayakawa et al. [95]). They applied Tchakaloff's theorem (Theorem 1) in the kernel setting via the Nyström method (Drineas et al. [65], Kumar et al. [142], and Williams et al. [231]). Given a set of M points $\mathbf{X}_{\text{nys}} = \{x_i\}_{i=1}^M \subset \mathcal{X}$, the Nyström approximation of the GP predictive covariance kernel $C_t(x, y)$ is given by:

$$C_t(x, y) \approx \hat{C}_t(x, y) := \sum_{i=1}^{n-1} \lambda_i^{-1} \varphi_i(x)\varphi_i(y) \quad \text{s.t.} \quad \forall i \ \lambda_i > 0, \tag{2.20}$$

where $\varphi_i(\cdot) := u_i^\top C_t(\mathbf{X}_{\text{nys}}, \cdot)$ $(i = 1, \ldots, n-1)$ are *test functions*, chosen from the larger M-dimensional space $\mathrm{span}\{C_t(x_i, \cdot)\}_{i=1}^M$. To compute this, we perform

the best rank-s approximation of the Gram matrix $C_t(\mathbf{X}_{\text{nys}}, \mathbf{X}_{\text{nys}})$, given by its eigendecomposition: $C_t(\mathbf{X}_{\text{nys}}, \mathbf{X}_{\text{nys}}) = U\Lambda U^\top$, where $U = [u_1, \ldots, u_M] \in \mathbb{R}^{M\times M}$ is a real orthogonal matrix, and $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_M)$ with $\lambda_1 \geq \ldots \geq \lambda_M \geq 0$.

The test functions can then be used to approximate the integral $\hat{Z} = \int_{\mathcal{X}} f(x)\mathrm{d}\mu(x)$. Let $\boldsymbol{\varphi} = \varphi_1, \ldots, \varphi_{n-1}{}^\top$ denote the vector of test functions that spans $\mathcal{H}_{\hat{C}_t}$, the RKHS associated with the approximated kernel $\hat{C}_t$. Assuming additional knowledge of expectations, such as: $\int_{\mathcal{X}} \boldsymbol{\varphi}(x)\mathrm{d}\mu(x) \approx \int_{\mathcal{X}} \boldsymbol{\varphi}(x)\mathrm{d}\mu_N(x) = \mathbf{w}_N^\top \boldsymbol{\varphi}(\mathbf{X}_N)$, where $\mu_N = \sum_{i=1}^N w_i\delta_{x_i} \approx \mu(x)$ represents an *empirical measure*, where $N \gg M, N \gg n$ and $w_i \in \mathbf{W}_N$ and $x_i \in \mathbf{X}_N$ are independently sampled from $\mu(x)$. We can construct a convex quadrature (Hayakawa et al. [98]) $Q_n = \mu_{\text{BQ}}(x) = \sum_{i=1}^{n-1} w_i\delta_{x_i}$ as the *compressed distribution*, where $w_i \in \mathbf{w}_{\text{BQ}}$ and $x_i \in \mathbf{X}_{\text{BQ}}$ are Bayesian quadrature weights and nodes:

$$\int_{\mathcal{X}} \boldsymbol{\varphi}(x)\mathrm{d}\mu_{\text{BQ}}(x) = \int_{\mathcal{X}} \boldsymbol{\varphi}(x)\mathrm{d}\mu_N(x) \approx \int_{\mathcal{X}} f(x)\mathrm{d}\mu(x) = \hat{Z}, \tag{2.21}$$

where $f \in \mathcal{H}_{C_t}$. Thus, the integral can be approximated using $n-1$ test functions and N empirical measure.

The first equality in Eq. (2.21) represents cubature, namely *lossless compression*. Thus, the only source of error arises from two factors; the empirical measure approximation, $\mu_N \approx \mu$, and kernel approximation, $C_t \approx \hat{C}_t$. Thus, the worst-case error is bounded by these two terms (Hayakawa et al. [see 99, Eq. (13)]):

Theorem 3 (**worst-case error bound**). *If an n-point convex quadrature Q_n satisfies $\int_{\mathcal{X}} \varphi_j(x)\mu_{BQ}(x) = \int_{\mathcal{X}} \varphi_j(x)\mu_N(x)$ for $1 \leq j \leq n-1$ and $\int_{\mathcal{X}} \sqrt{C_t(x) - \tilde{C}_t(x)}d\mu_{BQ}(x) \leq \int_{\mathcal{X}} \sqrt{C_t(x) - \tilde{C}_t(x)}d\mu_N(x)$, then we have:*

$$\begin{aligned}\text{wce}[\mu_{BQ}; \mathcal{H}_{C_t}, \mu] &\leq \text{MMD}_{\mathcal{H}_{C_t}}(\mu_{BQ}, \mu_N) + \text{MMD}_{\mathcal{H}_{C_t}}(\mu_N, \mu), \\ &\leq 2\underbrace{\int_{\mathcal{X}} \sqrt{C_t(x) - \tilde{C}_t(x)}d\mu_N(x)}_{\textit{Nyström approximation}} + \underbrace{\text{MMD}_{\mathcal{H}_{C_t}}(\mu_N, \mu)}_{\textit{empirical measure}},\end{aligned}$$

The empirical measure term asymptotically converges to zero with infinitely many samples N. However, Hayakawa et al. [97] demonstrates that the required number of samples is surprisingly small (up to $C_m d$ with some positive constant

$C_m > 0$) due to a property called *hypercontractivity* (Janson [see 114, Chapter 5]), which explains the strong practical performance of this approach. Additionally, we assume that sampling from μ is inexpensive (e.g., Gaussian prior). For enumerable discrete distributions, this approximation term naturally approaches zero.

For the kernel approximation term, the Nyström method relies on the eigendecomposition of the Gram matrix and, unlike random Fourier features [184], does not require the assumption of kernel stationarity. This makes it suitable for non-stationary kernels, such as polynomial kernels or kernels with input-dependent length scales (e.g., deep kernel learning [233]). The Nyström method can also approximate non-positive semi-definite or non-Mercer kernels[4], provided they are symmetric. However, the resulting approximation may not represent a valid kernel for tasks requiring an RKHS, making theoretical justification challenging. The performance of the Nyström method largely depends on the rank of the approximate Gram matrix. Smoother kernels, such as RBF kernels, often produce Gram matrices with faster eigenvalue decay, resulting in better approximations. Consequently, this Nyström-based quadrature approach extends Bayesian quadrature to richer combinations of the measure μ and kernel k beyond the original 'conjugate' sets in Bayesian quadrature, with additional error determined by the eigenvalue decay.

Furthermore, the convexity of the cubature construction can provide the robustness to the misspecified case (Hayakawa et al. [see 98, Appendix B.4]:

[4]Examples include non-continuous kernels such as kernels with singularities, or those defined on non-compact domains (Minh et al. [161]).

Proposition 3. *Under the setting* $|\mu_{BQ}|_{TV} = |\tilde{\pi}_t|_{TV} = 1$*, where* $|\cdot|_{TV}$ *denotes the total variation, let* $\mathcal{H}_{K_{mis}}$ *be the misspecified RKHS and* $\tilde{f} \in \mathcal{H}_{K_{mis}}$ *be a function in the misspecified RKHS, and* μ_{BQ} *be a quadrature rule applied to a function* $f \notin \mathcal{H}_{K_{mis}}$*, leading only to the following bound via triangle equality and standard integral estimates:*

$$\begin{aligned}
&\left| \int f(x) d\pi_{KQ}(x) - \int f(x) d\mu_N(x) \right| \\
&\leq (|\mu_{BQ}|_{TV} + |\tilde{\mu}_t|_{TV}) \sup_{x \in \mathcal{X}} |f(x) - \tilde{f}(x)| + \|\tilde{f}\|_{\mathcal{H}_{K_{mis}}} \mathrm{wce}[Q_n; \mathcal{H}_{K_{mis}}, \mu_N] \\
&= 2 \sup_{x \in \mathcal{X}} |f(x) - \tilde{f}(x)| + \|\tilde{f}\|_{\mathcal{H}_{K_{mis}}} \mathrm{wce}[Q_n; \mathcal{H}_{K_{mis}}, \mu_N].
\end{aligned}$$

The first inequality in Proposition 3 highlights the advantage of using convex weights within the KQ rule. Non-convex weights can inflate the total variation $|\mu_{\mathrm{BQ}}|_{\mathrm{TV}}$, potentially leading to significant integration errors. Unlike traditional BQ, which employs real-valued (and hence possibly negative) weights and thus the misspecification causes arbitrarily bad error [111], the convex weights can maintain uniform bounds. This underscores the robustness of our KQ approach against RKHS misspecification.

Importantly, the efficient algorithm for solving this convex kernel quadrature is known as *kernel recombination* (Cosentino et al. [54], Litterer et al. [149], and Tchernychova [210]), with $\mathcal{O}(nN + n^3 \log(N/n))$ computational steps. This complexity is linear with respect to the number of data points N, making it particularly suitable for large-scale datasets. Using this method, the next querying point for Bayesian quadrature can be obtained by solving the cubature task directly, eliminating the need for additional solvers such as multi-start optimizers or acquisition functions. As such, this approach enables robust, fast, and sample-efficient black-box inference.

Our approach brings the following advantages to the existing approach:

1. **Non-compact and exponential convergence rate**: Unlike existing analyses on adaptive Bayesian quadrature (Kanagawa et al. [117]), which assumes compact domains, our Theorem 1 demonstrates exponential convergence rates in non-compact spaces for some smooth functions. This result bridges the gap between theoretical guarantees and practical applications.

2. **Generality in combining prior and likelihood**: We solve Bayesian quadrature using the kernel recombination, which allows for the use of any prior distribution and likelihood function by general empirical measure and Nyström methods.

3. **Robustness against misspecidied RKHS**: Our convex cubature construction offers uniform bound for the misspecified RKHS. Thus, this is robust against misspecified kernel hyperparameters.

4. **Scalable active learning solver**: Simulations often require significant computational time, ranging from hours to days, necessitating efficient parallelization. Our kernel-based quadrature approach naturally supports massive parallelization, significantly speeding up computations.

5. **Separation of exploitation and exploration**: We utilize importance sampling: $Z = \int_{\mathcal{X}} {}^{\ell(\theta)}/_{\pi(\theta)} \mathrm{d}\nu(\theta)$, where $\nu(\theta) \propto \mu(\theta)\pi(\theta) \propto \mu(\theta)\sqrt{C_t(\theta,\theta)}$ is the proposal distribution that is the measure weighted by the predictive standard deviation. While the proposal distribution corresponds to exploitation, kernel quadrature sampling represents exploration within the domain, leading to improved convergence rates.

2.5 Adaptive Batch Sizes for Active Learning & Black-Box Optimisation

Table 2.1: Overview of the function f and measure μ defined for each task.

task	black-box function f	measure μ
Black-box integration	integrand function $f(x)$	Borel measure $\mu(x)$
Black-box inference	likelihood function $\mathbb{P}(\mathbf{D} \mid x)$	prior $\mathbb{P}(x)$
Active Learning	expected outcome $\mathbb{E}_{y\sim\mathbb{P}(y\|x)}[y]$	test data distribution $\mathbb{P}(\mathcal{S})$
Black-box optimization	objective function $f(x)$	probability of global maxima $\mathbb{P}(x^\star \mid \mathbf{D}_t)$

In Chapter 4, we generalized the Bayesian compression approach to Bayesian active learning and black-box optimization tasks. Table 2.1 summarizes how Bayesian quadrature can be applied to each task by interpreting the function f

and the measure μ (details are provided in the following section).

Since all these tasks relate to Bayesian inference, alternative interpretations are possible. For instance, these tasks can be viewed as maximizing the information gain for the objective task, a perspective often emphasized in the Bayesian experimental design community (Chaloner et al. [43] and Rainforth et al. [185]). However, our Bayesian compression viewpoint interprets these tasks as compressing the measure under model uncertainty. Both perspectives have their merits and are not inherently superior or inferior. Indeed, Hübotter et al. [108] confirms that information-theoretic and variance-reduction-based approaches can outperform one another depending on the dataset. Nonetheless, our approach naturally integrates well with its solver, kernel quadrature, offering a unique advantage in connecting model uncertainty with numerical precision. This connection is particularly valuable when numerical precision needs to be dynamically adjusted to accommodate uncertainty.

In this chapter, we explore ways to link numerical precision and batch size. In parallelization, batch sizes are typically predetermined and fixed throughout the experiment. This fixed approach is inefficient due to the dynamic trade-off between cost and speed—larger batches are more expensive but faster wall-clock run-times, while smaller batches are cheaper but slower. Moreover, the optimal trade-off may shift over time, with larger batches often being more effective in the earlier stages.

To address this issue, we modify the Nyström-based kernel quadrature algorithm to allow users to set the numerical precision of the quadrature approximation, ϵ. Instead of fixing the batch size, we fix the precision ϵ, allowing the algorithm to dynamically adjust batch sizes to meet predefined precision objectives. This approach is analogous to how typical optimization algorithms terminate based on convergence thresholds. We further extend this method to scenarios involving safe active learning and safe optimization, where constraint violations are interpreted as reductions in precision requirements. This enables the algorithm to adapt batch construction accordingly, maintaining efficiency while ensuring safety.

2.5.1 Task: Bayesian active learning.

The goal of Bayesian active learning is to make the model—specifically, the GP in this thesis—as accurate as possible within a limited budget T of evaluations, queries, or experiments. This setting is also known as Bayesian experimental design (Chaloner et al. [43] and Rainforth et al. [185]), Bayesian adaptive design (Cheng et al. [48] and Zhou et al. [244]), or adaptive design optimization (Cavagnaro et al. [40] and Myung et al. [167]). At each step t, this goal translates into selecting the optimal location x_t for querying y_t to maximize the accuracy of our estimate of f.

The informativeness of a set of sampling points $\mathbf{X}_t \subset \mathcal{X}$ about f is quantified by the information gain (Cover [55]), which measures the mutual information between f and the observations $\mathbf{Y}_t$ at these points: $\mathbb{I}(\mathbf{Y}_t; f) = \mathbb{H}(\mathbf{Y}_t) - \mathbb{H}(\mathbf{Y}_t \mid f)$, where $\mathbb{I}$ and $\mathbb{H}$ denote mutual information and differential entropy, respectively, quantifying the reduction in uncertainty about f from observing $\mathbf{Y}_t$. For a Gaussian distribution, $\mathbb{H}(\mathcal{N}(\mu, \Sigma)) = {}^1\!/_2 \log|2\pi e\Sigma|$, leading to: $\mathbb{I}(\mathbf{Y}_t; f) = 2\log|\mathbf{I}_{t\times t} + \lambda^{-1}\mathbf{K}_{X_t X_t}|$. Selecting the next point x_t to maximize the information gain provides an information-theoretically optimal choice.

However, directly maximizing information gain over $\mathbf{X}_t \subset \mathcal{X}$ under the constraint $|\mathbf{X}_t| \leq T$ is NP-hard (Ko et al. [133]). This problem can instead be approximated using a greedy algorithm. Defining $F(\mathbf{X}_t) = \mathbb{I}(\mathbf{Y}_t; f)$, the algorithm selects: $x_t = \arg\max_{x\in\mathcal{X}} F(\mathbf{X}_{t-1} \cup \{x\})$, which is equivalent to $x_t = \arg\max_{x\in\mathcal{X}} C_{t-1}(x, x)$, corresponding to uncertainty sampling. This approximation guarantees a nearly optimal solution at the T-th iteration: $F(\mathbf{X}_t) \geq (1 - 1/e) \max_{\mathbf{X}\subset\mathcal{X}, |\mathbf{X}|\leq T} F(\mathbf{X})$, where $(1 - 1/e)$ is the approximation ratio (Nemhauser et al. [168]). Since mutual information $F(\cdot)$ is submodular, this guarantees asymptotic convergence to the optimal value (Krause et al. [136]). However, extending to multiple-point selection, or batch active learning, introduces additional complexity and remains a challenging problem despite prior advancements (Kirsch et al. [131] and Pinsler et al. [181]).

2.5.2 Solution: Expected predictive variance

Active learning is broad field and we cannot cover all objectives in once, but we consider minimizing the expected predictive uncertainty:

$$\hat{Z} = \int_{\mathcal{X}} f_t(\theta) \mathrm{d}\mathbb{P}(\mathcal{S}) \overset{\text{IVR}}{\Longrightarrow} \max_{\mathbf{X}_t^n \subset \mathcal{X}} \Delta C_t(\mathbf{X}_t^n, \mathbb{P}(\mathcal{S})) \overset{\text{BQ}}{\Longrightarrow} \inf_{\nu_t^n} \mathrm{wce}(\nu_t^n; \mathcal{H}_{C_t}, \mathbb{P}(\mathcal{S}))^2, \quad (2.22)$$

where $\mathbb{P}(\mathcal{S})$ represents the test data distribution and $x \in \mathcal{S}$. While the test data distribution $\mathcal{S}$ corresponds to the space where the model is expected to generalize predictions effectively, the *actionable* (or train) data distribution $\mathcal{A}_t$ represents the space that can be sampled during the training phase.

Typical active learning research assumes $\mathcal{A}_t = \mathcal{S}$ or at least $\mathcal{A}_t \subseteq \mathcal{S}$. However, the more general condition $\mathcal{A}_t \neq \mathcal{S}$ often arises in practice. For example, safety concerns or discrepancies between screening and deployment phases can cause the actionable dataset $\mathcal{A}_t$ to differ from the test dataset $\mathcal{S}$. This scenario is referred to as directed active learning (MacKay [155]) or transductive active learning (Hübotter et al. [108]). In contrast, typical setting, often referred to as inductive active learning, represents a special case of transductive active learning where $\mathcal{A}_t = \mathcal{S}$.

Interestingly, even in inductive active learning (where $\mathcal{A}_t = \mathcal{S} = \mathcal{X}$), differences between the test dataset and the actionable dataset can still arise. For instance, if the data points are from a discrete domain with $|\mathcal{X}| < \infty$, the actionable data distribution can be expressed as: $\mathbb{P}(\mathcal{A}_t) = {}^1\!/\!_{|\mathcal{A}_t|} \sum_{i=1}^{|\mathcal{A}_t|} \delta_{x_i}$, where $x_i \in \mathcal{A}_t$, and $\mathcal{A}_t = \mathcal{S} \backslash \mathbf{X}_t$ represents the pool of candidate unlabeled data, which is called pool-based active learning (Sener et al. [197]). In a broader context, the data distributions are not restricted to the uniform case. Other methods, such as density estimation (Terrell et al. [211]) or generative models (Ho et al. [106]), can be employed.

The objective corresponds to the upper bound of the mean squared error (MSE) on the test dataset $\overline{\mathbf{X}}_t$:

$$\begin{aligned} \frac{1}{N} \sum_{i=1}^{N} (f(x_i) - m_t(x_i))^2 &= \int_{\mathcal{X}} |f(x) - m_t(x)|^2 \mathrm{d}\mathbb{P}(\mathcal{S}), \\ &= \mathbb{V}_{f_t \sim \mathcal{GP}(m_t, C_t)} \left[\int_{\mathcal{X}} f_t(x) \mathrm{d}\mathbb{P}(\mathcal{S}) \right]. \end{aligned}$$

As such, minimizing the integral variance of the expected predictive variance leads to the widely used MSE loss function and aligns with the Bayesian quadrature objective in Eq.(2.6) by setting $\mu(x) = \mathbb{P}(\mathcal{S})$. The actionable data distribution $\mathbb{P}(\mathcal{A}_t)$ can be interpreted as the proposal distribution, and the integral estimator can be unbiased using importance sampling, as discussed in Chapter 3. Consequently, the kernel quadrature solver selects the subset of unlabeled data from the pool that is most effective in reducing the expected predictive variance.

While the measure $\mu(x)$ in black-box inference tasks (Chapter 3) remains stationary across iterations t, the measure $\mathbb{P}(\mathcal{S})$ in pool-based active learning can be non-stationary. In safe active learning, the actionable space $\mathcal{A}_t$ may be subject to *unknown constraints*, which need to be estimated from pointwise queries. For example, in drug discovery, safety constraints are often black-box and can only be inferred through pointwise animal experiments. Consequently, the measure $\mathbb{P}(\mathcal{S})$ as *expected* actionable data distribution becomes time-varying[5].

2.5.3 Task: Black-box optimisation.

The goal of black-box optimization is to find the global maximum of a black-box objective function f, i.e., $x^\star \in \arg\max_{x\in\mathcal{X}} f(x)$. This problem is well-studied under the framework of Bayesian optimization, which formulates optimization as sequential decision-making under uncertainty.

At iteration t, we obtain the dataset $\mathbf{D}_t = \{x_\tau, y_\tau\}_{\tau=1}^t$, where $\hat{y}_t^\star = \max_{\tau\in[t]} y_\tau$ represents the highest observed value so far. We can define a utility function $u(x)$ as follows: $u(x) = 0$ if $f(x) < \hat{y}_t^\star$, and $u(x) = 1$ otherwise. This corresponds to receiving a unit reward if $f(x)$ is greater than or equal to $\hat{y}_t^\star$, and no reward otherwise. The expected utility over the belief about f is then:

$$\begin{aligned}\mathbb{P}(f(x) \geq \hat{y}_t^\star) = \mathbb{E}_{f_t\sim\mathcal{GP}(m_t,C_t)}[u(x)] &= \int_{-\infty}^{\hat{y}_t^\star} \mathcal{N}(f_t; m_t(x), C_t(x,x))\mathrm{d}f_t, \\ &= \Phi(\hat{y}_t^\star; m_t(x), C_t(x,x)),\end{aligned}$$

[5]The true constraints are not time-varying; the variability arises from the probabilistic surrogate model of the constraints f_t, which evolves over time as new data is acquired.

where Φ is the cumulative density function of the normal distribution. This acquisition function, known as the probability of improvement (Kushner [143]), selects the next query point x_t by maximizing the expected utility, i.e., $x_t = \arg\max_{x \in \mathcal{X}} \mathbb{P}(f(x) \geq \hat{y}_t^\star)$. PI represents a one-step Bayes optimal action under expected utility theory (Mongin [163]).

By defining alternative utility functions, various acquisition functions have been proposed. For example:

1. Expected Improvement: $u(x) = \max(0, f(x) - \hat{y}_t^\star)$ (Bull [37], Mockus [162], and Osborne et al. [174]).
2. Knowledge Gradient: $u(x) = m_{t+1}(x) - m_t^\star$, where $m_t^\star = \max_{x \in \mathcal{X}} m_t(x)$ (Frazier [73]).
3. Entropy Search: $u(x) = \mathbb{H}[x^\star \mid \mathbf{D}_t] - \mathbb{H}[x^\star \mid \mathbf{D}_t, x, f(x)]$, where $\mathbb{P}(x^\star \mid \mathbf{D}_t)$ represents the belief about the location of the global optimum (Hennig et al. [100], Hernández-Lobato et al. [102], and Wang et al. [226]).

Unfortunately, there is no closed-form solution to compute $\mathbb{P}(x^\star \mid \mathbf{D}_t)$, so it is typically approximated via Monte Carlo sampling. Specifically, we draw N function samples $\mathbf{f}_t = \{f_t^{(i)}(\tilde{\mathbf{X}})\}_{i=1}^N$ from the GP over a discretized subset $\tilde{\mathbf{X}} \subseteq \mathcal{X}$. For each sample, the maximum is computed as $\hat{x}_i^\star = \max_{x \in \tilde{\mathbf{X}}} f_t^{(i)}(x)$. The resulting samples $\{\hat{x}_i^\star\}_{i=1}^N$ represent an (approximate)[6] i.i.d. compression of $\mathbb{P}(x^\star \mid \mathbf{D}_t)$. This approach is known as Thompson sampling (Thompson [215]), which is itself an acquisition function, though it lacks a direct interpretation under expected utility theory.

Another popular acquisition function, GP-UCB (Srinivas et al. [204]), is defined as: $\alpha_{\text{GP-UCB}}(x) := m_t(x) + \beta_t \sqrt{C_t(x, x)}$, where $\beta_t > 0$ is a trade-off parameter in Theorem 2. Although GP-UCB cannot be interpreted within the framework of expected utility theory[7], it has strong theoretical guarantees. Specifically, the iterative application of GP-UCB is proven to converge to the true global maximum $x^\star$ with high probability.

[6] An additional approximation error in the i.i.d. compression arises from the discretization error, $\tilde{\mathbf{X}} \sim \mathcal{U}(\mathcal{X})$.

[7] This is because GP-UCB does not take the form of an expectation.

2.5.4 Solution: Probabilistic lifting

In our paper (Adachi et al. [8]), we propose a probabilistic lifting technique to reinterpret the optimization task as a Bayesian quadrature task:

$$x^* \in \operatorname*{argmax}_{x \in \mathcal{X}} f(x) \quad \overset{\text{dual}}{\Longleftrightarrow} \quad \delta_{x^*} \in \operatorname*{argmax}_{\mu \in \mathbb{P}(\mathcal{X})} \int f(x) \mathrm{d}\mu(x), \tag{2.23}$$

where δ_x denotes the delta distribution at x, making $\delta_{x^\star}$ the point mass at the global maximum.

Adachi et al. [8] transformed a non-convex optimization problem, $\max f(x)$, into an infinite-dimensional optimization problem over the set of probability measures $\mathbb{P}(\mathcal{X})$. In this approach, we move away from conventional pointwise updates, such as $\operatorname{plim}_{t\to\infty} x_t = x^\star$. Instead, we focus on *distributional* updates, aiming for $\operatorname{plim}_{t\to\infty} \mu_t = \delta_{x^\star}$, where $\operatorname{plim}_{t\to\infty} \mathbb{E}_x[\mu_t] = x^\star$ and $\operatorname{plim}_{t\to\infty} \mathbb{V}_x[\mu_t] = 0$.

This reformulation results in $\max \int f(x)\mathrm{d}\mu(x)$, which transforms the non-convex objective f into a linear and convex problem with respect to μ. This distributional perspective is particularly appealing due to its parallelizability and convexity, features that have made it a cornerstone of optimization techniques, ranging from traditional primal-dual interior-point methods (Vandenberghe et al. [220] and Wright [236]) to contemporary Bayesian machine learning frameworks (Rudi et al. [191] and Wild et al. [230]).

However, our probabilistic lifting approach transforms the original non-convex problem into an even more computationally demanding one. A possible remedy is to assume a predefined functional form for μ. We define $\mu_t(x) := \mathbb{P}(x^\star \mid \mathbf{D}_t)$ as the measure at the t-th iteration, which corresponds to the same target distribution as in entropy search. The entropy search can be roughly interpreted as aiming to:

$$\hat{Z} = \int_{\mathcal{X}} f_t(\theta) \mathrm{d}\mathbb{P}(x^\star \mid \mathbf{D}_t) \overset{\text{IVR}}{\Longrightarrow} \max_{\mathbf{X}_t^n \subset \mathcal{X}} \Delta C_t(\mathbf{X}_t^n, \mathbb{P}(x^\star \mid \mathbf{D}_t)) \overset{\text{BQ}}{\Longrightarrow} \inf_{\nu_t^n} \operatorname{wce}(\nu_t^n; \mathcal{H}_{C_t}, \mathbb{P}(x^\star \mid \mathbf{D}_t))^2, \tag{2.24}$$

for every iteration. Thus, this can be viewed as variance reduction over the *expected* global maximum, which corresponds to the Bayesian regret: $f(x^\star) - \sum_{i=1}^n f(x_i)$, where $x_i \sim \mu_t(x)$, and the second term represents the Monte Carlo estimate of the

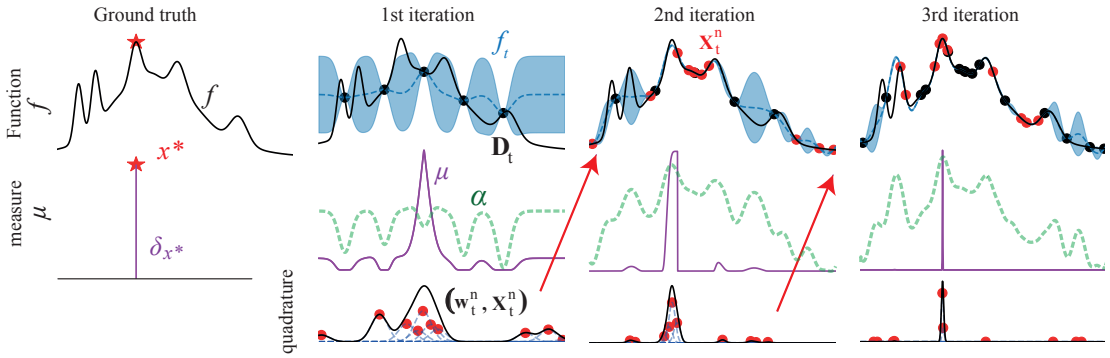


Figure 2.1: Finding the location of the global maximum $x^\star$ is equivalent to identifying the delta distribution $\delta_{x^\star}$. Using the surrogate model f_t, we set the measure μ as the belief about the global maximum $\mathbb{P}(\hat{x}^\star \mid \mathbf{D}_t)$. Additionally, a user-defined acquisition function α_t (e.g., UCB) can be employed to guide batch sample selection toward regions with higher rewards. The kernel quadrature algorithm provides a weighted point set $(\mathbf{w}_t^n, \mathbf{X}_t^n)$ that forms a discrete probability measure approximating the target measure π. To visualize this quantization, we use a weighted kernel density estimation based on $(\mathbf{w}_t^n, \mathbf{X}_t^n)$. As iterations progress, the measure μ contracts toward the global maximum, ideally converging to a delta function at the global maximum in single-maximum cases.

global maximum. To address this, we apply a Bayesian quadrature approach to construct n-point quadrature nodes that effectively minimize the integral variance. At each step t, the quadrature compresses the distribution as: $\int_{\mathcal{X}} f_t(x)\mathrm{d}\mu_t(x) \approx \int_{\mathcal{X}} f_t(x)\mathrm{d}\nu_t(x)$, where $\nu_t(x) = \sum_{i=1}^n w_i \delta_{x_i}$ is the compressed distribution. The Bayesian regret then becomes: $f(x^\star) - \int_{\mathcal{X}} f_t(x)\mathrm{d}\nu_t(x)$. As $\operatorname{plim}_{t\to\infty} \mu_t = \delta_{x^\star}$ and $\operatorname{plim}_{t\to\infty} \nu_t = \delta_{x^\star}$, it follows that: $\operatorname{plim}_{t\to\infty} \int_{\mathcal{X}} f_t(x)\mathrm{d}\nu_t(x) = x^\star$, and the Bayesian regret converges to zero. Figure 2.1 visualised the process of our approach. Over the iterations, the measure μ gradually shrink toward the delta measure $\delta_{x^\star}$.

Although full regret analysis of this approach is a significant challenge, this approach shows strong empirical performance. Figure 2.2 also shows the 2-dimensional example of our approach compared with popular baselines; batch Thompson sampling (Hernández-Lobato et al. [103] and Kandasamy et al. [121]) and hallucination (Azimi et al. [18] and González et al. [85]). While measure contraction avoids over-exploration unlike hallucination, Bayesian quadrature avoids over-exploitation unlike batch Thompson sampling.

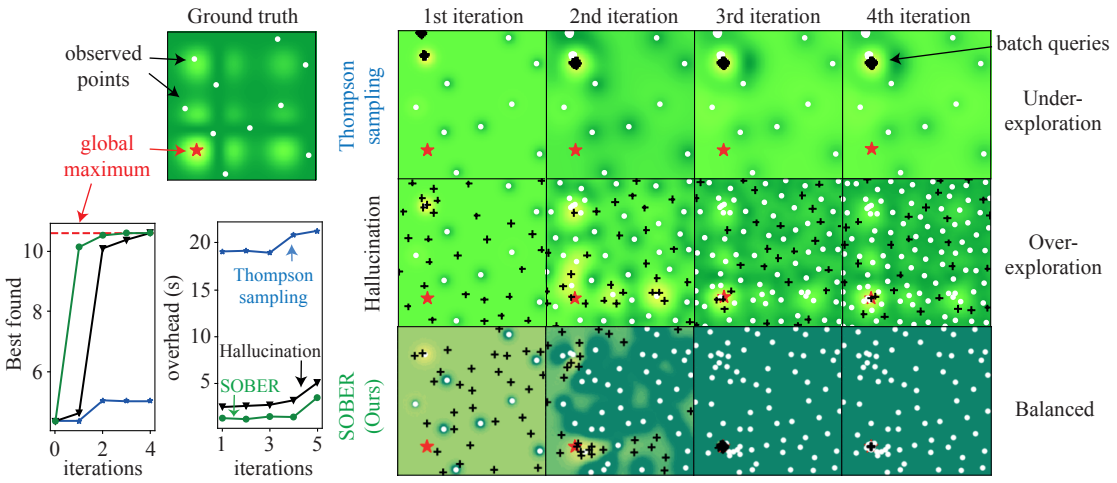


Figure 2.2: A demonstrating example featuring 2D Branin-Hoo function [63] with nine peaks and the global maximum at the bottom-left corner (red star). Initial 10 i.i.d. samples (white dots) unluckily misidentify the top-left peak as the promising area. Thompson sampling (blue lines) under-explores, erroneously focusing 30 queries (black crosses) near the top-left. Conversely, hallucination (black lines) over-explores, constantly venturing into new regions, yet allocating only a few queries towards the bottom-left area. Our approach (green lines), named SOBER [8], starts with wide exploration, then narrows down to the global maximum, demonstrating balanced exploration. The convergence plot illustrates that SOBER outperforms the baselines with the least wall-clock time overhead. The image's colour scheme represents different functions: upper confidence bound for Thompson and hallucination, $\log\mu$ for SOBER.

2.5.5 Task & Solution: Adaptive batch size

As shown in Table 2.1, most PN tasks can be reformulated as Bayesian quadrature tasks. Here, we aim to determine the batch size adaptively.

In Chapter 4, we introduced the concept of precision in the Nyström-based kernel quadrature approach:

$$\left|\int_{\mathcal{X}} \varphi_j(x)\mathrm{d}\mu_N(x) - \int_{\mathcal{X}} \varphi_j(x)\mathrm{d}\nu_t^n(x)\right| \leq \epsilon_{\mathrm{LP}}\sqrt{\frac{\lambda_j}{n-1}}, \tag{2.25}$$

where ϵ_{LP} denotes the tolerance parameter. Originally, as stated in Theorem 3, cubature construction was a lossless compression. However, we *intentionally* introduced an additional numerical precision term, ϵ_{LP}, to the cubature construction, enabling adaptive batch size selection. Intuitively:

1. Higher precision demands result in smaller quadrature error tolerances, requiring a larger sample set for more accurate integration.

2. Conversely, lower precision requirements allow fewer quadrature nodes to meet the desired accuracy.

More specifically, the batch size is tied to slack variables in linear programming solvers used in the kernel recombination algorithm. As the tolerance ϵ_{LP} increases, some inequality constraints are deactivated [60]. The batch size is determined by the number of active constraints, often resulting in sparse weights with $|\mathbf{X}_n| < n$. When constraints are loose, a large preset batch size becomes inefficient, as the desired precision can be achieved with fewer samples.

We fixed ϵ_{LP} instead of the batch size. Consequently, the batch size n adapts dynamically to the difficulty of the given quadrature task, resulting in larger batch sizes during the early stages due to high model uncertainty, and smaller batch sizes in the later stages as uncertainty decreases. As such, this approach enables the identification of an adaptive batch size n without requiring a brute-force search of all possible batch sizes.

2.6 Application to system identification in lithium-ion batteries

In Chapter 5, we apply the Bayesian quadrature approach to real-world problems in lithium-ion batteries. Lithium-ion batteries are considered a key enabler of a greener society, and their management significantly impacts battery lifetime [198]. Battery management systems typically rely on model-based control, where accurate system identification is crucial for control efficiency and precision. System identification essentially involves selecting the most plausible model from a set of candidate simulators.

We formulate the system identification task as a Bayesian model selection problem, where the goal is to select the model with the largest evidence. Since the simulators operate only in a forward manner, the likelihood function cannot be expressed in closed form, making this a black-box inference task.

A notable challenge in this setting is the large size of the observation dataset $\mathbf{D}$. This results in an extremely wide dynamic range for the likelihood, often reaching values as high as $\exp(10,000)$. To address this, it is standard practice to work with the logarithm of the likelihood. However, modeling the log-likelihood with a GP introduces another challenge: the exponential of Gaussian distribution is no longer Gaussian.

To overcome this, we adopted the moment-matching approximation proposed by Chai et al. [42] and Gunter et al. [92], which approximates the required transformations as follows:

$$\begin{aligned}
\ell \mid \mathbf{D}_{\log} \sim & \mathcal{GP}(\exp(m_t), \exp(C_t)), \\
\exp(m_t(x)) \approx & \exp\left(m_t(x) + \frac{1}{2} C_t(x,x)\right), \\
\exp(C_t(x,x')) \approx & m_t(x) \exp\left(C_t(x,x')\right) m_t(x') - m_t(x) m_t(x'),
\end{aligned}$$

where $\mathbf{D}_{\log}$ is the observed log-likelihood and corresponding location. This approximation allows us to condition the GP on log-likelihood values while maintaining a closed-form expression for the posterior predictive distribution in the original likelihood space.

2.7 Preference Learning for Human-AI Collaboration

Next, we consider the challenge that users may struggle to fully express their knowledge in the form of a Bayesian prior. To address this, we adopt the following viewpoint: scientific expert users have a latent belief distribution, but they may struggle to express this belief distribution quantitatively. Consequently, they require algorithmic support to elicit their beliefs in a form that machine learning algorithms can understand. This process is known as *prior elicitation* (Mikkola et al. [158], O'Hagan et al. [172], and Winkler [234]).

While various prior elicitation methods have been proposed, we employ a preference learning approach (Bradley et al. [31]). Humans are known to excel at making relative comparisons rather than quantifying absolute preferences for

a single choice (Kahneman et al. [116]). Therefore, instead of directly asking experts to provide a belief function in a functional form, we query them using relative comparisons, such as pairwise comparisons or rankings (Fürnkranz et al. [78]). From these comparisons, we estimate the latent utility function (Fishburn [72]) or preferential ranking function.

Eliciting scientific experts' intuition through preference learning has been shown to be effective. For instance, Choung et al. [49] demonstrated that a preferential ranking function elicited from 35 medicinal chemists was a better predictor for synthesizable and non-toxic drugs than conventional methods. Similarly, Llompart et al. [152] showed that ranking functions elicited from 92 scientists outperformed machine learning models trained solely on public datasets for hERG inhibition prediction. These results suggest that scientific experts' beliefs, elicited through preference learning, can serve as effective priors for Bayesian models.

2.7.1 Preference learning.

In this thesis, we focus on preference feedback in the form of pairwise comparisons. Humans express preferences between pairs of options $[x, x']$. A human prefers option x over x', denoted $x \succ_u x'$, if and only if $u(x) > u(x')$, where $\succ_u$ represents the human's preference relation. Thus, the utility function $u(\cdot)$ determines the preference relationship.

The Von Neumann–Morgenstern (VNM) utility theorem (Von Neumann et al. [222]) assumes two fundamental rationality axioms:

1. **Completeness**: The human has a well-defined preference, meaning the feedback is either $x \prec x'$ or $x \succ x'$.
2. **Transitivity**: Preferences are consistent across any three options, i.e., if $A \succ B$ and $B \succ C$, then $A \succ C$. This prevents cyclic preferences like rock-paper-scissors.

These axioms ensure that a utility function can be inferred from preferential feedback, allowing preference maximization to be formulated as the maximization of the underlying utility function.

Formally, given t-th pair of options (x_t, x'_t), there exists an oracle mechanism that returns a preference signal $\mathbf{1}_{x_t \prec x'_t}$ from the human where $\mathbf{1}_{x \prec x'} = 1$ if x_t is preferred and zero if x' is preferred. The feedback $\mathbf{1}_{x \prec x'}$ from the oracle follows the Bernoulli distribution with $\mathbb{P}(\mathbf{1}_{x \prec x'} = 1) = S(u(x) - u(x'))$, where $S(z) = (1 + \exp(-z))^{-1}$ is the sigmoid function. This preference likelihood is the so-called Bradley-Terry model (Bradley et al. [31]).

Similar to the black-box integration task, the utility function is unknown a priori, thus we need to estimate it from the preferential data, or *duel*, $D_t = (x_t, x'_t, \mathbf{1}_t)$. Similarly, we can assume the true utility function lies in RKHS, i.e., $u \in \mathcal{H}_k$. One distinction is that our likelihood is now Bernoulli distribution, which is not conjugate with Gaussian prior. Thus, the predictive posterior of preferential Gaussian process model is no more closed-form. One common approach is to approximate the posterior distribution by drawing function samples from GP prior, then transforming through the link function then we get the approximated predictive posterior distribution.

2.7.2 Human-AI collaboration.

We consider a scenario where computational agents and human experts collaborate on PN tasks, such as Bayesian optimization applied to battery design. Human expertise is valuable for accelerating these tasks, and our goal is to integrate this knowledge into the computational agent.

As one research direction, we might assume that expert knowledge is strong, as in physics-informed machine learning (Doumèche et al. [64] and Karniadakis et al. [123]), or when serving as a cost-effective proxy of an oracle mechanism (AV et al. [17]). However, in human-AI collaborative settings, we often assume that human expert knowledge is weaker—experts cannot explicitly express their knowledge in the form of functions, differential equations, or probability distributions. This scenario is particularly relevant in scientific applications, where experts themselves are engaged in trial and error at the forefront of discovery. Their knowledge is fluid and evolving, yet still informed by accumulated past experience. In such

cases, algorithmic support is essential for eliciting and incorporating this tacit knowledge into the computational framework.

How does human expertise benefit the PN algorithm? In a simple Bayesian inference scenario, this is analogous to specifying a prior distribution $\mathbb{P}(x)$: a stronger prior generally leads to a better posterior in principle. In a GP prior $f \sim \mathcal{GP}(m_0, k)$, this corresponds to the selection of appropriate kernel k. However, for a black-box function, it is not always easy to specify the appropriate kernel (Duvenaud [66]). And the experts' knowledge is not restricted to function space knowledge, e.g. a promising region to start for optimization tasks. Such experts' knowledge can significantly accelerate the optimization process by identifying a promising region, $\mathcal{X}_{\text{expert}} \subset \mathcal{X}$. Let $\text{vol}(\mathcal{X})$ denote the volume of the domain. By definition, $\text{vol}(\mathcal{X}_{\text{expert}}) \leq \text{vol}(\mathcal{X})$. As shown by Kandasamy et al. [120], the maximum information gain depends on the domain volume. Let $\Psi_t(\mathcal{X})$ represent the maximum information gain at iteration t over domain $\mathcal{X}$. Then, the ratio of maximum information gains between the expert-suggested region and the full domain is: $\Psi_t(\mathcal{X}_{\text{expert}})/\Psi_t(\mathcal{X}) = \text{vol}(\mathcal{X}_{\text{expert}})/\text{vol}(\mathcal{X})$. Thus, if expert input reduces the domain volume by a factor of 10, the convergence rate could improve by the same factor. While this explanation does not indicate an order-wise improvement in the convergence rate, the constant-wise improvement is particularly valuable under a limited budget T. Although asymptotic convergence rates matter in large-budget scenarios, constant-wise improvements are crucial during the early stages of expensive scientific experiments, where resources are constrained, and expert guidance can significantly accelerate progress.

However, expert knowledge elicitation faces several challenges:

1. **Elicitation efficiency**: To accurately elicit human knowledge, we may need to ask numerous pairwise comparisons. However, human experts prefer to minimize interactions with machines for convenience.
2. **Uncertainty in elicitation accuracy**: the limited pairwise dataset introduces uncertainty in estimating the utility function.
3. **Uncertainty in effectiveness**: human knowledge can be valuable but may also be completely erroneous.
4. **Uncertainty in comprehension**: humans may lack the background knowledge necessary to accurately evaluate given pairs, and their feedback can be more precise when explanations are provided.
5. **Incompleteness**: humans may only have confidence in their preferences for a limited number of pairs and may be unable to determine the preference relationships for all pairs.
6. **Inconsistency**: humans may exhibit inconsistent preference relationships that do not satisfy transitivity.

These challenge motivates us to develop the methods to solve these issues.

2.8 Human-AI Collaboration for Scientific Experts

In the latter half of this thesis, we investigate the human-AI collaboration setting for Bayesian optimization. Unlike the earlier sections, we adopt the standard GP-UCB approach and its theoretical framework. This shift reflects our focus on designing a cognitively intuitive user interface to elicit experts' tacit knowledge with precision, robustness, and efficiency. The algorithm developed in this chapter is designed to integrate seamlessly with the unified solver introduced in the first half of the thesis, discussed in the conclusion (Chapter 8)

2.8.1 Explainable Bayesian optimization for Human-AI collaboration

In Chapter 6, we proposed a human-AI collaborative BO algorithm, positioning humans as supervisors of the optimization process. The optimizer suggests a

pair of possible candidates, and the human selects the one they prefer, based on their belief that it is better.

Within this framework, we addressed the following cognitive challenges:

1. **Uncertainty in utility function**: Experts have an underlying utility function that governs their preferential relationships, but they cannot provide quantitative utility values for given pairs.
2. **Uncertainty in effectiveness**: Expert advice can be erroneous even if provided with confidence.
3. **Automation bias** (Cummings [59]): People often over-rely on suggestions from automated systems.

In response, we propose the following solutions:

1. **Preference learning**: Using a preferential GP (Chu et al. [51]), we infer the underlying utility function from the human's preferential feedback.
2. **Preset decaying factor**: The influence of the expert's utility function on the candidate generation algorithm is designed to decay over time, regardless of its effectiveness. This ensures it does not affect the asymptotic convergence rate of the regret bound.
3. **Explanability**: We incorporate closed-form Shapley values for the acquisition function (Chau et al. [46]) to explain the importance of each dimension in generating candidate pairs.

The Shapley value (Lundberg et al. [153] and Shapley [199]) is a popular explainability tool in machine learning. Explainability methods can be categorized into counterfactual explanations (Wachter et al. [223]), global explanations (Friedman [75]), and local explanations (Ribeiro et al. [188]). Shapley values fall under the local approach, providing feature importance attribution for specific data points.

Unlike other local explanation methods, such as variance-based (Cuevas et al. [58]) or gradient-based approaches (Ancona et al. [14]), the Shapley value is grounded in cooperative game theory and satisfies rationality axioms such as symmetry, efficiency, and linearity (Chau et al. [46]). While certain limitations of Shapley values have been identified, particularly from a human-centric explanation perspective

(Kumar et al. [140] and Kumar et al. [141]), they remain a de facto standard within the explainable AI community.

2.8.2 Principled collaboration by Optimistic MLE

While the previous approach introduced new concepts in explainable AI and a collaborative paradigm, several challenges remain unresolved:

1. **Inconsistency**: Human preferences may not rank all pairs consistently, leading to intransitive relationships or cyclic preferences (Budescu et al. [36]).
2. **Significant feedback effort**: Experts may be reluctant to provide preferential feedback for every query. They may prefer to stop providing feedback at some point and allow the algorithm to continue optimization autonomously.
3. **Fixed trust level**: There may be no prior knowledge of how reliable the experts' advice is, making it challenging to determine an appropriate trust level.

To address these issues, we propose the following solutions:

1. **Binary feedback**: Instead of pairwise comparisons, we adopt binary feedback to classify the next single-point candidate as either 'good' or 'bad.' This eliminates the possibility of intransitive or cyclic preferences, as comparisons are no longer required.
2. **Efficient elicitation**: Using the covering number theory (Zhou [243]), we derive a sublinear bound on the number of human feedback instances required. Initially, multiple labels per query are needed, but the frequency of binary feedback queries asymptotically converges to zero as the model learns.
3. **Data-driven trust level**: Instead of a predefined, fixed trust level, we introduce an adaptive trust level based on model uncertainty. As the model's confidence increases, it becomes increasingly self-reliant, requiring less human input over time.

To achieve this, we introduced the likelihood ratio model (Owen [176]) combined with optimistic MLE (Liu et al. [150]), as an alternative to preferential GP. The

primary challenge with preferential GP lies in the lack of a closed-form posterior predictive distribution, making it both theoretically and computationally challenging.

The core idea of optimistic MLE is to extend the original MLE point estimate into an interval estimate, bounded by the worst-case error with high probability guarantees. This interval is then used as a Bayesian prior. Importantly, this approach constrains the *function space*, $f \in \mathcal{F}$, rather than the input space, $x \in \mathcal{X}$.

Specifically, let the point MLE estimate be defined as $L_t = \max_{f \in \mathcal{H}_k} \ell_t(f)$, where $\ell_t(\cdot)$ is the log-likelihood function for a given binary dataset $\mathbf{D}_t$. For binary classification tasks, we use a Bernoulli likelihood. This objective can be interpreted as kernelized logistic regression (Zhu et al. [245]). The true function then lies in 'posterior' RKHS: $\mathcal{H}_{k,t} := \{f \in \mathcal{H}_k \mid \ell_t(f) \geq L_t - \alpha(t)\}$, where $\alpha(t)$ is a width hyperparameter for the optimistic MLE bound. This parameter is determined by the covering number, which depends on the kernel function and the number of observed data points t, similar to the information gain bound in Theorem 2.

Xu et al. [240] proved that this posterior function space, $\mathcal{H}_{k,t}$, contains the true function f with high probability, $1 - \delta$. Consequently, $\mathcal{H}_{k,t}$ can be viewed as a posterior function space, analogous to the GP predictive posterior class $\mathcal{H}_{C_t}$ described in Section 2.2. However, unlike in GP-based formulations, this approach does not assume a specific distribution within the interval, allowing for non-Gaussian distributions.

This flexibility is particularly important for preference modeling, as posterior predictive distributions with Bernoulli likelihoods are often skewed (Benavoli et al. [24]). Therefore, the optimistic MLE approach is well-suited for handling such cases.

Part I

Bayesian Quadrature for Probabilistic Numerics

The statement "if you want your convergence rate to be independent of problem dimension, do your integration with Monte Carlo" is much like the statement "If you want your nail-hammering to be independent of wall hardness, do your hammering with a banana." [101]

— Michael A. Osborne, Professor of Machine Learning

Everything that works works because it's Bayesian—David Duvenaud and I even snatched herding from Max Welling and Alex Smola when we established herding is just Bayesian quadrature done slightly wrong. [110]

— Ferenc Huszár, Professor of Machine Learning

3

Fast Bayesian inference with batch Bayesian quadrature via kernel recombination

This chapter is based on the following publication:

> **Masaki Adachi***, Satoshi Hayakawa*, Martin Jørgensen, Harald Oberhauser, and Michael A. Osborne. Fast Bayesian inference with batch Bayesian quadrature via kernel recombination. In *Advances in Neural Information Processing Systems (NeurIPS)* 35, 16533-16547, 2022.
>
> *Equal contribution

In Chapter 3, our goal is to further enhance the advantages of the Bayesian quadrature approach over classical MCMC methods for solving black-box inference problems. Unlike MCMC, which lacks a function prior, Bayesian quadrature restricts the function space to the GP covariance kernel RKHS, $f \in \mathcal{H}_{C_t}$.

We addressed two key challenges: (A) Practicality: The computational overhead of active sampling is justified only when each evaluation is expensive, and the limited range of 'conjugate' measure-kernel pairs (μ, k) restricts applicability. (B) Theoretical gap: Existing convergence analyses for adaptive Bayesian quadrature

are limited to compact domains (Kanagawa et al. [119]), which is problematic in practice (e.g., Gaussian priors are non-compact).

In response, we introduced the kernel quadrature approach, specifically kernel recombination (Hayakawa et al. [98]), to handle batch Bayesian quadrature tasks and solve black-box inference problems. This approach offers several advantages: (A) Scalable computation: Complexity is linear in the number of data points, N, enabling massive parallelization. (B) Versatility: Supports non-compact measures μ and a wide range of positive semi-definite kernels k, including non-stationary and non-Mercer kernels. Additionally, we also demonstrated that the importance sampling with proposal distribution $g(x) = \sqrt{C_t(x,x)}\mu(x)$ is optimal for Bayesian quadrature (see Theorem 1 and Lemma 1). However, constructing the empirical measure $\mu_N \approx \mu$ remains a challenge due to the difficulty of sampling from the square-root term in $g(x)$.

To address these issues, we introduced the factorization trick via WSABI approximation (Gunter et al. [92]). WSABI applies a square-root transformation to the GP function space, $f = \alpha + 1/2\tilde{f}^2$, where α is a constant, and $\tilde{f}$ is a GP conditioned on the square root of the observations, $\tilde{\mathbf{y}} = \sqrt{\mathbf{y} - \alpha}$. While f is no longer a GP (technically, it follows a Chi-squared distribution), WSABI approximates it as a GP using linearization. Although interpreting the RKHS of the WSABI-approximated GP is challenging, this approximation provides a closed-form posterior predictive distribution, allowing the construction of an approximate RKHS, $\mathcal{H}_{\tilde{C}_t}$.

$$\int_{\mathcal{X}} f(x)\mathrm{d}\mu(x) \approx \alpha + \int_{\mathcal{X}} \tilde{f}^2(x)\mathrm{d}\mu(x) = \alpha + \int_{\mathcal{X}} \tilde{f}(x)\mathrm{d}\tilde{f}(x)\mu(x) = \alpha + \int_{\mathcal{X}} \tilde{f}(x)\mathrm{d}g(x), \tag{3.1}$$

where the proposal distribution $g(x) = \tilde{C}_t(x,x)\mu(x)$ approximates the optimal distribution $\sqrt{C_t(x,x)}\mu(x)$. Since the predictive variance of the square-root GP, $\tilde{C}_t(x)$, is closed-form and behaves as a mixture of Gaussians for RBF kernels, it facilitates efficient sampling for constructing empirical measures.

Although this factorisation trick is applicable only to certain kernels that allow scalable sampling, we also provide a more general scheme using sampling importance resampling (SIR) (Kitagawa [132]).

Fast Bayesian Inference with Batch Bayesian Quadrature via Kernel Recombination

Masaki Adachi*
Machine Learning Research Group, University of Oxford
Toyota Motor Corporation
masaki@robots.ox.ac.uk

Satoshi Hayakawa*, **Harald Oberhauser**
Mathematical Institute, University of Oxford
{hayakawa,oberhauser}@maths.ox.ac.uk

Martin Jørgensen, Michael A. Osborne
Machine Learning Research Group, University of Oxford
{martinj, mosb}@robots.ox.ac.uk

Abstract

Calculation of Bayesian posteriors and model evidences typically requires numerical integration. Bayesian quadrature (BQ), a surrogate-model-based approach to numerical integration, is capable of superb sample efficiency, but its lack of parallelisation has hindered its practical applications. In this work, we propose a parallelised (batch) BQ method, employing techniques from kernel quadrature, that possesses an empirically exponential convergence rate. Additionally, just as with Nested Sampling, our method permits simultaneous inference of both posteriors and model evidence. Samples from our BQ surrogate model are re-selected to give a sparse set of samples, via a kernel recombination algorithm, requiring negligible additional time to increase the batch size. Empirically, we find that our approach significantly outperforms the sampling efficiency of both state-of-the-art BQ techniques and Nested Sampling in various real-world datasets, including lithium-ion battery analytics.[2]

1 Introduction

Many applications in science, engineering, and economics involve complex simulations to explain the structure and dynamics of the process. Such models are derived from knowledge of the mechanisms and principles underlying the data-generating process, and are critical for scientific hypothesis-building and testing. However, dozens of plausible simulators describing the same phenomena often exist, owing to differing assumptions or levels of approximation. Similar situations can be found in selection of simulator-based control models, selection of machine learning models on large-scale datasets, and in many data assimilation applications [28].

In such settings, with multiple competing models, choosing the best model for the dataset is crucial. Bayesian model evidence gives a clear criteria for such model selection. However, computing model evidence requires integration over the likelihood, which is challenging, particularly when the likelihood is non-closed-form and/or expensive. The ascertained model is often applied to

*Equal contribution
[2]Code: https://github.com/ma921/BASQ

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

produce posteriors for prediction and parameter estimation afterwards. There are many algorithms specialised for the calculation of model evidences or posteriors, although only a limited number of Bayesian inference solvers estimate both model evidence *and* posteriors in one go. As such, costly computations are often repeated (at least) twice. Addressing this concern, nested sampling (NS) [71, 46] was developed to estimate both model evidence and posteriors simultaneously, and has been broadly applied, especially amongst astrophysicists for cosmological model selection [63]. However, NS is based on a Monte Carlo (MC) sampler, and its slow convergence rate is a practical hindrance.

To aid NS, and other approaches, parallel computing is widely applied to improve the speed of wall-clock computation. Modern computer clusters and graphical processing units enable scientists to query the likelihood in large batches. However, parallelisation can, at best, linearly accelerate NS, doing little to counter NS's inherently slow convergence rate as a MC sampler.

This paper investigates batch Bayesian quadrature (BQ) [65] for fast Bayesian inference. BQ solves the integral as an inference problem, modelling the likelihood function with a probabilistic model (typically a Gaussian process (GP)). Gunter et al. [37] proposed Warped sequential active Bayesian integration (WSABI), which adopts active learning to select samples upon uncertainty over the integrand. WSABI showed that BQ with expensive GP calculations could achieve faster convergence in wall time than cheap MC samplers. Wagstaff et al. [78] introduced batch WSABI, achieving even faster calculation via parallel computing and became the fastest BQ model to date. We improve upon these existing works for a large-scale batch case.

2 Background

Vanilla Bayesian quadrature While BQ in general is the method for the integration, the functional approximation nature permits solving the following integral Z and obtaining the surrogate function of posterior $p(x)$ simultaneously in the Bayesian inference context:

$$p(x) = \frac{\ell_{\text{true}}(x)\pi(x)}{Z} = \frac{\ell_{\text{true}}(x)\pi(x)}{\int \ell_{\text{true}}(x)\pi(x)\,\mathrm{d}x}, \tag{1}$$

where both $\ell_{\text{true}}(x)$ (e.g. a likelihood) and $\pi(x)$ (e.g. a prior) are non-negative, and $x \in \mathbb{R}^d$ is a sample, and is sampled from prior $x \sim \pi(x)$. BQ solves the above integral as an inference problem, modelling a likelihood function $\ell(x)$ by a GP in order to construct a surrogate model of the expensive true likelihood $\ell_{\text{true}}(x)$. The surrogate likelihood function $\ell(x)$ is modelled:

$$\ell \,|\, \mathbf{y} \sim \mathcal{GP}(\ell; m_{\mathbf{y}}, C_{\mathbf{y}}), \tag{2a}$$

$$m_{\mathbf{y}}(x) = K(x, \mathbf{X})K(\mathbf{X}, \mathbf{X})^{-1}\mathbf{y}, \tag{2b}$$

$$C_{\mathbf{y}}(x, x') = K(x, x') - K(x, \mathbf{X})K(\mathbf{X}, \mathbf{X})^{-1}K(\mathbf{X}, x'), \tag{2c}$$

where $\mathbf{X} \in \mathbb{R}^{n \times d}$ is the matrix of observed samples, $\mathbf{y} \in \mathbb{R}^n$ is the observed true likelihood values, K is the kernel. [3] Due to linearity, the mean and variance of the integrals are simply

$$\mathbb{E}[Z \,|\, \mathbf{y}] = \int m_{\mathbf{y}}(x)\pi(x)\,\mathrm{d}x, \tag{3a}$$

$$\mathbb{V}\mathrm{ar}[Z \,|\, \mathbf{y}] = \iint C_{\mathbf{y}}(x, x')\pi(x)\pi(x')\,\mathrm{d}x\,\mathrm{d}x'. \tag{3b}$$

In particular, (3) becomes analytic when $\pi(x)$ is Gaussian and K is squared exponential kernel, $K(\mathbf{X}, x) = v\sqrt{|2\pi\mathbf{W}|}\mathcal{N}(\mathbf{X}; x, \mathbf{W})$, where v is kernel variance and $\mathbf{W}$ is the diagonal covariance matrix whose diagonal elements are the lengthscales of each dimension. Since both the mean and variance of the integrals can be calculated analytically, posterior and model evidence can be obtained simultaneously. Note that non-Gaussian prior and kernel are also possible to be chosen for modelling via kernel recombination (see Supplementary). Still, we use this combination throughout this paper for simplicity.

[3] In GP modelling, the GP likelihood function is modelled as $\mathcal{GP}(0, K)$, and (2) is the resulting posterior GP. Throughout the paper, we refer to a symmetric positive semi-definite kernel just as a kernel. The notations $\sim$ and $|$ refer to being sampled from and being conditioned, respectively.

Warped Bayesian quadrature (WSABI) WSABI with linearisation approximation (WSABI-L) adopts the square-root warping GP for non-negativity with linearisation approximation of the transform $\tilde{\ell} \mapsto \ell = \alpha + \frac{1}{2}\tilde{\ell}^2$. [4] The square-root GP is defined as $\tilde{\ell} \sim \mathcal{GP}(\tilde{\ell}; \tilde{m}_{\mathbf{y}}, \tilde{C}_{\mathbf{y}})$, and we have the following linear approximation:

$$\ell \,|\, \mathbf{y} \sim \mathcal{GP}(\ell; m^L_{\mathbf{y}}, C^L_{\mathbf{y}}), \tag{4a}$$

$$m^L_{\mathbf{y}}(x) = \alpha + \frac{1}{2}\tilde{m}_{\mathbf{y}}(x)^2, \tag{4b}$$

$$C^L_{\mathbf{y}}(x, x') = \tilde{m}_{\mathbf{y}}(x)\tilde{C}_{\mathbf{y}}(x, x')\tilde{m}_{\mathbf{y}}(x'). \tag{4c}$$

Gaussianity implies the model evidence Z and posterior $p(x)$ remain analytical (see Supplementary).

Kernel quadrature in general kernel quadrature (KQ) is the group of numerical integration rules for calculating the integral of function classes that form the reproducing kernel Hilbert space (RKHS). With a KQ rule $Q_{\mathbf{w},\mathbf{X}}$ given by weights $\mathbf{w} = (w_i)_{i=1}^n$ and points $\mathbf{X} = (x_i)_{i=1}^n$, we approximate the integral by the weighted sum

$$Q_{\mathbf{w},\mathbf{X}}(h) := \sum_{i=1}^n w_i h(x_i) \approx \int h(x)\pi(x)\,\mathrm{d}x, \tag{5}$$

where h is a function of RKHS $\mathcal{H}$ associated with the kernel K. We define its worst-case error by $\mathrm{wce}(Q_{\mathbf{w},\mathbf{X}}) := \sup_{\|h\|_{\mathcal{H}} \leq 1} |Q_{\mathbf{w},\mathbf{X}}(h) - \int h(x)\pi(x)\,\mathrm{d}x|$. Surprisingly, it is shown in [48] that we have

$$\mathbb{V}\mathrm{ar}[Z \,|\, \mathbf{y}] = \inf_{\mathbf{w}} \mathrm{wce}(Q_{\mathbf{w},\mathbf{x}})^2. \tag{6}$$

Thus, the point configuration in KQ with a small worst-case error gives a good way to select points to reduce the integral variance in Bayesian quadrature.

Random Convex Hull Quadrature (RCHQ) Recall from (5) and (6) that we wish to approximate the integral of a function h in the current RKHS. First, we prepare $n-1$ test functions $\varphi_1, \ldots, \varphi_{n-1}$ based on M sample points using the Nyström approximation of the kernel: $\varphi_i(x) := u_i^\top K(\mathbf{X}_{\mathrm{nys}}, x)$, where $u_i \in \mathbb{R}^M$ is the i-th eigenvector of $K(\mathbf{X}_{\mathrm{nys}}, \mathbf{X}_{\mathrm{nys}})$. If we let λ_i be the i-th eigenvalue of the same matrix, the following gives a practical approximation [55]:

$$K_0(x, y) := \sum_{i=1}^{n-1} \lambda_i^{-1} \varphi_i(x)\varphi_i(y). \tag{7}$$

Next, we consider extracting a weighted set of n points $(\mathbf{w}_{\mathrm{quad}}, \mathbf{X}_{\mathrm{quad}})$ from a set of N points $\mathbf{X}_{\mathrm{rec}}$ with positive weights $\mathbf{w}_{\mathrm{rec}}$. We do it by the so-called kernel recombination algorithm [59, 75], so that the measure induced by $(\mathbf{w}_{\mathrm{quad}}, \mathbf{X}_{\mathrm{quad}})$ exactly integrates the above test functions $\varphi_1, \ldots, \varphi_{n-1}$ with respect to the measure given by $(\mathbf{w}_{\mathrm{rec}}, \mathbf{X}_{\mathrm{rec}})$ [44].

In the actual implementation of multidimensional case, we execute the kernel recombination not by the algorithm [75] with the best known computational complexity $\mathcal{O}(C_\varphi N + n^3 \log(N/n))$ (where C_φ is the cost of evaluating $(\varphi_i)_{i=1}^{n-1}$ at a point), but the one of [59] using an LP solver (Gurobi [38] for this time) with empirically faster computational time. We also adopt the randomized SVD [40] for the Nyström approximation, so we have a computational time empirically faster than $\mathcal{O}(NM + M^2 \log n + Mn^2 \log(N/n))$ [44] in practice.

3 Related works

Bayesian inference for intractable likelihood Inference with intractable likelihoods is a long-standing problem, and a plethora of methods have been proposed. Most infer posterior and evidence separately, and hence are not our fair competitors, as solving both is more challenging. For posterior inference, *Markov Chain Monte Carlo* [62, 43], particularly *Hamilton Monte Carlo* [47], is the gold standard. In a *Likelihood-free inference* context, kernel density estimation (KDE) with Bayesian

[4] $\alpha := 0.8 \times \min(\mathbf{y})$. See Supplementary for the details on α

optimisation [39] and neural networks [36] surrogates are proposed for simulation-based inference [25]. In a *Bayesian coreset* context, scalable Bayesian inference [60], sparse variational inference [19, 20], active learning [68] have been proposed for large-scale dataset inference. However, all of the above only calculate posteriors, not model evidences. For evidence inference, *Annealed Importance Sampling* [64], and *Bridge Sampling* [9], Sequential Monte Carlo (SMC) [27] are popular, but *only* estimate evidence, not the posterior.

Bayesian quadrature The early works on BQ, which directly replaced the likelihood function with a GP [65, 66, 69], did not explicitly handle non-negative integrand constraints. Osborne et al. [67] introduced logarithmic warped GP to handle the non-negativity of the integrand, and introduced active learning for BQ, a method that selects samples based on where the variance of the integral will be minimised. Gunter et al. [37] introduced square-root GP to make the integrand function closed-form and to speed up the computation. Furthermore, they accelerated active learning by changing the objective from the variance of the integral $\mathbb{V}\mathrm{ar}[Z|\mathbf{y}]$ to simply the variance of the integrand $\mathbb{V}\mathrm{ar}[\ell(x)\pi(x)]$. Wagstaff et al. [78] introduced the first batch BQ. Chai et al. [22] generalised warped GP for BQ, and proposed probit-transformation. BQ has been extended to machine learning tasks (model selection [21], manifold learning [30], kernel learning [41]) with new acquisition function (AF) designed for each purpose. For posterior inference, VBMC [1] has pioneered that BQ can infer posterior and evidence in one go via variational inference, and [2] has extended it to the noisy likelihood case. This approach is an order of magnitude more expensive than the WSABI approach because it essentially involves BQ inside of an optimisation loop for variational inference. This paper employs WSABI-L and its AF for simplicity and faster computation. Still, our approach is a natural extension of BQ methods and compatible with the above advances. (e.g. changing the RCHQ kernel into the prospective uncertainty sampling AF for VMBC)

Kernel quadrature There are a number of KQ algorithms from herding/optimisation [23, 6, 48] to random sampling [5, 8]. In the context of BQ, Frank-Wolfe Bayesian quadrature (FWBQ) [13] using kernel herding has been proposed. This method proposes to do BQ with the points given by herding, but the guaranteed exponential convergence $\mathcal{O}(\exp(-cn))$ is limited to finite-dimensional RKHS, which is not the case in our setting. For general kernels, the convergence rate drops to $\mathcal{O}(1/\sqrt{n})$ [6]. Recently, a random sampling method with a good convergence rate was proposed for infinite-dimensional RKHS [44]. Based on Carathéodory's theorem for the convex hull, it efficiently calculates a reduced measure for a larger empirical measure. In this paper, we call it *random convex hull quadrature* (RCHQ) and use it in combination with the BQ methods. (See Supplementary)

4 Proposed Method: BASQ

4.1 General idea

We now introduce our algorithm, named *Bayesian alternately subsampled quadrature* (BASQ).

Kernel recombination for batch selection Batch WSABI [78] selects batch samples based on the AF, taking samples having the maximum AF values greedily via gradient-based optimisation with multiple starting points. The computational cost of this sampling scheme increases exponentially with the batch size and/or the problem dimension. Moreover, this method is often only locally optimal [79]. We adopt a scalable, gradient-free optimisation via KQ algorithm based on kernel recombination [44]. Surprisingly, Huszár et al. [48] pointed out the equivalence between BQ and KQ with optimised weights. KQ can select n samples from the N candidate samples $\mathbf{X}_{\text{rec}}$ to efficiently reduce the worst-case error. The problem in batch BQ is selecting n samples from the probability measure $\pi(x)$ that minimises integral variance. When subsampling N candidate samples $\mathbf{X}_{\text{rec}} \sim \pi(x)$, we can regard this samples $\mathbf{X}_{\text{rec}}$ as an empirical measure $\pi_{\text{emp}}(x)$ approximating the true probability measure $\pi(x)$ if $n \ll N$. Therefore, applying KQ to select n samples that can minimise $\mathbb{V}\mathrm{ar}[\ell(x)\pi_{\text{emp}}(x)]$ is equivalent to selecting n batch samples for batch BQ. As more observations make surrogate model $\ell(x)$ more accurate, the empirical integrand model $\ell(x)\pi_{\text{emp}}(x)$ approaches to the true integrand model $\ell_{\text{true}}(x)\pi(x)$. This subsampling scheme allows us to apply any KQ methods for batch BQ. However, such a dual quadrature scheme tends to be computationally demanding. Hayakawa et al. [44] proposed an efficient algorithm based on kernel recombination, RCHQ, which automatically

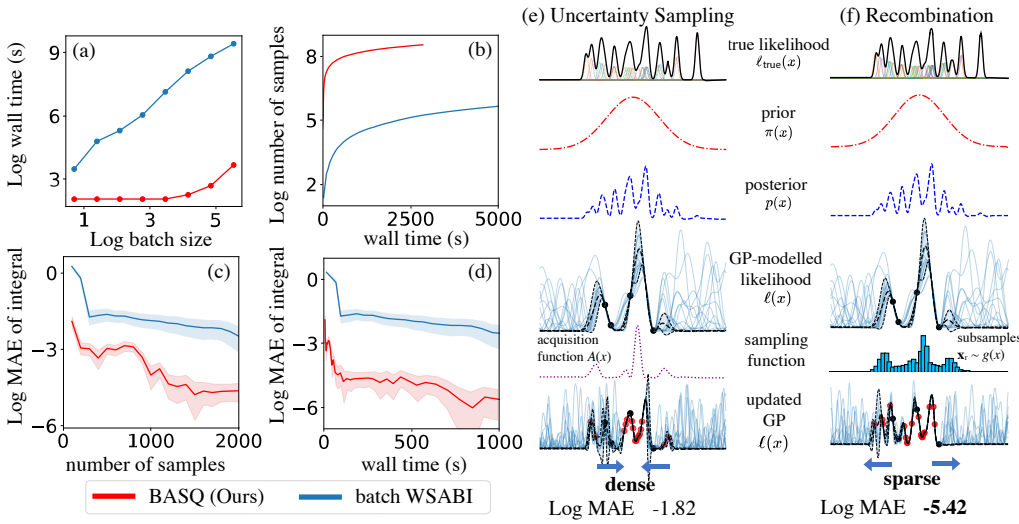


Figure 1: Performance comparison of our algorithm BASQ against batch WSABI [78]. All evaluation was performed with the likelihood of a mixture of N-dimensional Gaussians. (a), (b), (c), (d) 10-dimensional Gaussians, (e), (f) univariate Gaussians.

returns a sparse set of n samples based on Carathéodory's theorem. The computational cost of batch size n for our algorithm, BASQ, is lower than $\mathcal{O}(NM + M^2 \log n + Mn^2 \log(N/n))$ [44].

Alternately subsampling The performance of RCHQ relies upon the quality of a predefined kernel. Thus, we add BQ elements to KQ in return; making RCHQ an online algorithm. Throughout the sequential batch update, we pass the BQ-updated kernels to RCHQ.[5] This enables RCHQ to exploit the function shape information to select the best batch samples for minimising $\mathbb{V}\mathrm{ar}[\ell(x)\pi_{\mathrm{emp}}(x)]$. This corresponds to the batch maximisation of the model evidence for $\ell(x)$. Then, BQ optimises the hyperparameters based on samples from true likelihood $\ell_{\mathrm{true}}(x)$, which corresponds to an optimised kernel preset for RCHQ in the next round. These alternate updates characterise our algorithm, BASQ. (See Supplementary)

Importance sampling for uncertainty We added one more BQ element to RCHQ; *uncertainty sampling*. RCHQ relies upon the quality of subsamples from the prior distribution. However, sharp, multimodal, or high-dimensional likelihoods make it challenging to find prior subsamples that overlap over the likelihood distribution. This typically happens in Bayesian inference when the likelihood is expressed as the product of Gaussians, which gives rise to a very sharp likelihood peak. The likelihood of big data tends to become multimodal [70]. Therefore, we adopt *importance sampling*, gravitating subsamples toward the meaningful region, and correct the bias via its weights. We propose a mixture of prior and an AF as a guiding *proposal distribution*. The prior encourages global (rather than just local) optimisation, and the AF encourages sampling from uncertain areas for faster convergence. However, sampling from AF is expensive. We derive an efficient sparse Gaussian mixture sampler. Moreover, introducing square-root warping [37] enables the sampling distribution to factorise, yielding faster sampling.

Summary of contribution We summarised the key differences between batch WSABI and BASQ in Figure 1.[6] (a) shows that BASQ is more scalable in batch size. (b) clarifies that BASQ can sample

[5]For updating kernel, the kernel to be updated is $C_{\mathbf{y}}$, not the kernel K. The kernel K just corresponds to the *prior* belief in the distribution of ℓ, so once we have observed the samples $\mathbf{X}$ (and $\mathbf{y}$), the variance to be minimised becomes $C_{\mathbf{y}}$.

[6]We randomly varied the number of components between 10 and 15, setting their variance uniformly at random between 1 and 4, and setting their means uniformly at random within the box bounded by [-3,3] in all dimensions. The weights of Gaussians were randomly generated from a uniform distribution, but set to be one after integration. mean absolute error (MAE) was adopted for the evaluation of integral estimation.

Table 1: BASQ algorithm

Algorithm 1: Bayesian Alternately Subsampled Quadrature (BASQ)

Notation: x_{init}: initial guess, k: a convergence criterion,
n: batch size, M: Nyström sample size, N: recombination sample size,
ℓ: GP-modelled likelihood, ℓ_{true}: true likelihood,
$\pi(x)$: prior, $p(x)$: posterior, $A(x)$: AF, K: kernel,
S_{nys}, S_{rec}: samplers for Nyström and recombination, respectively

Input: prior $\pi(x)$, true likelihood function ℓ_{true}
Output: posterior $p(x)$, the mean and variance of model evidence $\mathbb{E}[Z \mid \mathbf{y}], \text{var}[Z \mid \mathbf{y}]$

1: $C_{\mathbf{y}}^L, A(x) \leftarrow \text{InitialiseGPs}(x_{\text{init}})$ # Initialise GPs with initial guess
2: $S_{\text{nys}}, S_{\text{rec}} \leftarrow \text{SetSampler}(A(x), \pi(x))$ # Set samplers
3: **while** $\text{var}[Z \mid \mathbf{y}] > k$:
4: $\quad \mathbf{X}_{\text{nys}} \sim S_{\text{nys}}(M)$ # Samples M points for test function φ
5: $\quad (\mathbf{w}_{\text{rec}}, \mathbf{X}_{\text{rec}}) \sim S_{\text{rec}}(N)$ # Samples N points for recombination
6: $\quad \varphi_1, ..., \varphi_{n-1} \leftarrow \text{Nyström}(\mathbf{X}_{\text{nys}}, C_{\mathbf{y}}^L)$ # Define $n-1$ test functions
7: $\quad$ **solve a kernel recombination problem**
8: $\quad\quad$ Find an n-point subset $\mathbf{X}_{\text{quad}} \subset \mathbf{X}_{\text{rec}}$ and $\mathbf{w}_{\text{quad}} \geq \mathbf{0}$
9: $\quad\quad$ s.t. $\mathbf{w}_{\text{quad}}^\top \varphi_i(\mathbf{X}_{\text{quad}}) = \mathbf{w}_{\text{rec}}^\top \varphi_i(\mathbf{X}_{\text{rec}})$, $\mathbf{w}_{\text{quad}}^\top \mathbf{1} = \mathbf{w}_{\text{rec}}^\top \mathbf{1}$
10: $\quad\quad$ **return** $\mathbf{X}_{\text{quad}}$ # The sparse set of n samples
11: $\quad \mathbf{y} = \text{Parallel}(\ell_{\text{true}}(\mathbf{X}_{\text{quad}}))$ # Parallel computing of likelihood
12: $\quad K \leftarrow \text{Update}(\mathbf{X}_{\text{quad}}, \mathbf{y})$ # Train GPs
13: $\quad C_{\mathbf{y}}^L, A(x) \leftarrow \text{OptHypersThenUpdate}(K)$ # Type II MLE optimisation
14: $\quad S_{\text{nys}}, S_{\text{rec}} \leftarrow \text{ResetSampler}(A(x), \pi(x))$ # Reset samplers with the updated $A(x)$
15: $\quad \mathbb{E}[Z \mid \mathbf{y}], \text{var}[Z \mid \mathbf{y}] \leftarrow \text{BayesQuad}(m_{\mathbf{y}}^L, C_{\mathbf{y}}^L, \pi(x))$ # Calculate via Eqs. (3) and (4)
16: **return** $p(x), \mathbb{E}[Z \mid \mathbf{y}], \text{var}[Z \mid \mathbf{y}]$

10 to 100 times as many samples in the same time budget as WSABI, supported by the efficient sampler and RCHQ. (c) states the convergence rate of BASQ is faster than WSABI, regardless of computation time. (d) demonstrates the combined acceleration in wall time. While the batch WSABI reached 10^{-1} after 1,000 seconds passed, BASQ was within seconds. (e) and (f), visualised how RCHQ selects sparser samples than batch WSABI. This clearly explains that gradient-free kernel recombination is better in finding the global optimal than multi-start optimisation. These results demonstrate that we were able to combine the merits of BQ and KQ (see Supplementary). We further tested with various synthetic datasets and real-world tasks in the fields of lithium-ion batteries and material science. Moreover, we mathematically analyse the convergence rate with proof in Section 6.

4.2 Algorithm

Table 1 illustrates the pseudo-code for our algorithm. Rows 4 - 10 correspond to RCHQ, and rows 11 - 15 correspond to BQ. We can use the variance of the integral $\mathbb{V}\text{ar}[Z|\mathbf{y}]$ as a convergence criterion. For hyperparameter optimisation, we adopt the type-II maximum likelihood estimation (MLE) to optimise hyperparameters via L-BFGS [18] for speed.

Importance sampling for uncertainty Lemma 1 in the supplementary proves the optimal upper bound of the proposal distribution $g(x) \approx \sqrt{K(x,x)}f(x) = \sqrt{C_{\mathbf{y}}^L(x,x)}f(x)$, where $f(x) := \pi(x)$. However, sampling from square-root variance is intractable, so we linearised to $g(x) \approx 0.5(1 + C_{\mathbf{y}}^L(x,x))f(x)$. To correct the linearisation error, the coefficient 0.5 was changed into the hyperparameter r, which is defined as follows:

$$g(x) = (1-r)f(x) + r\tilde{A}(x), \quad 0 \leq r \leq 1 \tag{8}$$

$$\tilde{A}(x) = \frac{C_{\mathbf{y}}^L(x,x)\pi(x)}{\int C_{\mathbf{y}}^L(x,x)\pi(x)dx}, \tag{9}$$

$$\text{w}_{\text{IS}}(x) = f(x)/g(x), \tag{10}$$

where w_{IS} is the weight of the importance sampling. While $r = 1$ becomes the pure uncertainty sampling, $r = 0$ is the vanilla MC sampler.

Efficient sampler Sampling from $A(x)$, a mixture of Gaussians, is expensive, as some mixture weights are negative, preventing the usual practice of weighted sampling from each Gaussian. As the warped kernel $C^L_{\mathbf{y}}(x, x)$ is also computationally expensive, we adopt a *factorisation trick*:

$$Z = \int \ell(x)\pi(x)\,\mathrm{d}x = \alpha + \frac{1}{2}\int |\tilde{\ell}(x)|^2\pi(x)\,\mathrm{d}x \approx \alpha + \frac{1}{2}\int |\tilde{\ell}(x)|f(x)\,\mathrm{d}x, \tag{11}$$

where we have changed the distribution of interest to $f(x) = |\tilde{m_{\mathbf{y}}}(x)|\pi(x)$. This is doubly beneficial. Firstly, the distribution of interest $f(x)$ will be updated over iterations. The previous $f(x) = \pi(x)$ means the subsample distribution eventually obeys prior, which is disadvantageous if the prior does not overlap the likelihood sufficiently. On the contrary, the new $f(x)$ narrows its region via $|\tilde{m_{\mathbf{y}}}(x)|$. Secondly, the likelihood function changed to $|\tilde{\ell}(x)|$, thus the kernels shared with RCHQ changed into cheap warped kernel $\tilde{C}_{\mathbf{y}}(x, x)$. This reduces the computational cost of RCHQ, and the sampling cost of $A(x)$. Now $A(x) = \tilde{C}_{\mathbf{y}}(x)\pi(x)$, which is also a Gaussian mixture, but the number of components is significantly lower than the original AF (9). As $\tilde{C}_{\mathbf{y}}(x)$ is positive, the positive weights of the Gaussian mixture should cover the negative components. Interestingly, in many cases, the positive weights vary exponentially, which means that limited number of components dominate the functional shape. Thus, we can ignore the trivial components for sampling.[7] Then we adopt SMC [53] to sample $A(x)$. We have a highly-calibrated proposal distribution of sparse Gaussian mixture, leading to efficient resampling from real $A(x)$ (see Supplementary).

Variants of proposal distribution Although (8) has mathematical motivation, sometimes we wish to incorporate prior information not included in the above procedure. We propose two additional "biased" proposal distributions. The first case is where we know both the maximum likelihood points and the likelihood's unimodality. This is typical in Bayesian inference because we can obtain (sub-)maximum points via a maximum a posteriori probability (MAP) estimate. In this case, we know exploring around the perfect initial guess is optimal rather than unnecessarily exploring an uncertain region. Thus, we introduce the initial guess believer (IGB) proposal distribution, $g_{\mathrm{IGB}}(x)$. This is written as $g_{\mathrm{IGB}}(x) = (1 - r)\pi(x) + r\sum_{i=1} w_{i,\mathrm{IGB}}\mathcal{N}(x; X_i, \mathbf{W})$, where $w_{i,\mathrm{IGB}} = \{0 \text{ if } y_i \leq 0, \text{ else } 1\}$, $X_i \in \mathbf{X}$. This means exploring only the vicinity of the observed data $\mathbf{X}$. The second case is where we know the likelihood is multimodal. In this case, determining all peak positions is most beneficial. Thus more explorative distribution is preferred. As such, we introduce the uncertainty believer (UB) proposal distribution, $g_{\mathrm{UB}}(x)$. This is written as $g_{\mathrm{UB}}(x) = A(x)$, meaning pure uncertainty sampling. To contrast the above two, we term the proposal distribution in Eq. (8) as integral variance reduction (IVR) $g_{\mathrm{IVR}}(x)$.

5 Experiments

Given our new model BASQ, with three variants of the proposal distribution, IVR, IGB, and UB, we now test for speed against MC samplers and batch WSABI. We compared with three NS methods [71, 29, 46, 14, 15], coded with [72, 17]. According to the review [16], MLFriends is the state-of-the-art NS sampler to date. The code is implemented based on [77, 34, 42, 76, 38, 31, 10, 7], and code around kernel recombination [24, 44] with additional modification. All experiments on synthetic datasets were averaged over 10 repeats, computed in parallel with multicore CPUs, without GPU for fair comparison.[8] The posterior distribution of NS was estimated via KDE with weighted samples [33]. For maximum speed performance, batch size was optimised for each method in each dataset, in fairness to the competitors. Batch WSABI needs to optimise batch size to balance the likelihood query cost and sampling cost, because sampling cost increases rapidly with batch size, as shown in Figure 1(a). Therefore, it has an optimal batch size for faster convergence. By wall time cost, we exclude the cost of integrand evaluation; that is, the wall time cost is the overhead cost of batch evaluation. Details can be found in the Supplementary.

[7] Negative elements in the matrices only exist in $K(\mathbf{X}, \mathbf{X})^{-1}$, which can be drawn from the memory of the GP regression model without additional calculation. The number of positive components is half of the matrix on average, resulting in $\mathcal{O}(n^2/2)$. Then, taking the threshold via the inverse of the recombination sample size N, the number of components becomes $n_{\mathrm{comp}} \ll n^2$, resulting in sampling complexity $\mathcal{O}(n^2/2 + n_{\mathrm{comp}}N)$.

[8] Performed on MacBook Pro 2019, 2.4 GHz 8-Core Intel Core i9, 64 GB 2667 MHz DDR4

5.1 Synthetic problems

We evaluate all methods on three synthetic problems. The goal is to estimate the integral and posterior of the likelihood modelled with the highly multimodal functions. Prior was set to a two-dimensional multivariate normal distribution, with a zero mean vector, and covariance whose diagonal elements are 2. The optimised batch sizes for each methods are BASQ: 100, batch WSABI: 16. The synthetic likelihood functions are cheap (0.5 ms on average). This is advantageous setting for NS: Within 10 seconds, the batch WSABI, BASQ, and NS collected 32, 600, 23225 samples, respectively. As for the metrics, posterior estimation was tested with Kullback-Leibler (KL) upon random 10,000 samples from true posterior. Evidence was evaluated with MAE, and ground truth was derived analytically.

Likelihood functions *Branin-Hoo* [49] is 8 modal function in two-dimensional space. *Ackley* [73] is a highly multimodal function with point symmetric periodical peaks in two-dimensional space. *Oscillatory function* [32] is a highly multimodal function with reflection symmetric periodical peaks of highly-correlated ellipsoids in two-dimensional space.

5.2 Real-world dataset

We consider three real-world applications with expensive likelihoods, which are simulator-based and hierarchical GP. We adopted the empirical metric due to no ground truth. For the posterior, we can calculate the true conditional posterior distribution along the line passing through ground truth parameter points. Then, evaluate the posterior with root mean squared error (RMSE) against 50 test samples for each dimension. For integral, we compare the model evidence itself. Expensive likelihoods makes the sample size per wall time amongst the methods no significant difference, whereas rejection sampling based NS dismiss more than 50% of queried samples. The batch sizes are BASQ: 32, batch WSABI: 8. (see Supplementary)

Parameter estimation of the lithium-ion battery simulator : The simulator is the SPMe [61], [9] estimating 3 simulation parameters at a given time-series voltage-current signal (the diffusivity of lithium-ion on the anode and cathode, and the experimental noise variance). Prior is modified to log multivariate normal distribution from [4]. Each query takes 1.2 seconds on average.

Parameter estimation of the phase-field model : The simulator is the phase-field model [61], [10] estimating 4 simulation parameters at given time-series two-dimensional morphological image (temperature, interaction parameter, Bohr magneton coefficient, and gradient energy coefficient). Prior is a log multivariate normal distribution. Each query takes 7.4 seconds on average.

Hyperparameter marginalisation of hierarchical GP model The hierarchical GP model was designed for analysing the large-scale battery time-series dataset from solar off-grid system field data [3].[8] For fast estimation of parameters in each GP, the recursive technique [74] is adopted. The task is to marginalise 5 GP hyperparameters at given hyperprior, which is modified to log multivariate normal distribution from [3]. Each query takes 1.1 seconds on average.

5.3 Results

We find BASQ consistently delivers strong empirical performance, as shown in Figure 2. On all benchmark problems, BASQ-IVR, IGB, or UB outperform baseline methods except in the battery simulator evidence estimation. The very low-dimensional and sharp unimodal nature of this likelihood could be advantageous for biased greedy batch WSABI, as IGB superiority supports this viewpoint. This suggests that BASQ could be a generally fast Bayesian solver as far as we investigated. In the multimodal setting of the synthetic dataset, BASQ-UB outperforms, whereas IVR does in a simulator-based likelihoods. When comparing each proposal distribution, BASQ-IVR was the performant. Our results support the general use of IVR, or UB if the likelihood is known to be highly multimodal.

[9] SPMe code used was translated into Python from MATLAB [11, 12]. This open-source code is published under the BSD 3-clause Licence. See more information on [11]

[10] Code used was from [54, 3]. All rights of the code are reserved by the authors. Thus, we do not redistribute the original code.

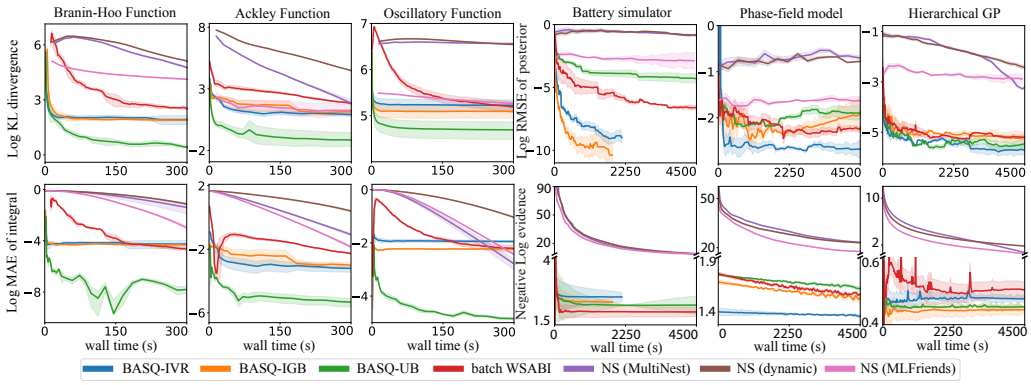


Figure 2: Time in seconds vs. KL divergence for posterior, and MAE for evidence in the synthetic datasets. Time in seconds vs. RMSE for posterior and evidence itself in the real-world dataset.

6 Convergence analysis

We analysed the convergence over single iteration on a simplified version of BASQ, which assumes the BQ is modelled with vanilla BQ, without batch and hyperparameter updates. Note that the kernel K on $\mathbb{R}^d$ in this section refers to the given covariance kernel of GP at each step. We discuss the convergence of BASQ in one iteration. We consider the importance sampling: Let f be a probability density on $\mathbb{R}^d$ and g be another density such that $f = \lambda g$ with a nonnegative function λ. Let us call such a pair, (f, g), a density pair with weight λ.

We approximate the kernel K with $K_0 = \sum_{i=1}^{n-1} c_i \varphi_i(x)\varphi_i(y)$. In general, we can apply the kernel recombination algorithm [59, 75] with the weighted sample $(\mathbf{w}_{\text{rec}}, \mathbf{X}_{\text{rec}})$ to obtain a weighted point set $(\mathbf{w}_{\text{quad}}, \mathbf{X}_{\text{quad}})$ of size n satisfying $\mathbf{w}_{\text{quad}}^\top \varphi_i(\mathbf{X}_{\text{quad}}) = \mathbf{w}_{\text{rec}}^\top \varphi_i(\mathbf{X}_{\text{rec}})$ $(i = 1, \dots, n-1)$ and $\mathbf{w}_{\text{quad}}^\top \mathbf{1} = \mathbf{w}_{\text{rec}}^\top \mathbf{1}$. By modifying the kernel recombination algorithm, we can require $\mathbf{w}_{\text{quad}}^\top k_1^{1/2}(\mathbf{X}_{\text{quad}}) \leq \mathbf{w}_{\text{rec}}^\top k_1^{1/2}(\mathbf{X}_{\text{rec}})$, where $k_1^{1/2}(x) := \sqrt{K(x,x) - K_0(x,x)}$ [44]. We call such $(\mathbf{w}_{\text{quad}}, \mathbf{X}_{\text{quad}})$ a *proper* kernel recombination of $(\mathbf{w}_{\text{rec}}, \mathbf{X}_{\text{rec}})$ with K_0.[11] We have the following guarantee (proved in Supplementary):

Theorem 1. *Suppose $\int \sqrt{K(x,x)} f(x)\,\mathrm{d}x < \infty$, $\ell \sim \mathcal{GP}(m, K)$, and we are given an $(n-1)$-dimensional kernel K_0 such that $K_1 := K - K_0$ is also a kernel. Let (f, g) be a density pair with weight λ. Let $\mathbf{X}_{\text{rec}}$ be an N-point independent sample from g and $\mathbf{w}_{\text{rec}} := \lambda(\mathbf{X}_{\text{rec}})$. Then, if $(\mathbf{w}_{\text{quad}}, \mathbf{X}_{\text{quad}})$ is a proper kernel recombination of $(\mathbf{w}_{\text{rec}}, \mathbf{X}_{\text{rec}})$ for K_0, it satisfies*

$$\mathbb{E}_{\mathbf{x}_{\text{rec}}}\left[\sqrt{\operatorname{var}[Z_f \mid \mathbf{x}_{\text{quad}}]}\right] \leq 2\left(\int K_1(x,x) f(x)\,\mathrm{d}x\right)^{1/2} + \sqrt{\frac{C_{K,f,g}}{N}}, \tag{12}$$

where $Z_f := \int \ell(x) f(x)\,\mathrm{d}x$ and $C_{K,f,g} := \int K(x,x)\lambda(x) f(x)\,\mathrm{d}x - \iint K(x,y) f(x) f(y)\,\mathrm{d}x\,\mathrm{d}y$.

The above approximation has one source of randomness which stems from sampling N points $\mathbf{x}_{\text{rec}}$ from g. One can also apply this estimate with a random kernel and thereby introduce another source of randomness. In particular, when we use the Nyström approximation for K_0 (that ensures K_1 is a kernel [44]), then one can show that $\int K_1(x,x) f(x)\,\mathrm{d}x$ can be bounded by

$$\int K_1(x,x) f(x)\,\mathrm{d}x \leq n\sigma_n + \sum_{m=n+1}^{\infty} \sigma_m + \mathcal{O}_p\left(\frac{n K_{\max}}{\sqrt{M}}\right), \tag{13}$$

where σ_n is the n-th eigenvalue of the integral operator $L^2(f) \ni h \mapsto \int K(\cdot, y) h(y) f(y)\,\mathrm{d}x$, $K_{\max} := \sup_x K(x,x)$. However, note that unlike Eq. (12), this inequality only applies with high probability due to the randomness of K_0; see Supplementary for details.

[11] Note that the inequality constraint on the diagonal value here is only needed for theoretical guarantee, and skipping it does not reduce the empirical performance [44].

If, for example, K is a Gaussian kernel on $\mathbb{R}^d$ and f is a Gaussian distribution, we have $\sigma_n = \mathcal{O}(\exp(-cn^{1/d}))$ for some constant $c > 0$ (see Supplementary). So in (12) we also achieve an empirically exponential rate when $N \gg C_{K,f,g}$. RCHQ works well with a moderate M in practise. Note that unlike the previous analysis [50], we do not have to assume that the space is compact. [12]

7 Discussion

We introduced a batch BQ approach, BASQ, capable of simultaneous calculation of both model evidence and posteriors. BASQ demonstrated faster convergence (in wall-clock time) on both synthetic and real-world datasets, when compared against existing BQ approaches and state-of-the-art NS. Further, mathematical analysis shows the possibility to converge exponentially-fast under natural assumptions. As the BASQ framework is general-purpose, this can be applied to other active learning GP-based applications, such as Bayesian optimisation [52], dynamic optimisation like control [26], and probabilistic numerics like ODE solvers [45]. Although it scales to the number of data seen in large-scale GP experiments, practical BASQ usage is limited to fewer than 16 dimensions (similar to many GP-based algorithms). However, RCHQ is agnostic to the input space, allowing quadrature in manifold space. An appropriate latent variable warped GP modelling, such as GPLVM [58], could pave the way to high dimensional quadrature in future work. In addition, while WSABI modelling limits the kernel to a squared exponential kernel, RCHQ allows to adopt other kernels or priors without a bespoke modelling BQ models. (See Supplementary). As for the mathematical proof, we do not incorporate batch and hyperparameter updates, which should be addressed in future work. The generality of our theoretical guarantee with respect to kernel and distribution should be useful for extending the analysis to the whole algorithm.

Acknowledgments and Disclosure of Funding

We thank Saad Hamid and Xingchen Wan for the insightful discussion of Bayesian quadrature, and Antti Aitio and David Howey for fruitful discussion of Bayesian inference for battery analytics and for sharing his codes on the single particle model with electrolyte dynamics, and hierarchical GPs. We would like to thank Binxin Ru, Michael Cohen, Samuel Daulton, Ondrej Bajgar, and anonymous reviewers for their helpful comments about improving the paper. Masaki Adachi was supported by the Clarendon Fund, the Oxford Kobe Scholarship, the Watanabe Foundation, the British Council Japan Association, and Toyota Motor Corporation. Satoshi Hayakawa was supported by the Clarendon Fund, the Oxford Kobe Scholarship, and the Toyota Riken Overseas Scholarship. Harald Oberhauser was supported by the DataSig Program [EP/S026347/1] and the Hong Kong Innovation and Technology Commission (InnoHK Project CIMDA). Martin Jørgensen was supported by the Carlsberg Foundation.

References

[1] Luigi Acerbi. Variational bayesian monte carlo. *Advances in Neural Information Processing Systems*, 31, 2018.

[2] Luigi Acerbi. Variational bayesian monte carlo with noisy likelihoods. *Advances in Neural Information Processing Systems*, 33:8211–8222, 2020.

[3] A. Aitio and D. A. Howey. Predicting battery end of life from solar off-grid system field data using machine learning. *Joule*, 5(12):3204–3220, 2021.

[4] A. Aitio, S. G. Marquis, P. Ascencio, and D. A. Howey. Bayesian parameter estimation applied to the Li-ion battery single particle model with electrolyte dynamics. *IFAC-PapersOnLine*, 53(2):12497–12504, 2020.

[5] F. Bach. On the equivalence between kernel quadrature rules and random feature expansions. *Journal of Machine Learning Research*, 18:714, 2017.

[12] The rate $N^{-1/2}$ for the expected integral variance on non-compact sets can also be found in [35] [Corollary 2.8]. In the case of Gaussian integration distribution and the Gaussian kernel exponential rates of convergence of the BQ integral variance can be found [8] [Theorem 1], [57] [Theorem 4.1], [56] [Theorem 3.2], [51] [Theorems 2.5 and 2.10]

[6] F. Bach, S. Lacoste-Julien, and G. Obozinski. On the equivalence between herding and conditional gradient algorithms. In *International Conference on Machine Learning (ICML)*, 2012.

[7] M. Balandat, B. Karrer, D. Jiang, S. Daulton, B. Letham, A. G. Wilson, and E. Bakshy. BoTorch: a framework for efficient Monte-Carlo Bayesian optimization. *International Conference on Neural Information Processing Systems (NeurIPS)*, 33:21524–21538, 2020.

[8] A. Belhadji, R. Bardenet, and P. Chainais. Kernel quadrature with DPPs. In *International Conference on Neural Information Processing Systems (NeurIPS)*, 2019. URL: `https://proceedings.neurips.cc/paper/2019/file/7012ef0335aa2adbab58bd6d0702ba41-Paper.pdf`.

[9] C. H. Bennett. Efficient estimation of free energy differences from Monte Carlo data. *Journal of Computational Physics*, 22:245 – 268, 1976.

[10] A. S. Berahas, J. Nocedal, and M. Takác. A multi-batch l-bfgs method for machine learning. *International Conference on Neural Information Processing Systems (NeurIPS)*, 29, 2016.

[11] A. M. Bizeray, J. Reniers, and D. A. Howey. Spectral Li-ion SPM. `"https://doi.org/10.5281/zenodo.212178"`, 2016.

[12] A. M. Bizeray, S. Zhao, S. R. Duncan, and D. A. Howey. Lithium-ion battery thermal-electrochemical model-based state estimation using orthogonal collocation and a modified extended Kalman filter. *Journal of Power Sources*, 296:400–412, 2015.

[13] F.-X. Briol, C. J. Oates, M. Girolami, and M. A. Osborne. Frank-Wolfe Bayesian quadrature: Probabilistic integration with theoretical guarantees. In *International Conference on Neural Information Processing Systems (NeurIPS)*, 2015. URL: `https://proceedings.neurips.cc/paper/2015/file/ba3866600c3540f67c1e9575e213be0a-Paper.pdf`.

[14] J. Buchner. A statistical test for nested sampling algorithms. *Statistics and Computing*, 26(1):383–392, 2016.

[15] J. Buchner. Collaborative nested sampling: Big data versus complex physical models. *Publications of the Astronomical Society of the Pacific*, 131(1004):108005, 2019.

[16] J. Buchner. Nested sampling methods. *arXiv preprint arXiv:2101.09675*, 2021.

[17] J. Buchner. UltraNest–a robust, general purpose Bayesian inference engine. *arXiv preprint arXiv:2101.09604*, 2021.

[18] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on scientific computing*, 16(5):1190–1208, 1995.

[19] T. Campbell and B. Beronov. Sparse variational inference: Bayesian coresets from scratch. *International Conference on Neural Information Processing Systems (NeurIPS)*, 32, 2019.

[20] T. Campbell and T. Broderick. Automated scalable Bayesian inference via Hilbert coresets. *The Journal of Machine Learning Research*, 20(1):551–588, 2019.

[21] H. Chai, J. F. Ton, M. A. Osborne, and R. Garnett. Automated model selection with Bayesian quadrature. In *International Conference on Machine Learning (ICML)*, volume 97, pages 931–940, 2019. URL: `https://proceedings.mlr.press/v97/chai19a.html`.

[22] H. R. Chai and R. Garnett. Improving quadrature for constrained integrands. In *In IEEE 16th International Conference on Data Mining (ICDM)*, page 1107, 2016. URL: `https://doi.org/10.1109/ICDM.2016.0144`.

[23] Y. Chen, M. Welling, and A. Smola. Super-samples from kernel herding. In *International Conference on Uncertainty in Artificial Intelligence (UAI)*, 2010.

[24] F. Cosentino, H. Oberhauser, and A. Abate. A randomized algorithm to reduce the support of discrete measures. *International Conference on Neural Information Processing Systems (NeurIPS)*, 33:15100–15110, 2020.

[25] K. Cranmer, J. Brehmer, and G. Louppe. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48):30055–30062, 2020.

[26] M. P. Deisenroth, C. E. Rasmussen, and J. Peters. Gaussian process dynamic programming. *Neurocomputing*, 72(7-9):1508–1524, 2009.

[27] X. Didelot, R. G. Everitt, A. M. Johansen, and D. J. Lawson. Likelihood-free estimation of model evidence. *Bayesian analysis*, 6(1):49–76, 2011.

[28] G. Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *Journal of Geophysical Research: Oceans*, 99(C5):10143–10162, 1994.

[29] F. Feroz, M. P. Hobson, and M. Bridges. MultiNest: an efficient and robust Bayesian inference tool for cosmology and particle physics. *Monthly Notices of the Royal Astronomical Society*, 398(4):1601–1614, 2009.

[30] C. Fröhlich, A. Gessner, P. Hennig, B. Schölkopf, and G. Arvanitidis. Bayesian quadrature on Riemannian data manifolds. In *International Conference on Machine Learning (ICML)*, 2021. URL: `https://icml.cc/virtual/2021/poster/9655`.

[31] J. Gardner, G. Pleiss, K. Q. Weinberger, D. Bindel, and A. G. Wilson. GPyTorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration. *International Conference on Neural Information Processing Systems (NeurIPS)*, 31, 2018.

[32] A. Genz. Testing multidimensional integration routines. In *In Proc. of international conference on Tools, methods and languages for scientific and engineering computation*, pages 81–94, 1984.

[33] F. J. G. Gisbert. Weighted samples, kernel density estimators and convergence. *Empirical Economics*, 28:335–351, 2003. URL: `https://doi.org/10.1007/s001810200134`.

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

[35] D.M. Gräf. *Efficient algorithms for the computation of optimal quadrature points on Riemannian manifolds*. PhD thesis, Chemnitz University of Technology, 2013.

[36] D. Greenberg, M. Nonnenmacher, and J. Macke. Automatic posterior transformation for likelihood-free inference. In *International Conference on Machine Learning (ICML)*, pages 2404–2414, 2019.

[37] T. Gunter, M. A. Osborne, R. Garnett, P. Hennig, and S. J. Roberts. Sampling for inference in probabilistic models with fast Bayesian quadrature. In *International Conference on Neural Information Processing Systems (NeurIPS)*, 2014. URL: `https://proceedings.neurips.cc/paper/2014/file/e94f63f579e05cb49c05c2d050ead9c0-Paper.pdf`.

[38] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2022. URL: `https://www.gurobi.com`.

[39] M. U. Gutmann, J. Corander, et al. Bayesian optimization for likelihood-free inference of simulator-based statistical models. *Journal of Machine Learning Research*, 2016.

[40] N. Halko, P. G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

[41] S. Hamid, S. Schulze, M. A. Osborne, and S. J. Roberts. Marginalising over stationary kernels with Bayesian quadrature. *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2021. URL: `https://proceedings.mlr.press/v151/hamid22a/hamid22a.pdf`.

[42] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. Kerkwijk, M. Brett, A. Haldane, J. F. Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, 2020. `doi:10.1038/s41586-020-2649-2`.

[43] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970. URL: `https://doi:10.1093/biomet/57.1.97`.

[44] S. Hayakawa, H. Oberhauser, and T. Lyons. Positively weighted kernel quadrature via subsampling. In *International Conference on Neural Information Processing Systems (NeurIPS)*, 2022. `doi:10.48550/arXiv.2107.09597`.

[45] P. Hennig, M. A. Osborne, and M. Girolami. Probabilistic numerics and uncertainty in computations. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 471(2179):20150142, 2015.

[46] E. Higson, W. Handley, M. Hobson, and A. Lasenby. Dynamic nested sampling: an improved algorithm for parameter estimation and evidence calculation. *Stat. Comput.*, 29:891–913, 2019. URL: https://doi.org/10.1007/s11222-018-9844-0.

[47] M. D. Hoffman and A. Gelman. The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(1):1593–1623, 2014.

[48] F. Huszár and D. Duvenaud. Optimally-weighted herding is Bayesian quadrature. In *International Conference on Uncertainty in Artificial Intelligence (UAI)*, 2012.

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

[50] M. Kanagawa and P. Hennig. Convergence guarantees for adaptive Bayesian quadrature methods. *Advances in Neural Information Processing Systems*, 32, 2019.

[51] T. Karvonen, C. Oates, and M. Girolami. Integration in reproducing kernel hilbert spaces of gaussian kernels. *Mathematics of Computation*, 90(331):2209–2233, 2021.

[52] T. Kathuria, A. Deshpande, and P. Kohli. Batched Gaussian process bandit optimization via determinantal point processes. *Advances in Neural Information Processing Systems*, 29, 2016.

[53] G. Kitagawa. A Monte Carlo filtering and smoothing method for non-Gaussian nonlinear state space models. In *Proceedings of the 2nd U.S.-Japan Joint Seminar on Statistical Time Series Analysis*, page 110, 1993. URL: https://www.ism.ac.jp/~kitagawa/1993_US-Japan.pdf.

[54] T. Koyama. *Computational Engineering for Materials Design, Computational Microstructures, New and Expanded Edition: Microstructure Formation Analysis by the Phase-Field Method*. Uchida Rokakuho, Tokyo, Japan, 2019.

[55] S. Kumar, M. Mohri, and A. Talwalkar. Sampling methods for the Nyström method. *The Journal of Machine Learning Research*, 13(1):981–1006, 2012.

[56] F. Kuo, I. Sloan, and H. Woźniakowski. Multivariate integration for analytic functions with gaussian kernels. *Mathematics of Computation*, 86(304):829–853, 2017.

[57] F. Y. Kuo and H. Woźniakowski. Gauss-Hermite quadratures for functions from Hilbert spaces with Gaussian reproducing kernels. *BIT Numerical Mathematics*, 52(2):425–436, 2012.

[58] N. Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. *Advances in neural information processing systems*, 16, 2003.

[59] C. Litterer and T. Lyons. High order recombination and an application to cubature on Wiener space. *The Annals of Applied Probability*, 22(4):1301–1327, 2012.

[60] D. Manousakas, Z. Xu, C. Mascolo, and T. Campbell. Bayesian pseudocoresets. *International Conference on Neural Information Processing Systems (NeurIPS)*, 33:14950–14960, 2020.

[61] S. G. Marquis, V. Sulzer, R. Timms, C. P. Please, and S. J. Chapman. An asymptotic derivation of a single particle model with electrolyte. *Journal of the Electrochemical Society*, 166(15):A3693–A3706, 2019. URL: https://doi.org/10.1149/2.0341915jes.

[62] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

[63] P. Mukherjee, D. Parkinson, and A. R. Liddle. A nested sampling algorithm for cosmological model selection. *The Astrophysical Journal*, 638(2):L51, 2006.

[64] R.M. Neal. Annealed importance sampling. *Statistics and Computing*, 11(2):125–139, 2001.

[65] A. O'Hagan. Bayes-Hermite quadrature. *Journal of Statistical Planning and Inference*, 29:245, 1991. URL: https://doi.org/10.1016/0378-3758(91)90002-V.

[66] A. O'Hagan. Bayesian quadrature with non-normal approximating functions. *Statistics and Computing*, 8:365, 1998. URL: https://doi.org/10.1023/A:1008832824006.

[67] M. A. Osborne, D. Duvenaud, R. Garnett, C. Rasmussen, S. Roberts, and Z. Ghahramani. Active learning of model evidence using Bayesian quadrature. In *International Conference on Neural Information Processing Systems (NeurIPS)*, 2012. URL: https://proceedings.neurips.cc/paper/2012/file/6364d3f0f495b6ab9dcf8d3b5c6e0b01-Paper.pdf.

[68] R. Pinsler, J. Gordon, E. Nalisnick, and J. M. Hernández-Lobato. Bayesian batch active learning as sparse subset approximation. *International Conference on Neural Information Processing Systems (NeurIPS)*, 32, 2019.

[69] C. E. Rasmussen and Z. Ghahramani. Bayesian Monte Carlo. In *International Conference on Neural Information Processing Systems (NeurIPS)*, 2003. URL: `https://mlg.eng.cam.ac.uk/zoubin/papers/RasGha03.pdf`.

[70] S. Roberts, M. Osborne, M. Ebden, S. Reece, N. Gibson, and S. Aigrain. Gaussian processes for time-series modelling. *Phil. Trans. R. Soc. A.*, 371:20110550, 2013.

[71] J. Skilling. Nested sampling for general Bayesian computation. *Bayesian Analysis*, 1(4):833 – 859, 2006. URL: `https://doi.org/10.1214/06-BA127`.

[72] J. S. Speagle. dynesty: a dynamic nested sampling package for estimating Bayesian posteriors and evidences. *Monthly Notices of the Royal Astronomical Society*, 493(3):3132–3158, 2020. URL: `https://doi.org/10.1093/mnras/staa278`.

[73] S. Surjanovic and D. Bingham. Virtual library of simulation experiments: Test functions and datasets. Retrieved May 12, 2022, from `http://www.sfu.ca/~ssurjano`.

[74] S. Särkkä and J. Hartikainen. Infinite-dimensional Kalman filtering approach to spatio-temporal Gaussian process regression. In *Artificial Intelligence and Statistics*, pages 993–1001, 2012.

[75] M. Tchernychova. *Carathéodory cubature measures*. PhD thesis, University of Oxford, 2015.

[76] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. `doi:10.1038/s41592-019-0686-2`.

[77] E. Wagstaff. Bayesian quadrature library. `https://github.com/OxfordML/bayesquad`, since 2019.

[78] E. Wagstaff, S. Hamid, and M. A. Osborne. Batch selection for parallelisation of Bayesian quadrature. arXiv preprint, 2018. URL: `https://doi.org/10.48550/arXiv.1812.01553`.

[79] J. Wilson, F. Hutter, and M. Deisenroth. Maximizing acquisition functions for Bayesian optimization. *International Conference on Neural Information Processing Systems (NeurIPS)*, 31, 2018.

Endnote

Clarification

The Appendix is provided in Section A. The integral variance reduction (IVR) used in Section 4.2 coincidentally shares its name with the method proposed by Gessner et al. [82], but the two approaches are distinct. Specifically, their IVR acquisition function defined in Section 2.2, Chapter 2, is designed for single-point selection with 'conjugate' pairs of (μ, k), while our approach extends this to multiple-point selection with optimally-weighted importance sampling $g(x)$ with generic combination of (μ, k).

Errata

In Eqs. (2)-(4), they should be conditioned on $\mathbf{D} = (\mathbf{X}, \mathbf{y})$, instead of $\mathbf{y}$. In Section 4.1, the term 'subsampling N candidates' should be 'supersampling,' as we sample a larger pool of candidates than necessary ($N \gg n$) and then subsample n queries, $\mathbf{X}_{\text{quad}} \subset \mathbf{X}_{\text{rec}}$.

Alternative approach

In our method, we applied a mixture of Gaussian approximation with WSABI to enable fast sampling. However, for more general sampling schemes, alternative approaches can be employed for supersampling without relying on WSABI or Gaussian assumptions. Although WSABI and Gaussian approximations are computationally efficient, as demonstrated in Figures 1 and 2, more generic fast supersamplers can extend the range of applicable scenarios.

In our recent work (Adachi et al. [8]), we employed a weighted kernel density estimator as an alternative. The steps are as follows:

1. Initial Sampling: Sample from the prior distribution, $\tilde{\mathbf{X}}_{\text{rec}} \sim \mu(x)$.

2. Compute Weights: Compute weights as: $\tilde{\mathbf{w}}_{\text{rec}} = \sqrt{\operatorname{Tr} C_t(\tilde{\mathbf{X}}_{\text{rec}}, \tilde{\mathbf{X}}_{\text{rec}})\mu(x)}/\mu(x) = \sqrt{\operatorname{Tr} C_t(\tilde{\mathbf{X}}_{\text{rec}}, \tilde{\mathbf{X}}_{\text{rec}})}$, where Tr denotes the trace of the covariance matrix. Normalize these weights so that $\tilde{\mathbf{w}}_{\text{rec}}^\top \mathbf{1} = 1$.

3. Weighted kernel density estimation: Approximate the proposal distribution: $\sqrt{C_t(x, x)} \approx \tilde{g}(x)$ using the weighted samples $(\tilde{\mathbf{w}}_{\text{rec}}, \tilde{\mathbf{X}}_{\text{rec}})$.

4. Secondary Sampling: Sample again from the estimated distribution, $\mathbf{X}_{\text{rec}} \sim \tilde{g}(x)$.

5. Correcting Weights: Compute the correcting weights: $\mathbf{w}_{\text{rec}} = \sqrt{\operatorname{Tr} C_t(\mathbf{X}_{\text{rec}}, \mathbf{X}_{\text{rec}})}/\tilde{g}(\mathbf{X}_{\text{rec}})$, and normalize these weights.

The resulting empirical measure, $\mu_N(x) = (\mathbf{w}_{\text{rec}}, \mathbf{X}_{\text{rec}})$, provides weighted samples from the proposal distribution $g(x) = \sqrt{C_t(x, x)\mu(x)}$. Using the fast Gauss transform (Yang et al. [241]), kernel density estimation operates with complexity $\mathcal{O}(2N)$, while sampling from a Gaussian distribution is $\mathcal{O}(N)$ (Scott [194]). Thus, this 'approximate' importance sampling approach[1] achieves scalable linear complexity, $\mathcal{O}(N)$.

Other sampling methods can also be applied. For instance:

1. Grosse et al. [91] proposed an inverse transform sampling method, albeit for specific kernels and unweighted uncertainty sampling.

2. Gaussian or Vine copulas (Huk et al. [109]) can serve as more generic extensions of inverse transform sampling, though they are not scalable to high dimensions.

3. Marteau-Ferey et al. [157] introduced a generic sampler for any distribution with positive semi-definite models, which could be an interesting avenue for future exploration.

[1] 'Approximate' in the sense that $\tilde{g}(x) \approx g(x)$.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Fast Bayesian inference with batch Bayesian quadrature via kernel recombination
Publication Status	☑Published □ Accepted for Publication □Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	Adachi, Masaki*, Satoshi Hayakawa*, Martin Jørgensen, Harald Oberhauser, and Michael A. Osborne. "Fast Bayesian inference with batch Bayesian quadrature via kernel recombination." *Advances in Neural Information Processing Systems* 35 (2022): 16533-16547.

Student Confirmation

Student Name:	Masaki Adachi		
Contribution to the Paper	Co-first author I developed the core idea of solving black-box inference as batch Bayesian quadrature using kernel quadrarture, derived the practical methodology with the WSABI formulation, and implemented the corresponding code. I conducted all experiments for the paper, performed additional analyses, and wrote the manuscript, focusing primarily on the experimental and engineering aspects.		
Signature 		Date	06 January 2025

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Michael A. Osborne, Professor of Machine Learning		
Supervisor comments I can confirm that, to the best of my knowledge, Masaki's description above is fair, and that I have great trust in Masaki.		
Signature 	Date	3 February 2025

This completed form should be included in the thesis, at the end of the relevant chapter.

Data is the fossil fuel of AI, and we used it all. [218]

— Ilya Sutskever, Computer Scientiest and FRS

4

Adaptive Batch Sizes for Active Learning: A Probabilistic Numerics Approach

This chapter is based on the following publication:

> **Masaki Adachi**, Satoshi Hayakawa, Martin Jørgensen, Xingchen Wan, Vu Nyugen, Harald Oberhauser, and Michael A. Osborne. Adaptive Batch Sizes for Active Learning: A Probabilistic Numerics Approach. In *International Conference on Artificial Intelligence and Statistics (AISTATS)* 238, 496-504, 2024.

In Chapter 4, we introduced adaptive batch sizes by fixing the quadrature precision for both active learning and Bayesian optimization, building upon our previous work (Adachi et al. [8]). We extended the basic setting outlined in Section 2.5 by addressing two practical factors: hyperparameter uncertainty in active learning and unknown constraints.

In active learning, we adopted Fully Bayesian Gaussian Process (FBGP) models, where: $f \sim \int_\Theta f(x \mid \mathbf{D}_t, \theta) \mathrm{d}\mathbb{P}(\theta \mid \mathbf{D}_t)$, and $\theta \in \Theta$ represents the kernel hyperparameters (e.g., lengthscale), $\Theta \subset \mathbb{R}^k$ is the k-dimensional hyperparameter domain, and $\mathbb{P}(\theta \mid \mathbf{D}_t) \propto \mathbb{P}(\mathbf{D}_t \mid \theta)\mathbb{P}(\theta)$ is the hyperposterior distribution with the marginal likelihood $\mathbb{P}(\mathbf{D}_t \mid \theta)$ and hyperprior $\mathbb{P}(\theta)$. Consequently, the integral

objective becomes:

$$\hat{Z} = \int_{\mathcal{X}} \mathbb{E}_{\theta\sim\mathbb{P}(\theta|\mathbf{D}_t)} \left[f_t(x \mid \theta)\right] \mathrm{d}\mathbb{P}(\mathcal{S}) \xRightarrow{\text{IVR}} \max_{\mathbf{X}_t^n \subset \mathcal{X}} \Delta\tilde{C}_t(\mathbf{X}_t^n, \mathbb{P}(\mathcal{S})) \tag{4.1}$$

This requires accounting for the additional expectation over the hyperposterior $\mathbb{P}(\theta \mid \mathbf{D}_t) \propto \mathbb{P}(\mathbf{D}_t \mid \theta)\mathbb{P}(\theta)$, where $\mathbb{P}(\theta)$ is the hyperprior and $\mathbb{P}(\mathbf{D}_t \mid \theta)$ is the marginal likelihood. Using a Gaussian approximation, the FBGP can be approximated as $\tilde{f}_t \sim \mathcal{GP}(\tilde{m}_t, \tilde{C}_t)$, where the expected predictive mean $\tilde{m}_t$ and covariance $\tilde{C}_t$ are given by:

$$\tilde{m}_t(x) = \mathbb{E}_{\theta\sim\mathbb{P}(\theta|\mathbf{D}_t)}[m_t(x \mid \theta)], \tag{4.2}$$

$$\tilde{C}_t(x,x) = \mathbb{E}_{\theta\sim\mathbb{P}(\theta|\mathbf{D}_t)} \left[C_t(x,x \mid \theta) + (m_t(x \mid \theta) - \tilde{m}_t(x \mid \theta))(m_t(x \mid \theta) - \tilde{m}_t(x))^\top\right]. \tag{4.3}$$

These terms correspond to the expected predictive mean and covariance for the Gaussian mixture. Then, the FBGP can be approximated as:

$$f_t(x) \sim \mathbb{E}_{\theta\sim\mathbb{P}(\theta|\mathbf{D}_t)}[f_t(x \mid \theta)] \approx \mathcal{GP}(\tilde{m}_t(x), \tilde{C}_t(x,x)). \tag{4.4}$$

This formulation enables FBGP active learning by reducing the integral variance over the *expected* function space $\mathcal{H}_{\tilde{C}_t}$.

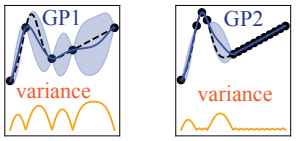
As introduced in Section 2.5, the unknown constraints are similar to black-box functions; we do not know the constraint functions form nor their gradients, and we need to estimate them only from the pointwise queries. We typically place other GP models on the constraints, then they becomes *probabilistic* constraint. As satisfying the probabilistic constraints with 100% safely is very challenging (at least we can only prove the high-probability guarantee), thus typical relaxation is to consider the probabilistic constraints as soft constraints, and we aim at minimizing the total violation, rather than zero violation.

Adaptive Batch Sizes for Active Learning: A Probabilistic Numerics Approach

Masaki Adachi[1,3] **Satoshi Hayakawa**[2] **Martin Jørgensen**[4] **Xingchen Wan**[1]
Vu Nguyen[5] **Harald Oberhauser**[2] **Michael A. Osborne**[1]
[1]Machine Learning Research Group, University of Oxford
[2]Mathematical Institute, University of Oxford
[3]Toyota Motor Corporation
[4]Department of Computer Science, University of Helsinki
[5]Amazon

Abstract

Active learning parallelization is widely used, but typically relies on fixing the batch size throughout experimentation. This fixed approach is inefficient because of a dynamic trade-off between cost and speed—larger batches are more costly, smaller batches lead to slower wall-clock run-times—and the trade-off may change over the run (larger batches are often preferable earlier). To address this trade-off, we propose a novel Probabilistic Numerics framework that adaptively changes batch sizes. By framing batch selection as a quadrature task, our integration-error-aware algorithm facilitates the automatic tuning of batch sizes to meet predefined quadrature precision objectives, akin to how typical optimizers terminate based on convergence thresholds. This approach obviates the necessity for exhaustive searches across all potential batch sizes. We also extend this to scenarios with constrained active learning and constrained optimization, interpreting constraint violations as reductions in the precision requirement, to subsequently adapt batch construction. Through extensive experiments, we demonstrate that our approach significantly enhances learning efficiency and flexibility in diverse Bayesian batch active learning and Bayesian optimization applications.



batch size	Worst-case error (precision $\varepsilon \leq$ 1e-3)	
n=2	1e-1	1e-3
n=3	1e-2	1e-4
n=4	1e-3	1e-5
n=5	1e-4	1e-6

Figure 1: We fix the quadrature precision instead of batch size. The batch size changes adaptively to meet the predefined precision requirement. Our method, AdaBatAL, efficiently determines the optimal number of batch sizes and their querying positions without requiring a brute-force search of all possible batch sizes. AdaBatAL also offers adaptive batch sizes for constrained active learning and constrained Bayesian optimization.

1 Introduction

Active Learning (AL) (Settles, 2009) is a machine learning concept where the algorithm selects its training data, which enhances accuracy based on fewer labels. Its use is widespread in deep learning models (Gal et al., 2017; Ren et al., 2021; Kirsch et al., 2019) and Gaussian processes (GPs) (Houlsby et al., 2011; Riis et al., 2022). Bayesian AL intertwines with Probabilistic Numerics (PN) (Hennig et al., 2015, 2022), that reinterprets numerical tasks as Bayesian machine learning. This allows uncertainty to interlink with real-world constraints, improving empirical performance, and algorithmic flexibility. In PN, AL enables sample-efficient procedures, with Bayesian optimization (BO) and Bayesian quadrature (BQ) being key instances,

Proceedings of the 27[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238.

applied in fields like drug discovery (Gómez-Bombarelli et al., 2018), materials (Adachi, 2021), and hyperparameter tuning (Feurer et al., 2015; Wu et al., 2020).

AL research can be classified into sequential and batch settings. While the sequential setting selects the next training data point one by one, the batch setting selects multiple points at the same time. We have two key metrics of performance: the number of iterations and the number of total queries. The number of iterations corresponds to the speed of model training, and the batch setting is advantageous as it can gain more feedback per iteration. In contrast, the number of total queries corresponds to the cost. For instance, labeling the data may involve expensive human evaluations. This total query metric is advantageous to the sequential setting as it can observe feedback for every single query to give a rational decision, whereas the batch setting needs to select multiple points without feedback.

However, situations arise where a balance between speed and cost is desirable. For instance, while training a model, renting cloud servers is an option, with charges applied based on the number of nodes (batch size) *and* duration (total queries). Another scenario is crowdsourcing annotation, where a balance is needed between the number of annotators (batch size) and the total working time (total queries). We aim to expedite model training while also saving on cost.

In addition to these situations, constraints often come into play in real-world applications, and often the constraints are also unknown a priori. Unknown constraints (Gelbart et al., 2014; Hernández-Lobato et al., 2016) are the constraints with which we must comply, but we do not know the constraint function a priori and are only observable pointwise. Hence, we had to estimate the true constraint function based on limited observations, resulting in uncertainty in the constraint estimation. For example, drug discovery needs to satisfy the safety constraints via animal experiments (Lipinski et al., 1997), but we do not know the functional form. Similarly, active learning with real physical experiments contains unknown constraints such as limitations from experimental apparatus or phase transition of measuring materials (Khatamsaz et al., 2023; Lookman et al., 2019). Training models on the cloud server may be halted due to errors or memory overflow, or annotators may pause annotation in cases of ambiguity in annotation guidelines or unclear samples. Avoiding querying samples that are likely to violate such unknown constraints is essential for the smooth execution of active learning. However, research on active learning under constraints is scarce, and no existing work considers adaptive batch size under constraints.

To address the said challenges, we propose a PN framework that adaptively adjusts the batch size. Figure 1 illustrates the concept. We hypothesize that an adaptive batch size can balance the trade-off between cost and speed. Fixed batch sizes might be ineffective because, as the shape of the acquisition function changes dynamically, the effectiveness of batch acquisition also shifts. In Figure 1, the left side displays four distinct peaks, indicating that four batch sizes would be suitable. Conversely, the right side exhibits only two prominent peaks, suggesting that two batch sizes would be more appropriate.

Given this intuition, we define batch construction as an approximation of a continuous target distribution (e.g., an acquisition function) using a discrete distribution (batch samples)—a process known as *quantization* applied in diverse machine learning fields (Graf & Luschgy, 2007; Karvonen, 2019; Teymur et al., 2021). The error in this approximation can be measured by the divergence between the target and the 'quantized' distribution. With this perspective, instead of fixing the batch size, we propose to fix the *precision* of the approximation. We reframe batch construction as a quantization task, assessing precision through divergence. We fix the precision requirement for iterations, allowing the batch size and locations to be adaptively adjusted. In essence, our approach quantifies numerical errors stemming from an insufficient batch sizes, strategically harnesses this computational uncertainty for decision-making, and embodies the essence of PN principles. Specifically, for GP models, this quantization links seamlessly to kernel quadrature (KQ), enabling the use of advanced KQ methods for efficient solutions. As such, we further re-cast the quantization task as a KQ task, using the worst-case integration error as our divergence metric. Our method, *adaptive batch active learning* (AdaBatAL), efficiently determines the optimal number of batch sizes and their querying positions without requiring a brute-force search of all possible batch sizes.

AdaBatAL also seamlessly handles AL in the presence of unknown constraints. We view the risk associated with these constraints as a 'varying precision requirement.' If querying points violate the constraints, we remove them from the valid dataset, thereby reducing precision. We interpret a high risk of constraint violation as a lower precision requirement and vice versa. Therefore, the constrained case serves as a 'preprocessing' step to determine the appropriate precision for AdaBatAL, with the constraint model estimating the precision requirement. The versatility of AdaBatAL provides a plug-and-play framework for AL, BQ, and BO, whether constraints are involved or not.

Contributions

1. **Adaptive batch size** We fixed *quadrature precision* via re-casting batch construction as a KQ, allowing batch size adaptively changing according to the acquisition function efficiently.
2. **Unknown constraints** We reinterpret the batch AL under unknown constraints as varying precision requirement. This allows adaptively changing the batch size and locations in accordance with the risks of constraint violation.
3. **Generality** Our adaptive batch construction scheme applies to AL, BO, and BQ by changing the target distribution of quantization with KQ. Moreover, it applies to non-continuous domains (e.g. combinatorial, mixed feature spaces).
4. **Significant improvement** is shown in both batch AL and batch BO tasks, outperforming 17 baselines over 6 synthetic and 7 real-world tasks.
5. **Open-source** we open-source the software on GitHub `https://github.com/ma921/AdaBatAL`.

2 Background

We start by providing the background on quantization and KQ. We then demonstrate the connection between GP, KQ, and BQ, leading to pure batch uncertainty sampling.

Quantization. Let μ be a probability distribution defined on a set $\mathcal{X}$. The *quantization* task is to find the discrete distribution $\nu := \frac{1}{n}\sum_{i=1}^{n}\delta_{x_i}$, which best approximates μ with n representative points x_i. Here, δ_x denotes a point mass (delta distribution) located at $x \in \mathcal{X}$. To solve the quantization task, one first identifies an optimality criterion, typically a notion of *discrepancy* between μ and ν, and then develops an algorithm to approximately minimize it.

Kernel Quadrature. KQ is a numerical integration for calculating the integral of a function belonging to a reproducing kernel Hilbert space (RKHS). The aim is to find a good approximation of an, otherwise intractable, integral with a weighted sum. A KQ rule, $Q_{\boldsymbol{w},\boldsymbol{x}}$, is given by weights $\boldsymbol{w} = \{w_i\}_{i=1}^n$ and points $\boldsymbol{x} = \{x_i\}_{i=1}^n$,

$$Q_{\boldsymbol{w},\boldsymbol{x}}(f) := \sum_{i=1}^{n} w_i f(x_i) \approx \int_{\mathcal{X}} f(x)\mathrm{d}\mu(x), \qquad (1)$$

where f is a function of RKHS $\mathcal{H}$ associated with the kernel K. The *worst-case error* given μ and $\mathcal{H}$ is

$$\mathrm{wce}(Q_{\boldsymbol{w},\boldsymbol{x}}) := \sup_{\|f\|_{\mathcal{H}}\leq 1}\left|Q_{\boldsymbol{w},\boldsymbol{x}}(f) - \int_{\mathcal{X}} f(x)\mathrm{d}\mu(x)\right|. \quad (2)$$

The aim is to find $Q_{\boldsymbol{w},\boldsymbol{x}}$ minimizing worst-case error.

Connection to quantization. When inspecting the KQ rule as integration against a discrete distribution $\nu := \sum_{i=1}^{n} w_i\delta_{x_i}$, namely, $Q_{\boldsymbol{w},\boldsymbol{x}}(f) = \int_{\mathcal{X}} f(x)\mathrm{d}\nu(x)$, the worst-case error can be viewed as the *divergence* between μ and ν. Indeed, there is a theoretical connection between KQ and quantization, as KQ is the *weighted* quantization under the maximum mean discrepancy (MMD) metric (Karvonen, 2019; Teymur et al., 2021). MMD is a widely used method to quantify the divergence between two distributions (Sriperumbudur et al., 2010; Muandet et al., 2017), defined as:

$$\mathrm{MMD}_{\mathcal{H}}(\nu,\mu) := \left\| \int K(\cdot,x)\mathrm{d}\nu(x) - \int K(\cdot,x)\mathrm{d}\mu(x) \right\|_{\mathcal{H}},$$

and we can rewrite as (Huszár & Duvenaud, 2012):

$$\mathrm{MMD}^2_{\mathcal{H}}(\nu,\mu) := \sup_{\|f\|_{\mathcal{H}}=1}\left| \int f(x)\mathrm{d}\nu(x) - \int f(x)\mathrm{d}\mu(x) \right|^2.$$

This squared formulation is the same with the worst-case error. Therefore, solving KQ is equivalent to finding the discrete distribution ν that best approximates μ with regard to MMD. Note that KQ is a weighted quantization, unlike in the previous section.

Connection to Gaussian Process. Assume a function f is modelled by GP, $f \sim \mathcal{GP}(m, C)$, with limited number of observed points, $\mathcal{D}_0 := \{\boldsymbol{x}_0, \boldsymbol{y}_0\}$, where $\boldsymbol{y}_0 = f_{\mathrm{true}}(\boldsymbol{x}_0) + \epsilon$ are the noisy observations. We wish to estimate the expectation of the function $\hat{Z} := \int_{\mathcal{X}} f(x)\mathrm{d}\mu(x)$. This setting is called Bayesian quadrature (BQ) (O'Hagan, 1991), one of the central methods of PN. The integral estimate are as follows:

$$\mathbb{E}_f[\hat{Z}] = \int m(x)\mathrm{d}\mu(x) = \boldsymbol{z}^\top \boldsymbol{K}^{-1}\boldsymbol{y}_0, \qquad (3)$$

$$\mathbb{V}\mathrm{ar}_f[\hat{Z}] = \int C(x,x')\mathrm{d}\mu(x)\mathrm{d}\mu(x') = z' - \boldsymbol{z}^\top \boldsymbol{K}^{-1}\boldsymbol{z}, \qquad (4)$$

where $\boldsymbol{z} := \int K(x,\boldsymbol{x}_0)\mathrm{d}\mu(x)$ and $z' := \int K(x,x')\mathrm{d}\mu(x)\mathrm{d}\mu(x')$ are kernel mean and variance, respectively.

Huszár & Duvenaud (2012) proved the worst-case error (Eq. (2)) equals to the variance in Eq. (4) if quadrature nodes are $\mathcal{D}_0$. BQ expectation in Eq. (3) is a weighted sum; $\boldsymbol{z}^\top \boldsymbol{K}^{-1}\boldsymbol{y}_0 = \sum_{i=1}^{n} w_{BQ,i} y_i$, where $w_{\mathrm{BQ,\,j}} := \sum_{i=1}^{n} \boldsymbol{z}_i^\top \boldsymbol{K}^{-1}_{i,j}$. We can further think these weights as a discrete distribition $\nu_{\mathrm{BQ}} := \sum_{i=1}^{n} w_{\mathrm{BQ},i}\delta_{x_i}$, then the variance of integral estimation becomes:

$$\mathbb{V}\mathrm{ar}_f[\hat{Z}] = \mathrm{MMD}^2(\mu,\nu_{\mathrm{BQ}}) = \inf_{\boldsymbol{w}} \mathrm{wce}(Q_{\boldsymbol{w},\boldsymbol{x}})^2. \qquad (5)$$

This shows that KQ and BQ are closely connected (see details in (Huszár & Duvenaud, 2012)). The variance of integral is the uncertainty of GP over μ, so quantizing this distribution using KQ can be understood as a 'pure

batch exploration' of GP uncertainty. This idea was applied to batch BQ (Adachi et al., 2022).

In summary. A quantization task can be viewed as a KQ task. The selected batch samples minimize the divergence between the target distribution μ and the batch samples ν, with a given kernel K. When we use the GP predictive covariance $C(\cdot, \cdot)$ as the kernel K for the MMD, the KQ becomes the pure batch exploration of GP uncertainty while also minimizing the divergence from the target distribution. Hence, batch construction via solving KQ can offer a quantization of the target distribution combined with uncertainty sampling.

3 Adaptive Batch Active Learning

Now, we introduce our method, AdaBatAL. Any KQ method can be used, but we employ the recombination (Hayakawa et al., 2022) for flexibility. We extend this to adaptive batch size under unknown constraints.

3.1 Problem Setting of Batch Active Learning

Batch Active Learning Consider we have a limited number of a labelled dataset $\mathcal{D}_0 = \{\boldsymbol{x}_k, \boldsymbol{y}_k\}_{k=1}^m$, and the large number of unlabelled pool set $\mathcal{X}_N = \{\boldsymbol{x}_l\}_{l=1}^N$, where $N \gg m$, an oracle can provide labels $\mathcal{Y}_N = \{\boldsymbol{y}_m\}_{m=1}^N$ for the corresponding inputs. We sequentially query the batch samples $\mathcal{D}_t^n = \{\boldsymbol{x}_j, \boldsymbol{y}_j\}_{j=1}^n$ with n batch sizes at t-th iteration, resulting in the total labelled dataset $\mathcal{D}_t = \mathcal{D}_{t-1} \cup \mathcal{D}_t^n = (\mathcal{X}_t, \mathcal{Y}_t)$, and repeat T times[1]. The batch AL task is to select the $\mathcal{D}_t$ to minimize the prediction error between true labels $\mathcal{Y}_N$ and the prediction conditioned on $\mathcal{X}_N$ at given budget T. Throughout this paper, we assume the model is an FBGP for AL and a normal GP for BO.

Following the works Pinsler et al. (2019); Adachi et al. (2023a), we can recast the batch AL and batch BO as a quantization task. The difference between AL and BO comes down to the target distributions μ: the candidate pool of unlabelled inputs for batch AL, and the probability of global optimum location for batch BO. Important takeaways from their works are that the quantization approach can outperform popular baselines, such as BALD (Houlsby et al., 2011) for batch AL, and hallucination Azimi et al. (2010) for batch BO. Yet, their approach only considers fixed batch size without constraints. We augment their approaches by adaptive batch sizes under unknown constraints.

Unknown Constraints Consider our labelling scheme is subject to the constraint $c(x) \geq 0$, where c is the constraint with which we must comply, otherwise the query x is eliminated from the labelled dataset $\mathcal{D}_t$ (e.g., a drug candidate that breaches a safety constraint will not be tested.). We further assume the constraints are unknown a priori and are only observable pointwise. Hence, probabilistic model estimates the function $\hat{c}(x)$ with its predictive uncertainty, providing the probability of constraint satisfaction $q(x)$ at given input x. Following Gelbart et al. (2014), we model the constraint by another GP.

3.2 Problem Setting of Kernel Quadrature

As a general situation, consider we are given a kernel K on $\mathcal{X}$ and an N-point samples $\mathbf{X}_{\mathrm{cand}} \in \mathcal{X}^N$ associated with a nonnegative weight $\mathbf{w}_{\mathrm{cand}}$ with $\mathbf{w}_{\mathrm{cand}}^\top \mathbf{1} = 1$[2]. We denote this as $\mu(x) := \sum_{i=1}^N w_i \delta_{x_i}$ as a discrete distribution, or $(\mathbf{w}_{\mathrm{cand}}, \mathbf{X}_{\mathrm{cand}})$ as the ordered pair. In a typical batch AL setting, μ is the candidate pool of unlabelled inputs with equal weights. The goal is to find a weighted subset $(\mathbf{w}_{\mathrm{batch}}, \mathbf{X}_{\mathrm{batch}})$, $\nu(x) := \sum_{j=1}^n w_j \delta_{x_j}$ which minimizes $\mathrm{MMD}_{\mathcal{H}}(\mu, \nu)$ given μ and kernel K[3]. Hence, this is a KQ task. The quantized subset ν, $\mathbf{X}_{\mathrm{batch}} \subset \mathbf{X}_{\mathrm{cand}}$, will give the batch samples for batch AL and batch BO. Unlike the existing setting (Hayakawa et al., 2022; Adachi et al., 2022, 2023a), we additionally work under the following conditions:

(a) The upper bound of batch size n is given but the actual batch size is adaptively changed to meet the precision under the given tolerance ϵ_{LP}.

(b) After we choose the batch querying points, $(\mathbf{w}_{\mathrm{batch}}, \mathbf{X}_{\mathrm{batch}})$, each point $x \in \mathbf{X}_{\mathrm{batch}}$ is subject to the probabilistic constraint $q(x)$[4] (and violated w.p. $1 - q(x)$), where $q : \mathcal{X} \to [0, 1]$ is given as GP. We query the true constraint $c(x)$, then we obtain the feasible points and corresponding weights $(\tilde{\mathbf{w}}_{\mathrm{batch}}, \tilde{\mathbf{X}}_{\mathrm{batch}})$, where $\tilde{\mathbf{X}}_{\mathrm{batch}} \subset \mathbf{X}_{\mathrm{batch}}$[5]. We use the feasible points for the quadrature.

(c) Additionally, a reward function $g : \mathcal{X} \to \mathbb{R}$ is given as additional flexibility that incorporates the other desideratum (e.g. soft constraint), and we want to make the expected reward $\tilde{\mathbf{w}}_{\mathrm{batch}}^\top g(\tilde{\mathbf{X}}_{\mathrm{batch}})$ as big as possible while making the worst-case error $\mathrm{wce}(Q_{\tilde{\mathbf{w}}_{\mathrm{batch}}, \tilde{\mathbf{X}}_{\mathrm{batch}}})$ [6] as small as possible.

[1] To clarify, $D_0 \subseteq D_t$ but $D_0 \not\subset D_t^n$.

[2] $\mathbf{1}$ is $[1, \ldots, 1]^N$, the vector of ones.

[3] We set the kernel K as the posterior predictive covariance $C(\cdot, \cdot)$ (recall the background section).

[4] The true constraint $c(x)$ is deterministic but $q(x)$ becomes probabilistic due to the predictive uncertainty.

[5] $\tilde{\mathbf{X}}_{\mathrm{batch}} = \mathbf{Z}^\top \mathbf{X}_{\mathrm{batch}}$, where $\mathbf{Z}$ is a vector of Bernoulli random variables with probabilities $q(\mathbf{X}_{\mathrm{batch}})$

[6] For brevity, b is batch, c is cand, then $\mathrm{wce}(Q_{\tilde{\mathbf{w}}_{\mathrm{b}}, \tilde{\mathbf{X}}_{\mathrm{b}}}) = \tilde{\mathbf{w}}_{\mathrm{b}}^\top K(\tilde{\mathbf{X}}_{\mathrm{b}}, \tilde{\mathbf{X}}_{\mathrm{b}}) \tilde{\mathbf{w}}_{\mathrm{b}} - 2\tilde{\mathbf{w}}_{\mathrm{b}}^\top K(\tilde{\mathbf{X}}_{\mathrm{b}}, \mathbf{X}_{\mathrm{c}}) \mathbf{w}_{\mathrm{c}} + \mathbf{w}_{\mathrm{c}}^\top K(\mathbf{X}_{\mathrm{c}}, \mathbf{X}_{\mathrm{c}}) \mathbf{w}_{\mathrm{c}}$.

3.3 Kernel Quadrature via Nyström Approximation

Although the Nyström method (Williams & Seeger, 2000; Drineas & Mahoney, 2005; Kumar et al., 2012) is primarily used for approximating a large Gram matrix by a low-rank matrix, it can also be used for directly approximating the kernel function itself. Given a set of M points $X_{\mathrm{nys}} = \{x_i\}_{i=1}^{M} \subset \mathcal{X}$, the Nyström approximation of $K(x, y)$ is given by:

$$K(x,y) \approx K_0(x,y) := \sum_{i=1}^{n-1} \lambda_i^{-1} \varphi_i(x)\varphi_i(y), \qquad (6)$$

where $\varphi_i(\cdot) := u_i^\top K(X_{\mathrm{nys}}, \cdot)$ $(i = 1, \ldots, n-1)$ are called *test functions*, chosen from a larger M dimensional space $\mathrm{span}\{K(x_i, \cdot)\}_{i=1}^{M}$. The Eq. (6) holds if $\lambda_s > 0$. To compute Eq. (6), we perform the best rank-s approximation of the Gram matrix $K(X_{\mathrm{nys}}, X_{\mathrm{nys}}) = U\Lambda U^\top$, given by eigendecomposition, where $U = [u_1, \ldots, u_M] \in \mathbb{R}^{M\times M}$ is a real orthogonal matrix and $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_M)$ with $\lambda_1 \geq \ldots \geq \lambda_M \geq 0$.

We can use the test functions for integration estimator $\hat{Z} = \int_{\mathcal{X}} f(x)\mathrm{d}\mu(x)$. When the spectral decay in eigenvalues is steep, the Nyström method can give a good approximation of the original kernel function with a small number of test functions. Let $\boldsymbol{\varphi} = \{\varphi_1, \ldots, \varphi_{n-1}\}^\top$ be the vector of test functions that spans $\mathcal{H}_{K_0}$, the RKHS associated with the approximated kernel K_0, we assume we have additional knowledge of expectations, namely, $\int_{\mathcal{X}} \boldsymbol{\varphi}(x)\mathrm{d}\mu(x) = \mathbf{w}_{\mathrm{cand}}^\top \boldsymbol{\varphi}(\mathbf{X}_{\mathrm{cand}})$ is given. We can actually construct a convex quadrature $Q_n = (w_i, x_i)_{i=1}^{n}$:

$$\sum_{i=1}^{n-1} w_i \varphi_i(x_i) = \int_{\mathcal{X}} \boldsymbol{\varphi}(x)\mathrm{d}\mu(x) \approx \int_{\mathcal{X}} f(x)\mathrm{d}\mu(x). \qquad (7)$$

Now, we can approximate the integral by $n-1$ test functions. Hence, Eq. (7) can be understood as $n-1$ *equality constraints* which w_i and x_i need to satisfy.

The benefit of this approximation is to incorporate the information of spectral decay of Gram matrix for faster convergence. If the target function f is smooth, the spectral decay is fast, then the small number of test functions can well represent the function, leading to batch-size efficient AL and BO (Hayakawa et al., 2022; Adachi et al., 2022).

3.4 Linear Programming Formulation

To solve the above problem, we introduce the following linear programming (LP) problem that aims to achieve both the reward maximization and the worst-case error minimization where possible, given by modifying the algorithm adopted in (Adachi et al., 2023a) ($n \geq 3$):

$$\begin{aligned}
&\underset{\mathbf{w}}{\text{maximize}} \quad \mathbf{w}^\top \left[g(\mathbf{X}_{\mathrm{cand}}) \odot q(\mathbf{X}_{\mathrm{cand}}) \right], \\
&\text{subject to} \\
&\left| (\mathbf{w} - \mathbf{w}_{\mathrm{cand}})^\top \varphi_j(\mathbf{X}_{\mathrm{cand}}) \right| \leq \epsilon_{\mathrm{LP}} \sqrt{\lambda_j/(n-2)}, \\
&\forall j : 1 \leq j \leq n-2, \\
&(\mathbf{w} - \mathbf{w}_{\mathrm{cand}})^\top q(\mathbf{X}_{\mathrm{cand}}) \geq 0, \\
&\mathbf{w}^\top \mathbf{1} = 1, \quad \mathbf{w} \geq \mathbf{0}, \quad |\mathbf{w}|_0 \leq n,
\end{aligned}$$

where $\epsilon_{\mathrm{LP}} \geq 0$ is a *tolerance* parameter, which can be interpreted as the quadrature precision requirements (smaller is more accurate), and (λ_j, φ_j) are given by the Nyström approximation (see 3.3)[7].

The intuition of this formulation is as follows:

(1) The solutions are the sparse weights $\mathbf{w}$, where the non-zero element of $\mathbf{w}$ corresponds to the batch selection, and the corresponding samples of $\mathbf{X}_{\mathrm{cand}}$ is the batch samples $\mathbf{X}_{\mathrm{batch}}$. We refer to the nonzero weights and corresponding samples as the solution $(\mathbf{w}_{\mathrm{batch}}, \mathbf{X}_{\mathrm{batch}})$[8], and its batch size is $|\mathbf{X}_{\mathrm{batch}}| \leq n$. As such, this LP problem is to subsample the batch samples ν from the given discrete distribution μ, namely, quantization.

(2) The objective is to maximize the expected reward g under the risk of constraint violation q. This promotes safe sampling by increasing the expected constraints satisfaction $\mathbf{w}^\top q(\mathbf{X}_{\mathrm{cand}})$.

(3) The first constraints correspond to equality constraints with test functions in Eq. (7). We relaxed the equality constraints to inequality constraints to accept the tolerance ϵ_{LP}. These $n-2$ inequality constraints restrict the solution space to where the approximation error of the expectations of test functions $|(\mathbf{w} - \mathbf{w}_{\mathrm{cand}})^\top \varphi_j(\mathbf{X}_{\mathrm{cand}})|$[9] is within the tolerance parameter ϵ_{LP}. These $n-2$ constraints are very restrictive; the flexibility to select the larger objective is much more restricted than the typical LP problem. ϵ_{LP} controls the trade-off between the accuracy for quadrature and relaxing solution space to find the larger objective.

(4) Other constraints assure the number of nonzero elements of the solution set $\mathbf{w}$ is fewer than the upper bound of batch size n, the convex and positive weights, and the probability of probabilistic constraints' satisfaction is positive.

Thus, in response to conditions (b)(c), the solution of this LP problem provides the batch samples that satisfy convex quadrature rules within the tolerance *and* maximizing the reward. The balance between

[7] $\odot$ refers to Hadamard product, and $|\cdot|_0$ denotes the number of nonzero entries.

[8] $\mathbf{X}_{\mathrm{batch}} \subset \mathbf{X}_{\mathrm{cand}}$, $\mathbf{w}_{\mathrm{batch}} \subset \mathbf{w}_{\mathrm{cand}}$, and $|\mathbf{X}_{\mathrm{batch}}| = |\mathbf{w}|_0$

[9] This is a quadrature error in Eq. (7). $\left|\mathbf{w}^\top \boldsymbol{\varphi}(\mathbf{X}_{\mathrm{cand}}) - \mathbf{w}_{\mathrm{cand}}^\top \boldsymbol{\varphi}(\mathbf{X}_{\mathrm{cand}})\right| \approx \left|\int_{\mathcal{X}} f(x)\mathrm{d}\nu(x) - \int_{\mathcal{X}} f(x)\mathrm{d}\mu(x)\right|$.

quadrature accuracy and reward maximization can be controlled by a single parameter ϵ_{LP}. To be clear, only within §3.4, the term 'constraints' refers to the ones in LP formulations. Otherwise, the constraints refer to the task-specific unknown constraints (e.g. safety constraints for drug discovery).

3.5 Adaptive Batch Sizes

The count of non-zero elements, denoted as $|\boldsymbol{w}|_0$, is adjusted based on the tolerance ϵ_{LP}. The intuition of the batch size adaptivity is explicated as:

1. Higher precision demands result in a smaller quadrature error tolerance. This necessitates a larger sample set for more precise integration.
2. Conversely, lower precision requirements needs fewer $|\boldsymbol{w}|_0$ to meet the desired accuracy.

Elaborating further, the batch size is tied to slack variables in LP solvers. As the tolerance ϵ_{LP} increases, some inequality constraints become deactivated (Dantzig, 2002). The batch size is determined by the number of active constraints, often leading to sparse weights with $|\boldsymbol{w}|_0 < n$. When constraints are loose, a large preset batch size is inefficient, as the desired precision can be achieved with fewer samples. As such, we can identify the adaptive batch size $|\boldsymbol{w}|_0$ without needing a brute-force search of all possible batch sizes.

Note that ϵ_{LP} controls *all* balances: the batch size, quadrature accuracy, and reward maximization. Interestingly, its behavior is not a monotonic decrease in its magnitude. As ϵ_{LP} approaches infinity, the batch size converges to 1, aligning with the sequential AL case. An increased ϵ_{LP} shrinks the batch size as observed in §5.1. This approach is essentially a heuristic for adaptive batch sizes. Although it satisfies a predefined worst-case error threshold, it does not guarantee optimal results based on other established metrics like mutual information (Krause & Guestrin, 2012). However, as Leskovec et al. (2007) highlighted, when greedily maximizing mutual information under the weighted candidates and a budget constraint (limitation in the number of the total queries), the approximation factor can be arbitrarily bad. Hence, even popular strategies, such as BALD (Houlsby et al., 2011), also cannot achieve a solution within $1 - 1/e$ of the optimal in our problem setting (Li et al., 2022).

3.6 Unknown Constraints As The Lowered Precision Requirement

In this further examination, we address the probabilistic constraint denoted as q. Given the uncertainty in predicting the true constraint c, the candidate solution, $\mathbf{X}_{\text{cand}}$, carries a risk of violation. We can estimate the

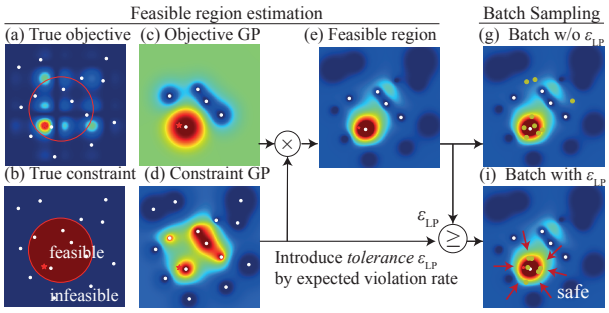


Figure 2: Constrained batch active learning. As the increased violation risk ϵ_{vio} propagates to the tolerance ϵ_{LP}, reward maximization is subsequently prioritized over quadrature, resulting in safe batch samples.

expected violation rate by $\epsilon_{\text{vio}} := 1 - \mathbf{w}_{\text{cand}}^{\top} q(\mathbf{X}_{\text{cand}})$. It is assumed that infeasible points are eliminated from quadrature nodes for computation, reducing quadrature accuracy. The expected violation rate ϵ_{vio} can be interpreted as the *risk* we cannot control. A high-risk scenario necessitates cautious exploration to avoid wasting valuable queries; this suggests smaller batch sizes and selecting queries where $\mathbf{x}_{\text{cand}}$ is more likely to satisfy the true constraint c. Conversely, a low risk allows for more optimistic exploration.

In response to varying risk levels, we advocate for an *adaptive* exploration strategy. Our proposed method is straightforward yet effective: setting $\epsilon_{\text{LP}} = \epsilon_{\text{vio}}$. This approach allows for automatic adjustment of exploration safety. When ϵ_{vio} is high, indicating greater risk, ϵ_{LP} is set higher. This results in looser quadrature precision, smaller batch sizes, and a solution space that is more likely to satisfy constraints[10]. Thus, a higher ϵLP leads to safer batch sampling. When the risk ϵ_{vio} is low, ϵ_{LP} is set lower, allowing for larger batch sizes and more explorative solutions. Figure 2 demonstrates this adaptive behavior: high-risk information ϵ_{vio} influences ϵ_{LP}, leading to safer batch samples. Adaptive safe exploration is not necessarily always safe. We need to explore uncertain regions at some point and we propose it is when the risk is low. Our PN framework effectively bridges computational uncertainty and real-world risk, providing an automated and adaptable balance between safety and exploration.

[10] Remember that a reduction in quadrature precision results in an expansion of the solution space, which in turn enables the identification of solutions with higher LP objective values, as denoted by $\mathbf{w}_{\text{cand}}^{\top}\left[g(\mathbf{X}_{\text{cand}}) \odot q(\mathbf{X}_{\text{cand}})\right]$, where $\mathbf{w}_{\text{cand}}^{\top} q(\mathbf{X}_{\text{cand}})$ represents the expected satisfaction of the constraint. Thus, maximizing the LP objective value leads to increasing the constraint satisfaction probability.

3.7 Error Bounds

The error estimate of KQ is essentially determined by the approximation error of the Nyström method, $\epsilon_{\text{nys}} := \max_{x \in \mathbf{X}_{\text{cand}}} |K_0(x,x) - K(x,x)|^{1/2}$. Error bounds for this approximation have been well studied in the literature (Drineas & Mahoney, 2005; Kumar et al., 2012; Hayakawa et al., 2023).

Proposition 1. *Under the above setting, let $\mathbf{w}_*$ be the optimal solution of the LP, and let $\mathbf{X}_{\text{batch}}$ be the subset of $\mathbf{X}_{\text{cand}}$, corresponding to the nonzero entries of $\mathbf{w}_*$ (denoted by $\mathbf{w}_{\text{batch}}$). Suppose that $\tilde{\mathbf{X}}_{\text{batch}}$ is given by a random subset of $\mathbf{X}_{\text{batch}}$, where each point x satisfies the constraints with probability $q(x)$, and let $\tilde{\mathbf{w}}_{\text{batch}}$ be the corresponding weights. Then, we have*

$$\mathbb{E}[\tilde{\mathbf{w}}_{\text{batch}}^\top g(\tilde{\mathbf{X}}_{\text{batch}})] \geq \mathbf{w}_{\text{cand}}^\top \big[g(\mathbf{X}_{\text{cand}}) \odot q(\mathbf{X}_{\text{cand}})\big], \tag{8}$$

and, for any function f in the RKHS with kernel K,

$$\begin{aligned} &\mathbb{E}\Big[\Big|\tilde{\mathbf{w}}_{\text{batch}}^\top f(\tilde{\mathbf{X}}_{\text{batch}}) - \mathbf{w}_{\text{cand}}^\top f(\mathbf{X}_{\text{cand}})\Big|\Big] \\ &\leq (\epsilon_{\text{vio}} K_{\max} + 2\epsilon_{\text{nys}} + \epsilon_{\text{LP}})\|f\|, \end{aligned} \tag{9}$$

where $\|f\|$ is the RKHS norm of f, $K_{\max} := \max_{x \in \boldsymbol{X}_{cand}} K(x,x)^{1/2}$, and $\epsilon_{\text{vio}} := 1 - \mathbf{w}_{\text{cand}}^\top q(\mathbf{x}_{\text{cand}})$ is the expected violation rate with respect to the empirical measure given by $(\mathbf{w}_{\text{cand}}, \mathbf{X}_{\text{cand}})$.

The proof is given in Supplementary A. This proposition indicates that we can obtain a quantitative estimate of the two tasks described in (c) concurrently. We can attain at least the expected reward of the original batch while ensuring that the resulting measure (which may not necessarily be probabilistic) integrates the functions in the RKHS within a proven error.

3.8 How to Solve The LP Problem

We used Gurobi (Gurobi Optimization, LLC, 2024) to solve the LP problem. We used the randomized singular value decomposition to eigendecompose the Gram matrix (Halko et al., 2011) with M-point samples $\mathbf{X}_{\text{nys}} \subset \mathbf{X}_{\text{cand}}$. We set $\epsilon_{\text{LP}} = 10^{-8}$ as the lower bound to avoid LP failure due to the randomness of μ. The complexity of this computation is lower than $\mathcal{O}(NM + M^2 \log n + Mn^2 \log(N/n))$ (Hayakawa et al., 2022).

Probability function q A probability function q can be a given constraint function (Gardner et al., 2014), or estimated function as another GP (Gelbart et al., 2014) . If there is no constraints, we can simply set $q(x) = 1$, then it becomes standard batch AL, BQ, or BO.

Reward function g A reward function g is for an additional flexibility to incorporate the information. If we do not have particularly informative information to add, we can simply set as $g = 1$. We can view g as the soft constraint of the objective. We can set g for another acquisition function, or prior knowledge of global optimum such as Hvarfner et al. (2022); Adachi et al. (2024).

4 Related Work

Batch Active Learning and Optimization There are a wide variety of batch methods has been proposed: (1) batch AL; for kernels (Kremer et al., 2014; Joshi et al., 2009; Leskovec et al., 2007; Riis et al., 2022), deep learning (Gal et al., 2017; Kirsch et al., 2019; Sener & Savarese, 2018; Pinsler et al., 2019). (2) batch BQ (Wagstaff et al., 2018; Adachi et al., 2022, 2023b), (3) batch BO, a greedy extension of sequential algorithms (Azimi et al., 2010; González et al., 2016; Eriksson et al., 2019; Balandat et al., 2020), diversified batch with determinantal point process (DPP) (Kathuria et al., 2016; Nava et al., 2022). Constrained batch sampling has been researched in BO (Hernández-Lobato et al., 2016; Letham et al., 2019; Eriksson & Poloczek, 2021). However, most do not discuss the quality of batch construction, like KQ methods. The adaptive batch size setting only found in BO (Nguyen et al., 2016), to the best of our knowledge.

Kernel Quadrature There are a number of KQ algorithms; herding/optimization (Chen et al., 2010; Bach et al., 2012; Huszár & Duvenaud, 2012), random sampling (Bach, 2017; Belhadji et al., 2019), DPP (Belhadji et al., 2019; Belhadji, 2021), kernel thinning (Dwivedi & Mackey, 2021, 2022), recombination (Hayakawa et al., 2022, 2023), kernel Stein discrepancy (Chen et al., 2018, 2019; Teymur et al., 2021), randomly pivoted Cholesky (Epperly & Moreno, 2023). While any KQ algorithms can be used to solve our problems, we focused on the recombination algorithm due to its flexibility.

5 Experiments

We evaluate our new algorithm, *AdaBatAL*, on synthetic and real-world tasks on batch AL and BO, with and without probabilistic constraints. We implemented AdaBatAL using PyTorch (Paszke et al., 2019), GPyTorch (Gardner et al., 2018), BoTorch (Balandat et al., 2020), and SOBER (Adachi et al., 2023a). All experiments were averaged over 10 repeats, and performed on a laptop[11]. We fix the number of initial random samples for objective queries to $n_{\text{obj}} = 10$. The details on experimental conditions and background on real-world examples are summarized in Supplementary B.

[11] Performed on MacBook Pro 2019, 2.4 GHz 8-Core Intel Core i9, 64 GB 2667 MHz DDR4

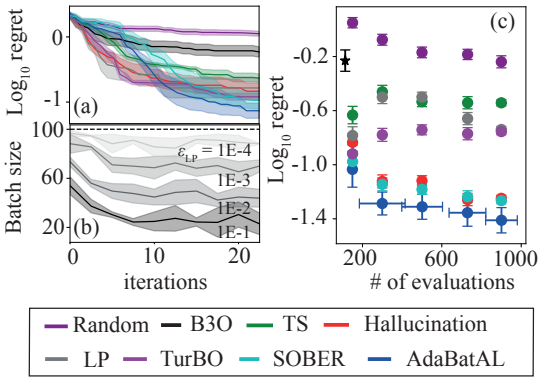


Figure 3: Batch Bayesian optimization results on Hartmann ($d = 6$): (a) convergence plot with ($n \leq 5$). (b) batch size variability ($n \leq 100$). The tolerance is set ($\epsilon = 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}$). (c) Total queries vs. simple regret at the last iteration results of (a)(b). For fixed batch size methods, the mean batch size of AdaBatAL is used ($n = 5, 30, 50, 73, 90$). The plot shows mean $\pm$ standard error of the mean.

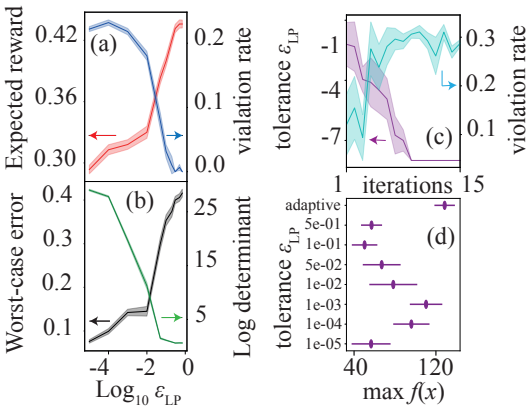


Figure 4: Tolerance effect on constrained batch BO on Branin ($d = 2$): the balance between (a) violation rate and expected reward, and (b) worst-case error and log determinant. (c) Tolerance adaptively controls violation rate, and (d) outperforms the fixed cases. (a)(b)(c) are the two Y-axis plots where the color and arrow indicate which Y axis to see.

5.1 Efficacy of Adaptive Batch Size

We first investigate the effect of the adaptive batch size itself *without* unknown constraints, namely, $q(x) = 1$. To compare with the only baseline of the adaptive batch size method, B3O (Nguyen et al., 2016), we selected the batch BO setting. We compared AdaBatAL with the 6 popular baselines of batch BO; B3O, Thompson sampling (TS) (Kandasamy et al., 2018), hallucination (Azimi et al., 2010), local penalization (LP)[12] (González et al., 2016), TurBO (Eriksson et al., 2019), SOBER (Adachi et al., 2023a).

Figure 3 illustrates that AdaBatAL consistently outperformed the baselines throughout the experiments. An increase in the tolerance ϵ_{LP} results in a reduced batch size. Over iterations, the batch size decreases for all values of ϵ_{LP}. This indicates that AdaBatAL initially needs more exploratory samples, then it squeezes its search space for exploitation. When matched against fixed batch size methods with a total cost, AdaBatAL achieves a lower regret for the same budget, even when compared to the original SOBER. While B3O tends to opt for a small batch size of around 4, AdaBatAL can adjust its batch size by ϵ_{LP}.

5.2 Efficacy of Expected Violation Rate

We empirically examine the role of expected violation rate ϵ_{vio} in constrained BO as the time-varying tolerance ϵ_{LP}. Figure 4 presents the main findings.

[12]Only within this section 5.1, LP refers to local penalization. Otherwise, LP means linear programming.

Four key metrics

(1) *The expected reward* (LP objective): the proxy for how safely we explore.
(2) *The violation rate* $1-|\tilde{\mathbf{X}}_{\mathrm{batch}}|/|\mathbf{X}_{\mathrm{batch}}|$: the proxy for actual results on how safely we explore.
(3) *The worst-case error* $\mathrm{wce}(Q_{\tilde{\mathbf{w}}_{\mathrm{batch}}, \tilde{\mathbf{X}}_{\mathrm{batch}}})$: the precision of quadrature.
(4) *log determinant* $\log|K(\tilde{\mathbf{X}}_{\mathrm{batch}}, \tilde{\mathbf{X}}_{\mathrm{batch}})|$: the proxy for how diversely we explore.

We examined the impact of ϵ_{LP} on four key metrics, as discussed in § 3.6. We aligned ϵ_{LP} with ϵ_{vio} to facilitate *adaptive* exploration relative to the specified risk level ϵ_{vio}. The x-axis in Figures (a) and (b) represents variations in ϵ_{vio}. At higher risk levels, it is essential to prioritize safety. Consequently, there is an increase in the expected reward, correlating with a higher likelihood of constraint satisfaction. This relationship is evident through a reduction in the violation rate, which signifies safer exploration practices. The numerical metrics give insights of safety exploration in the numerical level: High tisk leads to an increase in the worst-case error, which reflects a relaxation in precision requirements. A smaller log determinant suggests less diversity in batch samples, indicated by the proximity of the selected points $\mathbf{X}_{\mathrm{cand}}$ to each other. Conversely, at lower risk levels, we observe a trend towards more optimistic and exploratory sampling. Hence, our findings confirm that by setting $\epsilon_{\mathrm{LP}} = \epsilon_{\mathrm{vio}}$, our batch exploration successfully adapts to varying risk levels.

We further examined the evolution of the expected violation rate ϵ_{vio} during the optimization loop. As depicted in Figure 4 (c), the expected violation rate

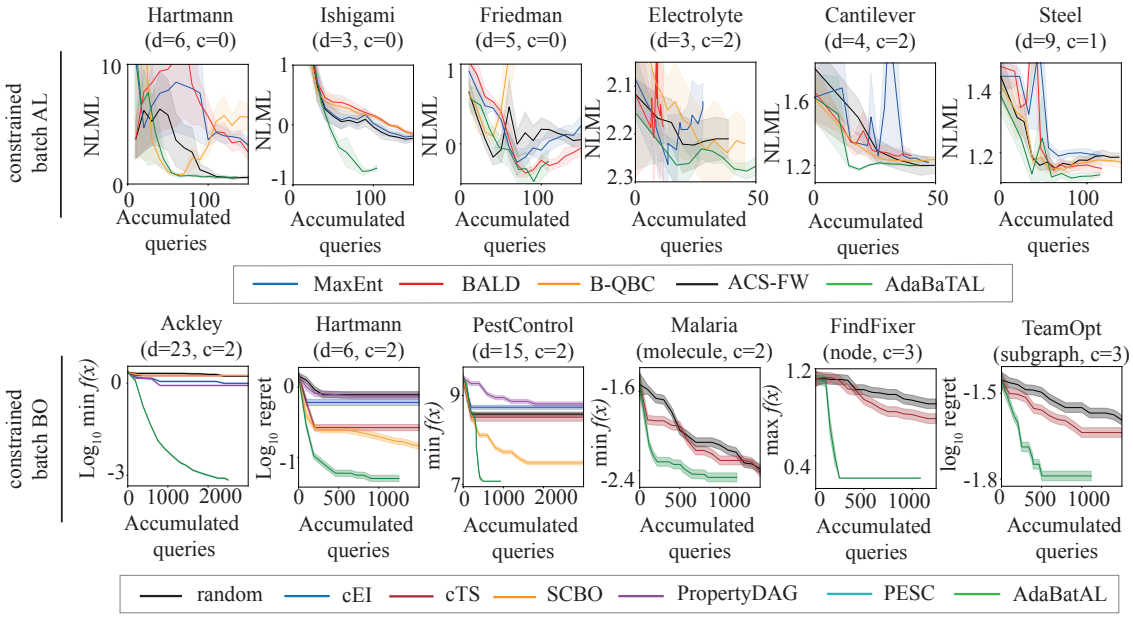


Figure 5: Convergence plot of both constrained batch active learning and Bayesian optimization results across 5 synthetic functions and 7 real-world tasks . d is the dimension, c is the number of unknown constraints. Negative log marginal likelihood (NLML) for active learning tasks, log regret or log best observations for optimization task. Lines and shaded area denote mean $\pm$ 1 standard error.

$\epsilon_{\text{vio}} = \epsilon_{\text{LP}}$ begins high and diminishes to a minimal value over time. This trend suggests an initial emphasis on safely gathering data, transitioning to greater exploration later on. This approach mirrors strategies like 'safe' BO (Sui et al., 2015), which has demonstrated strong empirical performance (e.g., Figure 4 in Xu et al. (2023)) backed by theoretical guarantee. The adaptive tolerance inherently exhibits this behavior with adaptive batch size. Moreover, Figure 4 (d) indicates that adaptive tolerance converges more rapidly than fixed versions. Notably, the most effective fixed tolerance was $\epsilon_{\text{LP}} = 10^{-3}$, suggesting that even in the absence of adaptive tolerance, AdaBatAL outperforms the original SOBER ($\epsilon_{\text{LP}} = 0$) under constraints.

5.3 Empirical Evaluation

We tested AdaBatAL's empirical performance across diverse tasks. For batch AL, we compared against five baselines: MaxEnt (MacKay, 1992), BALD (Houlsby et al., 2011; Kirsch et al., 2019), B-QBC (Riis et al., 2022), and ACS-FW (Pinsler et al., 2019). We evaluated on three synthetic and three real-world tasks. For batch BO, we also explored constrained batch BO and compared against five popular baselines: random, cEI (Letham et al., 2019), PESC (Hernández-Lobato et al., 2016), SCBO (Eriksson & Poloczek, 2021), and cTS (Eriksson & Poloczek, 2021). The Malaria, FindFixer, and TeamOpt tasks involve non-continuous inputs over non-Euclidean spaces, each requiring specialized kernels (Tanimoto kernel (Ralaivola et al., 2005) for molecules and the diffusion graph kernel (Zhi et al., 2023) for graphs). Due to this unique and crucial real-world setting, the only comparable baselines were random and cTS. Others utilized a standard GP with an RBF kernel. It is important to note that this is *constrained* batch BO, which differs from normal batch BO. Typically, constrained batch BO extends the standard acquisition function with regular batch methods. We chose cEI and cTS as representative methods for these approaches. More details are available in Supplementary B. Figure 5 shows AdaBatAL's strong empirical performance.

6 Discussion and Limitations

We introduced AdaBatAL, a versatile approach capable of adaptive batch sizes under probabilistic constraints for both AL and BO. It is also applicable for non-continuous inputs (e.g., strings for drug discovery and graphs for social data) and arbitrary acquisition functions as the reward function. AdaBatAL is best suited for batch sizes larger than three and does not support asynchronous batch settings (Kandasamy et al., 2018). Its efficacy in high-dimensional BO, which often faces challenges with slow eigenvalue decay, remains an open problem. However, the error bounds of the Nyström method are not directly related to dimensionality; rapid convergence is possible if the function exhibits fast eigenvalue decay, as in the case of the Ackley function.

Acknowledgements

We thank Samuel Daulton, Siu Lun Chau, Wenjie Xu, and Pierre Osselin for helpful feedback on our paper, and anonymous reviewers who gave useful comments. Masaki Adachi was supported by the Clarendon Fund, the Oxford Kobe Scholarship, the Watanabe Foundation, and Toyota Motor Corporation. Satoshi Hayakawa was supported by the Clarendon Fund, the Oxford Kobe Scholarship, and the Toyota Riken Overseas Scholarship. Harald Oberhauser was supported by the DataSig Program [EP/S026347/1], the Hong Kong Innovation and Technology Commission (InnoHK Project CIMDA), and the Oxford-Man Institute. Martin Jørgensen was partly supported by the Research Council of Finland (grant 356498).

References

Masaki Adachi. High-dimensional discrete Bayesian optimization with self-supervised representation learning for data-efficient materials exploration. In *NeurIPS 2021 AI for Science Workshop*, 2021. doi: https://openreview.net/forum?id=xJhjehqjQeB.

Masaki Adachi, Satoshi Hayakawa, Martin Jørgensen, Harald Oberhauser, and Michael A Osborne. Fast Bayesian inference with batch Bayesian quadrature via kernel recombination. *Advances in Neural Information Processing Systems*, 35, 2022. doi: https://doi.org/10.48550/arXiv.2206.04734.

Masaki Adachi, Satoshi Hayakawa, Saad Hamid, Martin Jørgensen, Harald Oberhauser, and Michael A Osborne. SOBER: Highly parallel Bayesian optimization and Bayesian quadrature over discrete and mixed spaces. In *ICML 2023 Workshop: Sampling and Optimization in Discrete Space*, 2023a. doi: https://doi.org/10.48550/arXiv.2301.11832.

Masaki Adachi, Yannick Kuhn, Birger Horstmann, Arnulf Latz, Michael A Osborne, and David A Howey. Bayesian model selection of lithium-ion battery models via Bayesian quadrature. *IFAC-PapersOnLine*, 56(2):10521–10526, 2023b. doi: https://doi.org/10.1016/j.ifacol.2023.10.1073.

Masaki Adachi, Brady Planden, David A Howey, Krikamol Muandet, Michael A Osborne, and Siu Lun Chau. Looping in the human: Collaborative and explainable Bayesian optimization. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2024. doi: https://doi.org/10.48550/arXiv.2310.17273.

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

Javad Azimi, Alan Fern, and Xiaoli Fern. Batch Bayesian optimization via simulation matching. *Advances in Neural Information Processing Systems*, 23, 2010.

F. Bach. On the equivalence between kernel quadrature rules and random feature expansions. *Journal of Machine Learning Research*, 18:714, 2017.

F. Bach, S. Lacoste-Julien, and G. Obozinski. On the equivalence between herding and conditional gradient algorithms. In *International Conference on Machine Learning (ICML)*, 2012.

Maximilian Balandat, Brian Karrer, Daniel Jiang, Samuel Daulton, Ben Letham, Andrew G Wilson, and Eytan Bakshy. BoTorch: a framework for efficient Monte-Carlo Bayesian optimization. *Advances in neural information processing systems*, 33:21524–21538, 2020.

Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439): 509–512, 1999.

A. Belhadji, R. Bardenet, and P. Chainais. Kernel quadrature with DPPs. In *International Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

Ayoub Belhadji. An analysis of ermakov-zolotukhin quadrature using kernels. *Advances in Neural Information Processing Systems*, 34:27278–27289, 2021.

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. *The Journal of Machine Learning Research*, 20(1):973–978, 2019.

Mark S Butler. The role of natural product chemistry in drug discovery. *Journal of natural products*, 67 (12):2141–2153, 2004.

Jerry F Casteel and Edward S Amis. Specific conductance of concentrated solutions of magnesium salts in water-ethanol system. *Journal of Chemical and Engineering Data*, 17(1):55–59, 1972.

Wilson Ye Chen, Lester Mackey, Jackson Gorham, François-Xavier Briol, and Chris Oates. Stein points. In *International Conference on Machine Learning*, pp. 844–853. PMLR, 2018.

Wilson Ye Chen, Alessandro Barp, François-Xavier Briol, Jackson Gorham, Mark Girolami, Lester Mackey, and Chris Oates. Stein point Markov chain Monte Carlo. In *International Conference on Machine Learning*, pp. 1011–1021. PMLR, 2019.

Y. Chen, M. Welling, and A. Smola. Super-samples from kernel herding. In *International Conference on Uncertainty in Artificial Intelligence (UAI)*, 2010.

George B Dantzig. Linear programming. *Operations research*, 50(1):42–47, 2002.

Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. Differentiable expected hypervolume improvement for parallel multi-objective Bayesian optimization. *Advances in Neural Information Processing Systems*, 33:9851–9864, 2020.

Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. Parallel Bayesian optimization of multiple noisy objectives with expected hypervolume improvement. *Advances in Neural Information Processing Systems*, 34:2187–2200, 2021.

Petros Drineas and Michael W. Mahoney. On the Nyström method for approximating a Gram matrix for improved kernel-based learning. *Journal of Machine Learning Research*, 6(72):2153–2175, 2005.

Raaz Dwivedi and Lester Mackey. Kernel thinning. In *Proceedings of Thirty Fourth Conference on Learning Theory*, volume 134 of *Proceedings of Machine Learning Research*, pp. 1753–1753. PMLR, 15–19 Aug 2021.

Raaz Dwivedi and Lester Mackey. Generalized kernel thinning. In *International Conference on Learning Representations*, 2022.

Ethan N Epperly and Elvira Moreno. Kernel quadrature with randomly pivoted Cholesky. *arXiv preprint arXiv:2306.03955*, 2023.

David Eriksson and Matthias Poloczek. Scalable constrained Bayesian optimization. In *International Conference on Artificial Intelligence and Statistics*, pp. 730–738. PMLR, 2021.

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.

Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. Efficient and robust automated machine learning. *Advances in neural information processing systems*, 28, 2015.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep Bayesian active learning with image data. In *International conference on machine learning*, pp. 1183–1192. PMLR, 2017.

Jacob Gardner, Geoff Pleiss, Kilian Q Weinberger, David Bindel, and Andrew G Wilson. GPyTorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration. In *Advances in Neural Information Processing Systems*, pp. 7576–7586, 2018.

Jacob R Gardner, Matt J Kusner, Zhixiang Eddie Xu, Kilian Q Weinberger, and John P Cunningham. Bayesian optimization with inequality constraints. In *ICML*, volume 2014, pp. 937–945, 2014.

Michael A Gelbart, Jasper Snoek, and Ryan P Adams. Bayesian optimization with unknown constraints. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, pp. 250—-259, 2014. doi: https://doi.org/10.48550/arXiv.1403.5607.

Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.

Javier González, Zhenwen Dai, Philipp Hennig, and Neil Lawrence. Batch Bayesian optimization via local penalization. In *Artificial intelligence and statistics*, pp. 648–657. PMLR, 2016.

Siegfried Graf and Harald Luschgy. *Foundations of quantization for probability distributions*. Springer, 2007.

Ryan-Rhys Griffiths, Leo Klarner, Henry Moss, Aditya Ravuri, Sang T Truong, Bojana Rankovic, Yuanqi Du, Arian Rokkum Jamasb, Julius Schwartz, Austin Tripp, et al. GAUCHE: A library for Gaussian processes in chemistry. In *ICML 2022 2nd AI for Science Workshop*, 2022.

Carlos Guestrin, Andreas Krause, and Ajit Paul Singh. Near-optimal sensor placements in Ggaussian processes. In *Proceedings of the 22nd international conference on Machine learning*, pp. 265–272, 2005.

Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2024. URL `https://www.gurobi.com`.

N. Halko, P. G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

Satoshi Hayakawa, Harald Oberhauser, and Terry Lyons. Positively weighted kernel quadrature via subsampling. *Advances in Neural Information Processing Systems*, 35:6886–6900, 2022.

Satoshi Hayakawa, Harald Oberhauser, and Terry Lyons. Sampling-based Nyström approximation and kernel quadrature. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 12678–12699, 2023.

Philipp Hennig, Michael A Osborne, and Mark Girolami. Probabilistic numerics and uncertainty in computations. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 471(2179):20150142, 2015.

Philipp Hennig, Michael A Osborne, and Hans P Kersting. *Probabilistic Numerics: Computation as Machine Learning*. Cambridge University Press, 2022.

José Miguel Hernández-Lobato, Matthew W Hoffman, and Zoubin Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. *Advances in neural information processing systems*, 27, 2014.

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*, pp. 1699–1707. PMLR, 2015.

José Miguel Hernández-Lobato, Michael A. Gelbart, Ryan P. Adams, Matthew W. Hoffman, and Zoubin Ghahramani. A general framework for constrained Bayesian optimization using information-based search. *Journal of Machine Learning Research*, 17(1):5549–5601, 2016.

José Miguel Hernández-Lobato, James Requeima, Edward O Pyzer-Knapp, and Alán Aspuru-Guzik. Parallel and distributed Thompson sampling for large-scale accelerated exploration of chemical space. In *International conference on machine learning*, pp. 1470–1479. PMLR, 2017.

Matthew D Hoffman and Andrew Gelman. The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *J. Mach. Learn. Res.*, 15 (1):1593–1623, 2014.

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

Ferenc Huszár and David Duvenaud. Optimally-weighted herding is Bayesian quadrature. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, pp. 377—-386, 2012. doi: https://doi.org/10.48550/arXiv.1204.1664.

Carl Hvarfner, Danny Stoll, Artur Souza, Marius Lindauer, Frank Hutter, and Luigi Nardi. πBO: Augmenting acquisition functions with user beliefs for bayesian optimization. In *International Conference on Learning Representations*, 2022.

Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13:455–492, 1998.

Ajay J Joshi, Fatih Porikli, and Nikolaos Papanikolopoulos. Multi-class active learning for image classification. In *2009 ieee conference on computer vision and pattern recognition*, pp. 2372–2379. IEEE, 2009.

Kirthevasan Kandasamy, Akshay Krishnamurthy, Jeff Schneider, and Barnabás Póczos. Parallelised Bayesian optimisation via Thompson sampling. In *International Conference on Artificial Intelligence and Statistics*, pp. 133–142. PMLR, 2018.

Toni Karvonen. *Kernel-based and Bayesian methods for numerical integration*. PhD thesis, Aalto University, 2019.

Tarun Kathuria, Amit Deshpande, and Pushmeet Kohli. Batched Gaussian process bandit optimization via determinantal point processes. *Advances in Neural Information Processing Systems*, 29, 2016.

Danial Khatamsaz, Brent Vela, Prashant Singh, Duane D Johnson, Douglas Allaire, and Raymundo Arróyave. Bayesian optimization with active learning of design constraints using an entropy-based approach. *npj Computational Materials*, 9(1):49, 2023.

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.

Christine Kiss and Martin Bichler. Identification of influencers—measuring influence in customer networks. *Decision Support Systems*, 46(1):233–253, 2008.

Andreas Krause and Carlos E Guestrin. Near-optimal nonmyopic value of information in graphical models. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, pp. 324–331, 2012. doi: https://doi.org/10.48550/arXiv.1207.1394.

Jan Kremer, Kim Steenstrup Pedersen, and Christian Igel. Active learning with support vector machines. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4(4):313–326, 2014.

Sanjiv Kumar, Mehryar Mohri, and Ameet Talwalkar. Sampling methods for the Nyström method. *Journal of Machine Learning Research*, 13(Apr):981–1006, 2012.

Norbert Kuschel and Rüdiger Rackwitz. Two basic problems in reliability-based structural optimization. *Mathematical Methods of Operations Research*, 46: 309–333, 1997.

Vidhi Lalchand and Carl Edward Rasmussen. Approximate inference for fully Bayesian Gaussian process regression. In *Symposium on Advances in Approximate Bayesian Inference*, pp. 1–12. PMLR, 2020.

Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 420–429, 2007.

Benjamin Letham, Brian Karrer, Guilherme Ottoni, and Eytan Bakshy. Constrained Bayesian optimization with noisy experiments. *Bayesian Analysis*, 14 (2):495 – 519, 2019. doi: 10.1214/18-BA1110.

Shibo Li, Jeff M Phillips, Xin Yu, Robert Kirby, and Shandian Zhe. Batch multi-fidelity active learning with budget constraints. *Advances in Neural Information Processing Systems*, 35:995–1007, 2022.

Christopher A Lipinski, Franco Lombardo, Beryl W Dominy, and Paul J Feeney. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Advanced drug delivery reviews*, 23(1-3):3–25, 1997.

Dong C Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.

ER Logan, Erin M Tonita, KL Gering, Jing Li, Xiaowei Ma, LY Beaulieu, and JR Dahn. A study of the physical properties of Li-ion battery electrolytes containing esters. *Journal of The Electrochemical Society*, 165(2):A21, 2018.

Turab Lookman, Prasanna V Balachandran, Dezhen Xue, and Ruihao Yuan. Active learning in materials science with emphasis on adaptive sampling using uncertainties for targeted design. *npj Computational Materials*, 5(1):21, 2019.

David JC MacKay. Information-based objective functions for active data selection. *Neural computation*, 4(4):590–604, 1992.

Masahiro Mochizuki, Shogo D Suzuki, Keisuke Yanagisawa, Masahito Ohue, and Yutaka Akiyama. QEX: target-specific druglikeness filter enhances ligand-based virtual screening. *Molecular Diversity*, 23: 11–18, 2019.

Krikamol Muandet, Kenji Fukumizu, Bharath Sriperumbudur, Bernhard Schölkopf, et al. Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends® in Machine Learning*, 10 (1-2):1–141, 2017.

Elvis Nava, Mojmir Mutny, and Andreas Krause. Diversified sampling for batched Bayesian optimization with determinantal point processes. In *International Conference on Artificial Intelligence and Statistics*, pp. 7031–7054. PMLR, 2022.

Vu Nguyen, Santu Rana, Sunil K Gupta, Cheng Li, and Svetha Venkatesh. Budgeted batch Bayesian optimization. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pp. 1107–1112. IEEE, 2016.

Changyong Oh, Jakub Tomczak, Efstratios Gavves, and Max Welling. Combinatorial Bayesian optimization using the graph Cartesian product. *Advances in Neural Information Processing Systems*, 32, 2019.

Anthony O'Hagan. Bayes–Hermite quadrature. *Journal of statistical planning and inference*, 29(3):245–260, 1991.

Ji Won Park, Samuel Stanton, Saeed Saremi, Andrew Watkins, Henri Dwyer, Vladimir Gligorijevic, Richard Bonneau, Stephen Ra, and Kyunghyun Cho. PropertyDAG: Multi-objective Bayesian optimization of partially ordered, mixed-variable properties for biological sequence design. In *NeurIPS 2022 AI for Science: Progress and Promises*, 2022.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, and Luca Antiga. PyTorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Robert Pinsler, Jonathan Gordon, Eric Nalisnick, and José Miguel Hernández-Lobato. Bayesian batch active learning as sparse subset approximation. *Advances in neural information processing systems*, 32, 2019.

Liva Ralaivola, Sanjay J Swamidass, Hiroto Saigo, and Pierre Baldi. Graph kernels for chemical informatics. *Neural networks*, 18(8):1093–1110, 2005.

Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B Gupta, Xiaojiang Chen, and Xin Wang. A survey of deep active learning. *ACM computing surveys (CSUR)*, 54(9):1–40, 2021.

Robert R Richardson, Michael A Osborne, and David A Howey. Gaussian process regression for forecasting battery state of health. *Journal of Power Sources*, 357:209–219, 2017.

Christoffer Riis, Francisco Antunes, Frederik Hüttel, Carlos Lima Azevedo, and Francisco Pereira. Bayesian active learning with fully Bayesian Gaussian processes. *Advances in Neural Information Processing Systems*, 35:12141–12153, 2022.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018.

Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

H Sebastian Seung, Manfred Opper, and Haim Sompolinsky. Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 287–294, 1992.

Il'ya Meerovich Sobol'. On the distribution of points in a cube and the approximate evaluation of integrals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, 7(4):784–802, 1967.

Thomas Spangenberg, Jeremy N Burrows, Paul Kowalczyk, Simon McDonald, Timothy NC Wells, and Paul Willis. The open access malaria box: a drug discovery catalyst for neglected diseases. *PloS one*, 8(6):e62906, 2013.

Bharath K Sriperumbudur, Arthur Gretton, Kenji Fukumizu, Bernhard Schölkopf, and Gert RG Lanckriet. Hilbert space embeddings and metrics on probability measures. *The Journal of Machine Learning Research*, 11:1517–1561, 2010.

Yanan Sui, Alkis Gotovos, Joel Burdick, and Andreas Krause. Safe exploration for optimization with Gaussian processes. In *International conference on machine learning*, pp. 997–1005. PMLR, 2015.

Onur Teymur, Jackson Gorham, Marina Riabiz, and Chris Oates. Optimal quantisation of probability measures using maximum mean discrepancy. In *International Conference on Artificial Intelligence and Statistics*, pp. 1027–1035. PMLR, 2021.

Daniel F Veber, Stephen R Johnson, Hung-Yuan Cheng, Brian R Smith, Keith W Ward, and Kenneth D Kopple. Molecular properties that influence the oral bioavailability of drug candidates. *Journal of medicinal chemistry*, 45(12):2615–2623, 2002.

Ed Wagstaff, Saad Hamid, and Michael Osborne. Batch selection for parallelisation of Bayesian quadrature. *arXiv preprint arXiv:1812.01553*, 2018.

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 *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 10663–10674, 2021.

Xingchen Wan, Pierre Osselin, Henry Kenlay, Binxin Ru, Michael A. Osborne, and Xiaowen Dong. Bayesian optimisation of functions on graphs. *arXiv preprint arXiv:2306.05304*, 2023.

Christopher Williams and Matthias Seeger. Using the Nyström method to speed up kernel machines. *Advances in neural information processing systems*, 13, 2000.

Jian Wu, Saul Toscano-Palmerin, Peter I Frazier, and Andrew Gordon Wilson. Practical multi-fidelity Bayesian optimization for hyperparameter tuning. In *Uncertainty in Artificial Intelligence*, pp. 788–798. PMLR, 2020.

Y-T Wu, Youngwon Shin, Robert Sues, and Mark Cesare. Safety-factor based approach for probability-based design optimization. In *19th AIAA applied aerodynamics conference*, pp. 1522, 2001.

Wenjie Xu, Yuning Jiang, Bratislav Svetozarevic, and Colin Jones. Constrained efficient global optimization of expensive black-box functions. In *International Conference on Machine Learning*, pp. 38485–38498. PMLR, 2023.

Yin-Cong Zhi, Yin Cheng Ng, and Xiaowen Dong. Gaussian processes on graphs via spectral kernel learning. *IEEE Transactions on Signal and Information Processing over Networks*, 2023.

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes in Supplementary]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Not Applicable]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Not Applicable]

(c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]

(d) Information about consent from data providers/curators. [Not Applicable]

(e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

(a) The full text of instructions given to participants and screenshots. [Not Applicable]

(b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]

(c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Endnote

Clarification

The Appendix is found in Section B.

Errata

In Section 3.1, we explained that the Bayesian active learning formulation follows the same sparse sampling strategy as Pinsler et al. [181]. However, their formulation is based on integral variance reduction in hyperparameter space, not in function space, and thus differs from ours. Our approach focuses on IVR in the expected function space $\mathcal{H}_{\tilde{C}_t}$, as explained in the Preface. In Section 3.6, a typo ϵLP should be ϵ_{LP}. In Section 5.2, a typo 'high tisk' should be 'high risk'.

Future direction

As noted in the errata, our current formulation of Bayesian active learning operates in the expected function space, $\mathcal{H}_{\tilde{C}_t}$. Extending this to IVR in hyperparameter space could make our approach more aligned with Bayesian experimental design principles (Rainforth et al. [185]).

In the hyperparameter space approach, we aim to model the *hyperposterior predictive distribution*, $\mathbb{P}(\theta \mid x, \mathbf{D}_t)$. Typically, this does not make sense because the hyperposterior is usually independent of the input, i.e., $\mathbb{P}(\theta \mid x, \mathbf{D}_t) = \mathbb{P}(\theta \mid \mathbf{D}_t)$. However, in active learning, the next query point x_t is generated by the acquisition policy, i.e., IVR $\mathbb{P}(x_t \mid \theta, \mathbf{D}_{t-1}) \propto \Delta C_t(\mathbf{X}_t^n, \mathbb{P}(\mathcal{S}) \mid \theta)$. Consequently, the generative distribution of the input can be conditioned on the hyperparameter, i.e.,

$$\mathbb{P}(\theta \mid x, \mathbf{D}_{t-1}) \propto \mathbb{P}(x_t \mid \theta, \mathbf{D}_{t-1})\mathbb{P}(\theta \mid \mathbf{D}_{t-1}) \propto \Delta C_t(\mathbf{X}_t^n, \mathbb{P}(\mathcal{S}) \mid \theta)\mathbb{P}(\theta \mid \mathbf{D}_{t-1}) \quad (4.5)$$

Using this, we can formulate the IVR in hyperparameter space as $\Delta H_t(\mathbf{X}_t^n) := \hat{Z}_{H_t}(\mathbf{X}_t^n) - \hat{Z}_{H_{t+1}}(\mathbf{X}_t^n \mid \mathbf{X}_{t+1}^n)$, as such:

$$\hat{Z}_{H_t} = \int_{\mathcal{X}} \mathbb{E}_{\mathbb{P}(\theta|\mathbf{D}_t)} \left[\mathbb{P}(\theta \mid x, \mathbf{D}_t)\right] \mathrm{d}\mathbb{P}(\mathcal{S}), \tag{4.6}$$

$$\propto \int_{\mathcal{X}} \mathbb{E}_{\mathbb{P}(\theta|\mathbf{D}_t)} \left[\Delta C_t(\mathbf{X}_t^n, \mathbb{P}(\mathcal{S}) \mid \theta)\mathbb{P}(\theta \mid \mathbf{D}_{t-1})\right] \mathrm{d}\mathbb{P}(\mathcal{S}), \tag{4.7}$$

$$\propto \int_{\mathcal{X}} \mathbb{E}_{g(\theta)} \left[\Delta C_t(\mathbf{X}_t^n, \mathbb{P}(\mathcal{S}) \mid \theta)\right] \mathrm{d}\mathbb{P}(\mathcal{S}), \tag{4.8}$$

$$\approx \int_{\mathcal{X}} \hat{C}_t(\mathbf{X}_t^n) \mathrm{d}\mathbb{P}(\mathcal{S}), \tag{4.9}$$

where

$$g(\theta) = \frac{\mathbb{P}(\theta \mid \mathbf{D}_t)^2}{\int_{\mathcal{X}} \mathbb{P}(\theta \mid \mathbf{D}_t)^2 \mathrm{d}\theta}, \tag{4.10}$$

$$\hat{C}_t(\mathbf{X}_t^n) = \mathbb{E}_{g(\theta)} \left[\Delta C_t(\mathbf{X}_t^n, \mathbb{P}(\mathcal{S}) \mid \theta)\right] \tag{4.11}$$

While this objective is more aligned with the Bayesian experimental approach Rainforth et al. [185], its adaptation to our quadrature approach is non-trivial, particularly because $\hat{Z}_{H_{t+1}}(\mathbf{X}_t^n \mid \mathbf{X}_{t+1}^n)$ requires a two-step lookahead. We anticipate that suitable approximation heuristics will be necessary to address this challenge. Or we can adopt Fisher kernel approach introduced in Khanna et al. [128].

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Adaptove Batch Sizes for Active Learning: A Probabilistic Numerics Approach
Publication Status	☑Published □ Accepted for Publication □Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	Masaki Adachi, Satoshi Hayakawa, Martin Jørgensen, Xingchen Wan, Vu Nyugen, Harald Oberhauser, and Michael A. Osborne. "Adaptive Batch Sizes for Active Learning: A Probabilistic Numerics Approach." In International Conference on Artificial Intelligence and Statistics (AISTATS) 238, 496-504, 2024.

Student Confirmation

Student Name:	Masaki Adachi		
Contribution to the Paper	first author I developed the core idea of addressing black-box optimization and Bayesian active learning as Bayesian quadrature through probabilistic lifting, incorporating the Probabilistic Numerics interpretation of quadrature precision tolerance as adaptive batch sizing. I implemented the corresponding code, conducted all experiments for the paper, performed additional analyses, and authored the manuscript, focusing primarily on the experimental and engineering aspects.		
Signature 		Date	06 January 2025

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Michael A. Osborne, Professor of Machine Learning		
Supervisor comments I can confirm that, to the best of my knowledge, Masaki's description above is fair, and that I have great trust in Masaki.		
Signature 	Date	3 February 2025

This completed form should be included in the thesis, at the end of the relevant chapter.

All models are wrong, but some are useful [30]

— George Box, Statistician and FRS

Everything should be made as simple as possible, but not simpler. [69]

— Albert Einstein, Nobel Laureate in Physics

5

Bayesian model selection of lithium-ion battery models via Bayesian quadrature

This chapter is based on the following publication:

> **Masaki Adachi**, Yannick Kuhn, Birger Horstmann, Arnulf Latz, Michael A. Osborne, and David A. Howey. Bayesian model selection of lithium-ion battery models via Bayesian quadrature. *IFAC-PapersOnLine* 56(2), 10521-10526, 2023.

In Chapter 5, we introduced our black-box inference approach to the Bayesian model selection task. Given simulators $\mathbb{P}(y \mid \theta, M)$ that generate the physical observations $\mathbf{D}$, our goal is to select the most plausible model based on the model evidence $\mathbb{P}(\mathbf{D} \mid M)$. We extended the method developed in Chapter 3 to adapt the log-likelihood space, effectively addressing the challenges posed by the extreme dynamic range of likelihood values. The estimated model evidence is then compared against popular selection metrics through correlation and variance analysis, confirming consistency with existing knowledge in the battery community. Additionally, our method achieves both the fastest computation and the highest sample efficiency. This contributes to advancing a greener society by addressing one of the suboptimality issues in battery control systems.

Bayesian Model Selection of Lithium-Ion Battery Models via Bayesian Quadrature

Masaki Adachi [*,**,***]
Yannick Kuhn [****,†] **Birger Horstmann** [****,†]
Arnulf Latz [****,†] **Michael A. Osborne** [*] **David A. Howey** [**,‡]

[*] *Machine Learning Research Group, University of Oxford, OX2 6ED, UK (e-mail: masaki@robots.ox.ac.uk).*
[**] *Battery Intelligence Lab, University of Oxford, OX1 3PJ, UK*
[***] *Toyota Motor Corporation, Shizuoka 410-1193, Japan*
[****] *German Aerospace Center (DLR), Pfaffenwaldring 38-40, 70569 Stuttgart, Germany*
Helmholtz Institute Ulm, Helmholtzstraße 11, 89081 Ulm, Germany
[†] *Universität Ulm, Albert-Einstein-Allee 47, 89081 Ulm, Germany*
[‡] *The Faraday Institution, Harwell Campus, Didcot OX11 0RA, UK*

Abstract: A wide variety of battery models are available, and it is not always obvious which model 'best' describes a dataset. This paper presents a Bayesian model selection approach using Bayesian quadrature. The model evidence is adopted as the selection metric, choosing the simplest model that describes the data, in the spirit of Occam's razor. However, estimating this requires integral computations over parameter space, which is usually prohibitively expensive. Bayesian quadrature offers sample-efficient integration via model-based inference that minimises the number of battery model evaluations. The posterior distribution of model parameters can also be inferred as a byproduct without further computation. Here, the simplest lithium-ion battery models, equivalent circuit models, were used to analyse the sensitivity of the selection criterion to given different datasets and model configurations. We show that popular model selection criteria, such as root-mean-square error and Bayesian information criterion, can fail to select a parsimonious model in the case of a multimodal posterior. The model evidence can spot the optimal model in such cases, simultaneously providing the variance of the evidence inference itself as an indication of confidence. We also show that Bayesian quadrature can compute the evidence faster than popular Monte Carlo based solvers.

Keywords: Bayesian, identifiability, system identification, estimation, battery, lithium-ion

1. INTRODUCTION

The lithium-ion battery is key to decarbonising power grids and electrifying vehicles. However, its behaviour can be challenging to model, control, and diagnose, and this is a practical hindrance to obtaining the optimal performance. This is compounded by the available data from operational batteries being typically limited to just three measurements: voltage, current, and temperature. Estimating internal states from these time-varying three variables is challenging or even mathematically impossible due to parameter identifiability issues (Bizeray et al., 2018). Degradation further complicates matters since the number of parameters to be identified becomes larger when considering long-term ageing.

There are dozens of plausible models for Li-ion batteries, owing to differing assumptions and levels of approximation. While electrochemists might prefer continuum models, such as the Doyle-Fuller-Newman model (Doyle et al., 1993), that give understanding of internal chemical reactions and transport, control engineers prefer simpler approaches such as equivalent circuit models (ECMs) (He et al., 2011), for fast control and fewer parameters. Other models exist in a spectrum between these (e.g. from simple to more complex: ECM → EHM (Milocco et al., 2014) → SPM (Santhanagopalan et al., 2006) → SPMe (Kemper et al., 2013) → DFN). System identification is the foundation of an estimation and control system, determining predictive accuracy, quick response, and reliability.

However, the 'best' model should be ascertained based on quantifiable performance metrics. Importantly, the optimal model strongly depends on the dataset $\mathbf{D}$ and user requirements. A widely accepted approach for defining 'good' models is Occam's razor, where the simplest model to reasonably reproduce a given dataset is considered the best. Simplest here relates to the number of parameters to be identified. Rasmussen et al. (2000) showed that such a metric could be evaluated via Bayesian model evidence, obtained for a model M by integrating out (i.e. averaging over) the parameters θ from the likelihood,

[⋆] The authors acknowledge support by Toyota Motor Corporation, Oxford Clarendon Fund, Oxford Kobe scholarship, the German Aerospace Center (DLR), and the Helmholtz Association through grant no KW-BASF-6, and contributes to the research performed at CELEST

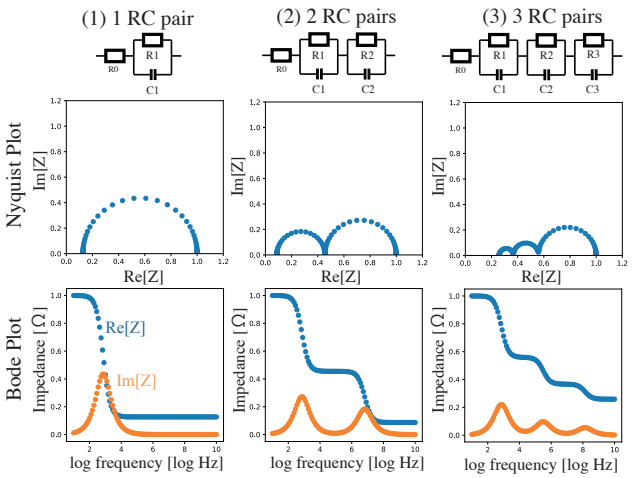


Fig. 1. Model selection from three RC pair models.

$$p(\mathbf{D}|M) = \int p(\mathbf{D}|\theta, M)\mathrm{d}p(\theta), \tag{1}$$

where $p(\theta)$ is the prior distribution and $p(\mathbf{D}|\theta, M)$ is the likelihood. The mean evidence $\mathbb{E}[p(\mathbf{D}|M)]$ gives the probability of reproducing a given dataset $\mathbf{D}$ with a given model M, the degree of model fit penalised by model complexity. The variance quantifies its uncertainty.

Surprisingly, Bayesian model selection of battery models has barely been reported, except for Miyazaki et al. (2020). Although Bayesian parameter estimation (Aitio et al., 2020; Escalante et al., 2021), and probabilistic modelling works (Huang et al., 2021; Liu et al., 2020) exist, most Bayesian approaches in the battery community use Markov chain Monte Carlo (MCMC) (Metropolis et al., 1953; Hastings, 1970), a user-friendly but sample-inefficient approach for inference. Recent work (Kuhn et al., 2022) on parameterisation applied a sample-efficient solver with Bayesian optimisation, none of the above solvers offer evidence computation. This is because estimating the evidence requires prohibitive integral computation, and this is particularly challenging when the likelihood is non-closed-form and/or expensive. A typical practice in such cases is to adopt the Bayesian information criterion (BIC), which is a coarse approximation of the evidence that assumes the posterior is a unimodal Gaussian. Unfortunately battery parameter estimation can produce multimodal or non-Gaussian posterior distributions (Aitio et al., 2020; Escalante et al., 2021), and ignoring this may cause overconfidence—previous work (Miyazaki et al., 2020) demonstrates that the identification of the best model using a variant of BIC gradually worsens as the posterior multimodality increases. This paper introduces Bayesian quadrature (BQ) as a novel technique for sample-efficient model evidence *and* parameter posterior estimation, and applies this to battery equivalent circuit models using synthetic data for demonstration purposes.

2. BATTERY MODEL FORMULATION

We selected ECMs for proof-of-concept here since they are relatively simple battery models that nonetheless offer identification challenges. Parameter identifiability for ECMs is often examined in the frequency domain, for example via electrochemical impedance spectroscopy (EIS) data, although time domain data may also be used. Several plausible ECMs are usually compared when fitting EIS data, but the process is subjective and based on the user's electrochemical understanding of the target battery. For simplicity, we chose a simple resistance-capacitance (RC) pair model—this may represent various physical processes, for example kinetics and double layer capacitance, or an approximation of diffusion. Fig. 1 illustrates the circuit configurations and typical Nyquist plots of three variations of RC circuit models. The number of RC parallel connection components corresponds to the number of the semi-circles in a Nyquist plot. This correspondence is key for identifying the model from spectra. As the semi-circle shape implies, the real and imaginary parts of spectra have a mathematical relationship (Kramers-Kronig relations in Debye relaxation), where one part of spectra can be derived from the other via an equation.

We extend this formulation to make the model better suited for statistical inference using the hyperbolic formulation (Calderwood, 2003); we improve this, permitting non-dimensionalised parameterisation without positivity constraint, as follows. For a general circuit with N total RC pairs plus an additional series resistance R_0, where R_i is the resistance of i-th RC pair [Ω], $\ln(\omega\tau_i)$ is the rescaled frequency scale to make the scale independent of the given frequency range of the dataset, rescaled with the breakpoint frequency $\omega_i := 1/\tau_i$ [rad/s], $\tau_i := R_i C_i$ is the time constant of the i-th RC pair [s], C_i is the capacitance of the i-th RC-pair [F], f is the frequency [Hz] and $\omega := 2\pi f$ is the angular frequency [rad/s], one can define the total resistance R_{total}, the log of this r_{total} (which is positive), the dimensionless resistance of i-th RC-pair r_i (constrained between zero and one), the unconstrained dimensionless resistance r'_i, the scaling factor R_{im}, and the weight of i-th hyperbolic secant distribution w_i, as follows:

$$R_{\text{re}} := R_0 + \sum_{i=1}^{N} R_i := R_{\text{total}} := \exp(r_{\text{total}}), \tag{2}$$

$$r_i := \frac{R_i}{R_{\text{total}}} := \exp\left[-\exp(r'_i)\right], \tag{3}$$

$$R_{\text{im}} := \frac{\pi}{2}\sum_{i=1}^{N} R_i \tag{4}$$

$$\lambda_i := \frac{R_i}{\sum_{i=1}^{N} R_i}. \tag{5}$$

From this, the real and imaginary parts of the impedance (Re[Z], Im[Z]) , are given by (see Appendix A)

$$\mathrm{Re}[Z] = R_{\text{re}}\left[r_0 + \sum_{i=1}^{N}\frac{r_i}{2}\left[1 - \tanh(\ln\omega\tau_i)\right]\right], \tag{6}$$

$$\mathrm{Im}[Z] = \underbrace{R_{\text{im}}}_{\text{scaling factor}}\underbrace{\left[\sum_{i=1}^{N}\frac{\lambda_i}{\pi}\mathrm{sech}\left(\ln\omega\tau_i\right)\right]}_{\substack{\text{mixture of hyperbolic}\\ \text{secant distributions}}}. \tag{7}$$

The frequency range is also standardised according to the frequencies in the available dataset,

$$\mu_\omega, \sigma_\omega := \mathbb{E}[\ln\omega], \sqrt{\mathbb{V}\mathrm{ar}[\ln\omega]}, \tag{8}$$

$$\omega^{\text{std}}, \tau_i^{\text{std}} := \frac{\ln\omega - \mu_\omega}{\sigma_\omega}, -\frac{\ln\tau_i + \mu_\omega}{\sigma_\omega}, \tag{9}$$

where the mean μ_ω and standard deviation σ_ω of logarithmic angular frequency [1], $\ln\omega$ can be calculated from the given frequency range of the dataset, and from this we define a standardised frequency scale ω^{std} and standardised time constants τ_i^{std}. These may be related to the actual time constants and capacitances (noting that τ_i is the unstandardised form of the time constant) via

$$\ln\omega\tau_i := \ln\omega - \sigma_\omega\tau_i^{\text{std}} - \mu_\omega, \tag{10}$$

$$C_i = \frac{\tau_i}{r_i R_{\text{total}}}. \tag{11}$$

The parameters to be fitted are unconstrained standardised ones $\theta := \{r_{\text{total}}, r_i', \tau_i^{\text{std}}\}$. This formulation is similar to the distribution of relaxation times modelling. This canonical form provides three benefits: separation of scaling factor, unconstrained prior distribution selection for all parameters, and integral-friendly formulation. Separating the scaling factors can decompose parameter estimation problems into problems of estimating magnitudes (R_{re}) and ratios (r_i), permitting fair comparison over varied magnitudes of resistance. Logarithmically transformed parameters enable non-negativity constraints over resistance, allowing arbitrary prior distributions to be used for Bayesian inference (for instance, r_i is constrained between zero and one, but r_i' is unconstrained). The mixture of hyperbolic secant distributions offers several integral identities to analytically calculate the expectation and variance (see Appendix C). Moreover, this formulation interprets the imaginary part as a probability distribution function, allowing statistical analysis (see section 5).

3. BAYESIAN INFERENCE FORMULATION

We wish to select the likeliest model from the abovementioned three RC pair options. In Bayesian inference, we need to assume a prior distribution $p(\Theta) := \pi(\Theta)$ and a likelihood function $p(\mathbf{D}|\Theta, M) := \ell_{\text{true}}(\Theta)$. The prior distribution is a probability distribution reflecting one's prior assumptions about possible parameters. For instance, we adopt here a multivariate normal distribution $\pi(\Theta) := \mathcal{N}(\Theta; \mu_\pi, \mathbf{\Sigma}_\pi)$. The mean vector μ_π represents our guess of plausible parameter values and the covariance matrix $\mathbf{\Sigma}_\pi$ reflects our assumption on the uncertainty of each parameter, and correlations between parameters. The likelihood function $\ell_{\text{true}}(\Theta)$ is a probability distribution to evaluate how the selected parameter set Θ can reproduce the given dataset $\mathbf{D}$. Here we assume a univariate Gaussian with zero mean $\mathbf{0}$ and homoskedastic noise, meaning the noise variance σ_{noise} does not vary over frequency. The squared error evaluates how similar the observed data y_{obs} and ECM predicted data y_{ecm} are. Now, with the assumed prior $p(\Theta)$ and likelihood function $p(\mathbf{D}|\Theta, M)$, Bayes' rule defines the parameter posterior as $p(\Theta, M|\mathbf{D})$ and the model evidence $p(\mathbf{D}|M)$, all as follows:

$$p(\mathbf{D}|\Theta, M) := \ell_{\text{true}}(\Theta) := \prod_j^m \mathcal{N}(\text{err}_j(\theta); \mathbf{0}, \sigma^2_{\text{noise}}), \tag{12}$$

$$p(\mathbf{D}|M) := \mathcal{N}\left(\mathbb{E}_\pi[\ell_{\text{true}}(\Theta)], \mathbb{V}\text{ar}_\pi[\ell_{\text{true}}(\Theta)]\right), \tag{13}$$

$$p(\Theta|\mathbf{D}, M) = \frac{p(\mathbf{D}|\Theta, M)p(\Theta)}{p(\mathbf{D}|M)} = \frac{\ell_{\text{true}}(\Theta)\pi(\Theta)}{\mathbb{E}_\pi[\ell_{\text{true}}(\Theta)]}, \tag{14}$$

[1] Where necessary we assume arguments of logarithms are divided by appropriate units, e.g. 1 [rad/s], to ensure they are dimensionless.

where

$$\mathbf{D} := \{\mathbf{y}_{\text{obs}}, \omega^{\text{std}}\} \in \mathbb{R}^{m\times 2}, \tag{15}$$

$$\theta := \{r_{\text{total}}, r_i', \tau_i^{\text{std}}\} \in \mathbb{R}^{d-1}, \tag{16}$$

$$\Theta := \{\theta, \sigma^2_{\text{noise}}\} \in \mathbb{R}^d, \tag{17}$$

$$y_{\text{ecm},j}(\theta) := \{y_{\text{re},j}, y_{\text{im},j}\} = M(\theta, \omega_j^{\text{std}}), \tag{18}$$

$$\text{err}_j(\theta) := \left[y_{\text{obs},j} - y_{\text{ecm},j}(\theta)\right]^2, \tag{19}$$

where subscript 'obs' refers to measured data, subscript 'ecm' to modelled data, and M is the model (equations (6)-(7)). The posterior $p(\Theta|\mathbf{D}, M)$ is a conditional probability distribution that reflects our updated estimate of the parameter space based on the observed data $\mathbf{D}$. We use dimensionless and unconstrained r_i' and τ_i^{std} as inputs of the model for arbitrary prior selection and fair comparison of models. The number of parameters to be estimated is $d = 2 + 2N$, as the scaling factor r_{total} and experimental noise variance σ^2_{noise} are shared over all models.

4. BAYESIAN QUADRATURE MODELLING

We wish to estimate both the parameter posterior distribution $p(\Theta|\mathbf{D}, M)$ and the evidence $p(\mathbf{D}|M)$. We also wish to minimise the number of times that the likelihood $\ell_{\text{true}}(\Theta)$ must be queried, as this could be a computationally demanding operation in a more complex model. This problem requires a sample-efficient Bayesian inference solver. Bayesian quadrature (BQ) offers sample efficiency and solves for the posterior and the evidence in one go. This is a surrogate-model-based numerical integration approach, solving the integral as an inference problem by modelling the likelihood function $\ell_{\text{true}}(\Theta)$ with a Gaussian process (GP). Define $\ell(\Theta)$ as the surrogate likelihood function modelled by a GP. The key result is that BQ can recast the problem of Bayesian inference into one of function approximation. The more accurately $\ell(\Theta)$ can predict $\ell_{\text{true}}(\Theta)$, the more accurately the posterior and evidence can be estimated via replacing $\ell_{\text{true}}(\Theta)$ with $\ell(\Theta)$ in Eqs. (13) - (14). To achieve this, Adachi et al. (2022) proposed *BASQ*, a discrete approximation of the kernel integral using a kernel recombination method (Hayakawa et al., 2022), yielding the following evidence computations:

$$\text{LEM} := \ln\mathbb{E}_\pi[\ell(\Theta)] \approx \ln\sum_k^L W_k\mu_f(X_k) + \beta, \tag{20}$$

$$\text{LEV} := \ln\mathbb{V}\text{ar}_\pi[\ell(\Theta)] \approx \ln\sum_{k,l}^L W_k W_l\sigma_f(X_k, X_l) + 2\beta, \tag{21}$$

where LEM and LEV refer to log evidence mean and log evidence variance, μ_f and σ_f are the predictive mean and covariance of the likelihood surrogate model $\ell(\Theta)$, β is the scaling constant, W_k, W_l and X_k, X_l are the positive weights and point configurations discretised by the kernel recombination. Recall that LEM gives the degree of model fit and the LEV quantifies the uncertainty of the fit. However, the prior work on this (Adachi et al., 2022) assumed a narrower dynamic range of likelihood, whereas the battery model typically produces 10^{700} likelihood values. This is way beyond a typical numerical overflow limit. Thus, we improve here the prior work by adopting a four-layered warped GP method to accommodate the wide dynamic range of likelihood. (See Appendix B)

5. IDENTIFIABILITY

To evaluate the model evidence as a model selection criterion, we compare results against three classical metrics related to identifiability: number of data points m, signal-to-noise ratio (SNR), and Jensen-Shannon divergence (JS). Owing to the integral-friendly model formulation, most parts of these can be calculated analytically. The number of data points is controllable here because data are synthetically generated and equispaced over log angular frequency space. Both SNR and JS are calculated using the imaginary part of the impedance. As the canonical form can be regarded as a mixture of hyperbolic secant distributions, such statistical analysis can be applied. While SNR evaluates the identifiability along the impedance magnitude axis, JS does so along the frequency axis.

5.1 Signal-to-noise ratio

The SNR is the log fraction of the impedance variance over the noise variance, representing how much the signal is more distinct than the noise, defined as:

$$\mathrm{SNR} := \ln \frac{\mathbb{V}\mathrm{ar}_{P(\ln\omega)}[\mathrm{Im}[Z]]}{\sigma^2_{\mathrm{noise}}}. \tag{22}$$

Larger SNR means a more distinct and identifiable signal. The canonical form of the model provides an analytical form for the SNR (see derivation in Appendix C).

5.2 Jensen-Shannon divergence

The JS divergence is a distance metric quantifying how one probability distribution $P_i(x)$ is similar to a second reference probability distribution $P_j(x)$, defined as:

$$\begin{aligned}\mathrm{JS} := &\frac{1}{2}\int \ln\left(\frac{P_i(x)}{M_{ij}(x)}\right)\mathrm{d}P_i(x) \\ &+ \frac{1}{2}\int \ln\left(\frac{P_j(x)}{M_{ij}(x)}\right)\mathrm{d}P_j(x),\end{aligned} \tag{23}$$

where

$$M_{ij}(x) := \frac{1}{2}\Big(P_i(x) + P_j(x)\Big) \tag{24}$$

As the JS is defined for pairwise comparisons, the number of criteria required increases combinatorially per the number of RC pairs. For simplicity, we only consider the case of two RC pairs, which produces only one JS divergence. This represents how much the selected two peaks in the imaginary parts overlap. A smaller JS divergence means a more distinguishable and identifiable signal. While SNR is determined by the noise variance $\sigma^2_{\mathrm{noise}}$ and scaling factor r_{total}, JS is dominated by the time constant difference $\Delta\tau_{ij}$. Again, the canonical form helps solve the integration. Note that this is formulated as noise-free. The integration calculation procedure can be seen in Appendix C.3. The extended JS to include noise $\sigma^2_{\mathrm{noise}}$ is also guided, but the results shown in this paper are consistently used with noise-free formulation for simplicity.

6. NUMERICAL RESULTS

6.1 Selection criteria comparison

We now demonstrate our modified version of BASQ over several cases. We compare the model evidence metric (LEM and LEV [2] , eqs. (20) - (21)) with root-mean-square error (RMSE), Bayesian information criterion (BIC), and expected log predictive density (ELPD), based on the maximum a posteriori (MAP) parameter estimates, defined as:

$$\Theta_{\mathrm{MAP}} := \mathrm{argmax}\,\ell_{\mathrm{true}}(\Theta), \tag{25}$$

$$\mathrm{RMSE} := \sqrt{\frac{1}{m}\sum_j^m \mathrm{err}_j(\theta_{\mathrm{MAP}})}, \tag{26}$$

$$\mathrm{BIC} := d\ln m - 2\ln\ell_{\mathrm{true}}(\Theta_{\mathrm{MAP}}), \tag{27}$$

$$\mathrm{ELPD} := \sum_j^m \ln\int \ell_{\mathrm{true}}(\Theta)\mathrm{d}p(\Theta|\mathbf{D}, M). \tag{28}$$

The RMSE is a noise-free formulation that does not consider parameter uncertainty. BIC is an asymptotic approximation of evidence, so it cannot evaluate multimodal likelihoods. ELPD is a similar formulation to the log mean evidence, but the probability measure is changed from prior to posterior. The motivation behind ELPD is to estimate the alternative evidence from MCMC samples, as it cannot estimate evidence when solving Bayesian inference. However, it relies on Monte Carlo (MC) integration, which requires a significant amount of posterior samples, meaning that a plethora of model evaluations $\ell_{\mathrm{true}}(\Theta)$ will run. All these alternative criteria were calculated from the BQ estimated posteriors by post-processing. Moreover, none of these criteria quantify their own uncertainty except BQ.

We demonstrate the behaviours of the selection criteria on two different datasets—an easy case ($\Delta\tau_{ij} = 9.1$, $\ln\sigma^2_{\mathrm{noise}} = -9.97$) with results in Table 1, and a hard case ($\Delta\tau_{ij} = 0.36$, $\ln\sigma^2_{\mathrm{noise}} = -1.6$) detailed in Table 2. The easy case is clean data generated with 2 well-separated semi-circles, and the hard case is noisy data generated with an additional third semi-circle with more overlap. The "better" column shows which upward or downward direction is better for each criterion. As expected, separated peaks (large $\Delta\tau_{ij}$) and lower noise $\sigma^2_{\mathrm{noise}}$ boost identifiability. While all criteria selected the true model in the easy case, only the evidence can select the true model in the hard case.

Table 1. Easy case

true model	1 RC pair	2 RC pairs ✔	3 RC Pairs	4 RC pairs
LEM	-2809233	**703.6569**	289.2976	225.1602
LEV	-33.52068	-27.31169	-31.91129	-31.38766
RMSE	1.147527	**0.006677**	0.031770	0.062151
BIC	5766999	**-1405.553**	-572.7390	-432.4597
ELPD	-2883641	**713.7332**	293.2492	213.7417

Table 2. Hard case

true model	1 RC pair	2 RC pairs	3 RC Pairs ✔	4 RC pairs
LEM	-150.3634	-151.8002	**-147.4257**	-151.9208
LEV	-15.74094	-19.07997	-19.45956	-26.57386
RMSE	**0.492191**	0.492643	0.492269	0.492260
BIC	**302.8601**	313.8921	321.0089	310.8289
ELPD	**-145.7505**	-148.8758	-148.4754	-146.1485

[2] LEV values in the tables are standardised via subtracting 2β from eq. (21) for a fair comparison between models.

Table 3. Linear correlation matrix

factors	m	JS	SNR	LEM	LEV
m	-	-0.0268	0.0058	0.4096	-0.1806
JS	-0.0268	-	-0.0993	-0.0297	0.2243
SNR	0.0058	-0.0993	-	0.7299	-0.4882
LEM	0.4096	-0.0297	0.7299	-	-0.2867
LEV	-0.1806	0.2243	-0.4882	-0.2867	-

The other metrics were unsuccessful in the hard case because of a multimodal posterior in the one RC pair model. As three RC pairs were used to generate the dataset, the posterior distribution of one RC pair parameter inevitably becomes multimodal, such as the peak intensity (λ_i). While the evidence correctly incorporates the multimodal distribution shape, RMSE and BIC consider only the largest peak. The BIC estimates the whole distribution from the local curvature at the maximum, which becomes erroneously overconfident in the multimodal case (Murphy, 2012). ELPD's failure could be due to its rough integral approximation. As the convergence rate of MC integration is $\mathcal{O}(1/\sqrt{n})$, the posterior samples ($n = 1{,}000$) is too few. This means more model evaluations $\ell_{\text{true}}(\Theta)$ are required, which would not scale to slower simulation models.

In contrast, the evidence can be estimated simultaneously during training. Moreover, the variance of the evidence successfully points out the lower confidence in the one RC pair model in the hard case, suggesting multimodality. This uncertainty over the selection criterion could avoid overconfidence toward a simpler model. Moreover, the evidence variance in the hard case is generally higher than in the easy case. This also tells us that the hard case dataset is almost unidentifiable, suggesting we should not trust these comparisons. For instance, the evidence mean and ELPD for one RC pair in the easy case are much lower than in the hard case. However, the integral variance is the opposite. Thus, only this metric quantifies its own uncertainty, suggesting the dataset or model is less informative. A similar notion can be found in Jeffreys' scale for the Bayes factor (Jeffreys, 1998), which claims the evidence is not strong when the difference between the log evidence of two models is lower than 10. This explains that the hard case is unreliable, as the difference in the log evidence shows insufficient plausibility. Contrary to Jeffreys' scale, log evidence variance is self-contained and does not require the comparison of models. Instead, it can independently spot the unreliability of the estimation.

In such an uncertain case, a typical practice is Bayesian model averaging. Rather than selecting one definite model, we sample from a *mixture of models* with probability proportional to their mean evidence. Averaging can boost predictive accuracy and reduce the uncertainty over predictions, where only evidence offers this method. As such, while the easy cases do not require advanced methods, the evidence with self-check on reliability can assist in deciphering minor differences in hardly identifiable problems.

6.2 Sensitivity analysis

A sensitivity analysis of the evidence metric was performed. We generated 1,024 datasets using two-RC-pair models while varying the following five parameters; the number of data points m, the scaling factor r_{total}, the first resistance r'_1, the first time constant τ_1^{std}, and noise variance σ^2_{noise}. We calculated the SNR, JS, and the number of data points m for each dataset. In the first step of the analysis, we compared the linear correlations between the evidence estimates. Table 3 shows Pearson's correlation coefficients. This result aligns with our intuition—for instance, larger data size m and SNR can boost the evidence LEM and confidence (inverse of LEV). However, while the large correlation of evidence with SNR is instinctive, the small correlation with JS is counterintuitive.

Table 4. Functional ANOVA results

factors	LEM	LEV	residual
(m)	0.0012	0.0083	0.0012
(JS)	**0.3258**	**0.3335**	**0.3397**
(SNR)	0.2778	0.2002	0.0825
(m, JS)	0.0028	0.0088	0.0017
(m, SNR)	0.0599	0.1260	0.2427
(JS, SNR)	0.2702	0.1969	0.0879
(m, JS, SNR)	0.0623	0.1264	0.2443

Thus, we further investigated the variance analysis via functional ANOVA (Hutter et al., 2014), which models a partition of a functional response according to the main effects and interactions of input parameters. This method can attribute each parameter sensitivity in a nonlinear manner. Table 4 illustrates that the most significant influence over the mean and variance of the evidence is the JS, contrary to the linear correlation results. This can be interpreted as meaning that a smaller JS divergence (more overlapped peaks) destabilises the evidence estimation, resulting in a more considerable variance. This viewpoint is supported by the relatively large negative correlation coefficient between JS and LEV.

Further insights can be obtained via residual analysis. The residual is defined as follows:

$$Z_{\text{pred}} := \text{slope} \times \text{BIC} + \text{intercept}, \tag{29}$$

$$\text{residual} := \left(Z_{\text{pred}} - \log \mathbb{E}_\pi[\mu_e(\Theta)]\right)^2. \tag{30}$$

As the BIC is an approximation of the LEM, the BIC and LEV have a linear relationship. While a linear regression model with BIC can predict log evidence mean reasonably, it fails to predict in hard cases, as shown in the section 6.1. Residual refers to the squared error between the BIC and log evidence mean. Table 4 shows that the residual is mainly caused by the JS divergence and less influenced by SNR or the number of data points m. This also suggests that the BIC cannot distinguish between the models with overlapped peaks, namely, a multimodal posterior.

6.3 Computation efficiency

Lastly, we compared the computation efficiency of our modified version of BASQ with the existing MCMC solvers elliptical slice sampling (ESS) (Murray et al., 2010) and dynamic nested sampling (Speagle, 2020). Note that amongst MCMC samplers, only nested sampling can estimate the evidence. For ESS, we approximated the evidence using ELPD via posterior samples. Therefore, the estimation with ESS should converge to a larger value than the

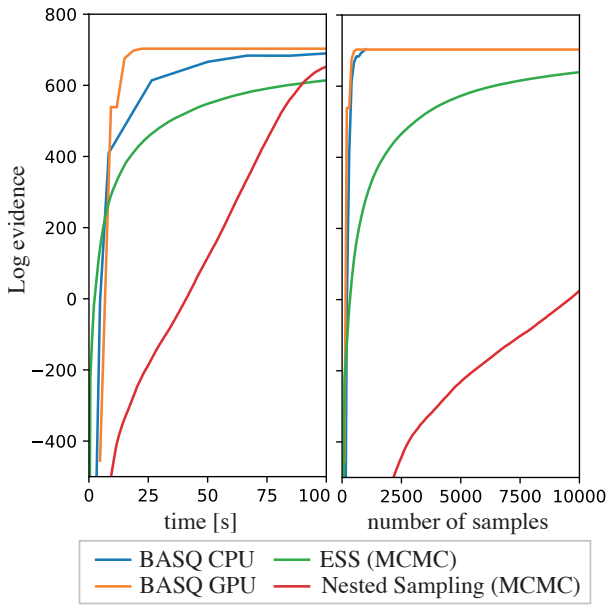


Fig. 2. The learning curve of log evidence over the computation time and the number of samples.

actual evidence. The BASQ computation was performed using both CPU and GPU. [3]

Fig. 2 compares the learning curve of the above four samplers versus computation time, using the easy case dataset shown in Table 1. While BASQ in a GPU converges at 18 seconds, BASQ in a CPU converges at 131 seconds. Both ESS and nested sampling do not converge in this time. Fig. 2 contrasts the sample efficiency of the samplers. As BASQ is a parallel sampler, we generate 100 samples per iteration. The sampling efficiency of BASQ does not change over computation modes and is the best of the selected solvers. This is expected—while the convergence rate of BASQ is $\mathcal{O}(\exp(-cn^{1/d}))$ in the Gaussian case (Adachi et al., 2022), that of MCMC is $\mathcal{O}(1/\sqrt{n})$. Furthermore, even this result does not fully represent BASQ's potential. While ECMs return model predictions in a millisecond order, more complex models (e.g. DFN model) take seconds to query. Therefore, BASQ for such complex models will be even more beneficial. Recent work shows even faster convergence than BASQ (Adachi et al., 2023).

REFERENCES

Adachi, M. et al. (2022). Fast Bayesian inference with batch Bayesian quadrature via kernel recombination. *NeurIPS*, 35.

Adachi, M. et al. (2023). SOBER: Scalable batch Bayesian optimization and quadrature using recombination constraints. *arXiv preprint arXiv:2301.11832*.

Aitio, A. et al. (2020). Bayesian parameter estimation applied to the Li-ion battery single particle model with electrolyte dynamics. *IFAC*, 53(2), 12497.

Bizeray, A.M. et al. (2018). Identifiability and parameter estimation of the single particle lithium-ion battery model. *IEEE Trans. Control. Syst. Technol.*, 27, 1862.

[3] Both MCMC samplers and BASQ in CPU were computed with a MacBook Pro 2019, 2.4 GHz 8-Core Intel Core i9, 64 GB 2667 MHz DDR4. BASQ in GPU was performed on Google Colaboratory.

Calderwood, J. (2003). A physical hypothesis for Cole-Davidson behavior. *IEEE Trans. Dielectr. Electr. Insul.*, 10(6), 1006.

Chai, H.R. et al. (2019). Improving quadrature for constrained integrands. In *AISTATS*, 2751. PMLR.

Doyle, M. et al. (1993). Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell. *J. Electrochem. Soc.*, 140(6), 1526.

Escalante, J.M. et al. (2021). On uncertainty quantification in the parametrization of Newman-type models of lithium-ion batteries. *J. Electrochem. Soc.*, 168, 110519.

Gunter, T. et al. (2014). Sampling for inference in probabilistic models with fast Bayesian quadrature. *NeurIPS*.

Hastings, W.K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1), 97.

Hayakawa, S. et al. (2022). Positively weighted kernel quadrature via subsampling. *NeurIPS*, 35.

He, H. et al. (2011). Evaluation of lithium-ion battery equivalent circuit models for state of charge estimation by an experimental approach. *Energies*, 4(4), 582.

Huang, J. et al. (2021). Towards robust autonomous impedance spectroscopy analysis: A calibrated hierarchical Bayesian approach for electrochemical impedance spectroscopy (EIS) inversion. *Electrochim. Acta*, 367, 137493.

Hutter, F. et al. (2014). An efficient approach for assessing hyperparameter importance. In *ICML*, 754. PMLR.

Jeffreys, H. (1998). *The theory of probability*. OUP Oxford.

Kemper, P. et al. (2013). Extended single particle model of Li-ion batteries towards high current applications. In *IEEE VPPC*, 1.

Kitagawa, G. (1993). A Monte Carlo filtering and smoothing method for non-Gaussian nonlinear state space models. In *Proceedings of the 2nd U.S.-Japan Joint Seminar on Statistical Time Series Analysis*, 110.

Kuhn, Y. et al. (2022). Bayesian parameterization of continuum battery models from featurized electrochemical measurements considering noise. *Batteries & Supercaps*.

Liu, J. et al. (2020). The Gaussian process distribution of relaxation times: A machine learning tool for the analysis and prediction of electrochemical impedance spectroscopy data. *Electrochim. Acta*, 331, 135316.

Metropolis, N. et al. (1953). Equation of state calculations by fast computing machines. *Chem. Phys.*, 21(6), 1087.

Milocco, R.H. et al. (2014). Generic dynamic model of rechargeable batteries. *J. Power Sources*, 246, 609.

Miyazaki, Y. et al. (2020). Bayesian statistics-based analysis of ac impedance spectra. *AIP Adv.*, 10, 045231.

Murphy, K.P. (2012). *Machine learning: a probabilistic perspective*. MIT press.

Murray, I. et al. (2010). Elliptical slice sampling. *AISTATS*, 541.

Osborne, M. et al. (2012). Active learning of model evidence using Bayesian quadrature. *NeurIPS*, 25.

Rasmussen, C. et al. (2000). Occam's razor. *NeurIPS*, 13.

Santhanagopalan, S. et al. (2006). Review of models for predicting the cycling performance of lithium ion batteries. *J. Power Sources*, 156(2), 620.

Speagle, J.S. (2020). dynesty: a dynamic nested sampling package for estimating Bayesian posteriors and evidences. *MNRAS*, 493(3), 3132.

Clarification

The Appendix is provided in Section C. The log-transformed Bayesian quadrature approach enables us to infer the closed-form posterior distribution, which is advantageous for further posterior analyses, such as marginal, conditional, and joint posterior analysis (e.g., Aitio et al. [see 13, Fig. 3]), graphical correlation analysis with Gaussian undirected graph (e.g., Edwards [68]), and functional ANOVA analysis (e.g., Kaufman et al. [125]). These analyses offer deeper insights into parameter fitting beyond typical sensitivity analyses using the Hessian.

Furthermore, the compressed posterior is also effective for estimating the predictive uncertainty of the simulator. By using quadrature, we obtain the compressed distribution $\tilde{\nu}(\theta) \approx \mathcal{P}(\theta \mid \mathbf{D})$, which allows us to estimate the predictive distribution with a smaller number of weighted samples.

While our approach is well-suited for Bayesian model selection in cases with extreme dynamic ranges, such as battery simulators, it is not a silver bullet. In cases with extremely large positive likelihood values, such as exp(703) in Table 1, the posterior effectively becomes similar to a point estimate due to the Bernstein-von Mises theorem (Freedman [74]). In such scenarios, the benefits of the aforementioned detailed analyses are effectively reduced to the typical Hessian-based (second-order Taylor approximation) approach. Thus, any approximated metric can correctly evaluate the easy case.

Future direction

Combining our approach with the adaptive model selection method proposed by Chai et al. [41] presents a promising direction. Their method uses mutual information between model evidences, $\mathbb{I}(\mathbf{D} \mid M_i)$, as the selection criterion and iteratively discards models unlikely to be the true model. This approach reallocates computational resources from unpromising models, thereby accelerating the entire selection process.

Another promising direction is to integrate likelihood-free inference methods, such as those proposed by Gutmann et al. [94]. Instead of adopting a log-transform

to handle the extreme range of likelihood values, their approach models a GP on the error space. Under Gaussian likelihood, the mean squared error, $1/n \sum_{i=1}^{n}(y_i - f(\theta_i))^2$, is linearly related to the log-likelihood. Therefore, an affine transformation can be applied to model the log-likelihood. Since the MSE does not take extreme values like the log-likelihood, it is easier to model with a GP. This idea is, in fact, already utilized in our work (Adachi et al. [8]).

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

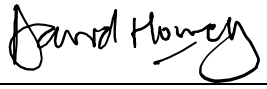
Title of Paper	Bayesian Model Selection of Lithium-ion Battery Models via Bayesian Quadrature
Publication Status	☑Published □ Accepted for Publication □Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	Masaki Adachi, Yannick Kuhn, Birger Horstmann, Arnulf Latz, Michael A. Osborne, and David A. Howey. Bayesian Model Selection of Lithium-ion Battery Models via Bayesian Quadrature. IFAC-PapersOnLine 56(2), 10521-10526, 2023.

Student Confirmation

Student Name:	Masaki Adachi		
Contribution to the Paper	first author I independently thought of, derived, and implemented this methodology. I ran the experiments for paper, conducted all additional analyses, formulated and proved the theoretical results, and wrote the manuscript. My co-authors played advisory and editorial roles.		
Signature 		Date	06 January 2025

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: David A. Howey, Professor of Engineering Science		
Supervisor comments Masaki led this work and the description given above is accurate.		
Signature 	Date	14th March 2025

This completed form should be included in the thesis, at the end of the relevant chapter.

Part II

Human-AI collaboration for Scientific Experts

Artificial intelligence is not a substitute for human intelligence; it is a tool to amplify human creativity and ingenuity. [79]

— Fei Fei Li, Professor of Computer Science

People are not accustomed to thinking hard, and are often content to trust a plausible judgment that comes to mind. [115]

— Daniel Kahneman, Nobel Laureate in Economics

6

Looping in the human: collaborative and explainable Bayesian optimization

This chapter is based on the following publication:

> **Masaki Adachi**, Brady Planden, David A. Howey, Michael A. Osborne, Sebastian Orbell, Natalia Ares, Krikamol Muandet, and Siu Lun Chau. Looping in the human: collaborative and explainable Bayesian optimization. In *International Conference on Artificial Intelligence and Statistics (AISTATS)* 238, 505-513, 2024.

In Chapter 6, we introduced a human-AI collaborative framework for Bayesian optimization, particularly tailored for scientific experts. We first presented the preferential learning approach for collaboration tasks and incorporated expert advice as a supervisory element in the optimization process. Human experts' advice was framed as a prior belief about the global maximum, $\mathbb{P}(f(x) = f(x^\star))$. Simultaneously, we considered the 'objective' belief derived from experimental results $\mathbf{D}_{\text{obj}_t}$, expressed as $\mathbb{P}(f(x) = f(x^\star) \mid \mathbf{D}_{\text{obj}_t})$. In contrast, the expert's prior represents a 'subjective' belief based on their preferential feedback $\mathbf{D}_{\text{pref}_t}$, defined as $\mathbb{P}(f(x) = f(x^\star) \mid \mathbf{D}_{\text{pref}_t})$. A significant challenge arises when $\mathbf{D}_{\text{pref}_t}$ is entirely incorrect, as relying solely on the subjective belief prior may lack guarantees of global convergence.

Inspired by Hvarfner et al. [112], we assumed that human advice is most effective at the beginning of the process, acting as a warm starter for the optimizer. To address this, we manually discounted the influence of human belief over iterations using a hyperparameter γ. This ensures that the effect of human advice becomes negligible in the asymptotic sense. Since the convergence rate and regret analysis of Bayesian optimization are typically offered as asymptotic guarantees, this approach ensures a no-harm guarantee: the convergence rate remains identical to that of standard Bayesian optimization, regardless of the efficacy of human advice.

Experimentally, this approach demonstrated significant acceleration of the optimization process. Additionally, the inclusion of explainability features enhanced the accuracy of human feedback, further accelerating the process.

Looping in the Human: Collaborative and Explainable Bayesian Optimization

Masaki Adachi[1,2,3] **Brady Planden**[2] **David A. Howey**[2,4] **Michael A. Osborne**[1,2]
Sebastian Orbell[2] **Natalia Ares**[2] **Krikamol Muandet**[5] **Siu Lun Chau**[5]
[1]Machine Learning Research Group, University of Oxford, OX2 6ED, United Kingdom
[2]Department of Engineering Science, University of Oxford, OX1 3PJ, United Kingdom
[3]Toyota Motor Corporation, Shizuoka 410-1107, Japan
[4]The Faraday Institution, Harwell Campus, Didcot OX11 0RA, United Kingdom
[5]CISPA Helmholtz Center for Information Security, 66123 Saarbrücken, Germany

Abstract

Like many optimizers, Bayesian optimization often falls short of gaining user trust due to opacity. While attempts have been made to develop human-centric optimizers, they typically assume user knowledge is well-specified and error-free, employing users mainly as supervisors of the optimization process. We relax these assumptions and propose a more balanced human-AI partnership with our Collaborative and Explainable Bayesian Optimization (CoExBO) framework. Instead of explicitly requiring a user to provide a knowledge model, CoExBO employs preference learning to seamlessly integrate human insights into the optimization, resulting in algorithmic suggestions that resonate with user preference. CoExBO explains its candidate selection every iteration to foster trust, empowering users with a clearer grasp of the optimization. Furthermore, CoExBO offers a no-harm guarantee, allowing users to make mistakes; even with extreme adversarial interventions, the algorithm converges asymptotically to a vanilla Bayesian optimization. We validate CoExBO's efficacy through human-AI teaming experiments in lithium-ion battery design, highlighting substantial improvements over conventional methods. Code is available `https://github.com/ma921/CoExBO`.

Proceedings of the 27[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238.

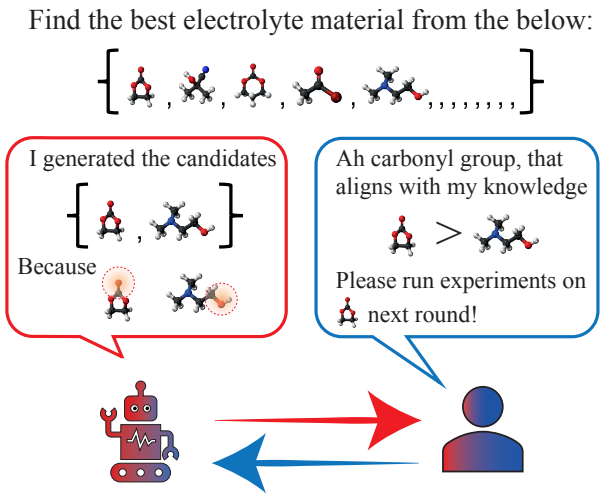


Figure 1: In Collaborative and Explainable Bayesian Optimization (CoExBO), a human expert collaborates with BO to refine electrolyte materials. While experts excel in discerning material differences rather than identifying the best one, pairwise comparisons and explanations boost their feedback accuracy and trust. This guides the BO to produce better candidates, ensuring quicker convergence.

1 Introduction

Bayesian optimization (BO) is a popular blackbox optimizer for expensive-to-evaluate tasks. While it is widely applied in diverse domains (Feurer et al., 2015; Wu et al., 2020; Adachi, 2021), it has yet to fully gain human users' trust. Surveys from NeurIPS2019/ICLR2020 (Bouthillier & Varoquaux, 2020) found that most AI researchers prefer manually tuning hyperparameters. This is surprising given that Bergstra & Bengio (2012) has shown manual search performs worse than simple random search and lacks

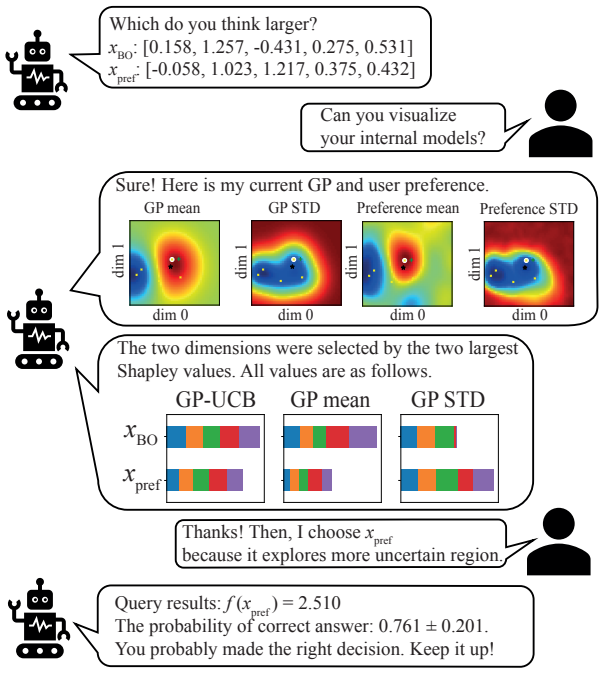


Figure 2: Explanation flow: **Spatial relation:** BO visualizes the surrogate model's predictive distribution and estimated human preference models for the two primary dimensions determined by Shapley values. **Feature importance:** Users' values are provided for both candidates' predictive mean, standard deviation, and acquisition function. **Selection accuracy feedback:** After observing the function value, a post-hoc evaluation of the correct selection probability is given.

global convergence guarantees (Gupta et al., 2023).

To make BO trustworthy, recent research has moved towards human-AI collaborative paradigms (Kanarik et al., 2023). These methods often make contrasting assumptions about human exploration abilities. Suppose humans are superior to BO (Colella et al., 2020; AV et al., 2022), their intervention can enhance convergence—but this potent assumption also implies manual search would be superior to BO, undermining the core justification for using BO. Conversely, if humans are viewed as imperfect agents, existing works such as Gupta et al. (2023); Khoshvishkaie et al. (2023) treat humans as a central optimizer, and BO supports human manual search via exploratory adjustment, that can assure the global convergence even with erroneous human selection. When one side is better, the inferior side's selection is wasteful, leading to a worse convergence rate than vanilla BO (Khoshvishkaie et al., 2023).

To build a balanced human-BO partnership, we believe an ideal method should satisfy the following criteria: (a) **Explainability:** The method should enhance user understanding of the optimization process, promoting transparency. While Li & Adams (2020) introduced explainability by limiting the search space, it may overly restrict it, lacking a global convergence guarantee. (b) **User-centric knowledge integration:** An ideal approach should seamlessly incorporate human insights from user interactions without requiring users to define an exact knowledge model. For instance, materials often exist in a high-dimensional feature space, whereas BO typically uses a reduced low-dimensional feature set due to limited chemistry data (Jordan, 2019). Chemists have discernment to assess materials using information inaccesible to the model but struggle to articulate it quantitatively (Cisse et al., 2023). Hence, users face challenges with existing methods; e.g., Hvarfner et al. (2022) mandates a prior function capturing the users' optimal location belief, while AV et al. (2022) requires users to select the next query point quantitatively. In short, knowledge elicitation in high-dimensional domains is notoriously intricate (Rousseau, 2001; Garthwaite et al., 2005; Mikkola et al., 2023a). (c) **Robustness:** The method should be robust against human errors, offering a no-harm guarantee to ensure that even adversarial interventions do not adversely impact the vanilla BO convergence rate. To the best of our knowledge, only Hvarfner et al. (2022) can theoretically assure the no-harm guarantee. Crucially, none encompass all three comprehensively.

This paper introduces the *Collaborative and Explainable Bayesian Optimization* (CoExBO) framework to tackle the above challenges. For criterion (a), CoExBO employs Shapley values (Shapley, 1953), a cornerstone of explainable AI, to ensure users can effectively understand and interpret the candidate acquisition mechanism. Addressing criterion (b), CoExBO deviates from conventional methods that require users to input an exact knowledge model. Instead, it presents users with candidate pairs, empowering them to select the perceived optimal one. This approach allows CoExBO to implicitly assimilate human insights via preference learning (Bradley & Terry, 1952). This is grounded that humans excel at relative comparisons rather than quantifying an absolute preference for a singular choice (Kahneman & Tversky, 1979). For criterion (c), inspired by Hvarfner et al. (2022), our candidate generation strategy prioritizes expert knowledge in the early optimization stages. As more experimental data accumulates and refines the surrogate model, the influence of human input gradually diminishes. Theorem 2 proves this offers a no-harm guarantee.

Our contributions are summarized as:

1. We introduce CoExBO, a novel framework promoting a balanced human-BO partnership. CoExBO is characterized by its transparency, capacity to assimilate human insights seamlessly, and resilience against human errors.

2. We establish the efficacy of CoExBO via real-world optimization tasks on lithium-ion battery design problems, and the expert-BO team can gain significant speedup over eight popular baselines.

2 Bayesian optimization and existing human-in-the-loop extensions

Bayesian optimization. We aim to optimize the function f when f can only be queried pointwise.

$$x^*_{\text{true}} = \underset{x\in\mathcal{X}}{\text{argmax}}\, f(x), \tag{1}$$

where $\mathcal{X} \subseteq \mathbb{R}^d$ is the d dimensional continuous input domain, and x^*_{true} is the global optimum. We assume that $f(x)$ is costly to query and can only observe a noisy estimate $y = f(x) + \epsilon$ with i.i.d. zero-mean Gaussian noise ϵ. The goal is to find the optimal $f(x)$ under a given number of queries.

BO (Mockus, 1998; Garnett, 2023) is a model-based black box optimizer that employs a Gaussian process (GP) (Williams & Rasmussen, 2006) as a surrogate model. At optimization step t, the GP approximates the true function f using the current observations $\mathbf{D}_{\text{obj}_t} = (\mathbf{X}_{\text{obj}_t}, \mathbf{y}_{\text{obj}_t})$ as $f_t \sim \mathcal{GP}(\mu_t, \kappa_t)$, with

$$\mu_t(x) = k(x, \mathbf{X}_{\text{obj}_t})\mathbf{K}'^{-1}_{\mathbf{XX}t}\mathbf{y}_{\text{obj}_t},$$
$$\kappa_t(x, x') = k(x, x') - k(x, \mathbf{X}_{\text{obj}_t})\mathbf{K}'^{-1}_{\mathbf{XX}t}k(\mathbf{X}_{\text{obj}_t}, x),$$

where μ_t and κ_t are the GP's posterior predictive mean and covariance functions at round t. k is the kernel, $\lambda > 0$ is the Gaussian likelihood variance, $\mathbf{K}_{\mathbf{XX}t} := k(\mathbf{X}_{\text{obj}_t}, \mathbf{X}_{\text{obj}_t})$, $\mathbf{K}'^{-1}_{\mathbf{XX}t} := (\mathbf{K}_{\mathbf{XX}t} + \lambda\mathbf{I})^{-1}$, and $\mathbf{I} \in \mathbb{R}^{d\times d}$ is the identity matrix. Using GP predictive uncertainty, BO solves the blackbox optimization problem as active learning and selects the next query point by maximising an acquisition function (AF). One popular class of AF is the upper confidence bound (UCB) (Srinivas et al., 2010): $\alpha_{f_t}(x) := \mu_t(x) + \beta_t^{1/2}\sigma_t(x)$, where $\sigma_t(x) := \sqrt{\kappa_t(x, x)}$ is the standard deviation of the GP predictive posterior, β_t is the user-specified parameter indicating the trade-off between exploitation (using current knowledge of optimum from μ_t) and exploration (considering the uncertainty from σ_t).

Human-in-the-loop extensions. There are four prevailing approaches to integrate human knowledge into BO: (1) By treating human knowledge as a prior over the input space (Souza et al., 2021; Ramachandran et al., 2020; Hvarfner et al., 2022; Cisse et al., 2023). (2) By adopting a hyperprior over the function space (Hutter et al., 2011; Snoek et al., 2014; Wang et al., 2021). (3) By considering it as a multi-fidelity information source (Song et al., 2019; Huang et al., 2022). (4) Implementing human knowledge as hard constraints (Gelbart et al., 2014; Hernández-Lobato et al., 2015; Adachi et al., 2024). Among these categories, only the work of Hvarfner et al. (2022), belonging to the first category, provides a no-harm guarantee against potential human errors. Notably, all these methods operate under the assumption that human knowledge can be well specified to the algorithm.

πBO. CoExBO is inspired by the πBO algorithm (Hvarfner et al., 2022), which characterizes human knowledge as a prior distribution representing their belief in the global optimum location, i.e., $\pi(x) := \mathbb{P}\big(f(x) = \max_{x'\in\mathcal{X}} f(x')\big)$. This prior can then be incorporated into an AF α_t to act as a soft constraint for a warmer start on the optimization. Specifically, at round t, we search for $x_{\text{next}} = \text{argmax}_{x\in\mathcal{X}}\ \alpha_t(x)\pi(x)^{\gamma/t}$ where $\gamma > 1$ controls the decay rate of this constraint. This decay of human contribution is justified as follows: at the start of the BO, expert knowledge can help substantially, whereas, at later stages, the BO will likely have enough data to reach the optimum confidently. Furthermore, this decaying property is the key reason behind the no-harm guarantee in Corollary 1 in Hvarfner et al. (2022).

While πBO's formulation is straightforward, requiring the user to specify a prior over high-dimensional input space could be very challenging in practice (Garthwaite et al., 2005). The following section demonstrates how CoExBO can relax this assumption by interacting with users through preference elicitation.

3 Collaborative and Explainable BO

In this section, we present our Collaborative and Explainable Bayesian Optimization (CoExBO) algorithm. While its objective aligns with the conventional BO objective (Eq. 1), CoExBO differentiates itself by explaining the acquisition process and incorporating human knowledge through preference learning. Specifically, the query procedure consists of the following steps: At round $t > 1$ with surrogate GP f_t and preference model $\hat{\pi}_t$, we have

$$\Gamma(f_t, \hat{\pi}_t) \to (x_1, x_2), \quad \text{(Acquire candidates)}$$
$$\text{E}(f_t, x_1, x_2) \to (\boldsymbol{\phi}_1, \boldsymbol{\phi}_2) \quad \text{(Explain acquisition)}$$
$$\text{H}\left(\{(x_i, \boldsymbol{\phi}_i)\}_{i=1}^2\right) \to \tilde{x} \in (x_1, x_2), \quad \text{(Elicit preference)}$$
$$\boldsymbol{\Pi}\left(\hat{\pi}_t, \tilde{x}, x_1, x_2\right) \to \hat{\pi}_{t+1}, \quad \text{(Update } \hat{\pi}_t\text{)}$$

and finally we run the experiment with $\tilde{x}$ to obtain $y_{\text{next}} = f(\tilde{x})$. We denote Γ as the candidate generation function (see §3.2) that takes in the surrogate and current preference models and generates a pair of candidates. E is an explanation function that explains the acquisition process (see §3.3) and returns

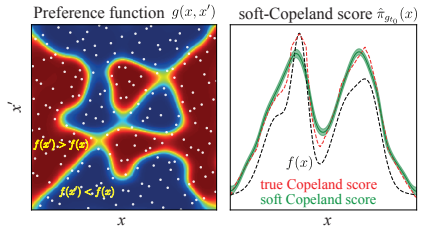


Figure 3: Preference learning concepts: we aim to model the ordinal relationship $f(x) < f(x')$ and its inverse with GP, utilizing the dataset $D^{t_0}_{\text{pref}}$, represented by white dots. Soft-Copeland score is used for the proxy of true function estimate.

explanation $\boldsymbol{\phi}_i$ for the i^{th} candidate. H denotes the human users' choice when pairs of candidates and the subsequent explanations are given. Preference function $\hat{\pi}_t$ is then updated to $\hat{\pi}_{t+1}$ by taking into account the human preference through an update function $\mathbf{\Pi}$ (see §3.1), and at last we end the iteration by running an experiment on the chosen candidate $\tilde{\boldsymbol{x}}$.

3.1 Model human knowledge through preference learning

While πBO requires the user to provide the preference function π explicitly—which might be challenging to elicit in practice—we relax this assumption and estimate π by $\hat{\pi} : \mathcal{X} \to \mathbb{R}_+$ using preference learning. At its core, preference learning aims to model the order relationships among a candidate set, $\mathcal{X}$. For this paper, our emphasis is on binary preference learning, as detailed in (Bradley & Terry, 1952; Chau et al., 2022b). However, this concept can be straightforwardly extended to more complex preference models such as choice functions (Benavoli et al., 2023).

To initiate the optimization process, we randomly select candidate pairs from $\mathcal{X}$, then solicit the users' opinion on which one is more likely the optimal location. Formally, at $t = t_0$, we sample J_{t_0} binary comparisons, denoted as $D^{t_0}_{\text{pref}} := \{x^{(j)}_1, x^{(j)}_2, y^{(j)}_{\text{pref}}\}^{J_{t_0}}_{j=1}$, where y_{pref} is 1 if $x^{(j)}_1$ is preferred over $x^{(j)}_2$, and 0 otherwise. Using $D^{t_0}_{\text{pref}}$, we can construct a binary preference function $g : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ based on the following likelihood model for any candidate pair x_1, x_2:

$$\mathbb{P}(y_{\text{pref}} \mid x_1, x_2) = S(y_{\text{pref}}; g(x_1, x_2)),$$

where $S(y_{\text{pref}}; z) := z^{y_{\text{pref}}}(1-z)^{1-y_{\text{pref}}}$ is the Bernoulli likelihood and $g(x_1, x_2)$ denotes the users degree of preference of x_1 over x_2. There are various ways to learn such a g (Bradley & Terry, 1952; Chau et al., 2022a). We choose to model g with a GP to encapsulate the inherent estimation uncertainty, pivotal for designing an acquisition in §3.2 that considers both surrogate and preference model's uncertainties. As the exact method we choose to learn g is not the primary focus of this work, we defer this discussion in Supplementary B.

Figure 3 visualizes the GP preference function g_{t_0} on the left, and $\hat{\pi}_{t_0}$ on the right, estimated by:

$$\hat{\pi}_{g_{t_0}}(x) := \int_{\mathcal{X}} g_{t_0}(x, x_2) dx_2. \tag{2}$$

This approach mirrors the soft-Copeland score[1] in González et al. (2017) and models the (unnormalized) likelihood of x being the Condorcet winner[2]. Hence, we can integrate out g_{t_0} and obtain the following representation of our estimated user preference

$$\hat{\pi}_{g_{t_0}}(x) \sim \mathcal{N}(\mathbb{E}_{g_{t_0}}[\hat{\pi}_{g_{t_0}}(x)], \mathbb{V}_{g_{t_0}}[\hat{\pi}_{g_{t_0}}(x)]).$$

Importantly, while the soft-Copeland score does not replicate the original function $f(x)$, the location of its maximum still corresponds to the maximum of $f(x)$. Furthermore, the maximum of the soft-Copeland score converges to the true maximum as the dataset size increases, namely $\lim_{t\to\infty} \operatorname{argmax}_{x\in\mathcal{X}} \pi_{g_t}(x) = \operatorname{argmax}_{x\in\mathcal{X}} f(x)$ if $|\mathcal{X}| < \infty$. As the optimization progresses and the acquisition of more binary comparison data, we can iteratively update the posterior of g_{t-1} via Bayes' theorem and recalibrate the preference model $\hat{\pi}_t$ at each iteration t.

3.2 Candidate generation with no-harm guarantee

New acquisition function. Following Hvarfner et al. (2022), we can use $\mathbb{E}_{g_t}[\hat{\pi}_{g_t}]$ as a discounting factor for α_{f_t} and redefine it as $\alpha_{f_t}(\cdot)\mathbb{E}_{g_t}[\hat{\pi}_{g_t}(\cdot)]^{\frac{\gamma}{t}}$. However, this approach does not consider the predictive uncertainty of $\hat{\pi}_{g_t}$ represented by the GP model g, i.e., $\mathbb{V}_{g_t}[\hat{\pi}_{g_t}(x)] = \mathbb{E}_{g_t}[\hat{\pi}_{g_t}(x)(1-\hat{\pi}_{g_t}(x))]$. This could result in overly optimistic acquisitions. πBO's performance depends on the peak centre of π, which must be close to the true global optimum for speedup. However, the peak centre of $\mathbb{E}_{g_t}[\hat{\pi}_{g_t}]$ can be unreliable, especially when the user inputs are insufficient or inconsistent to confidently construct $\hat{\pi}_{g_t}$. Hence, we aim to utilize $\mathbb{E}_{g_t}[\hat{\pi}_{g_t}]$ information only when it is confident, i.e., when $\mathbb{V}_{g_t}[\hat{\pi}_{g_t}(x)]$ is sufficiently small.

To account for uncertainties in both the surrogate and preference model, we multiply the two Gaussians. For any $x \in \mathcal{X}$, $f_t(x)$ is a Gaussian random variable in the target space of f, and $\hat{\pi}_{g_t}$ is a Gaussian random variable indicating the likelihood of x being the Condorcet winner. We therefore scale the preference function to

[1] True Copeland score is calculated by Eq.(2) but using true π instead of estimated $\hat{\pi}$, thus there is no uncertainty.

[2] typically defined as the most favoured player within $\mathcal{X}$.

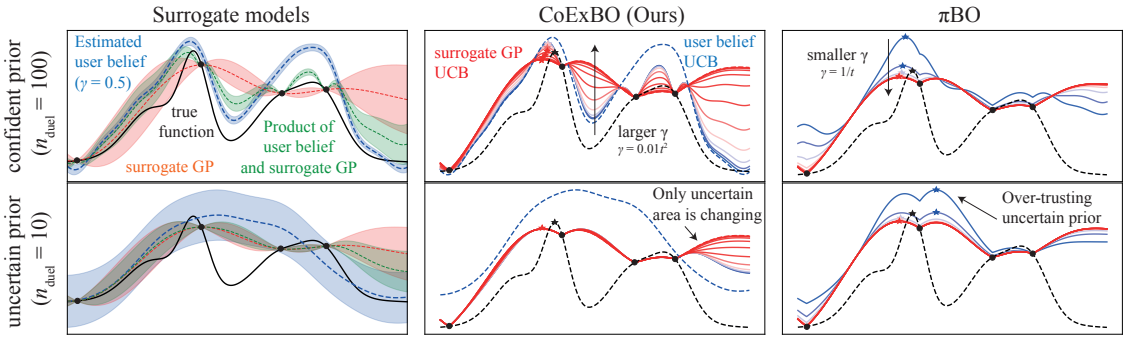


Figure 4: The CoExBO AF synthesizes GPs, utilizing one GP to represent the true function (red) and another to reflect user belief (blue), leading to the product GP (green). This product GP effectively assimilates the uncertainty inherent in user belief, adjusting the level of user belief integration during the acquisition process. Whereas the CoExBO AF is designed to adaptively manage the integration of uncertain user beliefs, the πBO approach tends to excessively depend on user belief, overlooking the uncertainty.

align with the surrogate model's scale. Using the property that the product of Gaussians is Gaussian, we derive a UCB-style AF.

Proposition 1. *Given* $f_t(x) \sim \mathcal{N}(\mu_{f_t}(x), \kappa_{f_t}(x,x))$, $\hat{\pi}_{g_t}(x) \sim \mathcal{N}(\mu_{g_t}(x), \sigma^2_{g_t}(x))$ *and a scaling function* ρ *that maps* $\hat{\pi}_{g_t}$ *to the scale of* f_t *and* $\gamma > 0$*, our new acquisition function* $\alpha_{f,\pi}$ *takes the following form:*

$$\alpha_{f_t,\hat{\pi}_t}(x) := \mu_{f_t,\hat{\pi}_t}(x) + \beta^{\frac{1}{2}}\sigma_{f_t,\hat{\pi}_t}(x) \tag{3}$$

where

$$\mu_{f_t,\hat{\pi}_t}(x) = \frac{\sigma^2_{f_t,\hat{\pi}_t}(x)}{\sigma^2_{\hat{\pi}_t}(x)}\mu_{\hat{\pi}_t}(x) + \frac{\sigma^2_{f_t,\hat{\pi}_t}(x)}{\sigma^2_{f_t}(x)}\mu_{f_t}(x), \tag{4}$$

$$\sigma^2_{f_t,\hat{\pi}_t}(x) = \frac{\sigma^2_{\hat{\pi}_t}(x)\sigma^2_{f_t}(x)}{\sigma^2_{\hat{\pi}_t}(x) + \sigma^2_{f_t}(x)}, \tag{5}$$

$$\mu_{\hat{\pi}_t}(x) := \rho(\mathbb{E}_{g_t}[\hat{\pi}_{g_t}(x)]), \tag{6}$$

$$\sigma^2_{\hat{\pi}_t}(x) := \rho^2(\mathbb{V}_{g_t}[\hat{\pi}_{g_t}(x)]) + \gamma t^2 \sigma^2_{f_t}(x), \tag{7}$$

$$\rho(x) := \mathbb{E}[\boldsymbol{y}_{obj_t}]x + \sqrt{\mathbb{V}[\boldsymbol{y}_{obj_t}]} \tag{8}$$

The new AF adheres to the following principle:

1. (Uncertainty in Preference Estimation) The information provided by $\hat{\pi}_t(x)$ becomes valuable only when its variance, denoted as $\mathbb{V}_{g_t}[\hat{\pi}_{g_t}(x)]$, is smaller than the variance of the surrogate model, represented as $\sigma^2_{f_t}(x)$.
2. (Uncertainty in Efficacy of User Information) $\hat{\pi}_{g_t}(x)$ becomes less significant as iterations proceed. This mirrors πBO's principle that human knowledge is most valuable in the early stages.

The simple product of two Gaussian distributions offers a heuristic solution to the above assumptions. Firstly, the resulting Gaussian mean is a weighted sum of two means weighted by their variances, aligning with our first assumption, where the mean $\mu_{f_t,\hat{\pi}_t}$ stays unchanged from μ_{f_t} when $\mathbb{V}_{g_t}[\hat{\pi}_{g_t}(x)] \gg \sigma^2_{f_t}(x)$. To address the second, we introduce a decay hyperparameter γ term in Eq. 7. When $\gamma t^2 \geq 1$, $\sigma^2_{\hat{\pi}_t}(x) > \sigma^2_{f_t}(x)$ holds, causing information from $\hat{\pi}_{g_t}$ to decay. While other methods could meet these principles, we choose the computationally simplest one.

Figure 4 illustrates typical behaviors of πBO and CoExBO. With a confident and accurate $\hat{\pi}_{g_t}$, both methods perform well. However, when dealing with uncertain $\hat{\pi}_{g_t}$, the peaks do not align with the true global maximum location. πBO tends to rely on user belief regardless of uncertainty, while CoExBO mitigates over-reliance on user belief in uncertain situations.

Candidate generation. Given f_t and $\hat{\pi}_t$, we generate a pair of candidates as follows:

$$x_1 = \arg\max_{x \in \mathcal{X}} \alpha_{f_t}(x) \qquad \text{(standard UCB)}$$

$$x_2 = \arg\max_{x \in \mathcal{X}} \alpha_{f_t,\hat{\pi}_t}(x) \qquad (\hat{\pi} \text{ incorporated UCB})$$

This approach is similar to other human-AI collaborative BO methods (Gupta et al., 2023; Khoshvishkaie et al., 2023), involving a direct comparison of BO with human recommendations. Opting for x_2 speeds up convergence if human input is superior while choosing x_1 is optimal if BO performs better. This represents a greedy approach to optimizing choices for both sides. A greedy approach is optimal in both scenarios since we assume that either human or BO is superior. The key difference lies in the decision-making process: in previous approaches, each agent independently selects their preferred option, necessitating separate queries. In contrast, our method employs BO to generate both candidates and then makes a selection. As our acquisition function $\alpha_{f,\hat{\pi}}$ gradually converges to the standard UCB, the selection process becomes equivalent over time. This prevents budget wastage resulting from suboptimal choices made by either side.

Note that our AF $\alpha_{f,\hat{\pi}}$ combines GP information and user beliefs, meaning it does not always align with user preferences. The GP component corrects any uncertainties or inaccuracies in user beliefs, preventing a persistent bias toward selecting x_2.

Regret analysis. By following Srinivas et al. (2010), we analyse the regret of preference-based AF $r_{\hat{\pi}_t} := f(x^*_{\text{true}}) - f(x_2)$ and the standrd UCB regret $r_t := f(x^*_{\text{true}}) - f(x_1)$. We assume good and bad user beliefs. A good user belief assumes to contain the true function within the standard deviation, whereas a bad user belief does with extra error with mean estimation.

Theorem 2. *Fix $t \geq 1$ and $\gamma > 0$. If $|f(\boldsymbol{x}) - \mu_{f_{t-1}}(\boldsymbol{x})| \leq \beta_t^{1/2}\sigma_{f_{t-1}}(\boldsymbol{x})$ for all $\boldsymbol{x} \in D$, $|D| < \infty$ hold, the ratio of regrets for with and without π augmentation $r_t, r_{\hat{\pi}_t}$ is bounded by:*
(Good user belief) If $|f(\boldsymbol{x}) - \mu_{f_{t-1},\hat{\pi}_{t-1}}(\boldsymbol{x})| \leq \beta_t^{1/2}\sigma_{f_{t-1},\hat{\pi}_{t-1}}(\boldsymbol{x})$ holds,

$$r_{\hat{\pi}_t} {r_t}^{-1} \leq R_{\hat{\pi}_t} < 1, \tag{9}$$

where
$R_{\hat{\pi}_t} = \sqrt{\frac{\rho^2(\mathbb{V}_{g_{t-1}}[\hat{\pi}_{g_{t-1}}(x_2)]) + \gamma(t-1)^2\sigma^2_{t-1}(x_2)}{\rho^2(\mathbb{V}_{g_{t-1}}[\hat{\pi}_{g_{t-1}}(x_2)]) + \gamma(t-1)^2\sigma^2_{t-1}(x_2) + \sigma^2_{t-1}(x_1)}}$.
(Bad user belief) If $|f(\boldsymbol{x}) - \mu_{f_{t-1},\hat{\pi}_{t-1}}(\boldsymbol{x})| \leq |\mu_{f_{t-1}}(\boldsymbol{x}) - \mu_{f_{t-1},\hat{\pi}_{t-1}}(\boldsymbol{x})| + \beta_t^{1/2}\sigma_{f_{t-1},\hat{\pi}_{t-1}}(\boldsymbol{x})$ holds,

$$r_{\hat{\pi}_t} {r_t}^{-1} \leq \Delta\mu_t + R_{\hat{\pi}_t t}, \tag{10}$$

where $\Delta\mu_t = \frac{|\mu_{t-1}(x_1) - \mu_{f_{t-1},\hat{\pi}_{t-1}}(x_2)|}{2\beta_t^{1/2}\sigma_{f_{t-1}}(x_1)}$.

The proof is given in Supplementary A. Using Theorem 2, we obtain the convergence rate of $\alpha_{f_{t-1},\hat{\pi}_{t-1}}$. This trivially follows the original convergence rate on UCB as in Srinivas et al. (2010):

Lemma 3. *(No harm guarantee) Given the regret in Theorem 2, The regret of a preference-based acquisition function, α_{pref_t}, asymptotically equals to the regret of an upper confidence bound strategy, UCB:*

$$\lim_{t\to\infty} r_t^{\pi} {r_t}^{-1} = 1, \tag{11}$$

so we obtain a convergence rate for $\alpha_{f_T,\hat{\pi}_T}$ of $\mathcal{O}(\sqrt{T\gamma_T \log T})$, an original UCB convergence rate.

Hence, we can ensure that the worst-case convergence rate remains unaffected, even with inaccurate user beliefs, for large t, where $\gamma t^2 \gg 1$. While short-term performance may not match the standard UCB, it often yields better empirical results. Particularly, a good user belief has a provably better regret bound. In our scenario, humans choose candidates from the UCB or $\alpha_{f_t,\hat{\pi}_t}$, reinforcing the no-harm guarantee. The human selection process does not impact the convergence rate because $\alpha_{f_t,\hat{\pi}_t}$ determines tighter or looser bounds than the UCB, depending on user beliefs. In practice, human knowledge evolves over iterations, positively influencing convergence, as demonstrated in our experiments section. The parameter γt^2 balances the integration of evolving human knowledge with the no-harm guarantee. It is worth noting that our approach differs from that of multitask GPs, as multitask GPs are vulnerable to unreliable low-fidelity GPs (Mikkola et al., 2023b).

3.3 Explaining candidate generation through Shapley values

To foster trust in the black-box optimizer among users, we employ Shapley values (Shapley, 1953), a popular solution concept from game theory adopted by the machine learning community (Lundberg & Lee, 2017; Chau et al., 2022c; Hu et al., 2022) to provide feature attributions for the acquisitions and the surrogate model. This provides users with a clearer understanding of the factors influencing the selection of candidates.

Shapley values follow a set of favourable rationality axioms, setting them apart from heuristic methods like extracting the length scale from a GP kernel. For a given function $h : \mathcal{X} \to \mathbb{R}$, a query location x, the Shapley value for feature j is expressed as

$$\phi_{j,x}(h) = \sum_{S \subseteq [d]\setminus\{j\}} c_{|S|} \left(\nu_{x,h}(S \cup i) - \nu_{x,h}(S)\right)$$

where $[d] := \{1, \ldots, d\}$, $c_{|S|} = \frac{1}{d}\binom{d-1}{|S|}^{-1}$, X_S is the subfeature vector of X for features in S and $\nu_{x,h}(S)$ measures a notion of contribution features S has to the prediction $h(x)$. We utilize the recently introduced GPSHAP (Chau et al., 2023) to explain the surrogate GP f. We illustrate how to estimate the Shapley values for α_f, but extending to $\alpha_{f,\pi}$ is straightforward. While in the Shapley explanation literature, it is suggested one should take the conditional expectation of the to-be-explained function, i.e. $\mathbb{E}_X[\alpha_f(X) \mid X_S = x_s]$, to structure the cooperative game. However, for computational reasons (see appendix), we opted to establish the game using its upper bound instead:

$$\nu_{x,f}(S) := \mathbb{E}[\mu_f(X) \mid X_S = x_S] + \beta_t^{1/2}\sqrt{\mathbb{E}[\kappa_f(X,X) \mid X_S = x_S]}.$$

$\nu_{x,f}(S)$ can be interpreted as the significance of the feature subset S measured by how much the upper bound has changed if we remove the contribution from other features in S^c by integration. This formulation allows us to estimate the quantity in analytical form:

Proposition 4. *Given $f \sim \mathcal{GP}(\mu, \kappa)$, for a given feature subset $S \subseteq \{1, \ldots, d\}$, and location $x, \nu_{x,f}(S)$ can*

Table 1: Comparisons between our proposed CoExBO with baselines used in the ablation study.

Baselines	Human selection	BO	π augmentation	Novel contributions: Interative π update	Novel contributions: Uncertainty in π estimation	Novel contributions: Explanation
random	✓	✗	✗	✗	✗	✗
manual search	✓	✗	✗	✗	✗	✗
UCB/TS	✗	✓	✗	✗	✗	✗
prior sampling	✗	✗	✓	✗	✗	✗
batch UCB/TS	✓	✓	✗	✗	✗	✗
πBO (Hvarfner et al., 2022)	✗	✓	✓	✗	✗	✗
CoExBO (πBO)	✓	✓	✓	✓	✗	✗
CoExBO	✓	✓	✓	✓	✓	✓

be estimated from observations as

$$\mathbf{B}_S(\mathbf{x})^\top \tilde{\mathbf{f}} + \beta_t^{1/2} \sqrt{\mathbf{B}_S(\mathbf{x})^\top \tilde{\mathbf{K}}_{\mathbf{XX}} \mathbf{B}_S(\mathbf{x})} \qquad (12)$$

where $\mathbf{B}_S(\mathbf{x}) = (\mathbf{K}_{\mathbf{X}_S\mathbf{X}_S} + \lambda_S I)^{-1} k_S(\mathbf{X}_S, \mathbf{x}_S)$, $\lambda_S > 0$ *is a regularisation parameter,* $\tilde{\mathbf{f}}$ *is the posterior mean, and* $\tilde{\mathbf{K}}_{\mathbf{X},\mathbf{X}}$ *is the posterior covariance matrix of the GP.*

To obtain this quantity, we utilized the fact that the conditional expectation of GPs also admits an analytical form; see Chau et al. (2021a,b) for further details. Other explanation features based on Shapley values are detailed in Supplementary E.

4 Experiments

CoExBO has been tested for synthetic and real-world tasks and is implemented using PyTorch (Paszke et al., 2019), GPyTorch (Gardner et al., 2018), BoTorch (Balandat et al., 2020), and SOBER (Adachi et al., 2023a). All experiments were averaged over 10 repeats, computed with a laptop PC[3]. We set the initial random samples for objective queries as $n_{\text{obj}} = 10$ and for preferential learning $n_{\text{pref}} = 100$, respectively.

Table 1 provides a summary of the baseline algorithms we evaluated. These include **Random**: This method generates a pair of i.i.d. samples uniformly, after which a human selects the preferred one. **Manual Search**: In this approach, a human selects the next query without any algorithmic assistance. **UCB** (Srinivas et al., 2010) and **Thompson Sampling (TS)** (Thompson, 1933): Both methods autonomously select the next query without human intervention. **Prior Sampling**: This technique involves choosing the next query point as an i.i.d. sample from the estimated prior $\hat{\pi}$, derived from the initial preference samples n_{pref}, without BO assistance. **BatchUCB** (Azimi et al., 2010) and **BatchTS** (Kandasamy et al., 2018): These algorithms generate pairs of candidates, from which a human selects one. They do not integrate human knowledge $\hat{\pi}$ in the candidate generation process. **πBO** (Hvarfner et al., 2022): This algorithm selects the next query point based on a $\hat{\pi}$-augmented AF, incorporating human knowledge through $\hat{\pi}$. However, it is not interactive (as it is fixed before running the BO) and does not account for uncertainty in $\hat{\pi}$ estimation or human interactive selection. Our proposed algorithm, **CoExBO**, incorporates all these elements. Its variant, **CoExBO (πBO)**, specifically analyzes the efficacy of our new AF by replacing it with the πBO's AF, $\alpha_{f_t}(\cdot)\mathbb{E}_{g_t}[\hat{\pi}_{g_t}(\cdot)]^{\frac{\gamma}{t}}$. Following the methodology in the original πBO paper, we set the decaying hyperparameter to 10 for πBO and γ to 0.01 for our algorithm.

[3]MacBook Pro 2019, 2.4 GHz 8-Core Intel Core i9, 64 GB 2667 MHz DDR4

4.1 Synthetic Functions with Synthetic Human Selection

Synthetic functions. First, CoExBO was tested with synthetic functions and a synthetic human selection as $\mathrm{H}(x_1, x_2)$ such that $f_{\text{human}}(x_1) > f_{\text{human}}(x_2)$, where $f_{\text{human}}(x) := f(x) + \epsilon_{\text{pref}}$, and $\epsilon_{\text{pref}} \sim \mathcal{N}(x; 0, \sigma^2_{\text{pref}})$. The correct human selection rate can be modified by changing σ^2_{pref}. $\sigma^2_{\text{pref}} = 0.1$ throughout this experiment. We have chosen the five commonly used test functions (Surjanovic & Bingham, 2023) (see details in Supplementary F.1).

Figure 5 shows the results on simple regret. CoExBO consistently outperforms four baselines except for the Rosenbrock function. This suggests human feedback is particularly effective for multimodal functions with many local minima. This is evident as there are no big differences between the algorithms with and without human intervention in the Rosenbrock function. Further details and computational complexity analysis can be found in Supplementary F.1.

Robustness evaluation. We tested CoExBO's robustness with the Ackley function regarding (a) uncertain prior and (b) incorrect human selections. We varied the prior confidence by changing the number of initial random samples ($n_{\text{pref}} = 10, 100, 500$). To vary human selection correctness, we adjusted

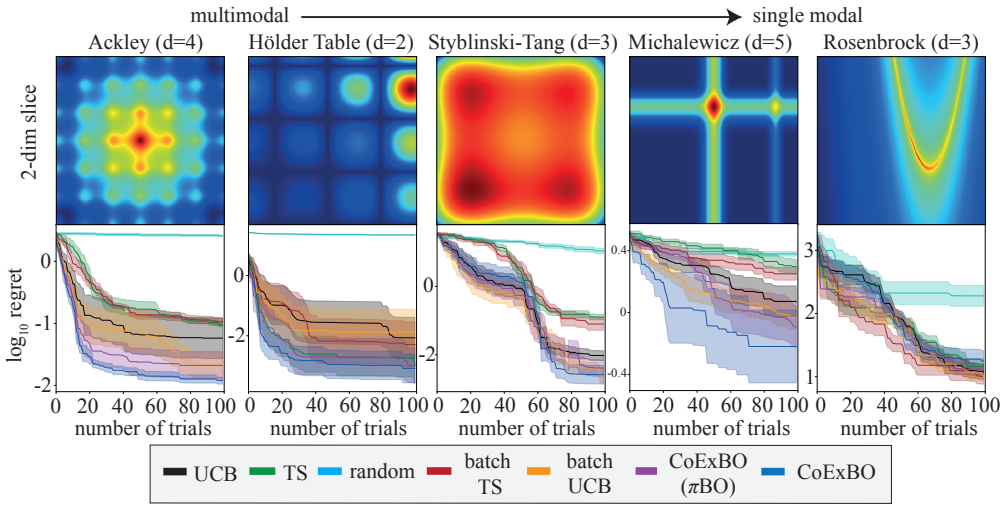


Figure 5: Convergence plot of simple regret for 5 synthetic functions with the synthetic selection accuracy ($\epsilon_{\text{pref}} := \mathcal{N}(0, 0.1^2)$). Lines and shaded area denote mean ± 1 standard error. CoExBO consistently outperforms all six baselines except for the Rosenbrock function. The dark red region is the global maximum.

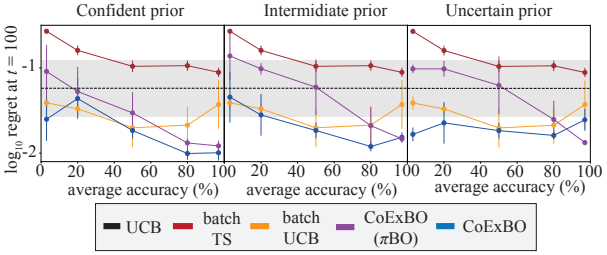


Figure 6: Convergence after 100 iterations on Ackley function ($d = 4$) with three prior confidence levels and five selection accuracy levels. Lines and error bars denote mean ± 1 standard error.

the noise variance of the synthetic human function ($\sigma^2_{\text{pref}} = 0.1, 1, 100$). We simulated adversarial selection cases by flipping the feedback from $\sigma^2_{\text{pref}} = 0.1, 1$.

Figure 6 demonstrates that CoExBO is robust against uncertain prior knowledge and incorrect human selections. While the optimistic πBO AF becomes less effective with reduced selection accuracy, CoExBO maintains its effectiveness better. The key distinction between πBO and CoExBO AFs is that the former only modifies the UCB in uncertain areas, whereas πBO adjusts it regardless of uncertainty (see Figure 4). In other words, once the wrongly believed position is queried, the uncertainty in that area decreases, leading to unbiased UCB. This feature offers greater resilience to incorrect and uncertain human selections compared to πBO AF. Note that both πBO and CoExBO are guaranteed to asymptotically approach the standard UCB, so longer iterations should yield similar results. Both batchUCB and CoExBO exhibited robustness against adversarial selection. However, the difference in adversarial selection falls within the standard error of UCB, indicating that it does not outperform the standard UCB in adversarial cases. Relative differences in tendencies provide more reliable insights. Further details are explained in Supplementary F.1.

4.2 Real-World Tasks with Human Experts

Lithium-ion batteries are the key to realizing the electrification of many sectors as a climate action. However, due to their complicated chemical nature, a perfect simulator that predicts required material properties under all operational and degradation conditions does not exist. Hence, researchers need to repeat costly laboratory experiments to find the best combinations of materials from which to build new lithium-ion batteries.

We assessed expert advice's effectiveness with four battery researchers and compared results with and without explainability features to gauge their impact on selection accuracy. The problem involves finding the best electrolyte material combination to maximize ionic conductivity. This task is challenging due to complex solvation and many-body effects (Gering, 2017), making prediction with a simulator-based approach difficult. We applied CoExBO to two electrolyte design problems: one involving four materials (EC-DMC-EMC-$LiPF_6$) (Dave et al., 2022), a well-known combination, and the other comprising MA-DMC-EMC-$LiPF_6$ (Logan et al., 2018), an unfamiliar composition to all participating experts. While materials science knowledge can deduce the effect on lithium-ion solvation states by changing from carbonate to acetate non-aqueous solvents, their knowledge is qualitative and not quantitative.

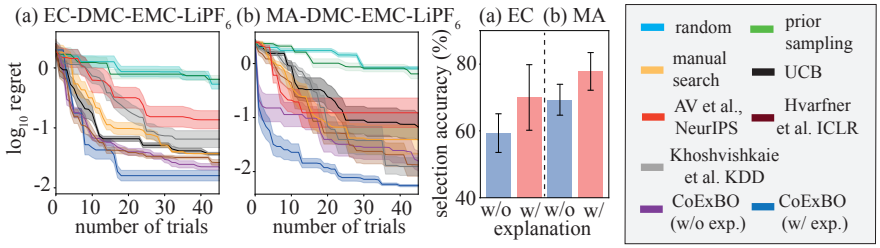


Figure 7: Convergence plot and selection accuracy of two battery material design tasks (a) EC (b) MA-based system. The explainability strengthens the selection accuracy and accelerates convergence.

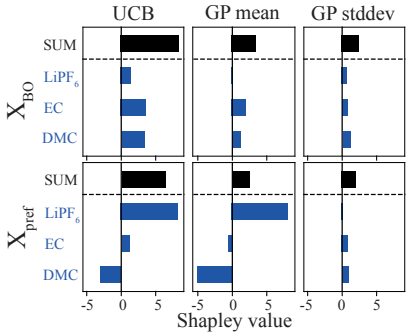


Figure 8: An example of Shapley values for UCB, GP predictive mean and standard deviation during the iteration at two recommended locations.

Figure 7 shows that CoExBO with expert knowledge accelerates convergence, even without explainability features. The addition of explainability results in a significant speedup in time-to-accuracy and enhancement in selection accuracy, which outperformed eight baselines. Figure 8 exemplifies a typical case of Shapley values. The black bar's sum of Shapley values shows that X_{BO} has a higher UCB value, indicating that selecting X_{BO} is natural in standard BO. However, the preference-based X_{pref} attributes more reasonable importance to conductivity, consistent with chemical expertise—highlighting $LiPF_6$ as the key material. We illustrate the CoExBO's effectiveness with two distinct cases: One participant initially relied heavily on BO suggestions, a phenomenon known as automation bias (Cummings, 2004). Goddard et al. (2012); Skitka et al. (2000) have shown that explaining the process and holding users accountable for their decision accuracy can reduce such biases, both of which CoExBO achieved. Another participant consistently trusted their own preferences, even when the GP improved. CoExBO's no-harm guarantee shepherds users to global convergence.

In summary, expert knowledge can be particularly helpful to the GP in two ways: (A) Assessing the reliability of noisy ionic conductivity measurements and disregarding noise to infer the true function shape. (B) Applying their chemical knowledge to roughly optimize the composition, exploiting the knowledge of each material's importance. On the other hand, CoExBO can help experts in three ways: (A) BO is better at fine-tuning more precisely than experts, as expert knowledge only focuses on the main effect. (B) Shapley values provide accurate importance rankings conditioned on x, which updates experts' knowledge. (C) The feedback and explaining feature can correct experts' wrong understanding and guide them in the true optimal direction.

5 Discussions and Limitations

Our approach accelerates convergence when experts possess accurate comparative knowledge that the GP cannot access, which remains a strong assumption, albeit weaker than the conventional ones with a well-defined and error-free belief function or the optimal query. Expert knowledge can be particularly helpful to the GP in three ways: (a) as a good warm starter, allowing it to begin from a promising region; (b) as a good encoder, compensating for the information lost during simplification to a low-dimensional search space; and (c) as a noise reducer, providing a more accurate estimation of experimental noise. These aspects align well with fields such as chemistry and scientific experiments, and experts can convey this complex information through simple selections. Our proofs are specific to the UCB setting, but the decay property can ensure convergence for any AF. In our experiments with experts, we observed that there is a tendency to expect both surrogate and explanation models to provide an 'oracle' understanding of the whole scientific process. We emphasise this is not the purpose of explainable BO as we are by definition, operating under a small data regime. However, our collaboration and explanation framework allows us to demystify the BO process and thus mitigate over-trust.

There is a growing interest in putting humans back into the optimization cycle. A prime example is the RLHF to fine-tune LLMs (Christiano et al., 2017; Rafailov et al., 2024). We also observe concurrent works centered on enhancing human-AI collaboration (AV et al., 2024) and explainable BO (Chakraborty et al., 2023, 2024), showcasing this as a promising direction of research.

Acknowledgements

We thank Philipp Dechent and Katie Lukow for participating in the real-world experiments, and anonymous reviewers who gave useful comments. Masaki Adachi was supported by the Clarendon Fund, the Oxford Kobe Scholarship, the Watanabe Foundation, and Toyota Motor Corporation.

References

Masaki Adachi. High-dimensional discrete Bayesian optimization with self-supervised representation learning for data-efficient materials exploration. In *NeurIPS 2021 AI for Science Workshop*, 2021. doi: https://openreview.net/forum?id=xJhjehqjQeB.

Masaki Adachi, Satoshi Hayakawa, Martin Jørgensen, Harald Oberhauser, and Michael A Osborne. Fast Bayesian inference with batch Bayesian quadrature via kernel recombination. *Advances in Neural Information Processing Systems*, 35, 2022. doi: https://doi.org/10.48550/arXiv.2206.04734.

Masaki Adachi, Satoshi Hayakawa, Saad Hamid, Martin Jørgensen, Harald Oberhauser, and Michael A Osborne. SOBER: Highly parallel Bayesian optimization and Bayesian quadrature over discrete and mixed spaces. In *ICML 2023 Workshop: Sampling and Optimization in Discrete Space*, 2023a. doi: https://doi.org/10.48550/arXiv.2301.11832.

Masaki Adachi, Yannick Kuhn, Birger Horstmann, Arnulf Latz, Michael A Osborne, and David A Howey. Bayesian model selection of lithium-ion battery models via Bayesian quadrature. *IFAC-PapersOnLine*, 56(2):10521–10526, 2023b. doi: https://doi.org/10.1016/j.ifacol.2023.10.1073.

Masaki Adachi, Satoshi Hayakawa, Martin Jørgensen, Xingchen Wan, Vu Nguyen, Harald Oberhauser, and Michael A Osborne. Adaptive batch sizes for active learning: A probabilistic numerics approach. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2024. doi: https://doi.org/10.48550/arXiv.2306.05843.

Arun Kumar AV, Santu Rana, Alistair Shilton, and Svetha Venkatesh. Human-AI collaborative Bayesian Optimisation. *Advances in Neural Information Processing Systems*, 35:16233–16245, 2022.

Arun Kumar AV, Alistair Shilton, Sunil Gupta, Santu Runa, Stewart Greenhill, and Svetha Venkatesh. Enhanced bayesian optimization via preferential modeling of abstract properties. *arXiv preprint arXiv:2402.17343*, 2024.

Javad Azimi, Alan Fern, and Xiaoli Fern. Batch Bayesian optimization via simulation matching. *Advances in Neural Information Processing Systems*, 23, 2010.

Maximilian Balandat, Brian Karrer, Daniel Jiang, Samuel Daulton, Ben Letham, Andrew G Wilson, and Eytan Bakshy. BoTorch: a framework for efficient Monte-Carlo Bayesian optimization. *Advances in neural information processing systems*, 33:21524–21538, 2020.

Alessio Benavoli, Dario Azzimonti, and Dario Piga. Learning choice functions with Gaussian processes. In *The 39th Conference on Uncertainty in Artificial Intelligence*, 2023.

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2), 2012.

Julian Berk, Sunil Gupta, Santu Rana, and Svetha Venkatesh. Randomised Gaussian process upper confidence bound for Bayesian optimisation. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pp. 2284–2290, 2020.

Xavier Bouthillier and Gaël Varoquaux. Survey of machine-learning experimental methods at NeurIPS2019 and ICLR2020. *HAL 02447823*, 2020. doi: https://hal.science/hal-02447823.

Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Jerry F Casteel and Edward S Amis. Specific conductance of concentrated solutions of magnesium salts in water-ethanol system. *Journal of Chemical and Engineering Data*, 17(1):55–59, 1972.

Tanmay Chakraborty, Christian Wirth, and Christin Seifert. Post-hoc rule based explanations for black box bayesian optimization. In *European Conference on Artificial Intelligence*, pp. 320–337. Springer, 2023.

Tanmay Chakraborty, Christin Seifert, and Christian Wirth. Explainable bayesian optimization. *arXiv preprint arXiv:2401.13334*, 2024.

Siu Lun Chau, Shahine Bouabid, and Dino Sejdinovic. Deconditional downscaling with Gaussian processes. *Advances in Neural Information Processing Systems*, 34:17813–17825, 2021a.

Siu Lun Chau, Jean-Francois Ton, Javier González, Yee Teh, and Dino Sejdinovic. Bayesimp: Uncertainty quantification for causal data fusion. *Advances in Neural Information Processing Systems*, 34:3466–3477, 2021b.

Siu Lun Chau, Mihai Cucuringu, and Dino Sejdinovic. Spectral ranking with covariates. In *Joint European*

Conference on Machine Learning and Knowledge Discovery in Databases, pp. 70–86. Springer, 2022a.

Siu Lun Chau, Javier González, and Dino Sejdinovic. Learning inconsistent preferences with Gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pp. 2266–2281. PMLR, 2022b.

Siu Lun Chau, Robert Hu, Javier Gonzalez, and Dino Sejdinovic. RKHS-SHAP: Shapley values for kernel methods. *Advances in Neural Information Processing Systems*, 35:13050–13063, 2022c.

Siu Lun Chau, Krikamol Muandet, and Dino Sejdinovic. Explaining the uncertain: Stochastic Shapley values for Gaussian process models. *arXiv preprint arXiv:2305.15167*, 2023.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Wei Chu and Zoubin Ghahramani. Preference learning with Gaussian processes. In *Proceedings of the 22nd International Conference on Machine Learning*, pp. 137–144, 2005.

Abdoulatif Cisse, Xenophon Evangelopoulos, Sam Carruthers, Vladimir V Gusev, and Andrew I Cooper. HypBO: Expert-guided chemist-in-the-loop Bayesian search for new materials. *arXiv preprint arXiv:2308.11787*, 2023.

Fabio Colella, Pedram Daee, Jussi Jokinen, Antti Oulasvirta, and Samuel Kaski. Human strategic steering improves performance of interactive optimization. In *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization*, pp. 293–297, 2020.

Mihai Cucuringu. Sync-rank: Robust ranking, constrained ranking and rank aggregation via eigenvector and SDP synchronization. *IEEE Transactions on Network Science and Engineering*, 3(1):58–79, 2016.

Mary Cummings. Automation bias in intelligent time critical decision support systems. In *AIAA 1st intelligent systems technical conference*, pp. 6313, 2004.

Adarsh Dave, Jared Mitchell, Sven Burke, Hongyi Lin, Jay Whitacre, and Venkatasubramanian Viswanathan. Autonomous optimization of non-aqueous Li-ion battery electrolytes via robotic experimentation and machine learning coupling. *Nature communications*, 13(1):5454, 2022.

Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. Efficient and robust automated machine learning. *Advances in neural information processing systems*, 28, 2015.

Jacob Gardner, Geoff Pleiss, Kilian Q Weinberger, David Bindel, and Andrew G Wilson. GPyTorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration. In *Advances in Neural Information Processing Systems*, pp. 7576–7586, 2018.

Roman Garnett. *Bayesian optimization*. Cambridge University Press, 2023.

Paul H Garthwaite, Joseph B Kadane, and Anthony O'Hagan. Statistical methods for eliciting probability distributions. *Journal of the American statistical Association*, 100(470):680–701, 2005.

Michael A Gelbart, Jasper Snoek, and Ryan P Adams. Bayesian optimization with unknown constraints. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, pp. 250—-259, 2014. doi: https://doi.org/10.48550/arXiv.1403.5607.

Kevin L Gering. Prediction of electrolyte conductivity: results from a generalized molecular model based on ion solvation and a chemical physics framework. *Electrochimica Acta*, 225:175–189, 2017.

Kate Goddard, Abdul Roudsari, and Jeremy C Wyatt. Automation bias: a systematic review of frequency, effect mediators, and mitigators. *Journal of the American Medical Informatics Association*, 19(1): 121–127, 2012.

Javier González, Michael Osborne, and Neil Lawrence. Glasses: Relieving the myopia of Bayesian optimisation. In *Artificial Intelligence and Statistics*, pp. 790–799. PMLR, 2016.

Javier González, Zhenwen Dai, Andreas Damianou, and Neil D Lawrence. Preferential Bayesian optimization. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 1282–1291, 2017.

Sunil Gupta, Alistair Shilton, Arun Kumar AV, Shannon Ryan, Majid Abdolshah, Hung Le, Santu Rana, Julian Berk, Mahad Rashid, and Svetha Venkatesh. BO-Muse: A human expert and AI teaming framework for accelerated experimental design. *arXiv preprint arXiv:2303.01684*, 2023.

Satoshi Hayakawa, Harald Oberhauser, and Terry Lyons. Positively weighted kernel quadrature via subsampling. *Advances in Neural Information Processing Systems*, 35:6886–6900, 2022.

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*, pp. 1699–1707. PMLR, 2015.

Robert Hu, Siu Lun Chau, Jaime Ferrando Huertas, and Dino Sejdinovic. Explaining preferences with shapley values. *Advances in Neural Information Processing Systems*, 35:27664–27677, 2022.

Daolang Huang, Louis Filstroff, Petrus Mikkola, Runkai Zheng, and Samuel Kaski. Bayesian optimization augmented with actively elicited expert knowledge. *arXiv preprint arXiv:2208.08742*, 2022.

Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *Learning and Intelligent Optimization: 5th International Conference, LION 5, Rome, Italy, January 17-21, 2011. Selected Papers 5*, pp. 507–523. Springer, 2011.

Carl Hvarfner, Danny Stoll, Artur Souza, Marius Lindauer, Frank Hutter, and Luigi Nardi. πBO: Augmenting acquisition functions with user beliefs for bayesian optimization. In *International Conference on Learning Representations*, 2022.

Michael I Jordan. Artificial intelligence—the revolution hasn't happened yet. *Harvard Data Science Review*, 1(1):1–9, 2019.

Daniel Kahneman and Amos Tversky. On the interpretation of intuitive probability: A reply to Jonathan Cohen. *Cognition*, 7(4):409–411, 1979.

Motonobu Kanagawa, Bharath K Sriperumbudur, and Kenji Fukumizu. Convergence guarantees for kernel-based quadrature rules in misspecified settings. *Advances in Neural Information Processing Systems*, 29, 2016.

Keren J Kanarik, Wojciech T Osowiecki, Yu Lu, Dipongkar Talukder, Niklas Roschewsky, Sae Na Park, Mattan Kamon, David M Fried, and Richard A Gottscho. Human–machine collaboration for improving semiconductor process development. *Nature*, 616 (7958):707–711, 2023.

Kirthevasan Kandasamy, Akshay Krishnamurthy, Jeff Schneider, and Barnabás Póczos. Parallelised Bayesian optimisation via Thompson sampling. In *International Conference on Artificial Intelligence and Statistics*, pp. 133–142. PMLR, 2018.

Ali Khoshvishkaie, Petrus Mikkola, Pierre-Alexandre Murena, and Samuel Kaski. Cooperative Bayesian optimization for imperfect agents. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 475–490. Springer, 2023.

Michael Y Li and Ryan P Adams. Explainability constraints for Bayesian optimization. In *6th ICML Workshop on Automated Machine Learning*, 2020.

Dong C Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.

ER Logan, Erin M Tonita, KL Gering, Jing Li, Xiaowei Ma, LY Beaulieu, and JR Dahn. A study of the physical properties of Li-ion battery electrolytes containing esters. *Journal of The Electrochemical Society*, 165(2):A21, 2018.

R Duncan Luce. On the possible psychophysical laws. *Psychological review*, 66(2):81, 1959.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, pp. 4765–4774, 2017.

Petrus Mikkola, Osvaldo A. Martin, Suyog Chandramouli, Marcelo Hartmann, Oriol Abril Pla, Owen Thomas, Henri Pesonen, Jukka Corander, Aki Vehtari, Samuel Kaski, Paul-Christian Bürkner, and Arto Klami. Prior Knowledge Elicitation: The Past, Present, and Future. *Bayesian Analysis*, pp. 1 – 33, 2023a. doi: 10.1214/23-BA1381.

Petrus Mikkola, Julien Martinelli, Louis Filstroff, and Samuel Kaski. Multi-fidelity Bayesian optimization with unreliable information sources. In *International Conference on Artificial Intelligence and Statistics*, pp. 7425–7454. PMLR, 2023b.

Dimitrios Milios, Raffaello Camoriano, Pietro Michiardi, Lorenzo Rosasco, and Maurizio Filippone. Dirichlet-based Gaussian processes for large-scale calibrated classification. *Advances in Neural Information Processing Systems*, 31, 2018.

Jonas Mockus. The application of Bayesian methods for seeking the extremum. *Towards global optimization*, 2:117, 1998.

Anthony O'Hagan. Bayes–Hermite quadrature. *Journal of statistical planning and inference*, 29(3):245–260, 1991.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, and Luca Antiga. PyTorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

Anil Ramachandran, Sunil Gupta, Santu Rana, Cheng Li, and Svetha Venkatesh. Incorporating expert prior in Bayesian optimisation via space warping. *Knowledge-Based Systems*, 195:105663, 2020.

Denise M Rousseau. Schema, promise and mutuality: The building blocks of the psychological contract. *Journal of occupational and organizational psychology*, 74(4):511–541, 2001.

Lloyd S Shapley. A value for n-person games. *Contributions to the Theory of Games*, 2(28):307–317, 1953.

Linda J Skitka, Kathleen Mosier, and Mark D Burdick. Accountability and automation bias. *International*

Journal of Human-Computer Studies, 52(4):701–717, 2000.

Jasper Snoek, Kevin Swersky, Rich Zemel, and Ryan Adams. Input warping for Bayesian optimization of non-stationary functions. In *International Conference on Machine Learning*, pp. 1674–1682. PMLR, 2014.

Il'ya Meerovich Sobol'. On the distribution of points in a cube and the approximate evaluation of integrals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, 7(4):784–802, 1967.

Jialin Song, Yuxin Chen, and Yisong Yue. A general framework for multi-fidelity Bayesian optimization with Gaussian processes. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 3158–3167. PMLR, 2019.

Artur Souza, Luigi Nardi, Leonardo B Oliveira, Kunle Olukotun, Marius Lindauer, and Frank 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 21*, pp. 265–296. Springer, 2021.

Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *International Conference on International Conference on Machine Learning*, pp. 1015–1022, 2010.

S. Surjanovic and D. Bingham. Virtual library of simulation experiments: Test functions and datasets. Retrieved October 6, 2023, from `http://www.sfu.ca/~ssurjano`, 2023.

Shion Takeno, Yu Inatsu, and Masayuki Karasuyama. Randomized Gaussian process upper confidence bound with tight Bayesian regret bounds. In *International Conference on Machine Learning*, volume 202, pp. 33490–33515, 2023.

William R 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.

Zi Wang, George E Dahl, Kevin Swersky, Chansoo Lee, Zelda Mariet, Zachary Nado, Justin Gilmer, Jasper Snoek, and Zoubin Ghahramani. Pre-trained Gaussian processes for Bayesian optimization. *arXiv preprint arXiv:2109.08215*, 2021.

Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006.

Jian Wu, Saul Toscano-Palmerin, Peter I Frazier, and Andrew Gordon Wilson. Practical multi-fidelity Bayesian optimization for hyperparameter tuning. In *Uncertainty in Artificial Intelligence*, pp. 788–798. PMLR, 2020.

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes in supplementary]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes at https://anonymous.4open.science/r/CoExBO-4B06/]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes/No/Not Applicable]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes in text and supplementary]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]

(d) Information about consent from data providers/curators. [Yes in supplementary]

(e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

(a) The full text of instructions given to participants and screenshots. [Yes]

(b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]

(c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Clarification

The Appendix is provided in Section D. Our proof of the no-harm guarantee is presented using simple regret for finite domain $|\mathcal{X}| < \infty$. However, it is typically expected to prove a cumulative regret bound, as it is more closely related to the convergence rate for continuous domain. Unfortunately, deriving an information-theoretic bound for a preferential GP is non-trivial due to the non-conjugate combination of a Gaussian prior and Bernoulli likelihood. As a result, its cumulative regret bound remains an open question in the Bayesian optimization community.

Errata

The proofs of Theorem 2 (Appendix A.2) should have been derived separately for the upper bounds of objective and subjective beliefs, rather than deriving the fraction simultaneously. However, this does not affect the results, which remain unchanged.

Alternative approach

Another potential approach is to use skewed GP, Benavoli et al. [24], which provides a conjugate predictive posterior for Bernoulli likelihood. Nevertheless, deriving the information-theoretic bound for its cumulative predictive covariance remains a significant challenge.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Looking in the Human: Collaborative and Explainable Bayesian Optimization
Publication Status	☑Published □ Accepted for Publication □Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	Masaki Adachi, Brady Planden, David A. Howey, Michael A. Osborne, Sebastian Orbell, Natalia Ares, Krikamol Muandet, and Siu Lun Chau. Looping in the Human: Collaborative and Explainable Bayesian Optimization. In \textit{International Conference on Artificial Intelligence and Statistics (AISTATS)} 238, 505-513, 2024.

Student Confirmation

Student Name:	Masaki Adachi		
Contribution to the Paper	first author I developed the core idea of a human-AI collaborative setting with explainable Bayesian optimization, proved most of the theoretical results, implemented the methodology. I conducted the experiments for the paper, performed additional analyses, and authored the manuscript.		
Signature 		Date	06 January 2025

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Michael A. Osborne, Professor of Machine Learning		
Supervisor comments I can confirm that, to the best of my knowledge, Masaki's description above is fair, and that I have great trust in Masaki.		
Signature 	Date	3 February 2025

This completed form should be included in the thesis, at the end of the relevant chapter.

It's frightening to think that you might not know something, but more frightening to think that, by and large, the world is run by people who have faith that they know exactly what's going on. [180]

— Amos Tversky, Cognitive and Mathematical Psychologist

Human rational behavior is shaped by a scissors whose two blades are the structure of task environments and the computational capabilities of the actor. [201]

— Herbert Simon, Nobel Prize in Economics and Turing Award winner

7

Principled Bayesian optimisation in collaboration with human experts

This chapter is based on the following publication:

> Wenjie Xu*, **Masaki Adachi***, Colin N. Jones, and Michael A. Osborne. Principled Bayesian optimisation in collaboration with human experts. In *Advances in Neural Information Processing Systems (NeurIPS)* 37, 2024.
>
> *Equal contribution

In Chapter 7, we redesign the human-AI collaborative framework for Bayesian optimization introduced in Chapter 6, making the process more principled and cognitively easier through an information-theoretic approach. In Chapter 6, the preferential GP framework limited our ability to derive a cumulative regret bound for continuous domain, which hindered the development of trust-level adjustments for expert advice and a stopping criterion for their involvement. To address these challenges, we employ a kernelised logistic regression model with an optimistic MLE approach. This methodology enables the derivation of the cumulative regret bound for the algorithm, facilitating both trust-level adjustments and a stopping criterion. Additionally, we modified the feedback form to binary recommendations

on the desirability of the next candidate points ('good' or 'bad'), thereby avoiding comparison-based utility estimation and relaxing the completeness assumption.

Principled Bayesian Optimisation in Collaboration with Human Experts

Wenjie Xu[*1,2], **Masaki Adachi**[*,3,4], **Colin N. Jones**[1], **Michael A. Osborne**[3],
[1] Automatic Control Laboratory, EPFL [2] Urban Energy Systems Laboratory, Empa
[3]Machine Learning Research Group, University of Oxford
[4] Toyota Motor Corporation
{wenjie.xu, colin.jones}@epfl.ch, {masaki, mosb}@robots.ox.ac.uk

Abstract

Bayesian optimisation for real-world problems is often performed interactively with human experts, and integrating their domain knowledge is key to accelerate the optimisation process. We consider a setup where experts provide advice on the next query point through binary accept/reject recommendations (labels). Experts' labels are often costly, requiring efficient use of their efforts, and can at the same time be unreliable, requiring careful adjustment of the degree to which any expert is trusted. We introduce the first principled approach that provides two key guarantees. (1) Handover guarantee: similar to a no-regret property, we establish a sublinear bound on the cumulative number of experts' binary labels. Initially, multiple labels per query are needed, but the number of expert labels required asymptotically converges to zero, saving both expert effort and computation time. (2) No-harm guarantee with data-driven trust level adjustment: our adaptive trust level ensures that the convergence rate will not be worse than the one without using advice, even if the advice from experts is adversarial. Unlike existing methods that employ a user-defined function that hand-tunes the trust level adjustment, our approach enables data-driven adjustments. Real-world applications empirically demonstrate that our method not only outperforms existing baselines, but also maintains robustness despite varying labelling accuracy, in tasks of battery design with human experts.

1 Introduction

Bayesian optimisation (BO) [60, 65, 33] is a successful approach to black-box optimisation that has been applied across a wide array of applications. BO is often praised for 'taking the human out of the loop' [80] by automating laborious optimisation processes, such as hyperparameter optimisation [29, 103] and neural architecture search [74, 99], thus relieving human users from these tasks. Nonetheless, a growing trend involves the opposite direction, which brings humans back into the loop and leverages human expertise as an adviser to the optimiser [7]. This human-in-the-loop approach is particularly relevant to scientific and explorative tasks, such as materials discovery [24, 2] and product design [48, 44, 7]. Experts have accumulated domain knowledge and should be helpful in accelerating the optimisation process, yet their experience and knowledge are often qualitative—they can struggle to express their knowledge in a functional form or to pinpoint the best candidates as an absolute quantity [47]. At the forefront of science, experts are also in the middle of trial and error; demanding well-defined and error-free inputs can limit the applicable range of BO. As such, a human-AI collaborative setting in BO has emerged, driven by practical demands, and has been gaining popularity in the literature [11, 43, 39, 50, 24, 7, 70, 12, 42].

[*]Equal contribution

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

A prevalent issue in this domain is the lack of both shared assumptions and theoretical guarantees, making fair comparisons challenging. Our community has yet to reach a consensus on acceptable assumptions, particularly in the following areas. **(a) The level of effectiveness of experts' knowledge:** assuming near oracle-like knowledge, e.g. in [11, 39, 12], collaborative settings can significantly surpass vanilla BO. However, if experts are entirely erroneous (yet confident)—which can happen [43, 50, 24, 7]—overreliance on experts' input cannot guarantee the global optimum convergence. **(b) Human interaction method:** ideally, humans prefer minimising interaction with machines for convenience. Minimising interaction leads to maximising the information at each query to human, which often ends up requesting error-free and quantitative information for humans [82, 11, 43, 42]. However, accurate knowledge elicitation remains a long-standing quest [79, 68, 58]. Inversely, when we assume human belief is also a black-box function and require the elicitation of the belief function through statistical modelling, e.g. [73, 34, 7], we will demand excessive queries of the experts.

Contributions. We propose an expert-advised algorithm with the contributions summarised below:

1. **Handover guarantee**: we model the expert's role as cognitively simple and qualitative—the expert serves as a black-box classifier, providing binary labels on the desirability of the next query location. Similar to the no-regret property, we establish a sublinear bound on the cumulative number of binary labels needed. Initially, multiple labels per query are needed, but the frequency of querying binary labels asymptotically converges to zero, thus saving both expert effort and computation time.
2. **No-harm guarantee**: we show that the convergence rate of our expert-advised algorithm will not be worse than that of vanilla BO (i.e. without expert advice), even if the advice from experts is adversarial. Our convergence is achieved through data-driven trust level adjustments, and is unlike existing methods that rely on hand-tuned user-defined functions.
3. **Real-world contribution**: empirically, our algorithm provides both fast convergence and resilience against erroneous inputs. It outperformed existing methods in both popular synthetic, and new real-world, tasks in designing lithium-ion batteries.

2 Problem Statement

We address the black-box optimization problem,

$$x^\star \in \arg\min_{x\in\mathcal{X}} \; f(x)\;, \tag{1}$$

while collaborating with an expert, where $\mathcal{X} \subset \mathbb{R}^d$ and d is the dimension.

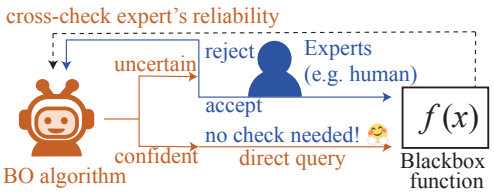


Figure 1: **BO-expert collaboration framework**: The algorithm (red) decides if an expert's (blue) label is necessary. If rejected, it generates a different candidate; otherwise, it directly queries.

Expert labelling model. We model an expert as a binary labeller (see Fig. 1). An expert labels a point $x \in \mathcal{X}$ as either 'accept' or 'reject'. An 'accept' label indicates that the point is worth sampling, while 'reject' label indicates it is not. These labels are binary, with 0 for 'accept' and 1 for 'reject'. In practice, the labelling process can be noisy, since humans may find some points hard to classify. Non-expert or incorrect belief may label the optimum $x^\star$ 'reject'. The distribution of the labels is determined by the expert's prior belief about the black-box function f, and we model the expert's belief through another unknown black-box function g.

Assumption 2.1. The notation $x \succ_g 0$ denotes the event where x is labelled as 'reject', based on the expert's belief function g. Additionally, the random indicator $\mathbf{1}_{x\succ_g 0} \in \{0, 1\}$ takes value 1 if $x \succ_g 0$ and 0 otherwise. The probability distribution of $\mathbf{1}_{x\succ_g 0} \in \{0, 1\}$ follows the Bernoulli distribution with $\mathbb{P}(\mathbf{1}_{x\succ_g 0} = 1) = p_{x\succ_g 0} = S(g(x))$, where $S(u) = 1/(1+e^{-u})$ is the sigmoid function.

Example 2.2. Let us define an example 'synthetic' expert's labelling response as $p_{x\succ_g 0} = S(a\rho(f(x)))$, where a is the accuracy coefficient and ρ is the linear scaling function from bound $[\min_{x\in\mathcal{X}} f(x), \max_{x\in\mathcal{X}} f(x)]$ to $[-3, 3]$. When $a = 1$, $\rho(f(x^*)) = -3$, $S(-3) \approx 0.05$, resulting in a Bernoulli distribution that yields an acceptance label of 0 with a 95% chance at the global minimum x^*. In this case, the sharpness of the belief $p_{x\succ_g 0}$ is influenced by both the shape of $f(x)$ and a; if $f(x)$ is peaky or $a \gg 1$, the expert can nearly pinpoint x^*.

However, in reality, the expert does not know the exact true f and therefore, we consider g to be a 'subjective' belief function representing f. This differs from a typical surrogate model $\hat{f}$ of f, which infers an 'objective' belief function from oracle queries. If g has better predictive ability than the surrogate model $\hat{f}$, exploiting g can accelerate convergence; otherwise it may decelerate the process. In the optimisation process, g may act as a regularizer function in addition to the objective function f. For simplicity, we use this Ex. 2.2 as synthetic human feedback. Readers interested in other examples are encouraged to refer to Appendix H.

Assumption 2.3. $\mathcal{X}$ is compact and non-empty.

Assumption 2.3 is reasonable because in many applications (e.g., continuous hyperparameter tuning) of BO, we are able to restrict the optimisation into certain ranges based on domain knowledge. Regarding the black-box function f and the function g, we assume that,

Assumption 2.4. $f \in \mathcal{H}_{k_f}, g \in \mathcal{H}_{k_g}$, where $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, representing k_f or k_g, is a symmetric, positive-semidefinite kernel function, and $\mathcal{H}_k$ is its corresponding reproducing kernel Hilbert space (RKHS, see [77]). Furthermore, we assume $\|f\|_{k_f} \leq B_f$ and $\|g\|_{k_g} \leq B_g$, where $\|\cdot\|_k$ is the norm induced by the inner product in the corresponding RKHS $\mathcal{H}_k$. We use $\mathcal{B}_g$ to denote the set $\left\{\tilde{g} \in \mathcal{H}_{k_g} \mid \|\tilde{g}\|_{k_g} \leq B_g\right\}$.

Assumption 2.4 requires that the objective f and the function g are regular in the sense that they have bounded norms in the corresponding RKHS, which is a common assumption.

Assumption 2.5. $k(x, x') \leq 1, x, x' \in \mathcal{X}$, and $k(x, x')$ is continuous on $\mathbb{R}^d \times \mathbb{R}^d$.

Assumption 2.6. At step t, if query point x_t is evaluated, we get a noisy evaluation of f (we refer to an oracle query), $y_t = f(x_t) + \xi_t$, where ξ_t is i.i.d σ-sub-Gaussian noise with fixed $\sigma > 0$.

Notation. We refer to $\mathbf{1}_\tau$ as data realisation of $\mathbf{1}_{x_\tau \succ_g 0}$ at step τ. We denote the following sequences of steps: iterations as $[t] := \{1, 2, \cdots, t\}$, f queries as $\mathcal{Q}_t^f := \{\tau \in [t-1] \mid \text{if} f \text{ is queried in step } \tau\}$, and expert queries as $\mathcal{Q}_t^g$, respectively $\left(t \geq |\mathcal{Q}_t^g|, t \geq |\mathcal{Q}_t^f|\right)$. We use capitals, e.g. $X_{\mathcal{Q}_t^f}$, for the set $(x_\tau)_{\tau \in \mathcal{Q}_t^f}$.

3 Confidence Set of the Surrogate Models

We introduce surrogate models for the objective f and the function g. We opted for a Gaussian process (GP; [86, 100]) for f and the likelihood ratio model [67, 27] for g.

3.1 Surrogate Model of the Objective f: Gaussian Process

Definitions. We employ a zero-mean GP regression model, with predictive posterior $\tilde{f}_t \mid D_t^f \sim \mathcal{GP}(\mu_{f_t}, \sigma_{f_t}^2)$,

$$\mu_{f_t}(x) = k_f(X_{\mathcal{Q}_t^f}, x)^\top \left(K_{\mathcal{Q}_t^f} + rI\right)^{-1} Y_{\mathcal{Q}_t^f}, \tag{2a}$$

$$\sigma_{f_t}^2(x) = k_f(x, x) - k_f(X_{\mathcal{Q}_t^f}, x)^\top \left(K_{\mathcal{Q}_t^f} + rI\right)^{-1} k_f\left(X_{\mathcal{Q}_t^f}, x\right), \tag{2b}$$

where $K_{\mathcal{Q}_t^f} = \left(k_f(x_{\tau_1}, x_{\tau_2})\right)_{\tau_1, \tau_2 \in \mathcal{Q}_t^f}$, $D_t^f := (X_{\mathcal{Q}_t^f}, Y_{\mathcal{Q}_t^f})$, r is the regularisation term [61].[2] The maximum information gain [84] for the objective f is,

$$\gamma_{|\mathcal{Q}_t^f|}^f := \max_{X \subset \mathcal{X};\, |X| = |\mathcal{Q}_t^f|} \frac{1}{2} \log\left|I + r^{-1} K_{f,X}\right|, \quad \text{where} \quad K_{f,X} := (k_f(x, x'))_{x, x' \in X}. \tag{3}$$

Lemma 3.1 (Theorem 2, [22])**.** *Let Assumptions 2.3, 2.4 and 2.6 hold. For any $\delta \in (0, 1)$, with probability at least $1 - \delta/2$, the following holds for all $x \in \mathcal{X}$ and $1 \leq t \leq T$, $T \in \mathbb{N}$,*

$$|\mu_{f_t}(x) - f(x)| \leq \beta_{f_t} \sigma_{f_t}(x), \qquad \beta_{f_t} := \left(B_f + \sigma\sqrt{2\left(\gamma_{|\mathcal{Q}_{t-1}^f|}^f + 1 + \ln(2/\delta)\right)}\right),$$

where $\mu_{f_t}(x), \sigma_{f_t}(x)$ and $\gamma_{|\mathcal{Q}_{t-1}^f|}^f$ are as given in Eq. (2) *and Eq.* (3)*, and $\gamma_0^f = 0$.*

[2] We follow the definition from [22].

For brevity, we denote the lower/upper confidence bound (LCB/UCB) functions $\underline{f}_t(x)$ and $\bar{f}_t(x)$ as,

$$\underline{f}_t(x) = \mu_{f_t}(x) - \beta_{f_t}\sigma_{f_t}(x)\ , \qquad \bar{f}_t(x) = \mu_{f_t}(x) + \beta_{f_t}\sigma_{f_t}(x).$$

3.2 Surrogate Model of the Expert Function g: Likelihood Ratio Model

While a GP classifier [63] is a popular choice, we opted for likelihood ratio model [67, 27]. The combination of a Gaussian prior with a Bernoulli likelihood in GP models presents challenges in estimating the posterior confidence bound both theoretically and computationally. Moreover, GPs assume strong rankability [38, 23], presuming humans can rank their preferences accurately in all cases, which often leads to inconsistent results [20]. To address these issues, we drew inspiration from classic expert elicitation methods using imprecise probability theory [10, 41]. Instead of estimating the predictive distribution, we estimate the 'interval' of the worst-case prediction only. This approach does not assume any distribution within the interval, thereby relaxing the rankability assumption [78]. This method is particularly well-suited to the GP-UCB algorithm [83], which only requires a confidence interval. We developed a kernel-based method to provably estimate the predictive interval.

Definitions. First, we introduce the function, $p_{\hat{g}}(x_\tau, \mathbf{1}_\tau) := \mathbf{1}_\tau S\left(\hat{g}(x_\tau)\right) + (1-\mathbf{1}_\tau)\left[1 - S(\hat{g}(x_\tau))\right]$, which is the likelihood of $\hat{g}$ over the event when $\mathbf{1}_{x_\tau \succ_g 0} = \mathbf{1}_\tau$ under the Assumption 2.1, and $\hat{g}$ is an estimate function of $g \in \mathcal{H}_{k_g}$ under the Assumption 2.4. We can then derive the likelihood function of a fixed function $\hat{g}$ over the historical dataset $\mathcal{D}_t^g := \{(x_\tau, \mathbf{1}_\tau)\}_{\tau \in \mathcal{Q}_t^g}$, which becomes the product, $\mathbb{P}_{\hat{g}}((x_\tau, \mathbf{1}_\tau)_{\tau\in\mathcal{Q}_t^g}) := \prod_{\tau\in\mathcal{Q}_t^g} p_{\hat{g}}(x_\tau, \mathbf{1}_\tau)$. The log-likelihood (LL) function,

$$\text{LL value: } \ell_t(\hat{g}) := \log \mathbb{P}_{\hat{g}}((x_\tau, \mathbf{1}_\tau)_{\tau\in\mathcal{Q}_t^g}), \tag{4}$$

reduces to $\ell_t(\hat{g}) = \sum_{\tau\in\mathcal{Q}_t^g} z_\tau \mathbf{1}_\tau - \sum_{\tau\in\mathcal{Q}_t^g} \log\left(1 + e^{z_\tau}\right)$, where $z_\tau = \hat{g}(x_\tau)$ (this equality can be checked as correct for either $\mathbf{1}_\tau = 1$ or $\mathbf{1}_\tau = 0$). We then introduce the maximum likelihood estimator (MLE), $\hat{g}_t^{\text{MLE}} \in \arg\max_{\tilde{g}\in\mathcal{B}_g} \log \mathbb{P}_{\tilde{g}}((x_\tau, \mathbf{1}_\tau)_{\tau\in\mathcal{Q}_t^g})$. Similar to [54, 27, 107], the *confidence set* can be derived as shown in Lemma 3.2.

Lemma 3.2 (Likelihood-based confidence set). *$\forall \epsilon, \delta > 0$, let,*

$$\mathcal{B}_g^{t+1} := \left\{\tilde{g} \in \mathcal{B}_g \mid \ell_t(\tilde{g}) \geq \ell_t(\hat{g}_t^{\text{MLE}}) - \alpha_1(\epsilon, \delta, |\mathcal{Q}_t^g|, t)\right\},$$

where $\alpha_1(\epsilon, \delta, |\mathcal{Q}_t^g|, t) := \sqrt{32|\mathcal{Q}_t^g| B_g^2 \log \frac{\pi^2 t^2 \mathcal{N}(\mathcal{B}_g, \epsilon, \|\cdot\|_\infty)}{6\delta}} + 2\epsilon t$. *We have,*

$$\mathbb{P}\left(g \in \mathcal{B}_g^{t+1}, \forall t \geq 1\right) \geq 1 - \delta.$$

The proof is in Appendix A. As introduced in Assumption 2.4, while the function g was originally in a broader set of RKHS functions $g \in \mathcal{B}_g$, it is now in a smaller set defined as $g \in \mathcal{B}_g^{t+1}$ conditioned on the expert labels $\mathcal{D}_t^g$. Intuitively, with limited data, the MLE may be imperfect. Hence, it is reasonable to suppose that $\mathcal{B}_g^{t+1}$, bounded by LL values 'slightly worse' than the MLE, contains the ground truth with high probability.

Remark 3.3 (**Choice of** ϵ). In Lemma 3.2, $\alpha_1(\epsilon, \delta, |\mathcal{Q}_t^g|, t)$ depends on a small positive value ϵ. It will be seen that ϵ can be selected to be $1/T$ in Appendix B, where T is the running horizon of the algorithm.

Remark 3.4 (**Confidence bound**). By Lemma 3.2, we define the pointwise confidence bound for unknown $g \in \mathcal{H}_{k_g}$, $\underline{g}_t(x) \leq g(x) \leq \bar{g}_t(x)$, where $\underline{g}_t(x) := \inf_{\tilde{g}\in\mathcal{B}_g^t} \tilde{g}(x)$ and $\bar{g}_t(x) := \sup_{\tilde{g}\in\mathcal{B}_g^t} \tilde{g}(x)$.

Remark 3.5 (**Pointwise predictive interval estimation**). At a given prediction point x, the predictive interval $\left[\underline{g}_t(x),\ \bar{g}_t(x)\right]$ can be estimated through two individual finite-dimensional optimisation problems (See Appendix B.3 for details). Subsequently, applying the sigmoid function yields the predictive interval in probability space $\left[S(\underline{g}_t(x)),\ S(\bar{g}_t(x))\right]$ (see Fig. 2 for visualisation).

4 Algorithm and Theoretical Guarantees

4.1 Mixing Two Surrogate Models f and g via Primal-Dual Method

Primal dual. We introduce the following primal-dual problem (5) as our acquisition policy,

$$\textbf{Primal}: x_t^c \in \arg\min_{x\in\mathcal{X}} \underline{f}_t(x) + \lambda_t \underline{g}_t(x),\ \textbf{Dual}: \lambda_{t+1} = [\lambda_t + \zeta \underline{g}_t(x_t^c)]^+, \tag{5}$$

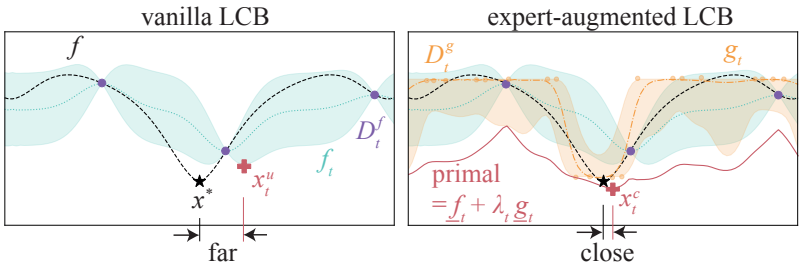


Figure 2: Visual explanation: While the vanilla LCB returns x_t^u, a far point from global minimum x^*, expert-augmented LCB can successfully navigate to closer point x_t^c by mixing f_t and g_t with $\underline{f}_t + \lambda_t \underline{g}_t$, where λ_t is the dual variable. In the figure, D_t^f is the set of the sample points of the objective function f and D_t^g is the set of human feedback.

where λ_t is the primal-dual weight at the t-th iteration and ζ is the step size for dual update. See Fig. 2 for the intuition: we prioritise the sample in the expert-preferred region (i.e., the region with small $\underline{g}_t(x)$). The primal-dual method is a classical algorithm for constrained optimisation [64] and has recently been applied to, for example, the constrained bandit problem [110]. In terms of constrained optimisation, Prob. (5) can be understood as solving $\min_{x \in \mathcal{X}} \underline{f}(x)$ s.t. $\underline{g}_t(x) \leq 0$. Interestingly, the primal-dual approach is also roughly analogous to Bayesian inference [25]. Just as the prior acts as a regulariser to the LL maximiser [94], expert belief $\underline{g}_t(x)$ regularises the $\underline{f}_t(x)$ minimiser. More specifically, the weight λ_{t+1} increases when $\underline{g}_t(x_t^c) > 0$; otherwise, λ_{t+1} decreases. The condition $\underline{g}_t(x_t^c) > 0$ indicates that the primal solution x_t^c is more likely to be rejected.[3] Under such a risk of rejection, increasing the weights λ_{t+1} is natural because it more strongly regularises the $\underline{f}_t$ minimiser to enhance feasibility in the next round, and vice versa.

Level of trust. Note that the primal-dual method is not the primary reason we achieve the no-harm guarantee. Indeed, its proof (detailed in Appendix B) does not rely on the primal-dual formulation. Therefore, technically speaking, our algorithm could employ a more aggressive exploitation of g_t (e.g., simply minimising g_t). Nevertheless, the primal-dual approach is our recommended policy for generating the expert-augmented candidate x_t^c to enhance resilience to erroneous inputs. The initial level of trust on g_t is determined by the initial weight λ_0, where larger λ_0 values correspond to greater trust in the expert. We compared the effect of λ_0 in the Fig. 3 of the experimental section.

Efficient computation. Leveraging the representer theorem [77, 107] due to the RKHS property, we further reformulate Prob. (5) to a $(|\mathcal{Q}_t^g| + d + 1)$-dimensional, tractable optimisation problem (6).

$$\begin{aligned} \min_{Z_{\mathcal{Q}_t^g} \in \mathbb{R}^{|\mathcal{Q}_t^g|},\ z \in \mathbb{R},\ x \in \mathcal{X}} \quad & \underline{f}_t(x) + \lambda_t z \\ \text{subject to} \quad & \begin{bmatrix} Z_{\mathcal{Q}_t^g} \\ z \end{bmatrix}^\top K_{\mathcal{Q}_t^g, x}^{-1} \begin{bmatrix} Z_{\mathcal{Q}_t^g} \\ z \end{bmatrix} \leq B_g^2, \\ & \ell(Z_{\mathcal{Q}_t^g} \mid \mathcal{D}_t^g) \geq \ell_t(\hat{g}_t^{\mathrm{MLE}}) - \alpha_1(\epsilon, \delta, |\mathcal{Q}_t^g|, t), \end{aligned} \tag{6}$$

where $K_{\mathcal{Q}_t^g, x} := (k_g(\tilde{x}, \tilde{x}'))_{\tilde{x}, \tilde{x}' \in X_{\mathcal{Q}_t^g} \cup \{x\}}$, and $\ell(Z_{\mathcal{Q}_t^g} \mid \mathcal{D}_t^g) = \sum_{\tau \in \mathcal{Q}_t^g} Z_\tau \mathbf{1}_\tau - \sum_{\tau \in \mathcal{Q}_t^g} \log\left(1 + e^{Z_\tau}\right)$ is the LL value when $\hat{g}(x_\tau) = Z_\tau, \forall \tau \in \mathcal{Q}_t^g$. We update $\lambda_{t+1} = \lambda_t + \zeta z^\star$, where $z^\star = \underline{g}_t(x_t^c)$ is the optimal z of Prob. (6).

Key hyperparameter estimation. A key hyperparameter in Prob. (6) is the norm bound B_g in the first constraint. Another hyperparameter, α_1, in the second constraint, also scales with B_g, (see Lemma 3.2). However, B_g may be unknown in practice, and its mis-specification leads to mis-calibrated uncertainty. We estimate B_g by starting with a small initial guess (e.g., 1) and doubling it when the following condition is met based on newly observed expert labels: $\alpha_1(\epsilon, \delta, |\mathcal{Q}_t^g|, t \mid 2\hat{B}_g) < \ell_t(\hat{g}_{t|2\hat{B}_g}^{\mathrm{MLE}}) - \ell_t(\hat{g}_{t|\hat{B}_g}^{\mathrm{MLE}})$, where $\hat{B}_g$ is our current guess. Intuitively, if the new likelihood $\ell_t(\hat{g}_{t|2\hat{B}_g}^{\mathrm{MLE}})$ is

[3] Recall that $S(g(x_t^c)) > S(0) = 0.5$ implies a higher chance of rejection than random (=0.5).

Algorithm 1 COllaborative **B**ayesian **O**ptimization with **L**abelling Experts (**COBOL**).

1: **Input and Initialization**: function space ball $\mathcal{B}_g$, trust weight η, and uncertainty threshold g_{thr}.
2: Set $\mathcal{B}_g^1 = \mathcal{B}_g$, $\mathcal{Q}_0^f = \emptyset$, and $\mathcal{Q}_0^g = \emptyset$.
3: **for** $t \in [T]$ **do**
4: Solve Prob. (5) via Prob. (6) to generate x_t^c. ▷ Expert-augmented LCB
5: Solve the unconstrained problem, $x_t^u \in \arg\min_{x \in \mathcal{X}} \underline{f}_t(x)$. ▷ Vanilla LCB
6: **if** $\underline{f}_t(x_t^c) \leq \min_{x \in \mathcal{X}} \bar{f}_t(x)$ **and** $\sigma_{f_t}(x_t^u) \leq \eta \sigma_{f_t}(x_t^c)$ **then** ▷ No-harm guarantee
7: Set $x_t = x_t^c$.
8: **if** $\bar{g}_t(x_t) - \underline{g}_t(x_t) > g_{\text{thr}}$ **then** ▷ Handover guarantee
9: Query the expert's label to get the feedback $\mathbf{1}_t$.
10: Update $\mathcal{Q}_t^g = \mathcal{Q}_{t-1}^g \cup \{t\}$ and the posterior confidence set $\mathcal{B}_g^{t+1}$.
11: **if** $\mathbf{1}_t = 1$ **then**
12: Set $\mathcal{Q}_t^f = \mathcal{Q}_{t-1}^f$, and continue the loop at line 4.
13: **else**
14: Set $\mathcal{Q}_t^g = \mathcal{Q}_{t-1}^g$, and $x_t = x_t^u$.
15: Evaluate the black-box function at the point x_t, and set $\mathcal{Q}_t^f = \mathcal{Q}_{t-1}^f \cup \{t\}$.
16: Update the posterior mean/variance of the objective f.

significantly larger, then $2\hat{B}_g$ is more likely a valid bound. We iterate this estimation online during optimisation and in pre-training with the initial dataset (see details in Appendix F).

4.2 Algorithm and Theoretical Guarantee

Algorithm. Our algorithm in Alg. 1 generates two candidates: the vanilla LCB x_t^u and the expert-augmented LCB x_t^c. (See App. I.3 on extension to other acquisition functions.) Always selecting the vanilla LCB guarantees no-harm but misses the chance to accelerate convergence using the expert's belief. Intuitively, this can be seen as a bandit problem regarding which arm to select. Line 8 corresponds to the *handover guarantee*, stating that our algorithm stops asking the expert once our model g becomes more confident than the predefined g_{thr}. Line 6 outlines the conditions for achieving the *no-harm guarantee* by assessing the reliability of the expert-augmented candidate x_t^c. The first condition ensures x_t^c is at least possibly better than the worst-case estimation of the optimal value. The second condition acts as active learning of human belief, exploring uncertain points to avoid inaccurate yet confident expert beliefs. The hyperparameter $\eta \geq 1$ represents the initial level of trust in the expert. A larger η indicates greater priority in exploring expert-preferred regions.

Theoretical guarantee. For Alg. 1, we mainly care about two metrics: cumulative regret $R_{\mathcal{Q}_T^f} := \sum_{t \in \mathcal{Q}_T^f} (f(x_t) - f(x^*))$ and cumulative queries $Q_T^g := |\mathcal{Q}_T^g|$. $R_{\mathcal{Q}_T^f}$ captures the cumulative regret over the query points to the black-box function. Q_T^g captures the number of queries to the expert. Since intuitively each query to the expert causes inconvenience, ideally, the frequency of query to an expert should be low (e.g., Q_T^g grows sublinearly in T).

Theorem 4.1. *Under Assumptions 2.1 to 2.6, with probability at least* $1 - \delta$*, Alg 1 satisfies,*

$$R_{\mathcal{Q}_T^f} \leq \mathcal{O}\left((2+\eta)\gamma^f_{|\mathcal{Q}_T^f|}\sqrt{|\mathcal{Q}_T^f|}\right), \quad \text{(7a)} \qquad Q_T^g \leq \mathcal{O}\left((\gamma_T^g)^2 \log \frac{T\mathcal{N}(\mathcal{B}_g, 1/T, \|\cdot\|_\infty)}{\delta}\right). \quad \text{(7b)}$$

See Appendix B for the proof of Thm. 4.1. Intuitively, Eq. (7a) shows the **no-harm guarantee**, since it provides a cumulative regret bound independent of the latent function g. Eq. (7b) shows the **handover guarantee**, since the bound on cumulative queries to the expert is sublinear for commonly-used kernel functions (See Table 1). This means that the frequency of querying the expert asymptotically converges to zero. We do not query human label for x_t^u to reduce human effort. Since $\mathcal{Q}_T^g \cup \mathcal{Q}_T^f = [T]$, $|\mathcal{Q}_T^f|$ grows linearly in T. There is a trade-off in η selection. A larger η can accelerate convergence when feedback is informative, but it may also cause the worse convergence rate for adversarial feedback (see Appendix B, which includes an additional constant factor of $(2+\eta)/4$ compared to the original UCB). In practice, setting $\eta = 3$ is sufficiently effective (see Figure 3).

Table 1: Kernel-specific bounds (fixed η is hidden) where ν is the smoothness parameter of the Matérn kernel that is assumed to satisfy $\nu > \frac{d}{4}(3 + d + \sqrt{d^2 + 14d + 17}) = \Theta(d^2)$.

Metric	Linear	Squared Exponential	Matérn
$R_{\mathcal{Q}_T^f}$	$\mathcal{O}\left(\sqrt{\lvert\mathcal{Q}_T^f\rvert}\log\lvert\mathcal{Q}_T^f\rvert\right)$	$\mathcal{O}\left(\sqrt{\lvert\mathcal{Q}_T^f\rvert}(\log\lvert\mathcal{Q}_T^f\rvert)^{d+1}\right)$	$\mathcal{O}\left(\lvert\mathcal{Q}_T^f\rvert^{\frac{2\nu+3d}{4\nu+2d}}\log^{\frac{2\nu}{2\nu+d}}(\lvert\mathcal{Q}_T^f\rvert)\right)$
Q_T^g	$\mathcal{O}\left((\log T)^3\right)$	$\mathcal{O}\left((\log T)^{3(d+1)}\right)$	$\mathcal{O}(T^{\frac{2d(d+1)}{2\nu+d(d+1)}}T^{\frac{d}{\nu}}(\log T)^3)$

By plugging in the maximum information gain bounds [84, 93] and covering number bounds [104, 105, 18, 109], we apply Thm. 4.1 to derive the kernel-specific bounds in Table 1. In practice, kernel choice and scalability to high dimensions are common challenges for BO. Existing generic techniques, such as decomposed kernels [49], can be applied in our algorithm to choose kernel functions and achieve scalability in high-dimensional spaces.

4.3 Related Works

Human-AI Collaborative BO. There are two primary approaches: the first approach assumes that human experts can express their beliefs through *quantitative* labels, such as well-defined distributions [69, 52, 82, 43, 24, 42] or pinpoint querying locations [11, 39, 50, 12, 70]. While this strong assumption is valid in specific cases, such as physics simulations [39], many experimental tasks—such as chemistry, which lacks the consensus on numerical representations of, e.g. molecules—require more relaxed assumptions [24, 46]. The *qualitative* approach, on the other hand, involves human experts providing pairwise comparisons [7] or binary recommendations (ours). The algorithm trains a surrogate model from experts' labels, thereby expanding applicable scenarios. Ours is the ***first-of-its-kind*** principled method with both no-harm and handover guarantee on a continuous domain.

Related BO tasks. Eliciting human preference from labels has been explored in preferential BO [28, 37, 59, 91, 9, 107]. However, this approach treats human preference as the main objective of BO, whereas our work uses experts' belief as an additional information source. Constrained BO [32, 35, 88, 87, 110, 106, 62, 44, 96, 57] is another line of research that investigates BO under unknown constraints, placing another surrogate model on the constraint inferred from queried labels. However, our approach does not treat human belief as a constraint that must be satisfied or a reward to maximise, given that expert knowledge can sometimes be unreliable (see details in Appendix G).

5 Experiments

We benchmarked the performance of the proposed algorithm against existing baselines in a collaborative setting with human experts. We employed an ARD RBF kernel for both f and g. In each iteration of the optimisation loop, the inputs were rescaled to the unit cube $[0, 1]^d$, and the outputs were standardised to have zero mean and unit variance. The initial datasets consisted of three random data points sampled uniformly from within the domain, and in each iteration, one data point was queried. Additionally, we collected initial expert labels by asking an expert to label 'accept' (= 0) or 'reject' (= 1) for 10 uniformly random points. All experiments were repeated ten times with different initial datasets and random seeds. We tuned hyperparameters online at each iteration. The GP hyperparameters were tuned by maximising the marginal likelihood on observed datasets using a multi-start L-BFGS-B method [53] (the default BoTorch optimiser [14]). The key hyperparameters of the confidence set, B_g and α_1, were optimised via the online method in Appendix F. Other hyperparameters were set as $\eta = 3$, $\lambda_0 = 1$, and $g_{\text{thr}} = 0.1$ by default throughout the experiments, with their sensitivity discussed later in Fig. 3 (see also Appendix J.1). The constrained optimisation in Prob. (6) was solved using the interior-point nonlinear optimiser IPOPT [95], which is highly scalable for solving the primal problem, via the symbolic interface CasADi [8]. The unconstrained optimisation (line 5) was solved using the default BoTorch optimiser [14]. More details for reproducing results are available on GitHub.[4] The models were implemented in GPyTorch [31]. All experiments were conducted on a laptop PC.[5] Computational time is discussed in Appendix J. In addition to cumulative regret and queries, we also consider simple regret defined as $\text{SR}_t := \min_{\tau \in \mathcal{Q}_t^f}(f(x_\tau) - f(x^\star))$.

[4] `https://github.com/ma921/COBOL/`

[5] MacBook Pro 2019, 2.4 GHz 8-Core Intel Core i9, 64 GB 2667 MHz DDR4

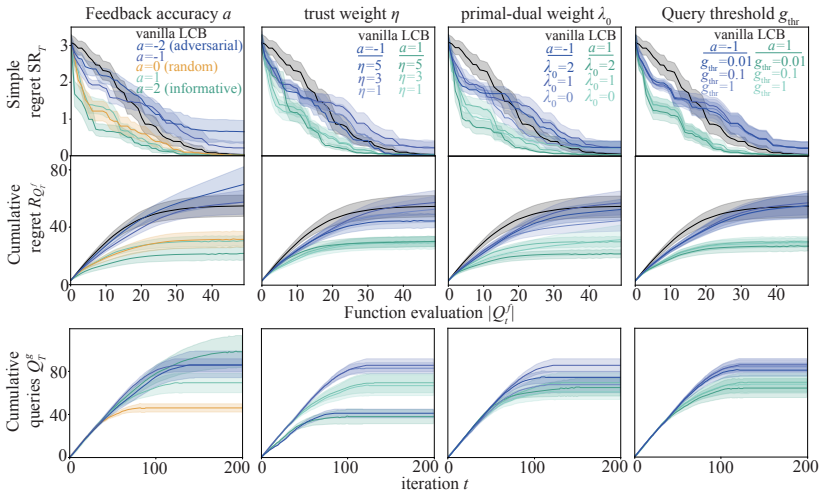


Figure 3: Robustness and sensitivity analysis using the Ackley function. Lines and shaded areas denote mean $\pm$ 1 standard error. The no-harm guarantee ensures the convergence rate is on par with vanilla LCB even in adversarial cases. Handover guarantee ensures that $\mathcal{Q}_t^g$ plateau, allowing optimisation without expert intervention once sufficient information has been elicited.

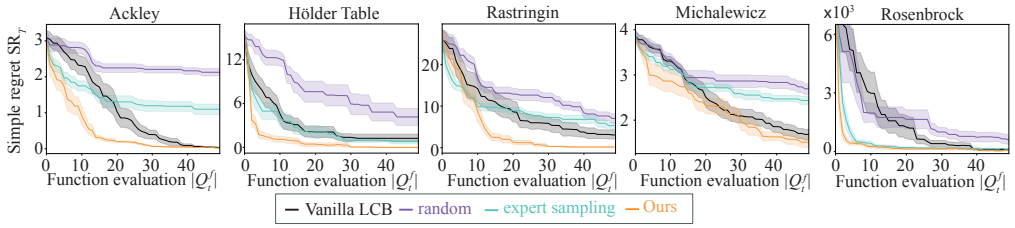


Figure 4: Ablation study on five common synthetic functions with synthetic expert labels ($a = 1$).

Robustness and sensitivity. First, we tested the robustness of our algorithm to the accuracy of the expert's labels using the 4-dimensional Ackley function [1]. We modelled the synthetic agent response according to Example 2.2. In particular, we examine the impact of feedback accuracy, denoted as a. Fig. 3 illustrates the robustness of our algorithm. When labels are informative ($a = 1, 2$), the convergence rate for both simple and cumulative regrets is accelerated in accordance with the accuracy. Even if the feedback is completely random ($a = 0$) or adversarial ($a = -1, -2$), the no-harm guarantee ensures that the algorithm converges at a rate on par with vanilla LCB by adjusting the level of trust to be lower over iterations. Refer to Appendix J.2.3 for additional confirmation of the no-harm guarantee based on more extensive experimental results. Handover guarantee ensures that our algorithm stops seeking label feedback once sufficient information has been elicited, as indicated by the plateau in the cumulative queries Q_T^g. We also tested the sensitivity to the optimisation parameters η, λ_0, and g_{thr}. The change in convergence at those parameters were varied mostly within the standard error, indicating that our algorithm is insensitive to these hyperparameters and that feedback accuracy is more dominant. For the primal-dual weight, $\lambda_0 = 0$ corresponds to starting optimisation without a primal-dual mixing objective, which performs worse than mixing cases ($\lambda_0 = 1, 2$), demonstrating the efficacy of incorporating the primal-dual mixing objective.

Synthetic dataset. We compared our algorithm against five common synthetic functions [89] (see details in Appendix J.2), using simple baselines for an ablation study: random sampling, vanilla LCB (unconstrained optimisation), and expert sampling. Expert sampling involves direct sampling from the expert belief distribution $p_{x \succ_g 0}$. We employ rejection sampling by generating a uniform random sample over the domain and then accepting it with the probability $1 - p_{x \succ_g 0}$. We fixed the feedback accuracy at $a = 1$ (as in Example 2.2.). The efficacy of expert labels is roughly estimated by how much faster expert sampling converges compared to random sampling. In all synthetic experiments, our algorithm outperformed the baselines. While expert sampling is at least more effective than

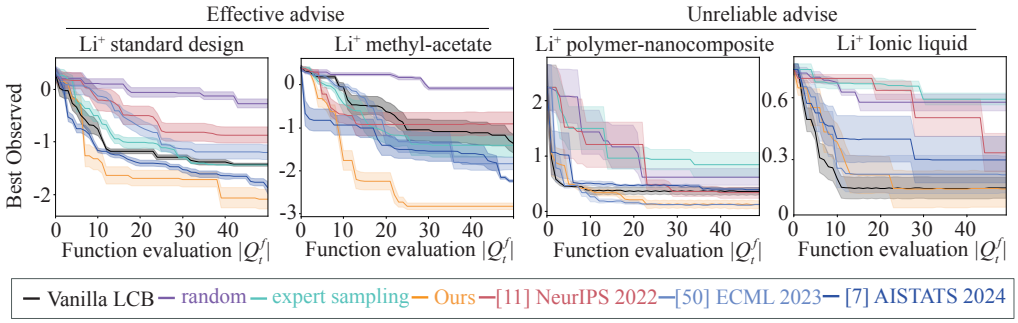


Figure 5: Real-world experiments with four human experts of lithium-ion batteries.

random sampling, it is not always better than vanilla LCB. For functions with a very sharp global optimum, such as Rosenbrock [71], $p_{x \succ_g 0}$ nearly pinpoints the global minimum. Still, our algorithm performs slightly better than expert sampling. See Appendix J.2.2 for computation time and query frequency. The overhead of our algorithm is comparable to that of other baselines.

Real-world experiments with human experts. We conducted real-world experiments in collaboration with four human experts who possess post-doctoral level knowledge on lithium-ion batteries. In this experiment, human labelling costs vary among experts but typically range from a few seconds to several minutes. In the real-world development of lithium-ion batteries, creating and testing a prototype cell requires at least a week, making the labelling cost negligible by comparison.

Lithium-ion batteries are crucial for realising a green society, a rapidly growing field where knowledge is continuously updated at an unprecedented rate. This field typically suffers from data scarcity [46] due to the ongoing development of new materials synthesised by chemists. Consequently, transfer learning approaches, e.g., [90, 101, 30, 21], are not effective in this setting. We prepared four cases for the experiments: the first is a standard task where we optimise the standard electrolyte composition [26, 36], and the second involves a slight modification of the first setup by changing one solvent material [56].[6] We expect the experts to have informative knowledge on these two tasks. The remaining two cases involve emerging new categories of materials: one is a polymer-nanocomposite electrolyte [108], and the other is an ionic liquid [72]. We anticipate that the experts' knowledge on these new materials will not be as effective as in the first two tasks (see more details in Appendix J.3). Given the scarcity of real experts, we conducted a pre-experimental step to elicit their knowledge for a fair baseline comparison. We asked them to label 50 random points uniformly from the domain, for all experiments before seeing the results. Then we fit the confidence set model to these results and used $\hat{g}_t^{\mathrm{MLE}}$ as the *estimated* human response. Additionally, we asked the participants to manually select the next query point without any assistance from BO, which we refer to as 'expert sampling' in the baseline. We also compared against state-of-the-art algorithms [11, 50, 7]. These methods have predefined levels of trust, roughly ranked from strong to weak: [11] $\rightarrow$ [7] $\rightarrow$ [50]. Ours can adjust the level of trust based on data, so we expect it to perform well in both effective and ineffective cases.

Fig. 5 summarises the results. For the first two tasks, our algorithm outperformed all baselines. Particularly in the second task, human sampling was better than vanilla LCB, indicating that we should trust their advice aggressively. Our algorithm can adapt to trust them over time, resulting in significantly accelerated convergence. On the other hand, expert sampling for the new materials tasks was, although unintentionally, worse than random, thereby discouraging trust. While trustful algorithms [11, 7] struggled to converge, the distrustful algorithm [50] was able to converge on par with vanilla LCB. Our no-harm guarantee worked in this situation, gradually equating to LCB, and showed identical performance to the distrustful algorithm [50]. See also Appendix J.4.1 for the complete experimental results on the number of queries and computation time.

6 Discussion

Feedback form. Other forms of feedback, such as pairwise comparisons [7] or preferential rankings [12], can be incorporated into our algorithm with slight modifications. However, we empirically

[6] This slight change makes optimal design challenging enough [36]. See Appendix J.4 for details.

found that the binary labelling approach performs best (see Fig. 5), and therefore, we recommend using binary feedback as the primary choice. For those interested in using alternative feedback forms, detailed instructions on how to adapt them to our algorithm are provided in Appendix H.

Time-varying human knowledge. We assume that expert knowledge is stationary, although it can be time-varying, e.g., experts' knowledge often evolves as more data is gathered. A simple extension to accommodate this is the use of windowing, where past queried data is forgotten. This can be easily implemented in our algorithm by removing old data beyond a predefined iteration window. However, our initial trials did not show significant performance gains from this approach, so it was not included in the main text. We suggest a dynamic model as a potential future direction, which is discussed in Appendix I.1 with additional experimental results. Similarly, we kept the trust weight η fixed throughout the optimization process. Since human knowledge can improve over time, an adaptive η could be employed to enhance both convergence and robustness. Nevertheless, our no-harm guarantee remains valid even without this adaptation. Further details are provided in Appendix I.2.

Acceleration vs. Robustness. One might seek to derive a theoretical guarantee on the acceleration of convergence when the feedback is helpful. However, we want to emphasize that theoretically guaranteeing both acceleration and robustness may be incompatible. From a theoretical perspective, they are in a trade-off relationship [92]. This can be intuitively explained by the no-free-lunch theorem [102]: if algorithm A outperforms B, it does so by exploiting 'biased' information. The 'bias' inherent in the acceleration is contradictory to robustness. Our setting is unbiased, meaning we do not have prior knowledge of helpful or adversarial human expert. Therefore, we must make a design choice between prioritizing robustness or acceleration as a theoretical contribution, depending on whether we assume that expert input can be adversarial (weak bias) or that it will always be helpful (strong bias). Indeed, there are lower bound results for the average-case regret of Bayesian optimization in the literature (e.g., see [76]). GP-UCB is already nearly rate-optimal in achieving this lower bound. This means theoretical acceleration is obtained in the price of worse robustness. In Appendix E, we present a slightly modified version, Algorithm 2, which offers an improvement guarantee based on strong bias. Our Algorithm 1 can be seen as a relaxed version of this algorithm (soft constraint), which helps explain the empirical success in accelerating convergence.

7 Conclusion

Our algorithm, with its data-driven adjustment of the level of trust, successfully accelerated convergence from effective advice while ensuring a no-harm guarantee from unreliable inputs. The handover guarantee also ensures that the BO can automate the optimisation process without assistance from human experts at a later stage. These features are particularly valuable for scientific applications, where researchers often face trial and error, making it challenging to determine the effectiveness of their prior knowledge before starting experiments. Our flexible and robust framework is also expected to be effective in collaboration with large language models (LLMs), which demonstrate remarkable sample-efficient performance by exploiting encoded priors [55, 75, 66], and can be regarded as 'expert knowledge'. Our safeguard features would be particularly effective for shared challenges, such as difficulty in eliciting knowledge [45, 16] and varying accuracy of advice due to hallucinations [81, 98, 85]. Although ours is the *first-of-its-kind* algorithm with a general theoretical guarantee in the expert-collaborative setting, it is still based on the GP-UCB algorithm [7] and shares its limitations (e.g., high dimensionality). One future direction is combining our approach with the high-dimensional BO methods [97, 51]. Additionally, our current setting does not consider the batch setting, yet one can easily extend with existing approaches, e.g. [4, 6, 3, 5]. Multiple expert scenario is also a promising future extension. While a simple expert aggregation approach (e.g., majority vote, adding multiple experts g) could work without modifications to the current algorithm, more advanced methods, such as choice functions [15], present promising directions for future work. Explainability is also key. [7] showed that Shapley value-based explanations improve human feedback accuracy, and this can be easily integrated into our framework. Our method can positively influence human experts by empirically demonstrating the value of their expertise, even amidst concerns about job security in the AI era [13]. On the negative side, more powerful LLMs may eventually replace the expert role in our algorithm in areas where data is sufficiently shared on websites or in papers, such as hyperparameter tuning [55].

[7] Maximization formulation is adopted in GP-UCB paper [84], while we consider minimization. So LCB in our paper essentially corresponds to UCB in GP-UCB algorithm.

Acknowledgments and Disclosure of Funding

We would like to thank Ondrej Bajgar, Juliusz Ziomek, and anonymous reviewers for their helpful comments about improving the paper. Wenjie Xu and Colin N. Jones were supported by the Swiss National Science Foundation under NCCR Automation, grant agreement 51NF40_180545. Masaki Adachi was supported by the Clarendon Fund, the Oxford Kobe Scholarship, the Watanabe Foundation, and Toyota Motor Corporation.

References

[1] David Ackley. *A connectionist machine for genetic hillclimbing*, volume 28. Springer science & business media, 1987.

[2] Masaki Adachi. High-dimensional discrete Bayesian optimization with self-supervised representation learning for data-efficient materials exploration. In *NeurIPS 2021 AI for Science Workshop*, 2021.

[3] Masaki Adachi, Satoshi Hayakawa, Martin Jørgensen, Saad Hamid, Harald Oberhauser, and Michael A Osborne. A quadrature approach for general-purpose batch Bayesian optimization via probabilistic lifting. *arXiv preprint arXiv:2404.12219*, 2024.

[4] Masaki Adachi, Satoshi Hayakawa, Martin Jørgensen, Harald Oberhauser, and Michael A Osborne. Fast Bayesian inference with batch Bayesian quadrature via kernel recombination. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:16533–16547, 2022.

[5] Masaki Adachi, Satoshi Hayakawa, Martin Jørgensen, Xingchen Wan, Vu Nguyen, Harald Oberhauser, and Michael A Osborne. Adaptive batch sizes for active learning: A probabilistic numerics approach. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 496–504. PMLR, 2024.

[6] Masaki Adachi, Yannick Kuhn, Birger Horstmann, Arnulf Latz, Michael A Osborne, and David A Howey. Bayesian model selection of lithium-ion battery models via Bayesian quadrature. *IFAC-PapersOnLine*, 56(2):10521–10526, 2023.

[7] Masaki Adachi, Brady Planden, David A Howey, Michael A Osborne, Sebastian Orbell, Natalia Ares, Krikamol Muandet, and Siu Lun Chau. Looping in the human: Collaborative and explainable Bayesian optimization. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2024.

[8] Joel A E Andersson, Joris Gillis, Greg Horn, James B Rawlings, and Moritz Diehl. CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36, 2019.

[9] Raul Astudillo, Zhiyuan Jerry Lin, Eytan Bakshy, and Peter Frazier. qEUBO: A decision-theoretic acquisition function for preferential Bayesian optimization. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1093–1114. PMLR, 2023.

[10] Thomas Augustin, Frank PA Coolen, Gert De Cooman, and Matthias CM Troffaes. *Introduction to imprecise probabilities*, volume 591. John Wiley & Sons, 2014.

[11] Arun Kumar AV, Santu Rana, Alistair Shilton, and Svetha Venkatesh. Human-AI collaborative Bayesian Optimisation. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:16233–16245, 2022.

[12] Arun Kumar AV, Alistair Shilton, Sunil Gupta, Santu Rana, Stewart Greenhill, and Svetha Venkatesh. Enhanced Bayesian optimization via preferential modeling of abstract properties. *arXiv preprint arXiv:2402.17343*, 2024.

[13] Hasan Bakhshi, Jonathan Downing, Michael Osborne, and Philippe Schneider. *The future of skills: Employment in 2030*. Pearson, 2017.

[14] Maximilian Balandat, Brian Karrer, Daniel Jiang, Samuel Daulton, Ben Letham, Andrew G Wilson, and Eytan Bakshy. BoTorch: a framework for efficient Monte-Carlo Bayesian optimization. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:21524–21538, 2020.

[15] Alessio Benavoli, Dario Azzimonti, and Dario Piga. Learning choice functions with Gaussian processes. In *Uncertainty in Artificial Intelligence (UAI)*, pages 141–151. PMLR, 2023.

[16] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[17] Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

[18] Adam D Bull. Convergence rates of efficient global optimization algorithms. *Journal of Machine Learning Research (JMLR)*, 12(10), 2011.

[19] Jerry F Casteel and Edward S Amis. Specific conductance of concentrated solutions of magnesium salts in water-ethanol system. *Journal of Chemical and Engineering Data*, 17(1):55–59, 1972.

[20] Siu Lun Chau, Javier Gonzalez, and Dino Sejdinovic. Learning inconsistent preferences with Gaussian processes. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 2266–2281. PMLR, 2022.

[21] Yutian Chen, Xingyou Song, Chansoo Lee, Zi Wang, Richard Zhang, David Dohan, Kazuya Kawakami, Greg Kochanski, Arnaud Doucet, Marc'aurelio Ranzato, et al. Towards learning universal hyperparameter optimizers with transformers. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:32053–32068, 2022.

[22] Sayak Ray Chowdhury and Aditya Gopalan. On kernelized multi-armed bandits. In *International Conference on Machine Learning (ICML)*, pages 844–853. PMLR, 2017.

[23] Wei Chu and Zoubin Ghahramani. Preference learning with Gaussian processes. In *Proceedings of the 22nd international conference on Machine learning*, pages 137–144, 2005.

[24] Abdoulatif Cisse, Xenophon Evangelopoulos, Sam Carruthers, Vladimir V Gusev, and Andrew I Cooper. HypBO: Expert-guided chemist-in-the-loop Bayesian search for new materials. *arXiv preprint arXiv:2308.11787*, 2023.

[25] Bo Dai, Hanjun Dai, Niao He, Weiyang Liu, Zhen Liu, Jianshu Chen, Lin Xiao, and Le Song. Coupled variational Bayes via optimization embedding. *Advances in Neural Information Processing Systems (NeurIPS)*, 31, 2018.

[26] Adarsh Dave, Jared Mitchell, Sven Burke, Hongyi Lin, Jay Whitacre, and Venkatasubramanian Viswanathan. Autonomous optimization of non-aqueous Li-ion battery electrolytes via robotic experimentation and machine learning coupling. *Nature communications*, 13(1):5454, 2022.

[27] Nicolas Emmenegger, Mojmir Mutny, and Andreas Krause. Likelihood ratio confidence sets for sequential decision making. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[28] Brochu Eric, Nando Freitas, and Abhijeet Ghosh. Active preference learning with discrete choice data. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 20, 2007.

[29] Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. Efficient and robust automated machine learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 28, 2015.

[30] Matthias Feurer, Benjamin Letham, and Eytan Bakshy. Scalable meta-learning for Bayesian optimization using ranking-weighted Gaussian process ensembles. In *AutoML Workshop at ICML*, volume 7, page 5, 2018.

[31] Jacob Gardner, Geoff Pleiss, Kilian Q Weinberger, David Bindel, and Andrew G Wilson. GPyTorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 7576–7586, 2018.

[32] Jacob R Gardner, Matt J Kusner, Zhixiang Eddie Xu, Kilian Q Weinberger, and John P Cunningham. Bayesian optimization with inequality constraints. In *International Conference on Machine Learning (ICML)*, volume 2014, pages 937–945, 2014.

[33] Roman Garnett. *Bayesian optimization*. Cambridge University Press, 2023.

[34] Paul H Garthwaite, Joseph B Kadane, and Anthony O'Hagan. Statistical methods for eliciting probability distributions. *Journal of the American Statistical Association*, 100(470):680–701, 2005.

[35] Michael A. Gelbart, Jasper Snoek, and Ryan P. Adams. Bayesian optimization with unknown constraints. In *Uncertainty in Artificial Intelligence (UAI)*, page 250–259, 2014.

[36] Kevin L Gering. Prediction of electrolyte conductivity: results from a generalized molecular model based on ion solvation and a chemical physics framework. *Electrochimica Acta*, 225:175–189, 2017.

[37] Javier González, Zhenwen Dai, Andreas Damianou, and Neil D Lawrence. Preferential Bayesian optimization. In *International Conference on Machine Learning (ICML)*, pages 1282–1291. PMLR, 2017.

[38] Shengbo Guo, Scott Sanner, and Edwin V Bonilla. Gaussian process preference elicitation. *Advances in Neural Information Processing Systems (NeurIPS)*, 23, 2010.

[39] Sunil Gupta, Alistair Shilton, Arun Kumar AV, Shannon Ryan, Majid Abdolshah, Hung Le, Santu Rana, Julian Berk, Mahad Rashid, and Svetha Venkatesh. BO-Muse: A human expert and AI teaming framework for accelerated experimental design. *arXiv preprint arXiv:2303.01684*, 2023.

[40] Kihyuk Hong, Yuhang Li, and Ambuj Tewari. An optimization-based algorithm for non-stationary kernel bandits without prior knowledge. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 3048–3085. PMLR, 2023.

[41] Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine learning*, 110(3):457–506, 2021.

[42] Carl Hvarfner, Frank Hutter, and Luigi Nardi. A general framework for user-guided Bayesian optimization. In *International Conference on Learning Representations (ICLR)*, 2024.

[43] Carl Hvarfner, Danny Stoll, Artur Souza, Marius Lindauer, Frank Hutter, and Luigi Nardi. πBO: Augmenting acquisition functions with user beliefs for Bayesian optimization. In *International Conference on Learning Representations (ICLR)*, 2022.

[44] Cole Jetton, Matthew Campbell, and Christopher Hoyle. Constraining the feasible design space in Bayesian optimization with user feedback. *Journal of Mechanical Design*, 146(4):041703, 2023.

[45] Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438, 2020.

[46] Michael I Jordan. Artificial intelligence—the revolution hasn't happened yet. *Harvard Data Science Review*, 1(1):1–9, 2019.

[47] Daniel Kahneman and Amos Tversky. On the interpretation of intuitive probability: A reply to Jonathan Cohen. *Cognition*, 7(4):409–411, 1979.

[48] Keren J Kanarik, Wojciech T Osowiecki, Yu Lu, Dipongkar Talukder, Niklas Roschewsky, Sae Na Park, Mattan Kamon, David M Fried, and Richard A Gottscho. Human–machine collaboration for improving semiconductor process development. *Nature*, 616(7958):707–711, 2023.

[49] Kirthevasan Kandasamy, Jeff Schneider, and Barnabás Póczos. High dimensional Bayesian optimisation and bandits via additive models. In *International Conference on Machine Learning (ICML)*, pages 295–304. PMLR, 2015.

[50] Ali Khoshvishkaie, Petrus Mikkola, Pierre-Alexandre Murena, and Samuel Kaski. Cooperative Bayesian optimization for imperfect agents. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML)*, pages 475–490. Springer, 2023.

[51] Johannes Kirschner, Mojmir Mutny, Nicole Hiller, Rasmus Ischebeck, and Andreas Krause. Adaptive and safe Bayesian optimization in high dimensions via one-dimensional subspaces. In *International Conference on Machine Learning (ICML)*, pages 3429–3438. PMLR, 2019.

[52] Cheng Li, Sunil Gupta, Santu Rana, Vu Nguyen, Antonio Robles-Kelly, and Svetha Venkatesh. Incorporating expert prior knowledge into experimental design via posterior sampling. *arXiv preprint arXiv:2002.11256*, 2020.

[53] Dong C Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.

[54] Qinghua Liu, Praneeth Netrapalli, Csaba Szepesvari, and Chi Jin. Optimistic MLE: A generic model-based algorithm for partially observable sequential decision making. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 363–376, 2023.

[55] Tennison Liu, Nicolás Astorga, Nabeel Seedat, and Mihaela van der Schaar. Large language models to enhance Bayesian optimization. In *The Twelfth International Conference on Learning Representations*, 2024.

[56] ER Logan, Erin M Tonita, KL Gering, Jing Li, Xiaowei Ma, LY Beaulieu, and JR Dahn. A study of the physical properties of Li-ion battery electrolytes containing esters. *Journal of The Electrochemical Society*, 165(2):A21, 2018.

[57] Arpan Losalka and Jonathan Scarlett. No-regret algorithms for safe Bayesian optimization with monotonicity constraints. In Sanjoy Dasgupta, Stephan Mandt, and Yingzhen Li, editors, *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 3232–3240. PMLR, 02–04 May 2024.

[58] Petrus Mikkola, Osvaldo A Martin, Suyog Chandramouli, Marcelo Hartmann, Oriol Abril Pla, Owen Thomas, Henri Pesonen, Jukka Corander, Aki Vehtari, Samuel Kaski, et al. Prior knowledge elicitation: The past, present, and future. *arXiv preprint arXiv:2112.01380*, 2112, 2021.

[59] Petrus Mikkola, Milica Todorović, Jari Järvi, Patrick Rinke, and Samuel Kaski. Projective preferential Bayesian optimization. In *International Conference on Machine Learning (ICML)*, pages 6884–6892. PMLR, 2020.

[60] Jonas Mockus, Vytautas Tiesis, and Antanas Zilinskas. The application of Bayesian methods for seeking the extremum. *Towards global optimization*, 2(117-129):2, 1978.

[61] Hossein Mohammadi, Rodolphe Le Riche, Nicolas Durrande, Eric Touboul, and Xavier Bay. An analytic comparison of regularization methods for Gaussian processes. *arXiv preprint arXiv:1602.00853*, 2016.

[62] Quoc Phong Nguyen, Wan Theng Ruth Chew, Le Song, Bryan Kian Hsiang Low, and Patrick Jaillet. Optimistic Bayesian optimization with unknown constraints. In *The Twelfth International Conference on Learning Representations*, 2024.

[63] Hannes Nickisch and Carl Edward Rasmussen. Approximations for binary Gaussian process classification. *Journal of Machine Learning Research (JMLR)*, 9(Oct):2035–2078, 2008.

[64] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 1999.

[65] Michael A Osborne, Roman Garnett, and Stephen J Roberts. Gaussian processes for global optimization. In *International Conference on Learning and Intelligent Optimization (LION3)*, 2009.

[66] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 27730–27744, 2022.

[67] Art Owen. Empirical likelihood ratio confidence regions. *The Annals of Statistics*, 18(1):90–120, 1990.

[68] Anthony O'Hagan. Expert knowledge elicitation: subjective but scientific. *The American Statistician*, 73(sup1):69–81, 2019.

[69] Anil Ramachandran, Sunil Gupta, Santu Rana, Cheng Li, and Svetha Venkatesh. Incorporating expert prior in Bayesian optimisation via space warping. *Knowledge-Based Systems*, 195:105663, 2020.

[70] Julian Rodemann, Federico Croppi, Philipp Arens, Yusuf Sale, Julia Herbinger, Bernd Bischl, Eyke Hüllermeier, Thomas Augustin, Conor J Walsh, and Giuseppe Casalicchio. Explaining Bayesian optimization by Shapley values facilitates human-AI collaboration. *arXiv preprint arXiv:2403.04629*, 2024.

[71] Howard Harry Rosenbrock. An automatic method for finding the greatest or least value of a function. *The computer journal*, 3(3):175–184, 1960.

[72] Zachary P Rosol, Natalie J German, and Stephen M Gross. Solubility, ionic conductivity and viscosity of lithium salts in room temperature ionic liquids. *Green Chemistry*, 11(9):1453–1457, 2009.

[73] Denise M Rousseau. Schema, promise and mutuality: The building blocks of the psychological contract. *Journal of occupational and organizational psychology*, 74(4):511–541, 2001.

[74] 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 (ICLR)*, 2021.

[75] Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations (ICLR)*, 2022.

[76] Jonathan Scarlett, Ilija Bogunovic, and Volkan Cevher. Lower bounds on regret for noisy Gaussian process bandit optimization. In *Conference on Learning Theory*, pages 1723–1742. PMLR, 2017.

[77] Bernhard Schölkopf, Ralf Herbrich, and Alex J Smola. A generalized representer theorem. In *International Conference on Computational Learning Theory*, pages 416–426. Springer, 2001.

[78] Robin Senge, Stefan Bösner, Krzysztof Dembczyński, Jörg Haasenritter, Oliver Hirsch, Norbert Donner-Banzhoff, and Eyke Hüllermeier. Reliable classification: Learning classifiers that distinguish aleatoric and epistemic uncertainty. *Information Sciences*, 255:16–29, 2014.

[79] Nigel R Shadbolt, Paul R Smart, J Wilson, and S Sharples. Knowledge elicitation. *Evaluation of human work*, pages 163–200, 2015.

[80] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015.

[81] Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235, Online, November 2020. Association for Computational Linguistics.

[82] Artur Souza, Luigi Nardi, Leonardo B Oliveira, Kunle Olukotun, Marius Lindauer, and Frank Hutter. Bayesian optimization with a prior for the optimum. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML)*, pages 265–296. Springer, 2021.

[83] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *International Conference on Machine Learning (ICML)*, pages 1015–1022, 2010.

[84] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias W Seeger. Information-theoretic regret bounds for Gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265, 2012.

[85] Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*, 2022.

[86] Michael L Stein. *Interpolation of spatial data*. Springer Science & Business Media, 1999.

[87] Yanan Sui, Joel Burdick, Yisong Yue, et al. Stage-wise safe Bayesian optimization with Gaussian processes. In *Proc. of the Int. Conf. on Mach. Learn.*, pages 4781–4789, 2018.

[88] Yanan Sui, Alkis Gotovos, Joel Burdick, and Andreas Krause. Safe exploration for optimization with Gaussian processes. In *Proc. of the Int. Conf. on Mach. Learn.*, pages 997–1005, 2015.

[89] S. Surjanovic and D. Bingham. Virtual library of simulation experiments: Test functions and datasets. Retrieved May 17, 2024, from `http://www.sfu.ca/~ssurjano`, 2024.

[90] Kevin Swersky, Jasper Snoek, and Ryan P Adams. Multi-task Bayesian optimization. *Advances in Neural Information Processing Systems (NeurIPS)*, 26, 2013.

[91] Shion Takeno, Masahiro Nomura, and Masayuki Karasuyama. Towards practical preferential Bayesian optimization with skew Gaussian processes. In *International Conference on Machine Learning (ICML)*, pages 33516–33533. PMLR, 2023.

[92] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations (ICLR)*, 2019.

[93] Sattar Vakili, Kia Khezeli, and Victor Picheny. On information gain and regret bounds in Gaussian process bandits. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 82–90. PMLR, 2021.

[94] Vladimir Naumovich Vapnik, Vlamimir Vapnik, et al. *Statistical learning theory*. wiley New York, 1998.

[95] Andreas Wächter and Lorenz T Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006.

[96] Shengbo Wang and Ke Li. Constrained Bayesian optimization under partial observations: Balanced improvements and provable convergence. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 15607–15615, 2024.

[97] Ziyu Wang, Frank Hutter, Masrour Zoghi, David Matheson, and Nando De Feitas. Bayesian optimization in a billion dimensions via random embeddings. *Journal of Artificial Intelligence Research*, 55:361–387, 2016.

[98] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:24824–24837, 2022.

[99] Colin White, Willie Neiswanger, and Yash Savani. Bananas: Bayesian optimization with neural architectures for neural architecture search. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 10293–10301, 2021.

[100] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006.

[101] Martin Wistuba and Josif Grabocka. Few-shot Bayesian optimization with deep kernel surrogates. In *International Conference on Learning Representations (ICLR)*, 2020.

[102] David H Wolpert and William G Macready. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1):67–82, 1997.

[103] Jian Wu, Saul Toscano-Palmerin, Peter I Frazier, and Andrew Gordon Wilson. Practical multi-fidelity Bayesian optimization for hyperparameter tuning. In *Uncertainty in Artificial Intelligence (UAI)*, pages 788–798. PMLR, 2020.

[104] Yihong Wu. Lecture notes on information-theoretic methods for high-dimensional statistics. *Lecture Notes for ECE598YW (UIUC)*, 16, 2017.

[105] Wenjie Xu, Yuning Jiang, Emilio T Maddalena, and Colin N Jones. Lower bounds on the noiseless worst-case complexity of efficient global optimization. *Journal of Optimization Theory and Applications*, pages 1–26, 2024.

[106] Wenjie Xu, Yuning Jiang, Bratislav Svetozarevic, and Colin Jones. Constrained efficient global optimization of expensive black-box functions. In *International Conference on Machine Learning (ICML)*, pages 38485–38498. PMLR, 2023.

[107] Wenjie Xu, Wenbin Wang, Yuning Jiang, Bratislav Svetozarevic, and Colin Jones. Principled preferential Bayesian optimization. In *Forty-first International Conference on Machine Learning*.

[108] Jingxian Zhang, Ning Zhao, Miao Zhang, Yiqiu Li, Paul K Chu, Xiangxin Guo, Zengfeng Di, Xi Wang, and Hong Li. Flexible and ion-conducting membrane electrolytes for solid-state lithium batteries: Dispersion of garnet nanoparticles in insulating polyethylene oxide. *Nano Energy*, 28:447–454, 2016.

[109] Ding-Xuan Zhou. The covering number in learning theory. *Journal of Complexity*, 18(3):739–767, 2002.

[110] Xingyu Zhou and Bo Ji. On kernelized multi-armed bandits with constraints. *Advances in Neural Information Processing Systems (NeurIPS)*, 35, 2022.

Clarification

The Appendix is provided in Section E. Our optimistic MLE approach is rooted in the RKHS viewpoint, which has a strong connection with GP, as discussed in Chapter 2. Compared to GP, a key drawback is the lack of a closed-form posterior predictive distribution. Instead, the optimistic MLE approach provides a confidence interval for a specified probability δ. Our primal-dual framework incorporates this interval-based constraint as a deterministic constraint with iterative updates, offering broad support for arbitrary acquisition functions by regularizing their maximizers. For example, with the expected improvement acquisition function $\alpha_{\text{EI}}(x)$, the expert-augmented suggestion is generated by $x_t^c = \min_{x \in \mathcal{X}} \mathbb{P}(\underline{g}(x) \leq 0 \mid x) \alpha_{\text{EI}}(x)$. The probability of this constraint is defined as $\mathbb{P}(\underline{g}(x) \leq 0 \mid x) = S(\underline{g}(x))$.

Applying this method to the Bayesian compression approach described in Part I is straightforward by imposing an upper-bounded constraint on the measure $\tilde{\mu}(x) \approx \mu(x)\mathbb{P}(\overline{g}(x) \geq 0 \mid x)$ [1] and iteratively updating the RKHS of g.

Alternative approach

Another potential approach is to frame the human belief model as low-fidelity information and apply multi-fidelity Bayesian optimization (Kandasamy et al. [120]). This approach can enhance the predictive ability of the objective model f_t, similar to the method described in Chapter 6. Empirical evidence suggests that this approach works well (e.g., Kristiadi et al. [137]). However, deriving an information-theoretic bound for a multi-output GP model is challenging, particularly when combining regression and classification models.

Furthermore, Kandasamy et al. [120] demonstrated that the convergence acceleration achieved through low-fidelity information is essentially equivalent to the interpretation of constrained optimization. Low-fidelity information helps constrain the search space to more promising regions, thereby accelerating optimization by reducing the domain volume. However, this introduces robustness concerns, as the optimization process becomes dependent on the reliability of the low-fidelity

[1] This follows the maximization formulation in Chapter 4.

information. Addressing this, Mikkola et al. [159] showed that robustification is necessary to ensure the same order of regret bounds, thereby maintaining theoretical guarantees.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Principled Bayesian Optimisation in Collaboration with Human Experts
Publication Status	□Published ☑ Accepted for Publication □Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	Wenjie Xu*, Masaki Adachi*, Colin N. Jones, and Michael A. Osborne. "Principled Bayesian Optimisation in Collaboration with Human Experts." In Advances in Neural Information Processing Systems (NeurIPS) 37, 2024.

Student Confirmation

Student Name:	Masaki Adachi		
Contribution to the Paper	Co-first author I developed the core idea of a human-AI collaborative setting using a preferential Bayesian optimization approach, implemented the methodology, conducted all experiments for the paper, performed additional analyses, and authored the manuscript.		
Signature 		Date	06 January 2025

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Michael A. Osborne, Professor of Machine Learning		
Supervisor comments I can confirm that, to the best of my knowledge, Masaki's description above is fair, and that I have great trust in Masaki.		
Signature 	Date	3 February 2025

This completed form should be included in the thesis, at the end of the relevant chapter.

8
Conclusion

We have explored Probabilistic Numerics algorithms that efficiently align human beliefs and desiderata with a computational agent's beliefs and policies. In the first half, we demonstrated that existing Probabilistic Numerics tasks can be unified through the lens of Bayesian data compression, where task differences are interpreted as differences in the probability measure. In the latter half, we showed how preference learning can seamlessly integrate human experts' tacit knowledge into computational agents while maintaining robustness against uninformative advice. This approach reduces cognitive load both in algorithm implementation for aligning desiderata and in deployment to safely incorporate the expert knowledge.

In conclusion, we will discuss the broader impacts of these methods in their application, as outlined in Section 8.1, and highlight potential future research directions beyond this thesis in Section 8.2.

8.1 Broader Applications

8.1.1 Applications in Science and Engineering

Battery science

Batteries are a key enabler for a decarbonized grid and electrified vehicles. Extending their lifetime and recycling them as second-life batteries is of paramount importance in achieving a circular economy. However, the degradation mechanisms of batteries are inherently complex, involving multi-chain chemical reactions, inhomogeneous microstructures, and non-equilibrium reactions that hinder principled understanding and accurate deterministic predictions using physics-based simulators. Nonetheless, the degradation trends exhibited over years are often empirically predictable using simple methods in many cases. Given this, lifetime prediction using machine learning has emerged as a popular topic in academia. However, lifetime prediction is just one of the many challenging tasks, and there are numerous other potential applications for our approaches.

In Chapter 5, we applied our black-box inference approach to system identification, specifically parameter estimation and model selection for physics-based models. In our recent work, we extended this to experimental design. Lifetime prediction requires data to train the model, but lifetime experiments take years to complete. Therefore, it is crucial to wisely select experimental conditions given limited resources, such as a limited number of iterations (T), experimental apparatus for running experiments (batch size n), and available battery samples (nT). This scenario naturally aligns with the active learning framework introduced in Chapter 4.

Beyond lifetime prediction, other tasks can also benefit from our approaches. For instance, cell matching involves matching n second-life batteries from N candidates ($N \gg n$). This is typically performed by selecting the n combinations of cells that yield the smallest performance differences, often based on current capacity. However, as each battery undergoes unique degradation conditions, future lifetime decay may vary over time. Our probabilistic approach can enable *predictive matching*, which minimizes $\int_{t=0}^{T}(Q_i(t) - Q_j(t))^2 \mathrm{d}t$, where Q is the capacity at time t, and

$i, j \in [n]$ represent the indices of the batteries. This approach focuses on integral variance reduction in probabilistic forecasting. Unlike the industry-standard metric that minimizes current capacity differences, $|Q_i(t=0) - Q_j(t=0)|$, our method minimizes the expected capacity differences over time, leading to more robust and reliable cell matching.

Astrophysics

Modern space observatories process increasingly large datasets, necessitating automation and acceleration in spectral analysis methods. Spectral analysis relies on simulators that reflect the current understanding of the universe's model. However, observed spectra are highly mixed with signals from different molecules, making it challenging to replace simulators with cheaper but less accurate surrogate predictors, such as deep learning-based models. Consequently, modern astrophysicists require scalable and reliable methods for simulation-based inference. Similar challenges are prevalent in other scientific domains, such as exoplanet detection (Parviainen [178]), particle physics (Cranmer et al. [56]) and materials science (Roussel et al. [190]).

In our work (Lin et al. [148]), we applied a highly parallelizable approach introduced in Chapters 3–4. The observed spectra are linked to their astronomical coordinates, resulting in 2D parameter maps. To reduce uncertainty in the visualization of these 2D maps, we applied Bayesian active learning to select the most informative spectra. Since similar astrophysical locations are expected to share similar molecular compositions, assuming smoothness in the 2D input space is natural. We modeled the 2D parameter map as a GP prior with many output dimensions (simulator parameters). To handle this efficiently, we employed a multi-output GP (Bonilla et al. [28]) with a Kronecker-structured kernel for fast computation (Maddox et al. [156]).

We applied this method to a spectral cube of 40,000 (200×200) pixels, where each pixel contains 138,016 frequency grids and 468 molecular parameter maps. While traditional MCMC-based approaches require approximately 2×10^6 hours, our approach achieves the same level of accuracy in just 20 hours, reducing computation

costs by a factor of $1/10^5$. This rapid visualization meets the demands of big data in modern astronomical surveys to efficiently screen out which data to focus.

Other applications in science and engineering

In addition to astrophysics, our work has begun to gain traction in several scientific and engineering communities. For instance, it has been applied to microcontroller design optimization (Gao et al. [80]), energy supply-demand matching (Lee [146]), chemical reaction optimization (Sin et al. [202]), self-driving laboratories (Kristiadi et al. [137]), physics-based inverse modeling (Philipp et al. [179]), mechanical engineering (Weia et al. [227]), reinforcement learning (Hayakawa et al. [96]), and AutoML (Theodorakopoulos et al. [214]). These applications highlight the growing interest and applicability of our approach across diverse domains.

We believe these represent just a fraction of the many potential applications of Probabilistic Numerics using the Bayesian data compression approach.

Extension to multiple experts

As a natural extension of Human-AI collaboration discussed in Chapters 6-7, we consider the case of multiple experts in our formulation. If we adopt a utilitarian perspective, the solution is straightforward: simply take the average of their feedback. In a preference learning framework, this is equivalent to assuming that the group feedback, or votes, are generated by a single individual with some randomness following a Bernoulli likelihood. Under this assumption, our method can be applied without modification by aggregating all feedback into one model.

However, in real-world scenarios, it is common for some experts' advice to be effective while others' may not. This necessitates the use of an aggregation function to assign weights to individual utility functions. The aggregation function, also referred to as a social welfare function (Coleman [53]) or choice function (Arrow [16]), can be viewed as a higher-order preference or rule that determines which experts should be prioritized.

In settings with objective feedback, such as in our collaborative framework, the objective model can eventually guide which experts to trust. However, in

the absence of objective feedback, the aggregation function must be determined a priori. Voting is inherently a social activity and can be influenced by interactions between voters, which may skew the outcomes. Such skewing can arise from various social factors, such as professional hierarchies.

In our work (Adachi et al. [5]), we examined the preference maximization task under social influence. We modeled social influence as a graph convolutional operator acting on utility functions. Under this influence model, we proved—using social choice theory—that achieving a social-influence-free consensus from influence-affected votes alone is theoretically impossible. To address this, we developed an algorithm that circumvents the impossibility theorem by relaxing certain assumptions. This algorithm achieves sample-efficient consensus by leveraging both corrupted votes and social-influence-free votes, enabling effective decision-making even in the presence of social influence.

Extension to unknown kernel hyperparameters

In theory, we assumed that the true RKHS, or kernel hyperparameters such as the lengthscale, are given. However, in practice, these are unknown a priori and must be estimated. While we demonstrated robustness to misspecified RKHS in Proposition 3, we do not claim that the uniform bound is superior to finding the true RKHS. In practice, the true RKHS is, of course, preferable to a misspecified one, as the error bound vanishes when the RKHS is correctly specified. Consequently, estimating the true hyperparameters from the given data is highly desirable. However, the common practice of type-II maximum likelihood estimation is known to be ill-posed (Karvonen et al. [124]). Therefore, more careful estimation methods are crucial to bridging the gap between theory and practice.

In our work (Ziomek et al. [246, 247]), we proposed two approaches to address unknown hyperparameters. In Ziomek et al. [247], we employed a 'learning-to-defer' approach, maintaining a set of possible hyperparameters throughout the iterations and eliminating implausible candidates when the queried y_t falls outside the confidence interval of the predictive distribution from the $(t-1)$-th iteration.

Conversely, in Ziomek et al. [246], we adopted a regret-balancing strategy, leveraging the suspected regret bound for a given lengthscale to select candidates from the set in a way that balances regret across options.

8.2 Future directions

8.2.1 Bayesian data compression

Connection with evolutionary search

The Bayesian optimization approach via probabilistic lifting, introduced in Chapters 2 and 4, shares a connection with evolutionary search methods, as both aim to optimize the search distribution ($\mu(x) = \mathbb{P}(x^\star \mid \mathbf{D}_t)$ in our terminology). While evolutionary search explicitly models the parametric form of the search distribution, our approach models the objective function f and subsequently estimates the search distribution.

In our new approach (Osselin et al. [175]), we employed a parametric search distribution based on a multivariate normal distribution, allowing us to derive the closed-form probabilistic structure of its gradient with respect to the natural gradient, applied to evolutionary search. The natural evolutionary search method (Wierstra et al. [229]) is also known to have a connection with the Bayesian learning rule (Khan et al. [126]), which is applicable to large deep learning models (e.g., Shen et al. [200]). This approach has the potential to scale up Probabilistic Numerics to high-dimensional and large-scale datasets through Bayesian deep learning.

Connection with time-series prediction

The signature transform maps sequentially ordered data (rough paths), such as time-series or genetic data, onto features in a Banach space, providing a powerful similarity measure between paths that may vary in length or be irregularly sampled. In this sense, the signature transform can be regarded as a form of 'compression' for sequentially ordered data. The signature kernel (Lee et al. [145]) is a kernelized version of the signature transform, enabling similarity measurement between paths using signature features. In time-series prediction tasks using GP regression models,

the signature kernel has demonstrated exceptional predictive performance (e.g., Toth et al. [216]).

The cubature on Wiener space represents an exact kernel quadrature or a truncated signature kernel (Király et al. [130]) with respect to the Wiener measure. This connection suggests that our kernel quadrature approach may extend naturally to the Wiener space, broadening the applicability of our compression methods. Moreover, GP-based time-series forecasting has a deep connection to state-space models and the Kalman filter, one of the foundational approaches to Bayesian filtering. This alignment hints at the potential for extending our approach to tackle differential equation solver tasks within the Probabilistic Numerics framework.

In our recent work (Tóth et al. [217]), we applied a variational approximation to accelerate computation, achieving effectiveness in large-scale experiments. This work demonstrated that our approximated signature kernel-based GP performs on par with state-of-the-art deep learning methods, including diffusion models (Kollovieh et al. [134]) and transformers (Vaswani [221]), while offering much faster computation through an efficient auto-regressive update method. These approximations could be further developed to create differential equation solvers with significantly improved computational efficiency.

Connection with continual learning

Continual learning involves developing models that can adaptively perform well with datasets continuously streamed in an online manner, potentially irregularly, with both input and output data distributions shifting or even changing completely. One significant challenge in this context is catastrophic forgetting, where the model loses its predictive ability on previously seen data by over-adapting to new data. When we assume that the test dataset is shifted, excessive adaptation can cause an undesirable shift in the predictive distribution.

Among the numerous approaches, replay-based methods (Pan et al. [177]) and test-time adaptation (Wang et al. [224] and Wang et al. [225]) are particularly relevant to our method. Replay-based methods involve using representative data

points from the previous data distribution, which are selectively stored and reused during training to remind. This data point selection scheme can be interpreted as a Bayesian compression task, as it seeks to summarize past data efficiently.

Test-time adaptation, on the other hand, does not update the model parameters globally. Instead, it adjusts only the final normalizing layer parameters during prediction (test) time by minimizing the entropy of the predictive distribution, $\mathbb{P}(y_t \mid x_t, \mathbf{D}_t)$. This adaptation is applied to each batch of prediction pairs. Since entropy is strongly connected to predictive variance (and thus errors), minimizing entropy can be understood as minimizing the expected error on the batch. This process can also be interpreted as reducing the expected predictive variance with a limited sample set. As a result, this task can be framed as a sample-efficient estimation of the integral for the objective function.

8.2.2 Human-AI collaboration

Connection with explainability

In Chapter 6, we applied Shapley value-based feature importance as an explanation of the acquisition process. The explainability tools can also be extended to enhance collaboration in other ways. For example, in the context of Shapley values for RKHS (Chau et al. [45]), the concept of a Shapley regularizer was introduced to control the contribution of specific features to the model. Using this approach, experts could directly indicate which features are more important or provide this information indirectly through pairwise comparison estimation. This feedback could then be incorporated as a "shrinking" prior over the function space, analogous to our optimistic MLE approach in Chapter 7. This idea aligns with projective preference optimization (Mikkola et al. [160]) but is extended to more general functions in RKHS.

Another interesting direction is to use explanations as the target of experimental design. In experimental design, the goal is often to understand feature importance rather than to create a highly accurate predictive model (as in active learning) or to identify extremes (as in Bayesian optimization). This approach has recently

been explored using gradient-based sensitivity analysis (Belakaria et al. [20]), but it could also be implemented with stochastic Shapley values, providing a novel and flexible framework for feature-focused experimental design.

Connection with large language models

In Chapter 7, we mentioned the possibility of using large language models (LLMs) as substitutes for human experts in well-defined tasks. Indeed, Liu et al. [151] has already employed LLMs to generate candidate solutions, and similar prompts could be used to elicit preferential labels from LLMs. However, sample-based elicitation can be inefficient. Given that LLMs are parameterized models built to output numerical values, it may be more natural to consider how to extract their encoded prior knowledge as numerical representations directly.

One straightforward idea is to query a large number of points at the outset, effectively creating an extensive dataset. Using this dataset, the entire function could be estimated accurately with a preference model, allowing the estimated near-deterministic model to serve as a constraint. This would make the primal-dual optimization objective more efficient and eliminate the need to query the LLM iteratively, point by point. Developing more efficient elicitation methods for LLMs represents an exciting and promising direction for future research.

Connection with Economics

Many advancements from economics can be applied to human-AI collaborative settings. While game-theoretic approaches have already been widely adopted in multi-objective optimization—such as Pareto Frontiers (Ngatchou et al. [169]), Shapley values (Sundararajan et al. [207]), and Nash bargaining solutions (Binmore et al. [26])—there remain numerous other theories that could be imported to further enrich this field.

For instance, nudge theory (Thaler et al. [213]) suggests that wisely selecting a default option (a 'nudge') can guide human behavior in a desirable direction without restricting freedom of choice. In the context of optimization, this can be interpreted as setting an effective initial seed, which, while significantly influencing

the convergence rate, remains independent of the theoretical rate itself. This effect is likely to have a profound impact in human-AI collaborative settings.

Additionally, explanations can also act as a form of nudge, helping humans calibrate their beliefs about AI suggestions and their own decision-making. A particularly relevant cognitive bias in human-AI collaboration is automation bias, where humans place excessive trust in AI suggestions. When explanations accompany AI decisions, humans can better understand the reliability of the suggestions. This enables them to recalibrate their decisions, especially in early stages when data is limited, allowing them to recognize situations where their own decisions may outperform the AI's recommendations.

This concept becomes particularly relevant in the context of multiple-agent scenarios. As discussed earlier, social influence can skew vote consensus. However, considering the opposite effect—where social influence debiases misspecified individual utility functions—we can interpret social influence as a form of nudge that guides the group toward a better consensus.

Another important yet underexplored area is social choice theory (Sen [196]). This theory takes a unique approach to understanding decision-making settings. It first seeks to establish impossibility theorems by presuming a combination of important axioms, and then explores appropriate relaxations of these axioms to make solutions feasible. Conceptually, this is similar to the approach in theoretical physics, where physicists test established physical laws under extreme conditions, study how those laws break, and develop fixes or extensions.

Social choice theory has recently been applied to areas such as collective intelligence, domain generalization, multi-modal learning, algorithmic fairness, and AI safety (Muandet [164]). It provides a framework to prove when these concepts break down and offers guidance on how to address such failures by relaxing certain axioms.

We applied social choice theory to preference aggregation tasks in the context of multiple-expert collaboration (Adachi et al. [5]). Our analysis proved the impossibility of achieving a social-influence-free consensus, demonstrating that it is

theoretically unachievable under the strict axiom of inaccessibility to social-influence-free votes. To address this, we relaxed this axiom, enabling the development of a feasible solution to the problem.

These ideas have the potential to be extended to alignment research more broadly, including the alignment of large language models and other AI systems, offering insights into how nudges and social dynamics can improve collective decision-making and system behavior.

References

[1] Luigi Acerbi. "Variational Bayesian Monte Carlo". In: *Advances in Neural Information Processing Systems* 31 (2018).

[2] Luigi Acerbi. "Variational Bayesian Monte Carlo with noisy likelihoods". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 8211–8222.

[3] Masaki Adachi. "High-dimensional discrete Bayesian optimization with self-supervised representation learning for data-efficient materials exploration". In: *NeurIPS 2021 AI for Science Workshop*. 2021. URL: https://openreview.net/forum?id=xJhjehqjQeB.

[4] Masaki Adachi. "Mixture-of-experts ensemble with hierarchical deep metric learning for spectroscopic identification". In: *NeurIPS Workshop: Machine Learning and the Physical Science* (2021).

[5] Masaki Adachi, Siu Lun Chau, Wenjie Xu, Anurag Singh, Michael A. Osborne, and Krikamol Muandet. "Bayesian optimization for building social-influence-free consensus". In: *arXiv preprint* (2025). URL: https://doi.org/10.48550/arXiv.2502.07166.

[6] Masaki Adachi, Masahiro Fujisawa, and Michael A Osborne. "Fixing the Pitfalls of Probabilistic Time-Series Forecasting Evaluation by Kernel Quadrature". In: *arXiv preprint arXiv:2503.06079* (2025).

[7] Masaki Adachi, Satoshi Hayakawa, Saad Hamid, Martin Jørgensen, Harald Oberhauser, and Michael A Osborne. "SOBER: highly parallel Bayesian optimization and Bayesian quadrature over discrete and mixed spaces". In: *ICML 2023 Workshop: Sampling and Optimization in Discrete Space*. 2023. DOI: https://doi.org/10.48550/arXiv.2301.11832.

[8] Masaki Adachi, Satoshi Hayakawa, Martin Jørgensen, Saad Hamid, Harald Oberhauser, and Michael A Osborne. "A quadrature approach for general-purpose batch Bayesian optimization via probabilistic lifting". In: *arXiv preprint arXiv:2404.12219* (2024).

[9] Masaki Adachi, Satoshi Hayakawa, Martin Jørgensen, Harald Oberhauser, and Michael A Osborne. "Fast Bayesian inference with batch Bayesian quadrature via kernel recombination". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 16533–16547. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/697200c9d1710c2799720b660abd11bb-Paper-Conference.pdf.

[10] Masaki Adachi, Satoshi Hayakawa, Martin Jørgensen, Xingchen Wan, Vu Nguyen, Harald Oberhauser, and Michael A Osborne. "Adaptive batch sizes for active learning: A probabilistic numerics approach". In: *International Conference on Artificial Intelligence and Statistics (AISTATS)*. Vol. 238. PMLR, 2024, pp. 496–504. URL: https://proceedings.mlr.press/v238/adachi24b.html.

[11] Masaki Adachi, Yannick Kuhn, Birger Horstmann, Arnulf Latz, Michael A Osborne, and David A Howey. "Bayesian model selection of lithium-ion battery models via Bayesian quadrature". In: *IFAC-PapersOnLine* 56.2 (2023). 22nd IFAC World Congress, pp. 10521–10526. DOI: https://doi.org/10.1016/j.ifacol.2023.10.1073.

[12] Masaki Adachi, Brady Planden, David Howey, Michael A. Osborne, Sebastian Orbell, Natalia Ares, Krikamol Muandet, and Siu Lun Chau. "Looping in the human: collaborative and explainable Bayesian optimization". In: *International Conference on Artificial Intelligence and Statistics (AISTATS)*. Vol. 238. PMLR, 2024, pp. 505–513. URL: https://proceedings.mlr.press/v238/adachi24a.html.

[13] A. Aitio, S. G. Marquis, P. Ascencio, and D. A. Howey. "Bayesian parameter estimation applied to the Li-ion battery single particle model with electrolyte dynamics". In: *IFAC-PapersOnLine* 53.2 (2020), pp. 12497–12504.

[14] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. "Towards better understanding of gradient-based attribution methods for Deep Neural Networks". In: *International Conference on Learning Representations*. 2018. URL: https://openreview.net/forum?id=Sy21R9JAW.

[15] Nachman Aronszajn. "Theory of reproducing kernels". In: *Transactions of the American mathematical society* 68.3 (1950), pp. 337–404.

[16] Kenneth J Arrow. "Rational choice functions and orderings". In: *Economica* 26.102 (1959), pp. 121–127.

[17] Arun Kumar AV, Santu Rana, Alistair Shilton, and Svetha Venkatesh. "Human-AI Collaborative Bayesian Optimisation". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 16233–16245.

[18] Javad Azimi, Alan Fern, and Xiaoli Fern. "Batch Bayesian optimization via simulation matching". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 23. 2010. URL: https://proceedings.neurips.cc/paper_files/paper/2010/file/e702e51da2c0f5be4dd354bb3e295d37-Paper.pdf.

[19] Francis Bach. "On the equivalence between kernel quadrature rules and random feature expansions". In: *Journal of Machine Learning Research* 18 (2017), p. 714. URL: https://jmlr.org/papers/volume18/15-178/15-178.pdf.

[20] Syrine Belakaria, Benjamin Letham, Jana Doppa, Barbara E Engelhardt, Stefano Ermon, and Eytan Bakshy. "Active Learning for Derivative-Based Global Sensitivity Analysis with Gaussian Processes". In: *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. 2024. URL: https://openreview.net/forum?id=da0ZJatRCN.

[21] Ayoub Belhadji. "An analysis of Ermakov-Zolotukhin quadrature using kernels". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 34. 2021, pp. 27278–27289. URL: https://proceedings.neurips.cc/paper_files/paper/2021/file/e531e258fe3098c3bdd707c30a687d73-Paper.pdf.

[22] Ayoub Belhadji, Rémi Bardenet, and Pierre Chainais. "Kernel interpolation with continuous volume sampling". In: *International Conference on Machine Learning (ICML)*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. 2020, pp. 725–735. URL: http://proceedings.mlr.press/v119/belhadji20a/belhadji20a.pdf.

[23] Ayoub Belhadji, Rémi Bardenet, and Pierre Chainais. "Kernel quadrature with DPPs". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 32. 2019. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/7012ef0335aa2adbab58bd6d0702ba41-Paper.pdf.

[24] Alessio Benavoli, Dario Azzimonti, and Dario Piga. "Skew Gaussian processes for classification". In: *Machine Learning* 109.9 (2020), pp. 1877–1902.

[25] Alain Berlinet and Christine Thomas-Agnan. *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media, 2011.

[26] Ken Binmore, Ariel Rubinstein, and Asher Wolinsky. "The Nash bargaining solution in economic modelling". In: *The RAND Journal of Economics* (1986), pp. 176–188.

[27] Ilija Bogunovic and Andreas Krause. "Misspecified Gaussian process bandit optimization". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 34. 2021, pp. 3004–3015. URL: https://proceedings.neurips.cc/paper_files/paper/2021/file/177db6acfe388526a4c7bff88e1feb15-Paper.pdf.

[28] Edwin V Bonilla, Kian Chai, and Christopher Williams. "Multi-task Gaussian process prediction". In: *Advances in neural information processing systems* 20 (2007).

[29] Nathanael Bosch, Adrien Corenflos, Fatemeh Yaghoobi, Filip Tronarp, Philipp Hennig, and Simo Särkkä. "Parallel-in-time probabilistic numerical ODE solvers". In: *Journal of Machine Learning Research* 25.206 (2024), pp. 1–27.

[30] George EP Box. "Science and statistics". In: *Journal of the American Statistical Association* 71.356 (1976), pp. 791–799.

[31] Ralph Allan Bradley and Milton E. Terry. "Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons". In: *Biometrika* 39.3/4 (1952), pp. 324–345.

[32] F.-X. Briol, C. J. Oates, M. Girolami, and M. A. Osborne. "Frank-Wolfe Bayesian Quadrature: Probabilistic Integration with Theoretical Guarantees". In: *International Conference on Neural Information Processing Systems (NeurIPS)*. 2015. URL: https://proceedings.neurips.cc/paper/2015/file/ba3866600c3540f67c1e9575e213be0a-Paper.pdf.

[33] François-Xavier Briol, Chris J Oates, Mark Girolami, Michael A Osborne, and Dino Sejdinovic. "Probabilistic integration: A role in statistical computation?" In: *Statistical Science* 34.1 (2019), pp. 1–22. URL: https://www.jstor.org/stable/26771026.

[34] J. Buchner. "A statistical test for nested sampling algorithms". In: *Statistics and Computing* 26.1 (2016), pp. 383–392.

[35] J. Buchner. "Collaborative nested sampling: Big data versus complex physical models". In: *Publications of the Astronomical Society of the Pacific* 131.1004 (2019), p. 108005.

[36] David V Budescu and Wendy Weiss. "Reflection of transitive and intransitive preferences: A test of prospect theory". In: *Organizational Behavior and Human Decision Processes* 39.2 (1987), pp. 184–202.

[37] Adam D Bull. "Convergence rates of efficient global optimization algorithms". In: *Journal of Machine Learning Research* 12.88 (2011), pp. 2879–2904. URL: http://jmlr.org/papers/v12/bull11a.html.

[38] Trevor Campbell and Tamara Broderick. "Automated scalable Bayesian inference via Hilbert coresets". In: *The Journal of Machine Learning Research* 20.1 (2019), pp. 551–588.

[39] Constantin Carathéodory. "Über den Variabilitätsbereich der Fourier'schen Konstanten von positiven harmonischen Funktionen". In: *Rendiconti Del Circolo Matematico di Palermo (1884-1940)* 32.1 (1911), pp. 193–217.

[40] Daniel R Cavagnaro, Jay I Myung, Mark A Pitt, and Janne V Kujala. "Adaptive design optimization: A mutual information-based approach to model discrimination in cognitive science". In: *Neural computation* 22.4 (2010), pp. 887–905.

[41] H. Chai, J. F. Ton, M. A. Osborne, and R. Garnett. "Automated Model Selection with Bayesian Quadrature". In: *International Conference on Machine Learning (ICML)*. Vol. 97. 2019, pp. 931–940. URL: https://proceedings.mlr.press/v97/chai19a.html.

[42] Henry R Chai and Roman Garnett. "Improving quadrature for constrained integrands". In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR. 2019, pp. 2751–2759.

[43] Kathryn Chaloner and Isabella Verdinelli. "Bayesian experimental design: A review". In: *Statistical science* (1995), pp. 273–304.

[44] Paul Edmund Chang, Prakhar Verma, ST John, Arno Solin, and Mohammad Emtiyaz Khan. "Memory-based dual Gaussian processes for sequential learning". In: *International Conference on Machine Learning*. PMLR. 2023, pp. 4035–4054.

[45] Siu Lun Chau, Robert Hu, Javier Gonzalez, and Dino Sejdinovic. "RKHS-SHAP: Shapley values for kernel methods". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 13050–13063.

[46] Siu Lun Chau, Krikamol Muandet, and Dino Sejdinovic. "Explaining the Uncertain: Stochastic Shapley Values for Gaussian Process Models". In: *arXiv preprint arXiv:2305.15167* (2023).

[47] Yutian Chen, Max Welling, and Alex Smola. "Super-samples from kernel herding". In: *International Conference on Uncertainty in Artificial Intelligence (UAI)*. 2010, pp. 109–116. URL: https://doi.org/10.48550/arXiv.1203.3472.

[48] Yi Cheng and Yu Shen. "Bayesian adaptive designs for clinical trials". In: *Biometrika* 92.3 (2005), pp. 633–646.

[49] Oh-Hyeon Choung, Riccardo Vianello, Marwin Segler, Nikolaus Stiefl, and José Jiménez-Luna. "Extracting medicinal chemistry intuition via preference machine learning". In: *Nature Communications* 14.1 (2023), p. 6651.

[50] Sayak Ray Chowdhury and Aditya Gopalan. "On kernelized multi-armed bandits". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 844–853.

[51] Wei Chu and Zoubin Ghahramani. "Preference learning with Gaussian processes". In: *Proceedings of the 22nd International Conference on Machine Learning*. 2005, pp. 137–144.

[52] Jon Cockayne, Chris J Oates, Timothy John Sullivan, and Mark Girolami. "Bayesian probabilistic numerical methods". In: *SIAM review* 61.4 (2019), pp. 756–789.

[53] James S Coleman. "The possibility of a social welfare function". In: *The American Economic Review* 56.5 (1966), pp. 1105–1122.

[54] Francesco Cosentino, Harald Oberhauser, and Alessandro Abate. "A randomized algorithm to reduce the support of discrete measures". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 33. 2020, pp. 15100–15110. URL: `https://proceedings.neurips.cc/paper_files/paper/2020/file/ac4395adcb3da3b2af3d3972d7a10221-Paper.pdf`.

[55] Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

[56] Kyle Cranmer, Johann Brehmer, and Gilles Louppe. "The frontier of simulation-based inference". In: *Proceedings of the National Academy of Sciences* 117.48 (2020), pp. 30055–30062.

[57] Katalin Csilléry, Michael GB Blum, Oscar E Gaggiotti, and Olivier François. "Approximate Bayesian computation (ABC) in practice". In: *Trends in Ecology & Evolution* 25.7 (2010), pp. 410–418. URL: `https://doi.org/10.1016/j.tree.2010.04.001`.

[58] Antonio Cuevas, Manuel Febrero, and Ricardo Fraiman. "An anova test for functional data". In: *Computational statistics & data analysis* 47.1 (2004), pp. 111–122.

[59] Mary Cummings. "Automation bias in intelligent time critical decision support systems". In: *AIAA 1st intelligent systems technical conference*. 2004, p. 6313.

[60] George B Dantzig. "Linear programming". In: *Operations Research* 50.1 (2002), pp. 42–47. URL: `https://doi.org/10.1287/opre.50.1.42.17798`.

[61] Philipp Dechent, Elias Barbers, Simon Clark, Susanne Lehner, Brady Planden, Masaki Adachi, David A Howey, and Sabine Paarmann. "Demonstrating Linked Battery Data To Accelerate Knowledge Flow in Battery Science". In: *arXiv preprint arXiv:2410.23303* (2024).

[62] P Diaconis. *Bayesian numerical analysis. Statistical Decision Theory and Related Topics IV, 1.(SS Gupta and J. Berger, eds.)* 1988.

[63] Laurence Charles Ward Dixon. "The global optimization problem: an introduction". In: *Towards Global Optimiation 2* (1978), pp. 1–15.

[64] Nathan Doumèche, Francis Bach, Gérard Biau, and Claire Boyer. "Physics-informed kernel learning". In: *arXiv preprint arXiv:2409.13786* (2024).

[65] Petros Drineas and Michael W. Mahoney. "On the Nyström Method for Approximating a Gram Matrix for Improved Kernel-Based Learning". In: *Journal of Machine Learning Research* 6.72 (2005), pp. 2153–2175. URL: http://jmlr.org/papers/v6/drineas05a.html.

[66] David Duvenaud. "Automatic model construction with Gaussian processes". PhD thesis. 2014.

[67] Raaz Dwivedi and Lester Mackey. "Kernel Thinning". In: *Annual Conference on Learning Theory (COLT)*. Vol. 134. 2021, pp. 1753–1753. URL: http://proceedings.mlr.press/v134/dwivedi21a/dwivedi21a.pdf.

[68] David Edwards. *Introduction to graphical modelling*. Springer Science & Business Media, 2000.

[69] Albert Einstein. *The ultimate quotable Einstein*. Princeton University Press, 2011.

[70] Ethan N Epperly and Elvira Moreno. "Kernel Quadrature with Randomly Pivoted Cholesky". In: *arXiv preprint arXiv:2306.03955* (2023).

[71] F. Feroz, M. P. Hobson, and M. Bridges. "MultiNest: an efficient and robust Bayesian inference tool for cosmology and particle physics". In: *Monthly Notices of the Royal Astronomical Society* 398.4 (2009), pp. 1601–1614.

[72] Peter C Fishburn. "Utility theory". In: *Management science* 14.5 (1968), pp. 335–378. URL: https://doi.org/10.1287/mnsc.14.5.335.

[73] Peter I Frazier. "Knowledge-gradient methods for statistical learning". PhD thesis. Princeton University Princeton, 2009.

[74] David Freedman. "Wald Lecture: On the Bernstein-von Mises theorem with infinite-dimensional parameters". In: *The Annals of Statistics* 27.4 (1999), pp. 1119–1141.

[75] Jerome H Friedman. "Greedy function approximation: a gradient boosting machine". In: *Annals of statistics* (2001), pp. 1189–1232.

[76] Masahiro Fujisawa, Masaki Adachi, and Michael A Osborne. "Scalable Valuation of Human Feedback through Provably Robust Model Alignment". In: *arXiv preprint arXiv:2505.17859* (2025).

[77] Masahiro Fujisawa, Takeshi Teshima, Issei Sato, and Masashi Sugiyama. "γ-ABC: Outlier-robust approximate Bayesian computation based on a robust divergence estimator". In: *International Conference on Artificial Intelligence and Statistics (AISTATS)*. Vol. 130. PMLR. 2021, pp. 1783–1791. URL: http://proceedings.mlr.press/v130/fujisawa21a/fujisawa21a.pdf.

[78] Johannes Fürnkranz and Eyke Hüllermeier. "Preference learning and ranking by pairwise comparison". In: *Preference learning*. Springer, 2010, pp. 65–82.

[79] Alex Galt. *Unlocking A World Of Possibilities: How AI Is Revolutionizing Real Estate And Empowering Your Life Choices.* 2023. URL: https://www.forbes.com/councils/forbesbusinesscouncil/2023/12/18/unlocking-a-world-of-possibilities-how-ai-is-revolutionizing-real-estate-and-empowering-your-life-choices/.

[80] Tianning Gao, Yifan Wang, Ming Zhu, Xiulong Wu, Dian Zhou, and Zhaori Bi. "An RISC-V PPA-Fusion Cooperative Optimization Framework Based on Hybrid Strategies". In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* (2024).

[81] Roman Garnett. *Bayesian Optimization*. Cambridge University Press, 2023.

[82] Alexandra Gessner, Javier Gonzalez, and Maren Mahsereci. "Active multi-information source Bayesian quadrature". In: *Uncertainty in Artificial Intelligence*. PMLR. 2020, pp. 712–721.

[83] Charles J Geyer. "Practical Markov chain Monte Carlo". In: *Statistical science* (1992), pp. 473–483.

[84] Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. "Automatic chemical design using a data-driven continuous representation of molecules". In: *ACS Central Science* 4.2 (2018), pp. 268–276. URL: https://doi.org/10.1021/acscentsci.7b00572.

[85] Javier González, Zhenwen Dai, Philipp Hennig, and Neil Lawrence. "Batch Bayesian optimization via local penalization". In: *International Conference on Artificial Intelligence and Statistics (AISTATS)*. Vol. 51. PMLR. 2016, pp. 648–657.

[86] Christian Gourieroux, Alain Monfort, and Eric Renault. "Indirect inference". In: *Journal of Applied Econometrics* 8.S1 (1993), S85–S118. URL: https://doi.org/10.1002/jae.3950080507.

[87] Siegfried Graf and Harald Luschgy. *Foundations of quantization for probability distributions*. Springer, 2007.

[88] Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex Smola. "A kernel method for the two-sample-problem". In: *Advances in neural information processing systems* 19 (2006).

[89] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. "A kernel two-sample test". In: *The Journal of Machine Learning Research* 13.1 (2012), pp. 723–773.

[90] Antoine Grosnit, Cedric Malherbe, Rasul Tutunov, Xingchen Wan, Jun Wang, and Haitham Bou Ammar. "Boils: Bayesian optimisation for logic synthesis". In: *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE. 2022, pp. 1193–1196.

[91] Julia Grosse, Rahel Fischer, Roman Garnett, and Philipp Hennig. "A Greedy Approximation for k-Determinantal Point Processes". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2024, pp. 3052–3060.

[92] T. Gunter, M. A. Osborne, R. Garnett, P. Hennig, and S. J. Roberts. "Sampling for Inference in Probabilistic Models with Fast Bayesian Quadrature". In: *International Conference on Neural Information Processing Systems (NeurIPS)*. 2014. URL: https://proceedings.neurips.cc/paper/2014/file/e94f63f579e05cb49c05c2d050ead9c0-Paper.pdf.

[93] Michael U Gutmann and Jun-ichiro Hirayama. "Bregman divergence as general framework to estimate unnormalized statistical models". In: *International Conference on Uncertainty in Artificial Intelligence (UAI)*. 2011, pp. 283–290. URL: https://doi.org/10.48550/arXiv.1202.3727.

[94] Michael U. Gutmann and Jukka Corander. "Bayesian optimization for likelihood-free inference of simulator-based statistical models". In: *Journal of Machine Learning Research* 17.125 (2016), pp. 1–47. URL: http://jmlr.org/papers/v17/15-017.html.

[95] S. Hayakawa, H. Oberhauser, and T. Lyons. "Positively Weighted Kernel Quadrature via Subsampling". In: *International Conference on Neural Information Processing Systems (NeurIPS)*. 2022. DOI: 10.48550/arXiv.2107.09597.

[96] Satoshi Hayakawa and Tetsuro Morimura. "Policy Gradient with Kernel Quadrature". In: *Transactions on Machine Learning Research* (2024). URL: https://openreview.net/forum?id=WFI9xhJrxF.

[97] Satoshi Hayakawa, Harald Oberhauser, and Terry Lyons. "Hypercontractivity meets random convex hulls: analysis of randomized multivariate cubatures". In: *Proceedings of the Royal Society A* 479.2273 (2023), p. 20220725. DOI: 10.1098/rspa.2022.0725. URL: https://royalsocietypublishing.org/doi/abs/10.1098/rspa.2022.0725.

[98] Satoshi Hayakawa, Harald Oberhauser, and Terry Lyons. "Positively weighted kernel quadrature via subsampling". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 35. 2022, pp. 6886–6900. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/2dae7d1ccf1edf76f8ce7c282bdf4730-Paper-Conference.pdf.

[99] Satoshi Hayakawa, Harald Oberhauser, and Terry Lyons. "Sampling-based Nyström Approximation and Kernel Quadrature". In: *International Conference on Machine Learning (ICML)*. Vol. 202. 2023, pp. 12678–12699. URL: https://proceedings.mlr.press/v202/hayakawa23a/hayakawa23a.pdf.

[100] Philipp Hennig, Michael A Osborne, and Mark Girolami. "Probabilistic numerics and uncertainty in computations". In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 471.2179 (2015), p. 20150142. URL: https://doi.org/10.1098/rspa.2015.0142.

[101] Philipp Hennig, Michael A Osborne, and Hans P Kersting. *Probabilistic Numerics: Computation as Machine Learning*. Cambridge University Press, 2022.

[102] José Miguel Hernández-Lobato, Matthew W Hoffman, and Zoubin Ghahramani. "Predictive entropy search for efficient global optimization of black-box functions". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 27. 2014. URL: https://proceedings.neurips.cc/paper_files/paper/2014/file/069d3bb002acd8d7dd095917f9efe4cb-Paper.pdf.

[103] José Miguel Hernández-Lobato, James Requeima, Edward O Pyzer-Knapp, and Alán Aspuru-Guzik. "Parallel and distributed Thompson sampling for large-scale accelerated exploration of chemical space". In: *International Conference on Machine Learning (ICML)*. PMLR. 2017, pp. 1470–1479. URL: http://proceedings.mlr.press/v70/hernandez-lobato17a/hernandez-lobato17a.pdf.

[104] E. Higson, W. Handley, M. Hobson, and A. Lasenby. "Dynamic nested sampling: an improved algorithm for parameter estimation and evidence calculation". In: *Stat. Comput.* 29 (2019), pp. 891–913. URL: https://doi.org/10.1007/s11222-018-9844-0.

[105] Geoffrey E Hinton. "Training products of experts by minimizing contrastive divergence". In: *Neural Computation* 14.8 (2002), pp. 1771–1800. URL: https://doi.org/10.1162/089976602760128018.

[106] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models". In: *Advances in neural information processing systems* 33 (2020), pp. 6840–6851.

[107] Daolang Huang, Ayush Bharti, Amauri Souza, Luigi Acerbi, and Samuel Kaski. "Learning Robust Statistics for Simulation-based Inference under Model Misspecification". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 36. 2023, pp. 7289–7310. URL: https://proceedings.neurips.cc/paper_files/paper/2023/file/16c5b4102a6b6eb061e502ce6736ad8a-Paper-Conference.pdf.

[108] Jonas Hübotter, Bhavya Sukhija, Lenart Treven, Yarden As, and Andreas Krause. "Transductive active learning: Theory and applications". In: *38th Annual Conference on Neural Information Processing Systems (NeurIPS 2024)*. 2024.

[109] David Huk, Yuanhe Zhang, Mark Steel, and Ritabrata Dutta. "Quasi-Bayes meets Vines". In: *arXiv preprint arXiv:2406.12764* (2024).

[110] Ferenc Huszár. *Everything that Works Works Because it's Bayesian: Why Deep Nets Generalize?* 2017. URL: https://www.inference.vc/everything-that-works-works-because-its-bayesian-2/.

[111] Ferenc Huszár and David Duvenaud. "Optimally-weighted herding is Bayesian quadrature". In: *International Conference on Uncertainty in Artificial Intelligence (UAI)*. 2012, pp. 377–386. URL: https://doi.org/10.48550/arXiv.1204.1664.

[112] Carl Hvarfner, Danny Stoll, Artur Souza, Marius Lindauer, Frank Hutter, and Luigi Nardi. "πBO: Augmenting Acquisition Functions with User Beliefs for Bayesian Optimization". In: *International Conference on Learning Representations (ICLR)*. 2022. URL: https://doi.org/10.48550/arXiv.2204.11051.

[113] Aapo Hyvärinen. "Estimation of Non-Normalized Statistical Models by Score Matching". In: *Journal of Machine Learning Research* 6.24 (2005), pp. 695–709. URL: http://jmlr.org/papers/v6/hyvarinen05a.html.

[114] Svante Janson. *Gaussian hilbert spaces*. 129. Cambridge university press, 1997.

[115] Daniel Kahneman. "Thinking, fast and slow". In: *Farrar, Straus and Giroux* (2011).

[116] Daniel Kahneman and Amos Tversky. "On the interpretation of intuitive probability: A reply to Jonathan Cohen". In: *Cognition* 7.4 (1979), pp. 409–411.

[117] Motonobu Kanagawa and Philipp Hennig. "Convergence guarantees for adaptive Bayesian quadrature methods". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 32. 2019. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/165a59f7cf3b5c4396ba65953d679f17-Paper.pdf.

[118] Motonobu Kanagawa, Philipp Hennig, Dino Sejdinovic, and Bharath K Sriperumbudur. "Gaussian processes and kernel methods: A review on connections and equivalences". In: *arXiv preprint arXiv:1807.02582* (2018).

[119] Motonobu Kanagawa, Bharath K Sriperumbudur, and Kenji Fukumizu. "Convergence analysis of deterministic kernel-based quadrature rules in misspecified settings". In: *Foundations of Computational Mathematics* 20 (2020), pp. 155–194. URL: https://doi.org/10.1007/s10208-018-09407-7.

[120] Kirthevasan Kandasamy, Gautam Dasarathy, Jeff Schneider, and Barnabás Póczos. "Multi-fidelity bayesian optimisation with continuous approximations". In: *International conference on machine learning*. PMLR. 2017, pp. 1799–1808.

[121] Kirthevasan Kandasamy, Akshay Krishnamurthy, Jeff Schneider, and Barnabás Póczos. "Parallelised Bayesian optimisation via Thompson sampling". In: *International Conference on Artificial Intelligence and Statistics (AISTATS)*. Vol. 84. PMLR. 2018, pp. 133–142. URL: http://proceedings.mlr.press/v84/kandasamy18a/kandasamy18a.pdf.

[122] Kirthevasan Kandasamy, Jeff Schneider, and Barnabás Póczos. "High dimensional Bayesian optimisation and bandits via additive models". In: *International Conference on Machine Learning (ICML)*. PMLR. 2015, pp. 295–304. URL: http://proceedings.mlr.press/v37/kandasamy15.pdf.

[123] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. "Physics-informed machine learning". In: *Nature Reviews Physics* 3.6 (2021), pp. 422–440.

[124] Toni Karvonen and Chris J Oates. "Maximum likelihood estimation in Gaussian process regression is ill-posed". In: *Journal of Machine Learning Research* 24.120 (2023), pp. 1–47.

[125] Cari G Kaufman and Stephan R Sain. "Bayesian Functional ANOVA Modeling Using Gaussian Process Prior Distributions". In: *Bayesian Analysis* 5.1 (2010), pp. 123–150.

[126] Mohammad Emtiyaz Khan and Håvard Rue. "The Bayesian learning rule". In: *Journal of Machine Learning Research* 24.281 (2023), pp. 1–46.

[127] Mohammad Emtiyaz E Khan and Siddharth Swaroop. "Knowledge-adaptation priors". In: *Advances in neural information processing systems* 34 (2021), pp. 19757–19770.

[128] Rajiv Khanna, Been Kim, Joydeep Ghosh, and Sanmi Koyejo. "Interpreting black box predictions using fisher kernels". In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR. 2019, pp. 3382–3390.

[129] El Mahdi Khribch and Pierre Alquier. "Convergence of Statistical Estimators via Mutual Information Bounds". In: *arXiv preprint arXiv:2412.18539* (2024).

[130] Franz J Király and Harald Oberhauser. "Kernels for sequentially ordered data". In: *Journal of Machine Learning Research* 20.31 (2019), pp. 1–45.

[131] Andreas Kirsch, Joost Van Amersfoort, and Yarin Gal. "BatchBALD: Efficient and diverse batch acquisition for deep Bayesian active learning". In: *Advances in neural information processing systems* 32 (2019).

[132] Genshiro Kitagawa. "A Monte Carlo filtering and smoothing method for non-Gaussian nonlinear state space models". In: *Proceedings of the 2nd US-Japan Joint Seminar on Statistical Time Series Analysis*. Vol. 110. 1993. URL: https://www.ism.ac.jp/~kitagawa/1993_US-Japan.pdf.

[133] Chun-Wa Ko, Jon Lee, and Maurice Queyranne. "An exact algorithm for maximum entropy sampling". In: *Operations Research* 43.4 (1995), pp. 684–691.

[134] Marcel Kollovieh, Abdul Fatir Ansari, Michael Bohlke-Schneider, Jasper Zschiegner, Hao Wang, and Yuyang Bernie Wang. "Predict, refine, synthesize: Self-guiding diffusion models for probabilistic time series forecasting". In: *Advances in Neural Information Processing Systems* 36 (2024).

[135] Nicholas Krämer, Nathanael Bosch, Jonathan Schmidt, and Philipp Hennig. "Probabilistic ODE solutions in millions of dimensions". In: *International Conference on Machine Learning*. PMLR. 2022, pp. 11634–11649.

[136] Andreas Krause and Carlos E Guestrin. "Near-optimal nonmyopic value of information in graphical models". In: *International Conference on Uncertainty in Artificial Intelligence (AISTATS)*. 2012, pp. 324–331. URL: https://doi.org/10.48550/arXiv.1207.1394.

[137] Agustinus Kristiadi, Felix Strieth-Kalthoff, Sriram Ganapathi Subramanian, Vincent Fortuin, Pascal Poupart, and Geoff Pleiss. "How Useful is Intermittent, Asynchronous Expert Feedback for Bayesian Optimization?" In: *Sixth Symposium on Advances in Approximate Bayesian Inference-Non Archival Track*. 2024.

[138] Yannick Kuhn, Masaki Adachi, Michael A. Osborne, David A. Howey, Arnulf Latz, and Birger Horstmann. "A Primer on Bayesian Parameter Estimation and Model Selection for Battery Simulators". In: *arXiv preprint* (2024).

[139] Alex Kulesza, Ben Taskar, et al. "Determinantal point processes for machine learning". In: *Foundations and Trends® in Machine Learning* 5.2–3 (2012), pp. 123–286.

[140] I Elizabeth Kumar, Suresh Venkatasubramanian, Carlos Scheidegger, and Sorelle Friedler. "Problems with Shapley-value-based explanations as feature importance measures". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 5491–5500.

[141] Indra Kumar, Carlos Scheidegger, Suresh Venkatasubramanian, and Sorelle Friedler. "Shapley Residuals: Quantifying the limits of the Shapley value for explanations". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 26598–26608.

[142] Sanjiv Kumar, Mehryar Mohri, and Ameet Talwalkar. "Sampling methods for the Nyström method". In: *Journal of Machine Learning Research* 13.34 (2012), pp. 981–1006. URL: http://jmlr.org/papers/v13/kumar12a.html.

[143] Harold J Kushner. "A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise". In: *Journal of Basic Engineering* 86 (1964), pp. 97–106. URL: https://doi.org/10.1115/1.3653121.

[144] FM Larkin. "Probabilistic Error Estimates in Spline Interpolation and Quadrature." In: *IFIP Congress*. 1974, pp. 605–609.

[145] Darrick Lee and Harald Oberhauser. "The Signature Kernel". In: *arXiv preprint arXiv:2305.04625* (2023).

[146] Hyun-Suk Lee. "Automated tariff design for energy supply–demand matching based on Bayesian optimization: Technical framework and policy implications". In: *Energy Policy* 188 (2024), p. 114102.

[147] Noureddine Lehdili, Pascal Oswald, and Othmane Mirinioui. "Leveraging Bayesian Quadrature for Accurate and Fast Credit Valuation Adjustment Calculations". In: *Mathematics* 12.23 (2024), p. 3779.

[148] Yuxin Lin, Masaki Adachi, Silvia Spezzano, Gordian Edenhofer, Vincent Eberle, Michael A. Osborne, and Paola Caselli. "Fast Broad-band Line-rich Spectralcube Fitting and Image Visualisation via Bayesian Quadrature". In: *arXiv preprint* (2024).

[149] C. Litterer and T. Lyons. "High order recombination and an application to cubature on Wiener space". In: *The Annals of Applied Probability* 22.4 (2012), pp. 1301–1327.

[150] Qinghua Liu, Praneeth Netrapalli, Csaba Szepesvari, and Chi Jin. "Optimistic mle: A generic model-based algorithm for partially observable sequential decision making". In: *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*. 2023, pp. 363–376.

[151] Tennison Liu, Nicolás Astorga, Nabeel Seedat, and Mihaela van der Schaar. "Large Language Models to Enhance Bayesian Optimization". In: *The Twelfth International Conference on Learning Representations*. 2024. URL: https://openreview.net/forum?id=OOxotBmGol.

[152] Pierre Llompart, Kwame Amaning, Marc Bianciotto, Bruno Filoche-Rommé, Yann Foricher, Pablo Mas, David Papin, Jean-Philippe Rameau, Laurent Schio, Gilles Marcou, and et al. "Harnessing Medicinal Chemical Intuition from Collective Intelligence". In: *ChemRxiv* (2024). DOI: 10.26434/chemrxiv-2024-0hww3.

[153] Scott M Lundberg and Su-In Lee. "A unified approach to interpreting model predictions". In: *Advances in neural information processing systems*. 2017, pp. 4765–4774.

[154] Terry Lyons and Nicolas Victoir. "Cubature on Wiener space". In: *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 460.2041 (2004), pp. 169–198.

[155] David JC MacKay. "Information-based objective functions for active data selection". In: *Neural computation* 4.4 (1992), pp. 590–604.

[156] Wesley J Maddox, Maximilian Balandat, Andrew G Wilson, and Eytan Bakshy. "Bayesian optimization with high-dimensional outputs". In: *Advances in neural information processing systems* 34 (2021), pp. 19274–19287.

[157] Ulysse Marteau-Ferey, Francis Bach, and Alessandro Rudi. "Sampling from arbitrary functions via psd models". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2022, pp. 2823–2861.

[158] Petrus Mikkola, Osvaldo A Martin, Suyog Chandramouli, Marcelo Hartmann, Oriol Abril Pla, Owen Thomas, Henri Pesonen, Jukka Corander, Aki Vehtari, Samuel Kaski, et al. "Prior knowledge elicitation: The past, present, and future". In: *Bayesian Analysis* 19.4 (2024), pp. 1129–1161.

[159] Petrus Mikkola, Julien Martinelli, Louis Filstroff, and Samuel Kaski. "Multi-Fidelity Bayesian Optimization with Unreliable Information Sources". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2023, pp. 7425–7454.

[160] Petrus Mikkola, Milica Todorović, Jari Järvi, Patrick Rinke, and Samuel Kaski. "Projective preferential bayesian optimization". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 6884–6892.

[161] Ha Quang Minh, Partha Niyogi, and Yuan Yao. "Mercer's theorem, feature maps, and smoothing". In: *International Conference on Computational Learning Theory*. Springer. 2006, pp. 154–168.

[162] Jonas Mockus. "On the Bayes Methods for Seeking the Extremal Point". In: *IFAC Proceedings Volumes* 8.1, Part 1 (1975), pp. 428–431. DOI: https://doi.org/10.1016/S1474-6670(17)67769-3.

[163] Philippe Mongin. *Expected utility theory*. 1998.

[164] Krikamol Muandet. "Impossibility of collective intelligence". In: *arXiv preprint arXiv:2206.02786* (2022).

[165] Krikamol Muandet, Kenji Fukumizu, Bharath Sriperumbudur, Bernhard Schölkopf, et al. "Kernel mean embedding of distributions: A review and beyond". In: *Foundations and Trends® in Machine Learning* 10.1-2 (2017), pp. 1–141.

[166] Alfred Müller. "Integral probability metrics and their generating classes of functions". In: *Advances in applied probability* 29.2 (1997), pp. 429–443.

[167] Jay I Myung, Daniel R Cavagnaro, and Mark A Pitt. "A tutorial on adaptive design optimization". In: *Journal of mathematical psychology* 57.3-4 (2013), pp. 53–67.

[168] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. "An analysis of approximations for maximizing submodular set functions—I". In: *Mathematical Programming* 14 (1978), pp. 265–294. URL: https://doi.org/10.1007/BF01588971.

[169] Patrick Ngatchou, Anahita Zarei, and A El-Sharkawi. "Pareto multi objective optimization". In: *Proceedings of the 13th international conference on, intelligent systems application to power systems*. IEEE. 2005, pp. 84–91.

[170] Anthony O'Hagan. "Bayes–Hermite quadrature". In: *Journal of Statistical Planning and Inference* 29.3 (1991), pp. 245–260. URL: https://doi.org/10.1016/0378-3758(91)90002-V.

[171] Anthony O'Hagan. "Monte Carlo is fundamentally unsound". In: *The Statistician* (1987), pp. 247–249.

[172] Anthony O'Hagan, Caitlin E Buck, Alireza Daneshkhah, J Richard Eiser, Paul H Garthwaite, David J Jenkinson, Jeremy E Oakley, and Tim Rakow. "Uncertain judgements: eliciting experts' probabilities". In: (2006).

[173] Michael Osborne, Roman Garnett, Zoubin Ghahramani, David K Duvenaud, Stephen J Roberts, and Carl Rasmussen. "Active learning of model evidence using Bayesian quadrature". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 25. 2012. URL: https://proceedings.neurips.cc/paper_files/paper/2012/file/6364d3f0f495b6ab9dcf8d3b5c6e0b01-Paper.pdf.

[174] Michael A Osborne, Roman Garnett, and Stephen J Roberts. "Gaussian processes for global optimization". In: *International Conference on Learning and Intelligent Optimization (LION3)*. 2009. URL: https://ora.ox.ac.uk/objects/uuid:7d2b38d0-43be-4bb4-852c-50001a28ead9/files/sq237ht37z.

[175] Pierre Osselin, Masaki Adachi, Xiaowen Dong, and Michael A. Osborne. "Natural Evolutionary Search meets Probabilistic Numerics". In: *arXiv preprint* (2024).

[176] Art Owen. "Empirical likelihood ratio confidence regions". In: *The annals of statistics* 18.1 (1990), pp. 90–120.

[177] Pingbo Pan, Siddharth Swaroop, Alexander Immer, Runa Eschenhagen, Richard Turner, and Mohammad Emtiyaz E Khan. "Continual deep learning by functional regularisation of memorable past". In: *Advances in neural information processing systems* 33 (2020), pp. 4453–4464.

[178] Hannu Parviainen. "Bayesian Methods for Exoplanet Science". In: *Handbook of Exoplanets* (2017), pp. 1–24.

[179] Micha CJ Philipp, Yannick Kuhn, Arnulf Latz, and Birger Horstmann. "Physics-based inverse modeling of battery degradation with Bayesian methods". In: *arXiv preprint arXiv:2410.19478* (2024).

[180] Dan Pilat and Sekoul Krastev. *Amos Tversky*. 2025. URL: https://thedecisionlab.com/thinkers/economics/amos-tversky.

[181] Robert Pinsler, Jonathan Gordon, Eric Nalisnick, and José Miguel Hernández-Lobato. "Bayesian batch active learning as sparse subset approximation". In: *Advances in neural information processing systems* 32 (2019).

[182] Leah F Price, Christopher C Drovandi, Anthony Lee, and David J Nott. "Bayesian synthetic likelihood". In: *Journal of Computational and Graphical Statistics* 27.1 (2018), pp. 1–11. URL: https://doi.org/10.1080/10618600.2017.1302882.

[183] J. Prüher and M. Šimandl. "Bayesian quadrature in nonlinear filtering". In: *International Conference on Informatics in Control, Automation and Robotics (ICINCO)*. 2015. URL: https://doi.org/10.5220/0005534003800387.

[184] Ali Rahimi, Benjamin Recht, et al. "Random Features for Large-Scale Kernel Machines." In: *NIPS*. Vol. 3. 4. Citeseer. 2007, p. 5.

[185] Tom Rainforth, Adam Foster, Desi R Ivanova, and Freddie Bickford Smith. "Modern Bayesian experimental design". In: *Statistical Science* 39.1 (2024), pp. 100–114.

[186] Carl Edward Rasmussen and Zoubin Ghahramani. "Bayesian Monte Carlo". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 15. 2002, pp. 505–512. URL: https://proceedings.neurips.cc/paper_files/paper/2002/file/24917db15c4e37e421866448c9ab23d8-Paper.pdf.

[187] Carl Edward Rasmussen, Christopher KI Williams, et al. *Gaussian processes for machine learning*. Vol. 1. Springer, 2006.

[188] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. ""Why should i trust you?" Explaining the predictions of any classifier". In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2016, pp. 1135–1144.

[189] Klaus Ritter. *Average-case analysis of numerical problems*. 1733. Springer Science & Business Media, 2000.

[190] Ryan Roussel, Juan Pablo Gonzalez-Aguilera, Young-Kee Kim, Eric Wisniewski, Wanming Liu, Philippe Piot, John Power, Adi Hanuka, and Auralee Edelen. "Turn-key constrained parameter space exploration for particle accelerators using Bayesian active learning". In: *Nature communications* 12.1 (2021), p. 5612.

[191] Alessandro Rudi, Ulysse Marteau-Ferey, and Francis Bach. "Finding global minima via kernel approximations". In: *arXiv preprint arXiv:2012.11978* (2020). URL: https://doi.org/10.48550/arXiv.2012.11978.

[192] Simo Särkkä and Lennart Svensson. *Bayesian filtering and smoothing*. Vol. 17. Cambridge university press, 2023.

[193] Joachim Schaeffer, Paul Gasper, Esteban Garcia-Tamayo, Raymond Gasper, Masaki Adachi, Juan Pablo Gaviria-Cardona, Simon Montoya-Bedoya, Anoushka Bhutani, Andrew Schiek, Rhys Goodall, Rolf Findeisen, Richard D. Braatz, and Simon Engelke. "Machine Learning Benchmarks for the Classification of Equivalent Circuit Models from Electrochemical Impedance Spectra". In: *Journal of The Electrochemical Society* 170.6 (2023), p. 060512. DOI: 10.1149/1945-7111/acd8fb.

[194] David W Scott. "Box–muller transformation". In: *Wiley Interdisciplinary Reviews: Computational Statistics* 3.2 (2011), pp. 177–179.

[195] Matthias W Seeger, Sham M Kakade, and Dean P Foster. "Information consistency of nonparametric Gaussian process methods". In: *IEEE Transactions on Information Theory* 54.5 (2008), pp. 2376–2382.

[196] Amartya Sen. "Social choice theory: A re-examination". In: *Econometrica: journal of the Econometric Society* (1977), pp. 53–89.

[197] Ozan Sener and Silvio Savarese. "Active Learning for Convolutional Neural Networks: A Core-Set Approach". In: *International Conference on Learning Representations*. 2018.

[198] Kristen A Severson, Peter M Attia, Norman Jin, Nicholas Perkins, Benben Jiang, Zi Yang, Michael H Chen, Muratahan Aykol, Patrick K Herring, Dimitrios Fraggedakis, et al. "Data-driven prediction of battery cycle life before capacity degradation". In: *Nature Energy* 4.5 (2019), pp. 383–391.

[199] Lloyd S Shapley. "A value for n-person games". In: *Contributions to the Theory of Games* 2.28 (1953), pp. 307–317.

[200] Yuesong Shen, Nico Daheim, Bai Cong, Peter Nickl, Gian Maria Marconi, Bazan Clement Emile Marcel Raoul, Rio Yokota, Iryna Gurevych, Daniel Cremers, Mohammad Emtiyaz Khan, and Thomas Möllenhoff. "Variational Learning is Effective for Large Deep Networks". In: *Forty-first International Conference on Machine Learning*. 2024. URL: https://openreview.net/forum?id=cXBv07GKvk.

[201] Herbert A Simon et al. "Invariants of human behavior". In: *Annual review of psychology* 41.1 (1990), pp. 1–20.

[202] Joshua W Sin, Siu Lun Chau, Ryan P Burwood, Kurt Püntener, Raphael Bigler, and Philippe Schwaller. "Highly Parallel Optimisation of Nickel-Catalysed Suzuki Reactions through Automation and Machine Intelligence". In: (2024).

[203] J. Skilling. "Nested sampling for general Bayesian computation". In: *Bayesian Analysis* 1.4 (2006), pp. 833–859. URL: https://doi.org/10.1214/06-BA127.

[204] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. "Gaussian process optimization in the bandit setting: No regret and experimental design". In: *International Conference on Machine Learning (ICML)*. 2010, pp. 1015–1022. URL: https://icml.cc/Conferences/2010/papers/422.pdf.

[205] Michael L Stein. *Interpolation of spatial data*. Springer Science & Business Media, 1999.

[206] Arthur H Stroud. *Approximate calculation of multiple integrals*. Prentice Hall, 1971.

[207] Mukund Sundararajan and Amir Najmi. "The many Shapley values for model explanation". In: *International conference on machine learning*. PMLR. 2020, pp. 9269–9278.

[208] Christian Szegedy. *Basically every problem can be formulated as compression by trying to compress (task, solution) pairs.* 2022. URL: https://x.com/ChrSzegedy/status/1563808001227571200.

[209] Vladimir Tchakaloff. "Formules de cubatures mécaniques à coefficients non négatifs". In: *Bull. Sci. Math* 81.2 (1957), pp. 123–134.

[210] Maria Tchernychova. "Carathéodory cubature measures". PhD thesis. University of Oxford, 2015.

[211] George R Terrell and David W Scott. "Variable kernel density estimation". In: *The Annals of Statistics* (1992), pp. 1236–1265.

[212] Onur Teymur, Jackson Gorham, Marina Riabiz, and Chris Oates. "Optimal quantisation of probability measures using maximum mean discrepancy". In: *International Conference on Artificial Intelligence and Statistics (AISTATS)*. Vol. 130. PMLR. 2021, pp. 1027–1035. URL: http://proceedings.mlr.press/v130/teymur21a/teymur21a.pdf.

[213] Richard H Thaler and Cass R Sunstein. *Nudge: The final edition*. Yale University Press, 2021.

[214] Daphne Theodorakopoulos, Frederic Stahl, and Marius Lindauer. "Hyperparameter Importance Analysis for Multi-Objective AutoML". In: *arXiv preprint arXiv:2405.07640* (2024).

[215] William R Thompson. "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples". In: *Biometrika* 25.3-4 (1933), pp. 285–294. URL: https://doi.org/10.1093/biomet/25.3-4.285.

[216] Csaba Toth and Harald Oberhauser. "Bayesian learning from sequential data using gaussian processes with signature covariances". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 9548–9560.

[217] Csaba Tóth, Masaki Adachi, Michael A Osborne, and Harald Oberhauser. "Learning to Forget: Bayesian Time Series Forecasting using Recurrent Sparse Spectrum Signature Gaussian Processes". In: *International Conference on Artificial Intelligence and Statistics (AISTATS)*. Ed. by Yingzhen Li, Stephan Mandt, Shipra Agrawal, and Emtiyaz Khan. Vol. 258. Proceedings of Machine Learning Research. PMLR, 2025, pp. 4654–4662. URL: https://proceedings.mlr.press/v258/toth25b.html.

[218] Alexandra Tremayne-Pengelly. *Ilya Sutskever Warns A.I. Is Running Out of Data—Here's What Will Happen Next*. 2024. URL: https://observer.com/2024/12/openai-cofounder-ilya-sutskever-ai-data-peak/.

[219] Sattar Vakili, Kia Khezeli, and Victor Picheny. "On information gain and regret bounds in Gaussian process bandits". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2021, pp. 82–90.

[220] Lieven Vandenberghe and Stephen Boyd. "Semidefinite programming". In: *SIAM review* 38.1 (1996), pp. 49–95. URL: https://doi.org/10.1137/1038003.

[221] A Vaswani. "Attention is all you need". In: *Advances in Neural Information Processing Systems* (2017).

[222] John Von Neumann and Oskar Morgenstern. *Theory of games and economic behavior*. Princeton university press, 1947.

[223] Sandra Wachter, Brent Mittelstadt, and Chris Russell. "Counterfactual explanations without opening the black box: Automated decisions and the GDPR". In: *Harv. JL & Tech.* 31 (2017), p. 841.

[224] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. "Tent: Fully test-time adaptation by entropy minimization". In: *arXiv preprint arXiv:2006.10726* (2020).

[225] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. "Continual test-time domain adaptation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 7201–7211.

[226] Zi Wang and Stefanie Jegelka. "Max-value entropy search for efficient Bayesian optimization". In: *International Conference on Machine Learning (ICML)*. Vol. 70. PMLR. 2017, pp. 3627–3635. URL: http://proceedings.mlr.press/v70/wang17e/wang17e.pdf.

[227] Pengfei Weia, Masaru Kitaharac, Matthias GR Faesd, and Michael Beere. "Probabilistic Calibration of Model Parameters with Approximate Bayesian Quadrature and Active Machine Learning". In: *Journal of Reliability Science and Engineering* (2024).

[228] Jonathan Wenger, Geoff Pleiss, Marvin Pförtner, Philipp Hennig, and John P Cunningham. "Posterior and computational uncertainty in Gaussian processes". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 10876–10890.

[229] Daan Wierstra, Tom Schaul, Tobias Glasmachers, Yi Sun, Jan Peters, and Jürgen Schmidhuber. "Natural evolution strategies". In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 949–980.

[230] Veit David Wild, Sahra Ghalebikesabi, Dino Sejdinovic, and Jeremias Knoblauch. "A rigorous link between deep ensembles and (variational) Bayesian methods". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 36. 2023, pp. 39782–39811. URL: https://proceedings.neurips.cc/paper_files/paper/2023/file/7d25b1db211d99d5750ec45d65fd6e4e-Paper-Conference.pdf.

[231] Christopher Williams and Matthias Seeger. "Using the Nyström method to speed up kernel machines". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 13. 2000. URL: https://proceedings.neurips.cc/paper/2000/file/19de10adbaa1b2ee13f77f679fa1483a-Paper.pdf.

[232] Sinead A Williamson and Jette Henderson. "Understanding collections of related datasets using dependent MMD coresets". In: *Information* 12.10 (2021), p. 392.

[233] Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing. "Deep kernel learning". In: *Artificial intelligence and statistics*. PMLR. 2016, pp. 370–378.

[234] Robert L Winkler. "The assessment of prior distributions in Bayesian analysis". In: *Journal of the American Statistical association* 62.319 (1967), pp. 776–800.

[235] Simon N Wood. "Statistical inference for noisy nonlinear ecological dynamic systems". In: *Nature* 466.7310 (2010), pp. 1102–1104. URL: https://doi.org/10.1038/nature09319.

[236] Stephen J Wright. *Primal-dual interior-point methods*. SIAM, 1997.

[237] Kaiwen Wu, Jonathan Wenger, Haydn T Jones, Geoff Pleiss, and Jacob Gardner. "Large-scale gaussian processes via alternating projection". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2024, pp. 2620–2628.

[238] Xiaoyue Xi, François-Xavier Briol, and Mark Girolami. "Bayesian quadrature for multiple related integrals". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 5373–5382.

[239] Wenjie Xu, Masaki Adachi, Colin N. Jones, and Michael A. Osborne. "Principled Bayesian Optimization in Collaboration with Human Experts". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Ed. by A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang. Vol. 37. Curran Associates, Inc., 2024, pp. 104091–104137. URL: https://proceedings.neurips.cc/paper_files/paper/2024/file/bc82dbfbfa43232be85b8d9838f49c3e-Paper-Conference.pdf.

[240] Wenjie Xu, Wenbin Wang, Yuning Jiang, Bratislav Svetozarevic, and Colin Jones. "Principled Preferential Bayesian Optimization". In: *Proceedings of the 41st International Conference on Machine Learning*. Vol. 235. Proceedings of Machine Learning Research. 2024, pp. 55305–55336. URL: https://proceedings.mlr.press/v235/xu24y.html.

[241] Yang, Duraiswami, and Gumerov. "Improved fast gauss transform and efficient kernel density estimation". In: *Proceedings ninth IEEE international conference on computer vision*. IEEE. 2003, pp. 664–671.

[242] Hector Zenil. "Compression is comprehension and the unreasonable effectiveness of digital computation in the natural world". In: *UNRAVELLING COMPLEXITY: The Life and Work of Gregory Chaitin*. World Scientific, 2020, pp. 201–238.

[243] Ding-Xuan Zhou. "The covering number in learning theory". In: *Journal of Complexity* 18.3 (2002), pp. 739–767.

[244] Xian Zhou, Suyu Liu, Edward S Kim, Roy S Herbst, and J Jack Lee. "Bayesian adaptive design for targeted therapy development in lung cancer—a step toward personalized medicine". In: *Clinical Trials* 5.3 (2008), pp. 181–193.

[245] Ji Zhu and Trevor Hastie. "Kernel logistic regression and the import vector machine". In: *Journal of Computational and Graphical Statistics* 14.1 (2005), pp. 185–205.

[246] Juliusz Ziomek, Masaki Adachi, and Michael A Osborne. "Time-varying Gaussian Process Bandits with Unknown Prior". In: *International Conference on Artificial Intelligence and Statistics (AISTATS)*. Ed. by Yingzhen Li, Stephan Mandt, Shipra Agrawal, and Emtiyaz Khan. Vol. 258. Proceedings of Machine Learning Research. PMLR, 2025, pp. 4294–4302. URL: https://proceedings.mlr.press/v258/ziomek25a.html.

[247] Juliusz Ziomek, Masaki Adachi, and Michael A. Osborne. "Bayesian Optimisation with Unknown Hyperparameters: Regret Bounds Logarithmically Closer to Optimal". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Ed. by A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang. Vol. 37. Curran Associates, Inc., 2024, pp. 86346–86374. URL: https://proceedings.neurips.cc/paper_files/paper/2024/file/9cf904c86cc5f9ac95646c07d2cfa241-Paper-Conference.pdf.

Appendices

A

Appendix of Chapter 3

Supplementary: Fast Bayesian Inference with Batch Bayesian Quadrature via Kernel Recombination

1 Convergence analysis

1.1 Proof of Theorem 1

We provide the proof of the following theorem given in the main text.

Theorem 1. *Suppose $\int \sqrt{K(x,x)}f(x)\,\mathrm{d}x < \infty$, $\ell \sim \mathcal{GP}(m, K)$, and we are given an $(n-1)$-dimensional kernel K_0 such that $K_1 := K - K_0$ is also a kernel. Let (f, g) be a density pair with weight λ. Let $\mathbf{x}_{\text{rec}}$ be an N-point independent sample from g and $\mathbf{w}_{\text{rec}} := \lambda(\mathbf{x}_{\text{rec}})$. Then, if $(\mathbf{w}_{\text{quad}}, \mathbf{x}_{\text{quad}})$ is a proper recombination of $(\mathbf{w}_{\text{rec}}, \mathbf{x}_{\text{rec}})$ for K_0, it satisfies*

$$\mathbb{E}_{\mathbf{x}_{\text{rec}}}\left[\sqrt{\text{var}[Z_f \mid \mathbf{x}_{\text{quad}}]}\right] \leq 2\left(\int K_1(x,x)f(x)\,\mathrm{d}x\right)^{1/2} + \sqrt{\frac{C_{K,f,g}}{N}} \tag{1}$$

where $Z_f := \int \ell(x)f(x)\,\mathrm{d}x$ and $C_{K,f,g} := \int K(x,x)\lambda(x)f(x)\,\mathrm{d}x - \iint K(x,y)f(x)f(y)\,\mathrm{d}x\,\mathrm{d}y$.

Recall $\mathcal{H}$ is the RKHS given by the kernel K. As the kernel satisfies $\int \sqrt{K(x,x)}f(x)\,\mathrm{d}x < \infty$, the mean embedding

$$\mu_K(f) := \int f(x)K(x,\cdot)\,\mathrm{d}x \tag{2}$$

is a well-defined element of $\mathcal{H}$. We first discuss its approximation via importance sampling.

Lemma 1. *Let f be a probability density on $\mathbb{R}^d$ and g be another density such that $f = \lambda g$ with a nonnegative function λ. Let $\mathbf{x}_{\text{rec}}$ be an N-point independent sample from g and $\mathbf{w}_{\text{rec}} = \lambda(\mathbf{x}_{\text{rec}})$ be the weights. If we define $\mu_r := \frac{1}{N}\mathbf{w}_{\text{rec}}^\top K(\mathbf{x}_{\text{rec}}, \cdot)$ then it satisfies*

$$\mathbb{E}[\|\mu_K(f) - \mu_r\|_{\mathcal{H}}^2] = \tfrac{1}{N}C_{K,f,g}$$
$$\text{where} \quad C_{K,f,g} = \int K(x,x)\lambda(x)f(x)\,\mathrm{d}x - \iint K(x,y)f(x)f(y)\,\mathrm{d}x\,\mathrm{d}y$$

Furthermore, the choice $g(x) \propto \sqrt{K(x,x)}f(x)$ minimises $C_{K,f,g}$, if $\lambda = K(x,x)^{-1/2}$ is well-defined.

Proof. Let $\mathbf{x}_{\text{rec}} = (X_1, \ldots, X_N)$, so $\mu_r = \frac{1}{N}\sum_{i=1}^N \lambda(X_i)K(X_i,\cdot)$. From (2), we have

$$\|\mu_K(f) - \mu_r\|_{\mathcal{H}}^2 = \|\mu_K(f)\|_{\mathcal{H}}^2 - 2\langle \mu_K(f), \mu_r\rangle_{\mathcal{H}} + \|\mu_r\|_{\mathcal{H}}^2 \tag{4}$$

$$= \iint K(x,y)f(x)f(y)\,\mathrm{d}x\,\mathrm{d}y - \frac{2}{N}\sum_{i=1}^N \int K(x,X_i)f(x)\lambda(X_i)\,\mathrm{d}x \tag{5}$$

$$+ \frac{1}{N^2}\sum_{i,j=1}^N K(X_i,X_j)\lambda(X_i)\lambda(X_j). \tag{6}$$

We have

$$\mathbb{E}\left[\int K(x,X_i)f(x)\lambda(X_i)\,\mathrm{d}x\right] = \iint K(x,y)f(x)\lambda(y)g(y)\,\mathrm{d}x\,\mathrm{d}y = \iint K(x,y)f(x)f(y)\,\mathrm{d}x\,\mathrm{d}y \tag{7}$$

and for $i \neq j$

$$\mathbb{E}[K(X_i,X_j)\lambda(X_i)\lambda(X_j)] = \iint K(x,y)\lambda(x)\lambda(y)g(x)g(y)\,\mathrm{d}x\,\mathrm{d}y = \iint K(x,y)f(x)f(y)\,\mathrm{d}x\,\mathrm{d}y, \tag{8}$$

so we in total have

$$\mathbb{E}[\|\mu_K(f) - \mu_r\|_{\mathcal{H}}^2] = \frac{1}{N^2}\sum_{i=1}^{N}\mathbb{E}[K(X_i, X_i)\lambda(X_i)^2] - \frac{1}{N}\iint K(x,y)f(x)f(y)\,\mathrm{d}x\,\mathrm{d}y \tag{9}$$

$$= \frac{1}{N}\left(\int K(x,x)\lambda(x)f(x)\,\mathrm{d}x - \iint K(x,y)f(x)f(y)\,\mathrm{d}x\,\mathrm{d}y\right) = \frac{C_{K,f,g}}{N}. \tag{10}$$

We next show the optimality of $g(x) \approx \sqrt{K(x,x)}f(x)$. It suffices to consider when $\int K(x,x)\lambda(x)f(x)\,\mathrm{d}x$ is minimised as the second term is independent of g. From the Cauchy-Schwarz, we have

$$\int K(x,x)\lambda(x)f(x)\,\mathrm{d}x = \int K(x,x)\lambda(x)f(x)\,\mathrm{d}x \int \frac{f(x)}{\lambda(x)}\,\mathrm{d}x \geq \left(\int \sqrt{K(x,x)}f(x)\,\mathrm{d}x\right)^2, \tag{11}$$

and the equality is satisfied if $g(x) = \frac{f(x)}{\lambda(x)} \propto \sqrt{K(x,x)}f(x)$. □

Proof of Theorem 1. Let $(\mathbf{w}_{\text{quad}}, \mathbf{x}_{\text{quad}})$ be a proper recombination of $(\mathbf{w}_{\text{rec}}, \mathbf{x}_{\text{rec}})$, and let Q_n be the quadrature formula given by points $\mathbf{x}_{\text{quad}}$ and weights $\frac{1}{N}\mathbf{w}_{\text{quad}}$, i.e, $Q(h) := \frac{1}{N}\mathbf{w}_{\text{quad}}^\top h(\mathbf{x}_{\text{quad}})$. We also define $\mu_n := \frac{1}{N}\mathbf{w}_{\text{quad}}^\top K(\mathbf{x}_{\text{quad}}, \cdot)$.

A well-known fact is that the worst-case error of Q_n (with respect to f here) $\mathrm{wce}(Q_n) = \sup_{\|h\|\leq 1}|Q_n(h) - \int h(x)f(x)\,\mathrm{d}x|$ satisfies $\mathrm{wce}(Q_n) = \|\mu_K(f) - \mu_n\|_{\mathcal{H}}$ for a kernel satisfying $\int \sqrt{K(x,x)}f(x)\,\mathrm{d}x < \infty$ [8, 16]. By using this and the relation between Bayesian quadrature and kernel quadrature in the main text, we have

$$\sqrt{\mathrm{var}[Z_f \mid \mathbf{x}_{\text{quad}}]} \leq \mathrm{wce}(Q_n) \leq \|\mu_K(f) - \mu_r\|_{\mathcal{H}} + \|\mu_r - \mu_n\|_{\mathcal{H}} \tag{12}$$

From Lemma 1 we have $\mathbb{E}[\|\mu_K(f) - \mu_r\|_{\mathcal{H}}] \leq \mathbb{E}[\|\mu_K(f) - \mu_r\|_{\mathcal{H}}^2]^{1/2} = \sqrt{C_{K,f,g}/N}$, so it now suffices to show

$$\mathbb{E}_{\mathbf{x}_{\text{rec}}}[\|\mu_r - \mu_n\|_{\mathcal{H}}] \leq 2\left(\int K_1(x,x)f(x)\,\mathrm{d}x\right)^{1/2}. \tag{13}$$

We first have

$$\|\mu_r - \mu_n\|_{\mathcal{H}}^2 = \frac{1}{N^2}\left(\mathbf{w}_{\text{rec}}^\top K(\mathbf{x}_{\text{rec}}, \mathbf{x}_{\text{rec}})\mathbf{w}_{\text{rec}} - 2\,\mathbf{w}_{\text{rec}}^\top K(\mathbf{x}_{\text{rec}}, \mathbf{x}_{\text{quad}})\mathbf{w}_{\text{quad}} + \mathbf{w}_{\text{quad}}^\top K(\mathbf{x}_{\text{quad}}, \mathbf{x}_{\text{quad}})\mathbf{w}_{\text{quad}}\right), \tag{14}$$

and from the recombination property we also have

$$\mathbf{w}_{\text{rec}}^\top K_0(\mathbf{x}_{\text{rec}}, \mathbf{x}_{\text{rec}})\mathbf{w}_{\text{rec}} - 2\,\mathbf{w}_{\text{rec}}^\top K_0(\mathbf{x}_{\text{rec}}, \mathbf{x}_{\text{quad}})\mathbf{w}_{\text{quad}} + \mathbf{w}_{\text{quad}}^\top K_0(\mathbf{x}_{\text{quad}}, \mathbf{x}_{\text{quad}})\mathbf{w}_{\text{quad}} = 0, \tag{15}$$

which follows from the fact that $(\mathbf{w}_{\text{rec}}, \mathbf{x}_{\text{rec}})$ and $(\mathbf{w}_{\text{quad}}, \mathbf{x}_{\text{quad}})$ give the same kernel embedding for the RKHS given by K_0 as the latter is a recombination of the former (see e.g. [9, Eq. 14]). By subtracting, we obtain

$$\|\mu_r - \mu_n\|_{\mathcal{H}}^2 \tag{16}$$

$$= \frac{1}{N^2}\left(\mathbf{w}_{\text{rec}}^\top K_1(\mathbf{x}_{\text{rec}}, \mathbf{x}_{\text{rec}})\mathbf{w}_{\text{rec}} - 2\,\mathbf{w}_{\text{rec}}^\top K_1(\mathbf{x}_{\text{rec}}, \mathbf{x}_{\text{quad}})\mathbf{w}_{\text{quad}} + \mathbf{w}_{\text{quad}}^\top K_1(\mathbf{x}_{\text{quad}}, \mathbf{x}_{\text{quad}})\mathbf{w}_{\text{quad}}\right) \tag{17}$$

$$= \|\mu_r^{(1)} - \mu_n^{(1)}\|_{\mathcal{H}_1}^2, \tag{18}$$

where $\mathcal{H}_1$ is the RKHS given by K_1 and

$$\mu_r^{(1)} := \frac{1}{N}\mathbf{w}_{\text{rec}}^\top K_1(\mathbf{x}_{\text{rec}}, \cdot), \qquad \mu_n^{(1)} := \frac{1}{N}\mathbf{w}_{\text{quad}}^\top K_1(\mathbf{x}_{\text{quad}}, \cdot). \tag{19}$$

Now, by letting $k_1^{1/2}(x) := \sqrt{K(x,x)}$, we have $\|K_1(x,\cdot)\|_{\mathcal{H}_1} = k_1^{1/2}(x)$ for a point x. So we have

$$\|\mu_r^{(1)}\|_{\mathcal{H}_1} \leq \frac{1}{N}\mathbf{w}_{\text{rec}}^\top k_1^{1/2}(\mathbf{x}_{\text{rec}}), \qquad \|\mu_n^{(1)}\|_{\mathcal{H}_1} \leq \frac{1}{N}\mathbf{w}_{\text{quad}}^\top k_1^{1/2}(\mathbf{x}_{\text{quad}}) \leq \frac{1}{N}\mathbf{w}_{\text{rec}}^\top k_1^{1/2}(\mathbf{x}_{\text{rec}}), \tag{20}$$

where the last inequality follows from the assumption that $(\mathbf{w}_{\text{quad}}, \mathbf{x}_{\text{quad}})$ is a proper recombination of $(\mathbf{w}_{\text{rec}}, \mathbf{x}_{\text{rec}})$. Therefore, we have the estimate

$$\|\mu_r - \mu_n\|_{\mathcal{H}} = \|\mu_r^{(1)} - \mu_n^{(1)}\|_{\mathcal{H}_1} \leq \|\mu_r^{(1)}\|_{\mathcal{H}_1} + \|\mu_n^{(1)}\|_{\mathcal{H}_1} \leq \frac{2}{N}\mathbf{w}_{\text{rec}}^\top k_1^{1/2}(\mathbf{x}_{\text{rec}}). \tag{21}$$

Finally, to prove (13), we recall that $\mathbf{x}_{\text{rec}}$ is an N-point independent sample from g and $\mathbf{w}_{\text{rec}} = \lambda(\mathbf{x}_{\text{rec}})$, so we obtain

$$\mathbb{E}_{\mathbf{x}_{\text{rec}}}[\|\mu_r - \mu_n\|_{\mathcal{H}}] \leq 2\,\mathbb{E}_{\mathbf{x}_{\text{rec}}}\left[\frac{1}{N}\mathbf{w}_{\text{rec}}^\top k_1^{1/2}(\mathbf{x}_{\text{rec}})\right] = 2\int \lambda(x)k_1^{1/2}(x)g(x)\,\mathrm{d}x \tag{22}$$

$$= 2\int \sqrt{K_1(x,x)}f(x)\,\mathrm{d}x \leq 2\left(\int K_1(x,x)f(x)\,\mathrm{d}x\right)^{1/2}, \tag{23}$$

where we have used Cauchy–Schwarz in the last inequality. □

1.2 Eigenvalue dacay of integral operators

Let us consider the integral operator

$$h \mapsto \int K(\cdot, y)h(y)f(y)\,\mathrm{d}y \tag{24}$$

where $h \in L^2(f) := \{\tilde{h} \mid \text{measurable}, \|\tilde{h}\|_{L^2(f)}^2 := \int \tilde{h}(x)^2 f(x)\,\mathrm{d}x < \infty\}$, and let $\sigma_1 \geq \sigma_2 \geq \cdots \geq 0$ be eigenvalues of this operator. This sequence of eigenvalues is known to be closely related to the convergence rate of kernel quadrature [3].

For the Nyström approximation, we have the following estimate represented by the eigenvalues:

Theorem 2 ([9]). *For a probability density function f on $\mathbb{R}^d$, let $\mathbf{x}_{\text{nys}}$ be an M-point independent sample from f. Let K_0 be the rank-$(n-1)$ approximate kernel using $\mathbf{x}_{\text{nys}}$ given by Eq. (10) in the main text. Then, $K_1 := K - K_0$ satisfies*

$$\int K_1(x,x)f(x)\,\mathrm{d}x \leq n\sigma_n + \sum_{m=n+1}^{\infty}\sigma_n + \frac{2(n-1)K_{\max}}{\sqrt{M}}\left(1 + \sqrt{2\log\frac{1}{\delta}}\right) \tag{25}$$

with probability at least $1-\delta$.

This gives a theoretical guarantee for one step of our algorithm, combined with Theorem 1.

Although the sequence of eigenvalues σ_n does not have an obvious expression when K is the kernel in the middle of our algorithms BASQ, when K is a multivariate Gaussian (RBF) kernel and f is also a Gaussian density, we have a concrete expression of eigenvalues [7].

Indeed, if $K(x,y) = \exp(-\epsilon^2|x-y|^2)$ and $f(x) \propto \exp(-\alpha^2|x|^2)$, in the case $d=1$, we have $\sigma_n = ab^n$ for some constants $a > 0$ and $0 < b < 1$ depending on ϵ and α. Thus, for the d-dimensional case, we can roughly estimate that $\sigma_n \leq a^d b^{m+1}$ if $n > m^d$. So, by only using n, we have

$$\sigma_n \leq a^d b^{\lceil n^{1/d}\rceil} \leq a^d b^{(n^{1/d})} = a^d \exp(-cn^{1/d}), \tag{26}$$

for $c = \log(1/b)$.

2 Model Analysis

2.1 Ablation study

2.1.1 Ablation study of sampling methods

We investigated the influence of each component using 10-dimensional Gaussian mixture likelihood. The performance is evaluated by taking the mean and standard deviation of five metrics when each model gathered 1,000 observations with $n = 100$ batch size. LogMAE is the natural logarithmic MAE between the estimated integral value and true one, and the logKL is the natural logarithmic of the KL divergence between the estimated posterior and true one. Wall time is the overhead time until

Table 1: Ablation study of sampling methods

Sampling		**Prop. dist.**		**Alternate update**		**Performance metric**				
Unc. sampl.	Factor. trick	Linear IVR	Optimal IVR	Kernel Update	RCHQ	logMAE	logKL	wall time (s)	logMAE per time	logKL per time
				✔		-1.8750	-8.1366	407.42	-0.0046	-0.0200
						±0.0435	±0.2049	±10.139	±0.0002	±0.0010
	✔			✔	✔	-3.3310	-9.5934	50.065	-0.0702	-0.1976
						±0.5265	±0.1697	±8.3153	±0.0222	±0.0362
✔		✔		✔	✔	-3.6743	-9.6108	367.63	-0.0100	-0.0263
						±0.0449	±0.1363	±26.565	±0.0008	±0.0023
✔	✔	✔			✔	-1.5936	-7.9967	**47.499**	-0.0346	-0.1735
						±0.0016	±0.0025	±8.1966	±0.0060	±0.0300
✔			✔	✔	✔	-3.4379	**-9.7877**	810.45	-0.0042	-0.0121
						±0.2345	**±0.4589**	±14.468	±0.0003	±0.0008
✔	✔	✔		✔	✔	**-4.0138**	-9.6222	48.75	**-0.0848**	**-0.2038**
						±0.0078	±0.17147	±8.2176	**±0.0144**	**±0.0379**

gathering 1,000 observations in seconds. LogMAE per time refers to the value that logMAE divided by the wall time. LogKL per time is the same.

Uncertainty sampling (Unc. sampl.) and factorisation trick (Factor. trick) refers to the technique explained in the section 4.2 in the main paper. Linearised IVR proposal distribution (Linear IVR) is the ones with Equations (8)-(10) in the main paper, whereas the optimal IVR is the square-root kernel IVR $g(x) = \sqrt{C^L_{\mathbf{y}}(x,x)}\pi(x)$ derived from the Lemma 1. Kernel update refers to the type-II MLE to optimise the hyperparameters. RCHQ means whether or not to adopt RCHQ, if not, it means multi-start optimisation (that is, the same with batch WSABI.)

Sampling from optimal IVR of the square root is intractable, so we adopted the SMC sampling scheme. Firstly, supersamples $\mathbf{X}_{\text{super}}$ are generated from prior $\pi(x)$, then we calculate the weights $w_i = \sqrt{C^L_{\mathbf{y}}(X_i,X_i)}\pi(x_i)/\pi(x_i) = \sqrt{C^L_{\mathbf{y}}(X_i,X_i)}$, then we normalise them via $w^n_i := \frac{w_i}{\sum_i w_i}$, where $\sum_i w^n_i = 1$. At last, we resample subsamples $\mathbf{X}_{\text{quad}}$ from the supersamples $\mathbf{X}_{\text{super}}$ with the weights w_i. By removing the identical samples from $\mathbf{X}_{\text{quad}}$, we can construct $\mathbf{X}_{\text{quad}}$. This removal reduces the size of subsamples to approximately 100 times smaller, so we need to supersample at least 100 times larger than the size of the subsample $\mathbf{X}_{\text{quad}}$. As the size of subsamples is already large, this SMC procedure is computationally demanding.

All components were examined by removing each. The ablation study of the sampling scheme in table 1 shows that all components are essential for reducing overhead or faster convergence. Alternate update and uncertainty sampling contribute to the fast convergence, and factorisation trick and linearised proposal distribution reduce overhead with a negligible effect on convergence.

2.1.2 Ablation study of BQ modelling

We investigated BQ modelling influence with the same procedure of the ablation study in the previous section. The compared models are WSABI-L, WSABI-M, Vanilla BQ (VBA), and log-warp BQ (BBQ). For the details of VBQ and BBQ modelling, see sections 3.3 and 3.4. With regard to the WSABI-M modelling, it has a disadvantageous formula in the mean posterior predictive distribution

Table 2: Ablation study of BQ modelling

BQ modelling				Performance metric				
WSABI -L	WSABI -M	VBQ (no warp)	BBQ (log warp)	logMAE	logKL	wall time (s)	logMAE per time	logKL per time
✔				-4.0138	-9.6222	**48.75**	**-0.0848**	**-0.2038**
				±0.0078	±0.1715	**8.2176**	**0.0144**	**0.0379**
	✔			**-4.0418**	**-10.083**	814.13	-0.0050	-0.0123
				0.0532	**0.2088**	±19.813	±0.0002	±0.0006
		✔		-2.5842	12.307	50.901	0.0534	0.2954
				±0.9978	±0.0158	±5.3634	±0.0252	±0.0254
			✔	-3.1278	9.0425	54.092	0.0617	-0.2038
				±1.7428	±0.3634	±5.4892	±0.0285	±0.0379

$m_{\mathbf{y}}^L(x)$. The acquisition function is expressed as:

$$m_{\mathbf{y}}^L(x) := \alpha + \frac{1}{2}\left(m_{\mathbf{y}}(x)^2 + C_{\mathbf{y}}(x,x)\right). \tag{27}$$

Then, the expectation of the WSABI-M is no more single term;

$$\mathbb{E}[m_{\mathbf{y}}^L(x)] := \alpha + \frac{1}{2}\mathbb{E}[m_{\mathbf{y}}(x)^2] + \frac{1}{2}\mathbb{E}[C_{\mathbf{y}}(x,x)] \tag{28}$$

As such, we cannot apply the factorisation trick for speed. Thus, we should adopt the SMC sampling scheme to sample from WSABI-M as the same procedure with the square root kernel IVR (Optimal IVR) explained in the previous section. This significantly slows down the computation with WSABI-M.

The ablation study result is shown in table 2. While WSABI-M achieves slightly better accuracy than WSABI-L, it also records the slowest computation. The WSABI-M intractable expression hinders to apply the quick sampling schemes we adopted. Vanilla BQ and BBQ (log warped BQ) shows larger errors. Therefore, the WSABI-L adoption is reasonable in this setting.

2.1.3 Ablation study of kernel modelling

We investigated kernel modelling influence with the same procedure of the ablation study in the previous section. The compared kernels are RBF (Radial Basis Function, as known as squared exponential, or Gaussian), Matérn32, Matérn52, Polynomial, Exponential, Rational quadratic, and exponentiated quadratic. The quadrature was performed via RCHQ with the weighted sum of the mean predictive distribution of the optimised GP. Exponential kernel marked the best accuracy in the evidence inference, whereas the KL divergence of posterior is embarrassingly erroneous. This is because all kernels examined in this section is not warped; thus, the GP-modelled likelihood is not non-negative.

2.2 Hyperparameter sensitivity analysis

2.2.1 Analysis results

The hyperparameter sensitivity analysis is performed in the same setting as the previous section that adopts a ten-dimensional Gaussian mixture. The analysis was performed with functional Analysis of Variance (ANOVA) [10]. The functional ANOVA is the statistical method to decompose the variance V of a black box function f into additive components VU associated with each subset of hyperparameters. [10] adopts random forest for efficient decomposition and marginal prediction

Table 3: Ablation study of kernel modelling

Kenel	logMAE	logKL	wall time (s)	logMAE per time	logKL per time
RBF	-2.9598 ± 0.3679	12.321 ± 0.0288	54.730 ± 1.2979	-0.0543 ± 0.0080	0.2349 ± 0.0048
Matérn32	-3.7328 ± 0.9608	12.312 ± 0.0577	53.502 ± 0.8661	-0.0701 ± 0.0191	0.2355 ± 0.0026
Matérn52	-4.3773 ± 1.4593	12.332 ± 0.0765	53.925 ± 0.9662	-0.0817 ± 0.0285	0.2341 ± 0.0027
Polynomial	-3.7828 ± 0.6803	12.208 ± 0.0673	52.281 ± 1.5491	-0.0728 ± 0.0152	0.2449 ± 0.0056
Exponential	**-4.6094 ± 1.3440**	12.268 ± 0.0291	**54.891 ± 0.3428**	**-0.0841 ± 0.0250**	0.2252 ± 0.0009
Rational quadratic	-3.2004 ± 0.6515	12.309 ± 0.0596	62.645 ± 1.7786	-0.0514 ± 0.0119	**0.2059 ± 0.0046**
Exponentiated quadratic	-3.3361 ± 1.4484	**11.046 ± 0.4465**	53.306 ± 0.1992	-0.0627 ± 0.0274	0.2072 ± 0.0076

Table 4: Sensitivity analysis with functional ANOVA

hyperparameters	logMAE	logKL	wall time	logMAE per time	logKL per time
N	0.0835	0.0465	**0.6347**	0.1729	0.1811
M	0.1041	0.0824	0.2497	**0.6401**	**0.6789**
r	0.0041	0.0226	0.0071	0.0040	0.0024
N,M	0.0710	0.1134	0.0903	0.1077	0.1206
N,r	0.1021	0.1299	0.0058	0.0062	0.0024
M,r	**0.4598**	**0.4381**	0.0059	0.0604	0.0123
N,M,r	0.1754	0.3116	0.1671	0.0065	0.0023

over each hyperparameter. The hyperparameters to be analysed are the number of subsamples for recombination N, the number of samples for the Nyström method, and the partition ratio in the IVR proposal distribution r. They need to satisfy the following relationship; $N \gg M > n$, where n is the batch size. We typically take $n = 100$, so M should be larger than at least 200, and N should be larger than at least 20,000. Grid search was adopted for the hyperparameter space, with the range of N = 20,000, 50,000, 100,000, 500,000, 1,000,000, M = 200, 500, 1,000, 5,000, 10,000, and r = 0.0, 0.25, 0.5, 0.75, 1.0, resulting in 125 data points. To compensate for the dynamic range difference, a natural logarithmic $\log N$ and $\log M$ were used as the input.

The result is shown in Table 4. Each value refers to the fraction of the decomposed variance, corresponding to the importance of each hyperparameter over the performance metric. Functional ANOVA evaluates the main effect and the pairwise interaction effect. Figure 1 shows the marginal predictions on the important two hyperparameters for each performance metric.

As the most obvious case, we will look into the wall time. The most important hyperparameter was N, and the second was M. This is well correlated to the theoretical aspect. The overhead computation time of BASQ can be decomposed into two components; subsampling and RCHQ. The N subsampling is dependent on N as $\mathcal{O}(n^2/2 + n_{\text{comp}}N)$. n_{comp} is way less than M or n^2 and is insensitive to the hyperparameter variation or GP updates as we designed it to be sparse. The SMC

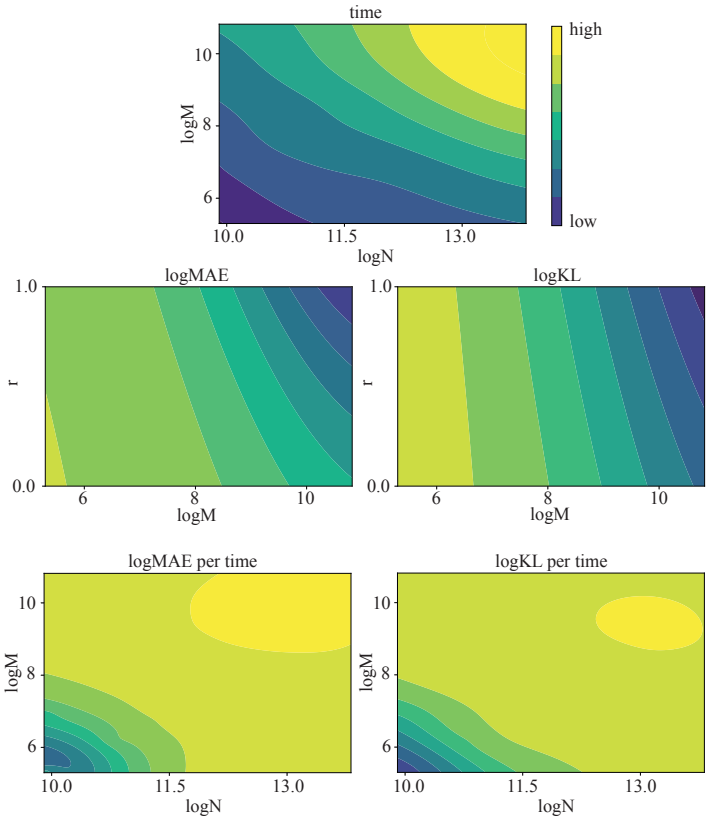


Figure 1: Sensitivity analysis of the hyperparameters over the metrics

sampling for M is negligible as $\mathcal{O}(N+M)$. The RCHQ is $\mathcal{O}(NM + M^2 \log n + Mn^2 \log(N/n))$. Comparing the complexity, the RCHQ stands out. Therefore, the whole BASQ algorithm complexity is dominated by RCHQ. While M has the squared component, N itself is as large as M^2. Therefore, the selected two hyperparameters N and M align with the theory. Figure 1 agrees the above analysis.

The logMAE and logKL metrics have a similar trend to each other. In fact, their correlation coefficient was 0.6494. This makes sense because both metrics are determined by the functional approximation accuracy of likelihood $\ell(x)$. While increasing M is always beneficial in any r, varying r is effective in larger M. At last, the metrics of performance per wall time is the combination of these effects. Obviously, time is the dominant factor, so we should limit the N and M as small as possible. The most important hyperparameter was M. This is a natural consequence because M affects the overhead increment less than N but contributes to reducing errors more.

2.2.2 A guideline to select hyperparameters

The main takeaway from the functional ANOVA analysis is that the accuracy with and without the time constraint has an opposite trend. Therefore, the best hyperparameter set is dependent on the overhead time allowance. The expensiveness of likelihood evaluation determines this. Per the likelihood query time per iteration, we should increase M and N for faster convergence.

In the cheap likelihood case, the most relevant metric is logMAE per time and logKL per time. As we should minimise the time, we choose the minimal size for N and M to minimise the overhead. As the typical hyperparameter set is $(n, N, M, r) = (100, 200, 20, 000, 0.5)$, and these are the minimum values for N and M at given n. The remained choice is the selection of r. As shown in Figure 1, $r = 1$, namely, UB proposal distribution, was the best selection in the Gaussian mixture likelihood case. A similar trend is observed in the synthetic dataset. However, as we observed in the real-world

dataset cases, some likelihoods showed that $r = 0.5$, namely IVR proposal distribution outperformed the UB. Therefore, the $r = 1$ might be the first choice., and $r = 0.5$ is the second choice.

In the expensive likelihood case, we can increase the N and M because the overhead time is less significant than the likelihood query time. The logMAE and logKL without time constraints are good guidelines for tuning the hyperparameters. As M is the most significant hyperparameter, we wish to increase M first. However, we have to increase N under the constraint $N \gg M$, necessary for RCHQ fast convergence. Empirically, we recommend increasing the M three times larger than N from the minimum set because the importance factor ratio in Table 4 is roughly three times. And the increment of M should be corresponded to the likelihood query time $t_{\text{likelihood}}$ over the RCHQ computation time t_{RCHQ}. We increase the M in accordance with the ratio $r_{\text{comp}} := t_{\text{likelihood}}/t_{\text{RCHQ}}$. Thus, the M and N is determined as $M = r_{\text{comp}} M_{\text{min}}$, $N = \frac{r_{\text{comp}}}{3} N_{\text{min}}$, where $M_{\text{min}} = 200, N_{\text{min}} = 20,000$.

3 Analytical form of integrals

3.1 Gaussian identities

$$\mathcal{N}(\mathbf{x}; \mathbf{m}_1, \mathbf{\Sigma}_1)\mathcal{N}(\mathbf{x}; \mathbf{m}_2, \mathbf{\Sigma}_2) = C_c \mathcal{N}(\mathbf{x}; \mathbf{m}_c, \mathbf{\Sigma}_c) \tag{29}$$

$$\int \mathcal{N}(\mathbf{x}; \mathbf{m}_1, \mathbf{\Sigma}_1)\mathcal{N}(\mathbf{x}; \mathbf{m}_2, \mathbf{\Sigma}_2) d\mathbf{x} = C_c \tag{30}$$

$$\mathcal{N}(\mathbf{Ax+b}; \mathbf{m}, \mathbf{\Sigma}) = \sqrt{\frac{|2\pi(\mathbf{A}^\top \mathbf{\Sigma}^{-1}\mathbf{A})^{-1}|}{|2\pi\mathbf{\Sigma}|}} \mathcal{N}\left(\mathbf{x}; \mathbf{A}^{-1}\mathbf{m} - \mathbf{b}, (\mathbf{A}^\top \mathbf{\Sigma}^{-1}\mathbf{A})^{-1}\right) \tag{31}$$

where

$$\mathbf{\Sigma}_c = \left[\mathbf{\Sigma}_1^{-1} + \mathbf{\Sigma}_2^{-1}\right]^{-1} \tag{32}$$

$$\mathbf{m}_c = \left[\mathbf{\Sigma}_1^{-1} + \mathbf{\Sigma}_2^{-1}\right]^{-1} \left(\mathbf{\Sigma}_1^{-1}\mathbf{m}_1 + \mathbf{\Sigma}_2^{-1}\mathbf{m}_2\right) \tag{33}$$

$$C_c = \mathcal{N}(\mathbf{m}_1; \mathbf{m}_2, \mathbf{\Sigma}_1 + \mathbf{\Sigma}_2) \tag{34}$$

In the finite number of product case:

$$\prod_{i=1}^{n} \mathcal{N}(\mathbf{x}; \mathbf{m}_i, \mathbf{\Sigma}_i) = C_m \mathcal{N}(\mathbf{x}; \mathbf{m}_m, \mathbf{\Sigma}_m) \tag{35}$$

$$\int \prod_{i=1}^{n} \mathcal{N}(\mathbf{x}; \mathbf{m}_i, \mathbf{\Sigma}_i) d\mathbf{x} = C_m \int \mathcal{N}(\mathbf{x}; \mathbf{m}_m, \mathbf{\Sigma}_m) d\mathbf{x} \tag{36}$$

$$= C_m \tag{37}$$

where

$$\mathbf{\Sigma}_m = \left[\sum_{i=1}^{n} \left(\mathbf{\Sigma}_i^{-1}\right)\right]^{-1} \tag{38}$$

$$\mathbf{m}_m = \left[\sum_{i=1}^{n} \left(\mathbf{\Sigma}_i^{-1}\right)\right]^{-1} \sum_{i=1}^{n} \left(\mathbf{\Sigma}_i^{-1}\mathbf{m}_i\right) \tag{39}$$

$$C_m = \exp\left[-\frac{1}{2}\left\{(n-1)d\log 2\pi - \sum_{i=1}^{n} \log\left|\mathbf{\Sigma}_i^{-1}\right| + \log\left|\sum_{i=1}^{n}\left(\mathbf{\Sigma}_i^{-1}\right)\right| \right.\right. \tag{40}$$

$$\left.\left. + \sum_{i=1}^{n}\left((\mathbf{\Sigma}_i^{-1}\mathbf{m}_i)^\top \mathbf{\Sigma}_i (\mathbf{\Sigma}_i^{-1}\mathbf{m}_i)\right) - \left(\sum_{i=1}^{n}\left(\mathbf{\Sigma}_i^{-1}\mathbf{m}_i\right)\right)^\top \left(\sum_{i=1}^{n}\left(\mathbf{\Sigma}_i^{-1}\right)\right)^{-1} \left(\sum_{i=1}^{n}\left(\mathbf{\Sigma}_i^{-1}\mathbf{m}_i\right)\right)\right\}\right] \tag{41}$$

3.2 Definitions

x_*: predictive data points, $x_* \in \mathbb{R}^d$
$\mathbf{X}$: the observed data points, $\mathbf{X} \in \mathbb{R}^{n \times d}$
$\mathbf{y} = \sqrt{2(\ell_{\text{true}}(\mathbf{X}) - \alpha)}$: the (warped) observed likelihood, $\mathbf{y} \in \mathbb{R}^n$
$\pi(x_*) = \mathcal{N}(x_*; \mu_\pi, \mathbf{\Sigma}_\pi)$: prior distribution,
$v^{'}$: a kernel variance,
l: a kernel lengthscale,
$K(x_*, \mathbf{X}) = v\mathcal{N}(x_*; \mathbf{X}, \mathbf{W})$: a RBF kernel,
$v = v^{'}\sqrt{|2\pi\mathbf{W}|}$: a normalised kernel variance,
$\mathbf{W}$: a diagonal covariance matrix whose diagonal elements are the lengthscales of each dimension,
$\mathbf{W} = \begin{bmatrix} l^2 & \mathbf{0} \\ \mathbf{0} & l^2 \end{bmatrix}$
$\mathbf{I}$: The identity matrix,
$\mathbf{K}_{XX} = K(\mathbf{X}, \mathbf{X})$: a kernel over the observed data points.

3.3 Warped GPs as Gaussian Mixture

3.3.1 Mean

$$
\begin{aligned}
m^L_{\mathbf{y}}(x_*) &= \alpha + \frac{1}{2}\tilde{m}_{\mathbf{y}}(x_*)^2 && (42)\\
&= \alpha + \frac{1}{2}\mathbf{y}^T\mathbf{K}^{-1}_{XX}K(\mathbf{X}, x_*)K(x_*, \mathbf{X})\mathbf{K}^{-1}_{XX}\mathbf{y} && (43)\\
&= \alpha + \frac{1}{2}\sum_{i,j}\boldsymbol{\omega}_i\boldsymbol{\omega}_j K(X_i, x_*)K(x_*, X_j) && (44)\\
&= \alpha + \sum_{i,j} w^m_{ij}\mathcal{N}\left(x_*; \frac{X_i + X_j}{2}, \frac{\mathbf{W}}{2}\right) && (45)
\end{aligned}
$$

where
Woodbury vector: $\boldsymbol{\omega} = \mathbf{K}^{-1}_{XX}\mathbf{y}$,
mean weight: $w^m_{ij} = \frac{1}{2}v^2\boldsymbol{\omega}_i\boldsymbol{\omega}_j\mathcal{N}(X_i; X_j, 2\mathbf{W})$

3.3.2 Variance

$$
\begin{aligned}
C^L_{\mathbf{y}}(x_*, x_*^{'}) &= \tilde{m}_{\mathbf{y}}(x)\tilde{C}_{\mathbf{y}}(x, x')\tilde{m}_{\mathbf{y}}(x')\\
&= [K(x_*, \mathbf{X})\boldsymbol{\omega}]^\top \left[K(x_*, x_*^{'}) - K(x_*, \mathbf{X})\mathbf{K}^{-1}_{XX}K(\mathbf{X}, x_*^{'})\right]\left[K(x_*^{'}, \mathbf{X})\boldsymbol{\omega}\right]\\
&= \boldsymbol{\omega}^\top K(\mathbf{X}, x_*)K(x_*, x_*^{'})K(x_*^{'}, X)\boldsymbol{\omega}\\
&\quad - \boldsymbol{\omega}^\top K(\mathbf{X}, x_*)K(x_*, \mathbf{X})\mathbf{K}^{-1}_{XX}K(\mathbf{X}, x_*^{'})K(x_*^{'}, \mathbf{X})\boldsymbol{\omega}\\
&= \sum_{i,j}\boldsymbol{\omega}_i\boldsymbol{\omega}_j K(X_i, x_*)K(x_*, x_*^{'})K(x_*^{'}, X_j)\\
&\quad - \sum_{i,j}\boldsymbol{\omega}_i\boldsymbol{\omega}_j\sum_{k,l}\mathbf{\Omega}_{kl}K(X_j, x_*)K(x_*, X_i)K(X_k, x_*^{'})K(x_*^{'}, X_l)\\
&= \sum_{i,j} w^v_{ij}\mathfrak{C}_{\mathfrak{v}}(i,j) - \sum_{i,j}\sum_{k,l} w^{vv}_{ijkl}\mathfrak{C}_{\mathfrak{v}\mathfrak{v}}(i,j,k,l)
\end{aligned}
\tag{46}
$$

where
The inverse kernel weight: $\mathbf{\Omega}_{kl} = \mathbf{K}^{-1}_{XX}(k,l)$
the first variance weight: $w^v_{ij} = v^3\boldsymbol{\omega}_i\boldsymbol{\omega}_j$
the second variance weight: $w^{vv}_{ijkl} = v^4\boldsymbol{\omega}_i\boldsymbol{\omega}_j\mathbf{\Omega}_{kl}$
the first variance Gaussian variable

$$\mathfrak{C}_{\mathfrak{v}}(i,j) = \mathcal{N}\left(\begin{bmatrix} x_* \\ x_*^{'} \\ x_*^{'} \end{bmatrix}; \begin{bmatrix} X_i \\ X_j \\ x_* \end{bmatrix}, \begin{bmatrix} \mathbf{W} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{W} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{W} \end{bmatrix}\right)$$

the second variance Gaussian variable

$$\mathfrak{C}_{\mathfrak{v}\mathfrak{v}}(i,j,k,l) = \mathcal{N}\left(\begin{bmatrix} x_* \\ x_* \\ x_*^{'} \\ x_*^{'} \end{bmatrix}; \begin{bmatrix} X_i \\ X_j \\ X_k \\ X_l \end{bmatrix}, \begin{bmatrix} \mathbf{W} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{W} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{W} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{W} \end{bmatrix}\right)$$

3.3.3 Symmetric variance

Here we consider $x_* = x'_*$ and x_* is a sample with dimension d:

$$
\begin{aligned}
C^L_{\mathbf{y}}(x_*, x_*) &= C^L_{\mathbf{y}}(x_*) \\
&= \boldsymbol{\omega}^\top K(\mathbf{X}, x_*)K(x_*, x_*)K(x_*, \mathbf{X})\boldsymbol{\omega} \\
&\quad - \boldsymbol{\omega}^\top K(\mathbf{X}, x_*)K(x_*, \mathbf{X})\mathbf{K}_{XX}^{-1}K(\mathbf{X}, x_*)K(x_*, \mathbf{X})\boldsymbol{\omega} \\
&= \sum_{i,j} \boldsymbol{\omega}_i\boldsymbol{\omega}_j K(X_i, x_*)K(x_*, x_*)K(x_*, X_j) \\
&\quad - \sum_{i,j} \boldsymbol{\omega}_i\boldsymbol{\omega}_j \sum_{k,l} \boldsymbol{\Omega}_{kl} K(X_j, x_*)K(x_*, X_i)K(X_k, x_*)K(x_*, X_l) \\
&= \frac{v}{\sqrt{|2\pi\mathbf{W}|}} \sum_{i,j} \boldsymbol{\omega}_i\boldsymbol{\omega}_j K(X_i, x_*)K(x_*, X_j) \\
&\quad - \sum_{i,j} \boldsymbol{\omega}_i\boldsymbol{\omega}_j \sum_{k,l} \boldsymbol{\Omega}_{kl} K(X_j, x_*)K(x_*, X_i)K(X_k, x_*)K(x_*, X_l) \\
&= \frac{v^3}{\sqrt{|2\pi\mathbf{W}|}} \sum_{i,j} \boldsymbol{\omega}_i\boldsymbol{\omega}_j \mathcal{N}(X_i; X_j, 2\mathbf{W})\mathcal{N}\left(x_*; \frac{X_i + X_j}{2}, \frac{\mathbf{W}}{2}\right) \\
&\quad - v^4 \sum_{i,j} \left[\boldsymbol{\omega}_i\boldsymbol{\omega}_j \sum_{k,l} \left\{\boldsymbol{\Omega}_{kl}\mathcal{N}(X_l; X_i, 2\mathbf{W})\mathcal{N}\left(x_*; \frac{X_l + X_i}{2}, \frac{\mathbf{W}}{2}\right)\right.\right. \\
&\qquad\qquad \left.\left. \cdot\mathcal{N}(X_k; X_j, 2\mathbf{W})\mathcal{N}\left(x_*; \frac{X_k + X_j}{2}, \frac{\mathbf{W}}{2}\right)\right\}\right] \\
&= \sum_{i,j} w'^{v}_{ij}\mathcal{N}\left(x_*; \frac{X_i + X_j}{2}, \frac{\mathbf{W}}{2}\right) - \sum_{i,j}\sum_{k,l} w'^{vv}_{ijkl}\mathcal{N}\left(x_*; \frac{X_i + X_j + X_k + X_l}{4}, \frac{\mathbf{W}}{4}\right)
\end{aligned}
\tag{47}
$$

where
$w'^{v}_{ij} = \frac{h^3}{\sqrt{|2\pi\mathbf{W}|}}\boldsymbol{\omega}_i\boldsymbol{\omega}_j\mathcal{N}(X_i; X_j, 2\mathbf{W})$
$w'^{vv}_{ijkl} = h^4\boldsymbol{\omega}_i\boldsymbol{\omega}_j\boldsymbol{\Omega}_{kl}\mathcal{N}(X_l; X_i, 2\mathbf{W})\mathcal{N}(X_k; X_j, 2\mathbf{W})\mathcal{N}\left(\frac{X_k+X_j}{2}; \frac{X_i+X_l}{2}, \mathbf{W}\right)$

3.4 Model evidence

The distribution over integral Z is given by:

$$
\begin{aligned}
p(Z|\mathbf{y}) &= \int p\big(Z|\ell(x_*)\big)p\big(\ell(x_*)|\mathbf{y}\big)dx && (48)\\
&= p\big(Z|\ell(x_*)\big)\mathcal{N}\big(\ell(x_*); m^L_{\mathbf{y}}(x_*), C^L_{\mathbf{y}}(x_*)\big) && (49)\\
&= \mathcal{N}\Big(Z; \mathbb{E}\big[Z|\mathbf{y}\big], \text{var}\big[Z|\mathbf{y}\big]\Big) && (50)
\end{aligned}
$$

3.4.1 Mean of the integral

$$\mathbb{E}\big[Z|\mathbf{y}\big] = \mathbb{E}\big[m_{\mathrm{y}}^{L}\big] \tag{51}$$

$$= \int m_{\mathbf{y}}^{L}(x_*)\pi(x_*)dx_* \tag{52}$$

$$= \alpha + \frac{1}{2}\int \tilde{m}_{\mathbf{y}}^{2}(x_*)\pi(x_*)dx_* \tag{53}$$

$$= \alpha + \sum_{i,j} w_{ij}^{m} \int \mathcal{N}\left(x_*; \frac{X_i + X_j}{2}, \frac{\mathbf{W}}{2}\right)\mathcal{N}(x_*; \mu_\pi, \mathbf{\Sigma}_\pi)dx_* \tag{54}$$

$$= \alpha + \sum_{i,j} w_{ij}^{m} \mathcal{N}\left(\frac{X_i + X_j}{2}; \mu_\pi, \frac{\mathbf{W}}{2} + \mathbf{\Sigma}_\pi\right) \tag{55}$$

3.4.2 Variance of the integral

$$\begin{aligned}
\mathrm{var}\big[Z|\mathbf{y}\big] &= \mathrm{var}\big[C_{\mathbf{y}}^{L}\big] \\
&= \iint \pi(x_*)C_{\mathbf{y}}^{L}(x_*, x_*^{'})\pi(x_*^{'})dx_*dx_*^{'} \\
&= \iint \Big(\boldsymbol{\omega}^{\top}K(\mathbf{X}, x_*)K(x_*, x_*^{'})K(x_*^{'}, \mathbf{X})\boldsymbol{\omega}\pi(x_*)\pi(x_*^{'}) \\
&\quad -\boldsymbol{\omega}^{\top}K(\mathbf{X}, x_*)K(x_*, \mathbf{X})\mathbf{K}_{XX}^{-1}K(\mathbf{X}, x_*^{'})K(x_*^{'}, \mathbf{X})\boldsymbol{\omega}\pi(x_*)\pi(x_*^{'})\Big)\, dx_*dx_*^{'} \\
&= \sum_{i,j}\boldsymbol{\omega}_i\boldsymbol{\omega}_j \iint K(X_i, x_*)K(x_*, x_*^{'})K(x_*^{'}, X_j)\pi(x_*)\pi(x_*^{'})dx_*dx_*^{'} \\
&\quad - \sum_{i,j}\boldsymbol{\omega}_i\boldsymbol{\omega}_j \sum_{k,l}\mathbf{\Omega}_{kl} \iint K(X_j, x_*)K(x_*, X_i)K(X_k, x_*^{'})K(x_*^{'}, X_l)\pi(x_*)\pi(x_*^{'})dx_*dx_*^{'} \\
&= \sum_{i,j}\boldsymbol{\omega}_i\boldsymbol{\omega}_j h^3 \int \left[\mathcal{N}(x_*^{'}; \mu_\pi, \Sigma_\pi)\mathcal{N}(x_*^{'}; X_j, \mathbf{W}) \int \mathcal{N}(x_*; X_i, \mathbf{W})\mathcal{N}(x_*; x_*^{'}, \mathbf{W})\mathcal{N}(x_*; \mu_\pi, \Sigma_\pi)dx_*\right] dx_*^{'} \\
&\quad - \sum_{i,j}\sum k, l\boldsymbol{\omega}_i\boldsymbol{\omega}_j\mathbf{\Omega}_{kl}h^4 \int \mathcal{N}(x_*; X_j, \mathbf{W})\mathcal{N}(x_*; X_i, \mathbf{W})\mathcal{N}(x_*; \mu_\pi, \Sigma_\pi)dx_* \\
&\qquad\qquad \cdot \int \mathcal{N}(x_*^{'}; X_k, \mathbf{W})\mathcal{N}(x_*^{'}; x_l, \mathbf{W})\mathcal{N}(x_*^{'}; \mu_\pi, \Sigma_\pi)dx_*^{'} \\
&= \sum_{i,j}\boldsymbol{\omega}_i\boldsymbol{\omega}_j h^3 \int \mathcal{N}(x_*^{'}; \mu_\pi, \Sigma_\pi)\mathcal{N}(x_*^{'}; X_j, \mathbf{W})\mathcal{N}(x_*^{'}; X_i, \mathbf{W})\mathcal{N}\left(\frac{X_i + x_*^{'}}{2}; \mu_\pi, \frac{\mathbf{W}}{2} + \mathbf{\Sigma}_\pi\right) dx_*^{'} \\
&\quad - \sum_{i,j}\sum k, l\boldsymbol{\omega}_i\boldsymbol{\omega}_j\mathbf{\Omega}_{kl}h^4 \left[\mathcal{N}(X_j; X_i, 2W)\mathcal{N}\left(\frac{X_i + X_j}{2}; \mu_\pi, \frac{\mathbf{W}}{2} + \mathbf{\Sigma}_\pi\right)\right] \\
&\qquad\qquad \cdot \left[\mathcal{N}(X_k; x_l, 2W)\mathcal{N}\left(\frac{X_k + X_l}{2}; \mu_\pi, \frac{\mathbf{W}}{2} + \mathbf{\Sigma}_\pi\right)\right] \\
&= \sum_{i,j}\boldsymbol{\omega}_i\boldsymbol{\omega}_j h^3 \int \mathcal{N}(x_*^{'}; \mu_\pi, \mathbf{\Sigma}_\pi)\mathcal{N}(x_*^{'}; X_i, \mathbf{W})\mathcal{N}(x_*^{'}; X_j, \mathbf{W})2^d\mathcal{N}\left(x_*^{'}; 2\mu_\pi - X_i/2, 2\mathbf{W} + 4\mathbf{\Sigma}_\pi\right) dx_*^{'} \\
&\quad - \sum_{i,j}\sum_{k,l} w_{ijkl}^{vv}\mathfrak{K}_{vv}(i, j, k, l) \\
&= \sum_{i,j} 2^d w_{ij}^{v}\mathfrak{K}_{v}(i, j) - \sum_{i,j}\sum_{k,l} w_{ijkl}^{vv}\mathfrak{K}_{vv}(i, j, k, l)
\end{aligned} \tag{56}$$

where

$$\mathfrak{K}_{\mathfrak{v}}(i,j) = \mathcal{N}(X_i; X_j, 2\mathbf{W})\mathcal{N}\left(\frac{X_i + X_j}{2}; \mu_\pi, \frac{\mathbf{W}}{2} + \mathbf{\Sigma}_\pi\right) \tag{57}$$

$$\mathcal{N}\left[\left(2\mathbf{W}^{-1} + \mathbf{\Sigma}_\pi^{-1}\right)^{-1}\left(\mathbf{W}^{-1}(X_i + X_j) + \mathbf{\Sigma}_\pi^{-1}\mu_\pi\right); 2\mu_\pi - \frac{X_i}{2}, \left(2\mathbf{W}^{-1} + \mathbf{\Sigma}_\pi^{-1}\right)^{-1} + 2\mathbf{W} + 2\mathbf{\Sigma}_\pi\right] \tag{58}$$

$$\mathfrak{K}_{\mathfrak{v}\mathfrak{v}}(i,j,k,l) = \left[\mathcal{N}(X_j; X_i, 2\mathbf{W})\mathcal{N}\left(\frac{X_i + X_j}{2}; \mu_\pi, \frac{\mathbf{W}}{2} + \mathbf{\Sigma}_\pi\right)\right]\left[\mathcal{N}(X_k; X_l, 2\mathbf{W})\mathcal{N}\left(\frac{X_k + X_l}{2}; \mu_\pi, \frac{\mathbf{W}}{2} + \mathbf{\Sigma}_\pi\right)\right] \tag{59}$$

3.5 Posterior inference

3.5.1 Joint posterior

$$p(x) = \frac{m_{\mathbf{y}}^L(x)\pi(x)}{\mathbb{E}[Z|\mathbf{y}]} \tag{60}$$

3.5.2 Marginal posterior

The marginal posterior can be on obtained from Gaussian mixture form of joint posterior. Thanks to the Gaussianity, marginal posterior can be easily derived by extracting the d-th element of matrices in the following mixture of Gaussians.

$$p(x) = \frac{\alpha}{\mathbb{E}[Z|\mathbf{y}]} + \sum_{i,j} w_{ij}^p \mathcal{N}(x_*; \mu_p, \mathbf{\Sigma}_p) \tag{61}$$

where
$w_{ij}^p = \frac{w_{ij}^m}{\mathbb{E}[Z|\mathbf{y}]}\mathcal{N}\left(\frac{X_i+X_j}{2}; \mu_\pi, \frac{\mathbf{W}}{2} + \mathbf{\Sigma}_\pi\right)$
$\mathbf{\Sigma}_p = (2\mathbf{W}^{-1} + \mathbf{\Sigma}_\pi^{-1})^{-1}$
$\mu_p = \mathbf{\Sigma}_p(\mathbf{W}^{-1}(X_i + X_j) + \mathbf{\Sigma}_\pi^{-1}\mu_\pi)$

3.5.3 Conditional posterior

The conditional posterior $p\left(x; d = d \,|\, d = \mathbf{D} \setminus \mathbf{D}(\geq d)\right)$ can be derived from the Gaussian mixture form of joint posterior. We can obtain the conditional posterior via applying the following relationship to each Gaussian: Assume $\mathbf{x} \sim \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \mathbf{\Sigma})$ where

$$\mathbf{x} = \begin{bmatrix}\mathbf{x}_a \\ \mathbf{x}_b\end{bmatrix} \qquad \boldsymbol{\mu} = \begin{bmatrix}\boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b\end{bmatrix} \qquad \mathbf{\Sigma} = \begin{bmatrix}\mathbf{\Sigma}_a & \mathbf{\Sigma}_c \\ \mathbf{\Sigma}_c^\top & \mathbf{\Sigma}_b\end{bmatrix} \tag{62}$$

Then

$$p(\mathbf{x}_a)|p(\mathbf{x}_b) = \mathcal{N}(\mathbf{x}_a; \hat{\boldsymbol{\mu}}_a, \hat{\mathbf{\Sigma}}_a) \qquad \begin{cases}\hat{\boldsymbol{\mu}}_a = \boldsymbol{\mu}_a + \mathbf{\Sigma}_c\mathbf{\Sigma}_b^{-1}(\mathbf{x}_b - \boldsymbol{\mu}_b) \\ \hat{\mathbf{\Sigma}}_a = \mathbf{\Sigma}_a - \mathbf{\Sigma}_c\mathbf{\Sigma}_b^{-1}\mathbf{\Sigma}_c^\top\end{cases} \tag{63}$$

$$p(\mathbf{x}_b)|p(\mathbf{x}_a) = \mathcal{N}(\mathbf{x}_b; \hat{\boldsymbol{\mu}}_b, \hat{\mathbf{\Sigma}}_b) \qquad \begin{cases}\hat{\boldsymbol{\mu}}_b = \boldsymbol{\mu}_b + \mathbf{\Sigma}_c\mathbf{\Sigma}_a^{-1}(\mathbf{x}_a - \boldsymbol{\mu}_a) \\ \hat{\mathbf{\Sigma}}_b = \mathbf{\Sigma}_b - \mathbf{\Sigma}_c\mathbf{\Sigma}_a^{-1}\mathbf{\Sigma}_c^\top\end{cases} \tag{64}$$

4 Uncertainty sampling

4.1 Analytical form of acquisiton function

4.1.1 Acquisiton function as Gaussian Mixture

We set the acquisition function $A(x)$ as the product of the variance and the prior. As is shown in Eq. (47), When we provide the predictive samples x_*:

$$
\begin{aligned}
A(x_*) &= C_{\mathbf{y}}^{L}(x_*, x_*)\pi(x_*) && (65)\\
&= C_{\mathbf{y}}^{L}(x_*)\pi(x_*) && (66)\\
&= \left(\sum_{i,j} w_{ij}^{'v} \mathcal{N}\left(x_*; \frac{X_i + X_j}{2}, \frac{\mathbf{W}}{2}\right) - \sum_{i,j}\sum_{k,l} w_{ijkl}^{'vv} \mathcal{N}\left(x_*; \frac{X_i + X_j + X_k + X_l}{4}, \frac{\mathbf{W}}{4}\right) \right) && (67)\\
&\quad \cdot \mathcal{N}\left(x_*; \mu_\pi, \mathbf{\Sigma}_\pi\right) && (68)\\
&= \sum_{i,j} w_{ij}^{'A} \mathcal{N}\left(x_*; \mu_{ij}^{A}, \mathbf{\Sigma}_A\right) - \sum_{i,j}\sum_{k,l} w_{ijkl}^{'AA} \mathcal{N}\left(x_*; \mu_{ijkl}^{AA}, \mathbf{\Sigma}_{AA}\right) && (69)
\end{aligned}
$$

where
$\mathbf{\Sigma}_A = (2\mathbf{W}^{-1} + \mathbf{\Sigma}_\pi^{-1})^{-1}$
$\mu_{ij}^{A} = \mathbf{\Sigma}_A\left(\mathbf{W}^{-1}(X_i + X_j) + \mathbf{\Sigma}_\pi^{-1}\mu_\pi\right)$
$w_{ij}^{'A} = w_{ij}^{'v}\mathcal{N}\left(\frac{X_i+X_j}{2}; \mu_\pi, \frac{\mathbf{W}}{2} + \mathbf{\Sigma}_\pi\right)$

$\mathbf{\Sigma}_{AA} = (4W^{-1} + \mathbf{\Sigma}_\pi^{-1})^{-1}$
$\mu_{ijkl}^{AA} = \mathbf{\Sigma}_A\left(\mathbf{W}^{-1}(X_i + X_j + X_k + X_l) + \mathbf{\Sigma}_\pi^{-1}\mu_\pi\right)$
$w_{ijkl}^{'AA} = w_{ijkl}^{'vv}\mathcal{N}\left(\frac{X_i+X_j+X_k+X_l}{4}; \mu_\pi, \frac{\mathbf{W}}{4} + \mathbf{\Sigma}_\pi\right)$

4.1.2 Normalising constant and PDF

The normalising constant can be obtained via the integral:

$$
\begin{aligned}
Z_A &= \int A(x_*)dx_* \\
&= \sum_{i,j} w_{ij}^{'A} - \sum_{i,j}\sum_{k,l} w_{ijkl}^{'AA}
\end{aligned} \tag{70}
$$

Thus, the normalised acquisition function as PDF p_A is as follows:

$$
\begin{aligned}
p_A(x_*) &= \tilde{A}(x_*) && (71)\\
&= \frac{C_{\mathbf{y}}^{L}(x_*)\pi(x_*)}{Z_A} && (72)\\
&= \sum_{i,j} w_{ij}^{A}\mathcal{N}(x_*; \mu_{ij}^{A}, \mathbf{\Sigma}_A) - \sum_{i,j}\sum_{k,l} w_{ijkl}^{AA}\mathcal{N}(x_*; \mu_{ijkl}^{AA}, \mathbf{\Sigma}_{AA}) && (73)
\end{aligned}
$$

where $w_{ij}^{A} = w_{ij}^{'A}/Z_A$
$w_{ijkl}^{AA} = w_{ijkl}^{'AA}/Z_A$

4.1.3 Factorisation trick

The factorisation trick is set in the conditions where the likelihood is $|\tilde{\ell}(x)|$, the distribution of interest $f(x)$ is $|\tilde{\ell}(x)|\pi(x)$, and the acquiition function is $\tilde{C}_{\mathbf{y}}(x, x)\pi(x)$. We will derive the Gaussian mixture form of this acquisition function.

$$
\begin{aligned}
A(x) &= \tilde{C}_{\mathbf{y}}(x, x)\pi(x) && (74)\\
&= \frac{v}{\sqrt{|2\pi\mathbf{W}|}}\mathcal{N}(x; \mu_\pi, \mathbf{\Sigma}_\pi) - v^2\sum_{ij}\mathbf{\Omega}_{ij}\mathcal{N}(x; \mu^f, \mathbf{\Sigma}^f) && (75)
\end{aligned}
$$

where
$\mathbf{\Sigma}^f = \left(2\mathbf{W}^{-1} + \mathbf{\Sigma}_\pi^{-1}\right)^{-1}$
$\boldsymbol{\mu}^f = \mathbf{\Sigma}^f\left(\mathbf{W}^{-1}(X_i + X_j) + \mathbf{\Sigma}_\pi^{-1}\mu_\pi\right)$

Then, normalising constant is:

$$Z_A^f = \int A(x)dx = \frac{v}{\sqrt{|2\pi\mathbf{W}|}} - v^2 \sum_{ij} \mathbf{\Omega}_{ij} \tag{76}$$

Therefore, the acquisition function as a probability distribution function $p_A(x)$ is:

$$p_A(x) = \frac{v}{Z_A^f\sqrt{|2\pi\mathbf{W}|}}\mathcal{N}(x;\mu_\pi,\mathbf{\Sigma}_\pi) - \frac{v^2}{Z_A^f}\sum_{ij}\mathbf{\Omega}_{ij}\mathcal{N}(x;\mu^f,\mathbf{\Sigma}^f) \tag{77}$$

4.2 Efficient sampler

4.2.1 Acquisition function as sparse Gaussian mixture sampler

Eq. (77) clearly explains the acquisition function can be written as a Gaussian mixture, but it also contains negative components. The first term is obviously positive, and the second term is a mixture of positive and negative components. The condition where the second term becomes positive is $\mathbf{\Omega}_{ij} < 0$. By checking the negativity of the element $\mathbf{\Omega}_{ij}$, we can reduce the number of components by half on average. Then, when we consider sampling from this non-negative acquisition function, the following steps will be performed: First, we sample the index of the component from weighted categorical distribution $\Pi(x)$, and the weights are the one in Eq. (77). Then, we sample from the normal distribution that has the same index identified in the first process. These sampling will be repeated until the accumulated number of the sample reaches the same as the recombination sample size N. This means the component whose weight is lower than $1/N$ is unlikely to be sampled even once. Therefore, we can dismiss these components with the threshold of $1/N$. Interestingly, the weights of Gaussians vary exponentially. The reduced number of Gaussians is much lower than n^2. As such, we can construct the efficient sparse Gaussian mixture sampler of the acquisition function $p'_A(x)$.

4.2.2 Sequential Monte Carlo

Recall from the Eqs (8) - (10) in the main paper, we wish to sample from $g(x) = (1-r)\pi(x)+rp_A(x)$. We have the efficient sampler $p'_A(x)$, but $p'_A(x) \neq p_A(x)$ because $p'_A(x)$ is the function which is constructed from only positive components of $p_A(x)$. Thus, we need to correct this difference via sequential Monte Carlo (SMC). The idea of SMC is simple:

1. sample $\mathbf{x} \sim p'_A(x)$, $\mathbf{x} \in \mathbb{R}^{rN}$
2. calculate weights $\mathbf{w}_{\text{smc}} = p_A(\mathbf{x})/p'_A(\mathbf{x})$
3. resample from the categorical distribution of the index of $\mathbf{x}$ based on $\mathbf{w}_{\text{smc}}$

If $p_A(\mathbf{x}) \approx p'_A(\mathbf{x})$, the rejected samples in the procedure 3 is minimised. As we formulate $p'_A(\mathbf{x})$ can approximate $p_A(\mathbf{x})$ well, the number of samples to be rejected is negligibly small. Thus, the number of samples from $p_A(x)$ is slightly smaller than rN. The number of samples for $\pi(x)$ in $g(x)$ is adjusted to this fluctuation to keep the partition ratio r.

5 Other BQ modelling

5.1 Non-Gaussian Prior

Non-Gaussian prior distributions can be applied via importance sampling.

$$\int \ell(x)\pi(x) = \int \ell(x)\frac{\pi(x)}{g(x)}g(x)dx \tag{78}$$

$$= \int \ell'(x)g(x)dx \tag{79}$$

where $\pi(x)$ is the arbitrary prior distribution of interest, $g(x)$ is the proposal distribution of Gaussian (mixture), $\ell'(x) = \ell(x)\pi(x)/g(x)$ is the modified likelihood. Then, we set the two independent GPs on each of $\ell(x)$ and $\ell'(x)$. Then, both the model evidence $Z = \int \ell'(x)g(x)dx$, and the posterior $p(x) = \ell(x)\pi(x)/Z$ becomes analytical.

5.2 Non-Gaussian kernel

WSABI-BQ methods are limited to the squared exponential kernel in the likelihood modelling. However, other BQ modelling permits the selection of different kernels. For instance, there are the existing works on tractable BQ modelling with kernels of Matérn [6], Wendland [14], Gegenbauer [6], Trigonometric (Integration by parts), splines [18] polynomial [5], and gradient-based kernel [13]. See details in [6].

5.3 RCHQ for Non-Gaussian prior and kernel

RCHQ permits the integral estimation via non-Gaussian prior and/or kernel without bespoke modelling like the above techniques.

$$\mathbf{X}_{\text{quad}}, \mathbf{w}_{\text{quad}} = \text{RCHQ}(\text{BQmodel}, \text{sampler}) \tag{80}$$

$$\mathbb{E}[\ell(x)\pi(x)] = \mathbf{w}_{\text{quad}} m^L_{\mathbf{y}}(\mathbf{X}_{\text{quad}}) \tag{81}$$

$$\mathbb{V}\text{ar}[\ell(x)\pi(x)] = \mathbf{w}_{\text{rec}}^\top C^L_{\mathbf{y}}(\mathbf{x}_{\text{rec}}, \mathbf{x}_{\text{rec}})\mathbf{w}_{\text{rec}} - 2\,\mathbf{w}_{\text{rec}}^\top C^L_{\mathbf{y}}(\mathbf{x}_{\text{rec}}, \mathbf{x}_{\text{quad}})\mathbf{w}_{\text{quad}} + \mathbf{w}_{\text{quad}}^\top C^L_{\mathbf{y}}(\mathbf{x}_{\text{quad}}, \mathbf{x}_{\text{quad}})\mathbf{w}_{\text{quad}} \tag{82}$$

5.4 Vanilla BQ model (VBQ)

5.4.1 Expectation

$$\int m_{\ell_0}(x)\pi(x)dx = v\int \mathcal{N}(x; \mathbf{X}, \mathbf{W})\mathcal{N}(x; \mu_\pi, \mathbf{\Sigma}_\pi)dx\boldsymbol{\omega} \tag{83}$$

$$= v\mathcal{N}(\mathbf{X}; \mu_\pi, \mathbf{W} + \mathbf{\Sigma}_\pi)\boldsymbol{\omega} \tag{84}$$

$$\tag{85}$$

5.4.2 Acquisition function

$$A_{\text{unnormalised}}(x) = C(x, x)\pi(x) \tag{86}$$

$$= K(x, x)\pi(x) - K(x, \mathbf{X})K(\mathbf{X}, \mathbf{X})^{-1}K(\mathbf{X}, x)\pi(x) \tag{87}$$

$$= \mathcal{N}(x; x, \mathbf{W})\mathcal{N}(x; \mu_\pi, \mathbf{\Sigma}_\pi) - v^2\mathcal{N}(x; \mu_\pi, \mathbf{\Sigma}_\pi)\mathcal{N}(x; \mathbf{X}, \mathbf{W})K(\mathbf{X}, \mathbf{X})^{-1}\mathcal{N}(x; \mathbf{X}, \mathbf{W})^\top \tag{88}$$

$$= \frac{v}{\sqrt{|2\pi\mathbf{W}|}}\mathcal{N}(x; \mu_\pi, \mathbf{\Sigma}_\pi) - v^2\sum_{i,j}\Omega_{ij}\mathcal{N}(\mu_\pi; X_i, \mathbf{W} + \mathbf{\Sigma}_\pi)\mathcal{N}(x; X'_i, \mathbf{W}')\mathcal{N}(x; X_j, \mathbf{W}) \tag{89}$$

$$= \frac{v}{\sqrt{|2\pi\mathbf{W}|}}\mathcal{N}(x; \mu_\pi, \mathbf{\Sigma}_\pi) - v^2\sum_{i,j}\Omega_{ij}\mathcal{N}(\mu_\pi; X_i, \mathbf{W} + \mathbf{\Sigma}_\pi)\mathcal{N}(X_j; X'_i, \mathbf{W} + \mathbf{W}')\mathcal{N}(x; X''_{ij}, \mathbf{W}'') \tag{90}$$

$$= \frac{v}{\sqrt{|2\pi\mathbf{W}|}}\mathcal{N}(x; \mu_\pi, \mathbf{\Sigma}_\pi) - \sum_{i,j}w_{ij}\mathcal{N}(x; X''_{ij}, \mathbf{W}'') \tag{91}$$

$$\tag{92}$$

where

$$\Omega_{ij} := K(\mathbf{X}, \mathbf{X})^{-1} \tag{93}$$

$$w_i := v^2\Omega_{ij}\mathcal{N}(\mu_\pi; X_i, \mathbf{W} + \mathbf{\Sigma}_\pi)\mathcal{N}(X_j; X'_i, \mathbf{W} + \mathbf{W}') \tag{94}$$

$$\mathbf{W}' = (\mathbf{W}^{-1} + \mathbf{\Sigma}_\pi^{-1})^{-1} \tag{95}$$

$$X'_i = \mathbf{W}'(\mathbf{W}^{-1}X_i + \mathbf{\Sigma}_\pi^{-1}\mu_\pi) \tag{96}$$

$$\mathbf{W}'' = \left(\mathbf{W}'^{-1} + \mathbf{W}^{-1}\right)^{-1} \tag{97}$$

$$X''_{ij} = \mathbf{W}''\left(\mathbf{W}'^{-1}X'_i + \mathbf{W}^{-1}X_j\right) \tag{98}$$

$$\tag{99}$$

$$\tag{100}$$

Then, the normalised acquisition function $p_A(x)$ as a probability distribution is as follows:

$$P_A(x) := A_{\text{unnormalised}}(x)/Z_A \tag{101}$$

$$= \frac{v}{Z_A\sqrt{|2\pi\mathbf{W}|}}\mathcal{N}(x;\mu_\pi,\mathbf{\Sigma}_\pi) - \sum_{i,j}\frac{w_i}{Z_A}\mathcal{N}(x;X''_{ij},\mathbf{W}'') \tag{102}$$

$$\tag{103}$$

where

$$Z_A = \int A_{\text{unnormalised}}(x)dx \tag{104}$$

$$= \frac{v}{\sqrt{|2\pi\mathbf{W}|}}\int \mathcal{N}(x;\mu_\pi,\mathbf{\Sigma}_\pi)dx - v^2\sum_{i,j}\Omega_{ij}\mathcal{N}(\mu_\pi;X_i,\mathbf{W}+\mathbf{\Sigma}_\pi)\int \mathcal{N}(x;X'_i,\mathbf{W}')\mathcal{N}(x;X_j,\mathbf{W})dx \tag{105}$$

$$= \frac{v}{\sqrt{|2\pi\mathbf{W}|}} - v^2\sum_{i,j}\Omega_{ij}\mathcal{N}(\mu_\pi;X_i,\mathbf{W}+\mathbf{\Sigma}_\pi)\mathcal{N}(X_j;X'_i,\mathbf{W}+\mathbf{W}') \tag{106}$$

$$\tag{107}$$

5.5 Log-GP BQ modelling (BBQ)

5.5.1 BBQ modelling

The doubly-Bayesian quadrature (BBQ) is modelled with log-warped GPs as follows (see details in the paper [15]):

Set three GPs

$$p(\ell_0|\mathbf{D}) \sim \mathcal{GP}(\ell_0; m_{\ell_0}(x), C_{\ell_0}(x,x')) \tag{108}$$

$$p(\log\ell_0|\mathbf{D}) \sim \mathcal{GP}(\log\ell_0; m_{\log\ell_0}(x), C_{\log\ell_0}(x,x')) \tag{109}$$

$$p(\Delta_{\log\ell_0}|\mathbf{D}) \sim \mathcal{GP}(\Delta_{\log\ell_0}; m_\Delta(x), C_\Delta(x,x')) \tag{110}$$

$$\tag{111}$$

Definitions

$$\exp(\log\ell(x)) \approx \exp(\log\ell_0(x)) + \exp(\log\ell_0(x))(\log\ell(x) - \log\ell_0(x)) \tag{112}$$

$$\ell_0 := m_{\ell_0} \tag{113}$$

$$\Delta_{\log\ell_0} := m_{\log\ell_0} - \log\ell_0 = m_{\log\ell_0} - \log(m_{\ell_0}) \tag{114}$$

$$m_\ell = m_{\ell_0} + m_{\ell_0}m_\Delta(x) \tag{115}$$

Expectation

$$\mathbb{E}[Z|\mathbf{D}] = \int m_{\ell_0}(x)\pi(x)dx + \int m_{\ell_0}(x)m_\Delta(x)\pi(x)dx \tag{116}$$

$$\tag{117}$$

The first term is as follows:

$$\int m_{\ell_0}(x)\pi(x)dx = v\int \mathcal{N}(x;\mathbf{X},\mathbf{W})\mathcal{N}(x;\mu_\pi,\mathbf{\Sigma}_\pi)dx\boldsymbol{\omega} \tag{118}$$

$$= v\mathcal{N}(\mathbf{X};\mu_\pi,\mathbf{W}+\mathbf{\Sigma}_\pi)\boldsymbol{\omega} \tag{119}$$

$$\tag{120}$$

where

$$\boldsymbol{\omega} = K(\mathbf{X},\mathbf{X})^{-1}\ell_0(\mathbf{X}) \tag{121}$$

$$K(x,\mathbf{X}) = v\mathcal{N}(x;\mathbf{X},\mathbf{W}) \tag{122}$$

The second term is as follows:

$$\int m_{\ell_0}(x)\Delta_{\log \ell_0}(x)p(x)dx \tag{123}$$

$$= vv^{\Delta}\boldsymbol{\omega}^{\top}\int \mathcal{N}(x;\mathbf{X},\mathbf{W})^{\top}\mathcal{N}(x;\mathbf{X}^{\Delta},\mathbf{W}^{\Delta})\mathcal{N}(x;\mu_{\pi},\boldsymbol{\Sigma}_{\pi})dx\boldsymbol{\omega}^{\Delta} \tag{124}$$

$$= vv^{\Delta}\boldsymbol{\omega}^{\top}\mathcal{N}(\mathbf{X}^{\top}-\mathbf{X}^{\Delta},\mathbf{0},\mathbf{W}+\mathbf{W}^{\Delta})\int \mathcal{N}(x;\boldsymbol{\mu}^{\Delta},\boldsymbol{\Sigma}^{\Delta})\mathcal{N}(x;\mu_{\pi},\boldsymbol{\Sigma}_{\pi})dx\boldsymbol{\omega}^{\Delta} \tag{125}$$

$$= vv^{\Delta}\boldsymbol{\omega}^{\top}\mathcal{N}(\mathbf{X}^{\top}-\mathbf{X}^{\Delta},\mathbf{0},\mathbf{W}+\mathbf{W}^{\Delta})\mathcal{N}(\boldsymbol{\mu}^{\Delta};\mu_{\pi},\boldsymbol{\Sigma}_{\pi}+\boldsymbol{\Sigma}^{\Delta})\boldsymbol{\omega}^{\Delta} \tag{126}$$

$$\tag{127}$$

where

$$\boldsymbol{\omega}^{\Delta} = K(\mathbf{X}^{\Delta},\mathbf{X}^{\Delta})^{-1}\Delta_{\log \ell_0}(\mathbf{X}^{\Delta}) \tag{128}$$

$$\mu^{\Delta} = [\mathbf{W}^{-1}+\mathbf{W}^{\Delta,-1}]^{-1}(\mathbf{W}^{-1}\mathbf{X}+\mathbf{W}^{\Delta,-1}\mathbf{X}^{\Delta}) \tag{129}$$

$$\boldsymbol{\Sigma}^{\Delta} = [\mathbf{W}^{-1}+\mathbf{W}^{\Delta,-1}]^{-1} \tag{130}$$

$$\tag{131}$$

$\mathbf{X}^{\Delta}$ is the observed data for the correlation factor $\Delta_{\log \ell_0}$, which includes not only $\mathbf{X}$ but also the additional data points via $m_{\log \ell_0} - \log(m_{\ell_0})$, with GPs calculation.

5.5.2 Sampling for BBQ

We apply BASQ-VBQ sampling scheme for log-GP $\log\ell_0$, then calculate the others as post-process. Therefore, the sampling cost is similar to the VBQ, whereas the integral estimation as post-process is more expensive than VBQ.

6 Experimental details

6.1 Synthetic problems

6.1.1 Quadrature hyperparameters

The initial quadrature hyperparameters are as follows:
A kernel length scale $l = 2$
A kernel variance $v' = 2$
Recombination sample size $N = 20,000$
Nyström sample size $M = N/100$
Supersample ratio $r_{\text{super}} = 100$
Proposal distribution $g(x)$ partition ratio $r = 0.5$

The supersample ratio r_{super} is the ratio of supersamples for SMC sampling of acquisition function against the recombination sample size N.

A kernel length scale and a kernel variance are important for selecting the samples in the first batch. Nevertheless, these parameters are updated via type-II MLE optimisation after the second round. Nyström sample size must be larger than the batch size n, and the recombination sample size is preferred to satisfy $N \gg M$. Larger N and M give more accurate sample selection via kernel quadrature. However, larger subsamples result in a longer wall-time. We do not need to change the values as long as the integral converged to the designated criterion. When longer computational time is allowed, or likelihood is expensive enough to regard recombination time as negligible, larger N, M will give us a faster convergence rate.

The partition ratio r is the only hyperparamter that affects the convergence sensitively. The optimal value depends the integrand and it is challenging to know the optimal value before running. As we derived in Lemma 1, $\sqrt{C_{\mathbf{y}}^{L}}\pi(x)$ gives the optimal upper bound. $r = 0.5$ is a good approximation of this optimal proposal distribution: $g(x) = (1-r)\pi(x) + rC_{\mathbf{y}}^{L}\pi(x) = \left\{(1-r) + rC_{\mathbf{y}}^{L}\right\}\pi(x)$. Here,

the linearisation gives the approximation $\sqrt{C_{\mathbf{y}}^L} = \sqrt{1 + (C_{\mathbf{y}}^L - 1)} \approx 1 + \frac{C_{\mathbf{y}}^L - 1}{2} = 0.5 + 0.5C_{\mathbf{y}}^L$. Therefore, $(0.5 + 0.5C_{\mathbf{y}}^L)\pi(x) \approx \sqrt{C_{\mathbf{y}}^L}\pi(x)$. Thus, $r = 0.5$ is a safe choice.

6.1.2 Gaussian mixture

The likelihood function of the Gaussian mixture used in Figure 1 in the main paper is expressed as:

$$\ell_{\text{true}}(x) = \sum_{i=1}^{n} w_i \mathcal{N}(x; \mu_i, \mathbf{\Sigma}_i) \tag{132}$$

$$w_i = \mathcal{N}(\mu_i; \mu_\pi, \mathbf{\Sigma}_i + \mathbf{\Sigma}_\pi)^{-1} \tag{133}$$

$$Z_{\text{true}} = \int \ell_{\text{true}}(x)\pi(x)dx, \tag{134}$$

$$= \sum_{i=1}^{n} w_i \int \mathcal{N}(x; \mu_i, \mathbf{\Sigma}_i)\mathcal{N}(x; \mu_\pi, \mathbf{\Sigma}_\pi)dx \tag{135}$$

$$= \sum_{i=1}^{n} w_i \mathcal{N}(\mu_i; \mu_\pi, \mathbf{\Sigma}_i + \mathbf{\Sigma}_\pi) \tag{136}$$

$$= 1 \tag{137}$$

where
$\mu_\pi = \mathbf{0}$
$\mathbf{\Sigma}_\pi = 2\mathbf{I}$
$\pi(x) = \mathcal{N}(x; \mu_\pi, \mathbf{\Sigma}_\pi)$

The prior is the same throughout the synthetic problems.

6.1.3 Branin-Hoo function

The Branin-Hoo function in Figure 2 in the main paper is expressed as:

$$\ell_{\text{true}}(x) = \prod_{i=1}^{2} \frac{\left[\sin(x_i) + \frac{1}{2}\cos(3x_i)\right]^2}{(\frac{1}{2}x_i)^2 + \frac{3}{10}}, \quad x \in \mathbb{R}^2 \tag{138}$$

$$Z_{\text{true}} = \int \ell_{\text{true}}(x)\pi(x)dx \tag{139}$$

$$= 0.955728^2 \tag{140}$$

$$\approx 0.913416 \tag{141}$$

6.1.4 Ackley function

The Ackley function in Figure 2 in the main paper is expressed as:

$$\ell_{\text{true}}(x) = -20\exp\left(-\frac{1}{5}\sqrt{\frac{1}{2}\sum_{i=1}^{2} x_i^2}\right) + \exp\left(\frac{1}{2}\sum_{i=1}^{2}\cos(2\pi x_i)\right) + 20, \quad x \in \mathbb{R}^2 \tag{142}$$

$$Z_{\text{true}} = \int \ell_{\text{true}}(x)\pi(x)dx \tag{143}$$

$$\approx 5.43478 \tag{144}$$

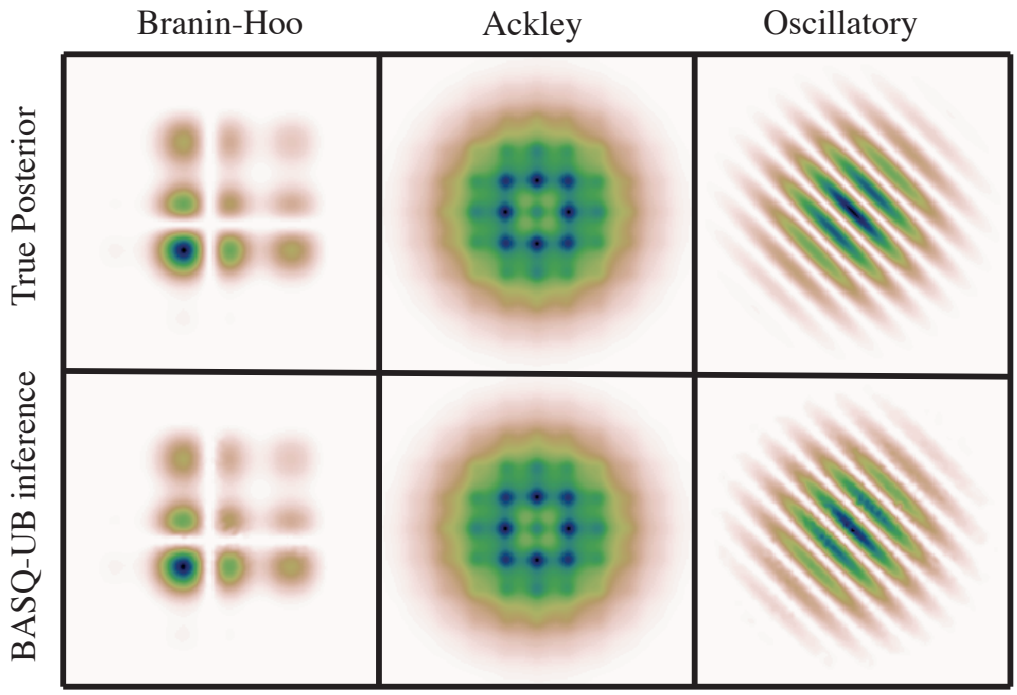


Figure 2: Performance comparison with N-dimensional Gaussian mixture likelihood function. (a) dimension study, (b) convergence rate, and (c) wall time vs MAE of integral. (a) varies from 2 to 16 dimensions, (b) and (c) are 10 dimensional Gaussian mixture.

6.1.5 Oscillatory function

The Oscillatory function in Figure 2 in the main paper is expressed as:

$$\ell_{\text{true}}(x) = \cos\left(2\pi + 5\sum_{i=1}^{2} x_i\right) + 1, \quad x \in \mathbb{R}^2 \tag{145}$$

$$Z_{\text{true}} = \int \ell_{\text{true}}(x)\pi(x)dx \tag{146}$$

$$= 1 \tag{147}$$

6.1.6 Additional experiments

Dimensional study in Gaussian mixture likelihood Figure 2(a) shows the dimension study of Gaussian mixture likelihood. The BASQ and BQ are conditioned at the same time budget (200 seconds). The higher dimension gives a more inaccurate estimation. From this result, we recommend using BASQ with fewer than 16 dimensions.

Ablation study We investigated the influence of the approximation we adopted using 10 dimensional Gaussian mixture likelihood. The compared models are as follows:

1. Exact sampler (without factorisation trick)
2. Provable recombination (without LP solver)

The exact sampler without the factorisation trick is the one that exactly follows the Eqs. (8) - (10) of the main paper. That is, the distribution of interest $f(x)$ is the prior $\pi(x)$. In addition, the kernel for the acquisition function is an unwarped $C^L_{\mathbf{y}}$, which is computationally expensive. Next, the provable recombination algorithm is the one introduced in [17] with the best known computational complexity. As explained in the Background section of the main paper, our BASQ implementation is based on an

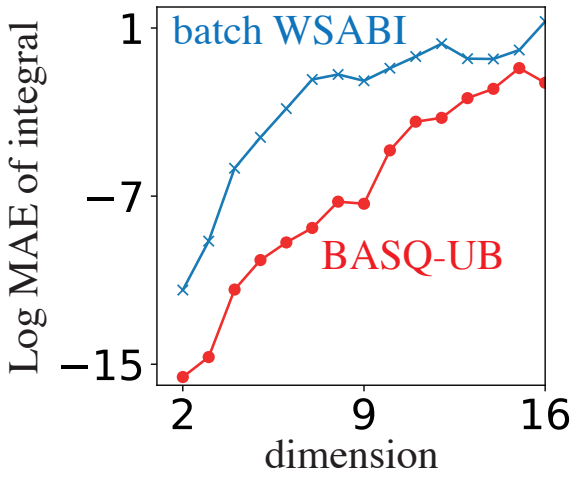


Figure 3: Qualitative evaluation of posterior inference in synthetic problems

LP solver (Gurobi [31] for this time) with empirically faster computational time. We compared the influence of these approximations.

Figure 2(b) illustrates that these approximations are not affecting the convergence rate in the sample efficiency. However, when compared to the wall-clock time (Figure 2(c)), the exact sampler without the factorisation trick is apparently slow to converge. Moreover, the provable recombination algorithm is slower than an LP solver implementation. Thus, the number of samples the provable recombination algorithm per wall time is much smaller than the LP solver. Therefore, our BASQ standard solver delivers solid empirical performance.

Qualitative evaluation of posterior inference Figure 3 shows the qualitative evaluation of joint posterior inference after 200 seconds passed against the analytical true posterior. The estimated posterior shape is exactly the same as the ground truth.

6.2 Real-world problems

6.2.1 Battery simulator

Background Single Particle Model with electrolyte dynamics (SPMe) is a commonly-used lithium-ion battery simulator to predict the voltage response at given excitation current time-series data. Estimating SPMe parameters from observations are well known for ill-conditioned problem because this model is overparameterised [4]. In the physical model, we need to separate the anode and cathode internal states to represent actual cell components. However, when it comes to predicting the voltage response, this separation into two components is redundant. Except for extreme conditions such as low temperature, most voltage responses can be expressed with a single component. Therefore, the parameters of cathode and anode often have a perfect negative correlation, meaning an arbitrary combination of cathode and anode parameters can reconstruct the exactly same voltage profile. As such, point estimation means nothing in these cases. Bayesian inference can capture this negative correlation as covariance. Therefore, Bayesian inference is a natural choice for parameter estimation in the battery simulator. Moreover, there are many plausible battery simulators with differing levels of approximation. Selecting the model satisfying both predictability and a minimal number of parameters is crucial for faster calculation, particularly in setting up the control simulator. Therefore, Bayesian model selection with model evidence is essential. The experimental setup is basically following [2].

Problem setting We wish to infer the posterior distribution of 3 simulation parameters (D_n, D_p, σ_n), where D_n is the diffusivity on anode, D_p is the diffusivity on cathode, σ_n is the

noise variance of the observed data. We have the observed time-series voltage **y** and exciation profiles **i** as training dataset.

The parameter inference is modelled as follows:

$$y_* = \text{Sim}(x_*, i_*) \tag{148}$$
$$\pi(x_*) = \text{Lognormal}(x_*; \mu_\pi, \mathbf{\Sigma}_\pi) \tag{149}$$
$$\ell_{\text{true}}(x_*) = \mathcal{N}\left[\text{Sim}(x_*, \mathbf{i}); \mathbf{y}, \sigma_n \mathbf{I}\right] \tag{150}$$

where
$\mu_\pi = [1.38, 0, -20.25]$
$\mathbf{\Sigma}_\pi = \text{diag}([0.03, 0.001, 0.001])$
in the logarithmic space.

Parameters The observed data **y** and **i** are generated by the simulator with multiharmonic sinusoidal excitation current defined as:

$$\mathbf{i} = 0.132671\left[\sin(1/5\pi t) + \sin(2\pi t) + \sin(20\pi t) + \sin(200\pi t)\right] \tag{151}$$
$$\mathbf{y} = \text{Sim}(x_{\text{true}}, \mathbf{i}) + \sqrt{\sigma_n}\mathcal{U}[0, 1] \tag{152}$$

where
t is discretised for 10 seconds with the sampling rate of 0.00025 seconds, resulting in 40,000 data points.
x_{true} = [$\exp(1.361) \times 10^{-14}$, $\exp(0) \times 10^{-13}$, $\exp(-20.25) \times 10^{-10}$]

Metric The posterior distribution is evaluated via RMSE between true and inferred conditional posterior on each parameter. The RMSE is calculated on 50 grid samples for each dimension so as to slice the maximum value of the joint posterior. Each 50 grid samples are equally-spaced and bounded with the following boundaries:
bounds $= [1.1, 1.7], [-0.075, 0.08], [-20.3, -20.2]$
where the boundaries are given by [lower, upper] in the logarithmic space.

6.2.2 Phase-field model

Background The PFM is a flexible time-evolving interfacial physical model that can easily incorporate the multi-physical energy [11]. In this dataset, the PFM is applied to the simulation of spinodal decomposition, which is the self-organised nanostructure in the bistable Fe-Cr alloy at high temperatures. Spinodal decomposition is an inherently stochastic process, making characterisation challenging [12]. Therefore, Bayesian model selection is promising for estimating its parameter and determining the model physics component.

Problem setting We wish to infer the posterior distribution of 4 simulation parameters (T, L_{cT}, n_B, L_g), where T is the temperature, L_{cT} is the interaction parameter that defines the interaction between composition and temperature, n_B is the number of Bohr magnetons per atom, and L_g is the gradient energy coefficient. We have the observed time-series 2-dimensional images **y**.

The parameter inference is modelled as follows:

$$y_* = \text{Sim}(x_*) \tag{153}$$
$$\pi(x_*) = \text{Lognormal}(x_*; \mu_\pi, \mathbf{\Sigma}_\pi) \tag{154}$$
$$\ell_{\text{true}}(x_*) = \mathcal{N}\left[\text{Sim}(x_*); \mathbf{y}, \sigma_n \mathbf{I}\right] \tag{155}$$

where
$\sigma_n = 10^{-4}$
$\mu_\pi = [1.91, 0.718, 0.798, 0.693]$
$\mathbf{\Sigma}_\pi = \text{diag}([0.0003, 0.00006, 0.0001, 0.0001])$
in the logarithmic space.

Parameters The observed data $\mathbf{y}$ is generated by the simulator defined as:

$$\mathbf{y} = \text{Sim}(x_{\text{true}}) + \sqrt{\sigma_n}\mathcal{U}[0,1] \tag{156}$$

where
$\mathbf{y}$ is discretised in both spatially and time-domain. Time domain is discretised for 5000 seconds with the sampling rate of 1 seconds, resulting in 5,000 data points. 2-dimensional space is discretised for 64×64 nm^2, with 64×64 nm^2 pixels. The total data points are $64 \times 64 \times 5,000 = 20,480,000$.
x_{true} = [$\exp(1.90657514) \times 10^2$, $\exp(0.71783979) \times 10^4$, $\exp(0.7975072)$, $\exp(0.69314718) \times 10^{-15}$]

Metric The posterior distribution is evaluated via RMSE between true and inferred conditional posterior on each parameter. The RMSE is calculated on 50 grid samples for each dimension so as to slice the maximum value of the joint posterior. Each 50 grid samples are equally-spaced and bounded with the following boundaries:
bounds $= [1.87, 1.94], [0.69, 0.73], [0.77, 0.83], [0.68, 0.73]$
where the boundaries are given by [lower, upper] in the logarithmic space.

6.2.3 Hyperparameter marginalisation of hierarchical GP

Background The hierarchical GP model was designed for analysing the large-scale battery time-series dataset from solar off-grid system field data all over the African continent [1]. The field data contains the information of time-series operating conditions (I, T, V), where I is the excitation current, T is the temperature, and V is the voltage. We wish to estimate the state of health (SoH) from these field data, achieving the preventive battery replacement before it fails for the convenience of those who rely on the power system for their living. However, estimating the state of health is challenging because the raw data (I, T, V) is not correlated to the battery health. There are several definitions of SoH, but the internal resistance of a battery R is adopted in [1]. In the usual circuit element, resistance can be easily calculated from $R = V/I$ via Ohm's law. However, the battery internal resistance R is way more complex. Battery internal resistance R is a function of (t, I, T, c), where t is time, c is the acid concentration. Furthermore, there are two factors of resistance variation; ionic polarisation and aging. To incorporate these physical insights to the machine learning model, [1] is adopted the hierarchical GP model. First, they adopted the additive kernel of a squared exponential kernel and a Wiener velocity kernel to divide the ionic polarisation effect and aging effect. Second, they adopted the hierarchical GPs to model V to divide into R-dependent GP and non-R-dependent GP to incorporate the Open Circuit Voltage-State of Charge (OCV-SOC) relationship.

Problem setting We wish to infer the hyperposterior distribution of 5 GP hyperparameters $(l_T, l_I, l_c, \sigma_0, \sigma_1)$, where l_T, l_I, l_c are the a squared exponential kernel lengthscale of temperature T, current I, and acid concentration c, respectively, and σ_0, σ_1 are the kernel variances of a squared exponential kernel and a Wiener velocity kernel, respectively. We have the observed time-series dataset of (I, T, V) as $\mathbf{y}$.

The hyperposterior inference is based on the energy function $\Phi(x)$ (The details can be found in [1], Equation (15) in the Appendix information).

$$\Phi x = -\log p(\mathbf{y}|x) - \log p(x) \tag{157}$$

$$= -\log p(x) + \frac{1}{2}\sum_t \log|S_t(x)| + \frac{1}{2}\sum_t e_t^{\mathsf{T}} S_t(x)^{-1} e_t + \sum_t \frac{n_t}{2}\log 2\pi \tag{158}$$

where
$p(x) = \text{Lognormal}(x_*; \mu_\pi, \mathbf{\Sigma}_\pi)$ is a hyperprior.
e_t is the error vector for each charging segment.
n_t is the number of observations in the charging segment.
$S_t(x)$ is the innovation covariance for the segment.
$\mu_\pi = [3.96, 1.94, 2.79, 2.26, 0.34]$
$\mathbf{\Sigma}_\pi = \text{diag}([1, 1, 1, 1, 1])$
in the logarithmic space.

Metric The posterior distribution is evaluated via RMSE between true and inferred conditional posterior on each parameter. The RMSE is calculated on 50 grid samples for each dimension so as to slice the maximum value of the joint posterior. Each 50 grid samples are equally-spaced and bounded with the following boundaries:
bounds $= [-10, 10], [-10, 10], [-10, 10], [-10, 10], [-10, 10]$
where the boundaries are given by [lower, upper] in the logarithmic space.

References

[1] A. Aitio and D. A. Howey. Predicting battery end of life from solar off-grid system field data using machine learning. *Joule*, 5(12):3204–3220, 2021.

[2] A. Aitio, S. G. Marquis, P. Ascencio, and D. A. Howey. Bayesian parameter estimation applied to the Li-ion battery single particle model with electrolyte dynamics. *IFAC-PapersOnLine*, 53(2):12497–12504, 2020.

[3] F. Bach. On the equivalence between kernel quadrature rules and random feature expansions. *Journal of Machine Learning Research*, 18:714, 2017.

[4] A. M. Bizeray, J. H. Kim, S. R. Duncan, and D. A. Howey. Identifiability and parameter estimation of the single particle lithium-ion battery model. *IEEE Transactions on Control Systems Technology*, 27(5):1862–1877, 2019. URL: `https://doi.org/10.1109/TCST.2018.2838097`.

[5] F.-X. Briol, C. J. Oates, M. Girolami, and M. A. Osborne. Frank-Wolfe Bayesian quadrature: Probabilistic integration with theoretical guarantees. In *International Conference on Neural Information Processing Systems (NeurIPS)*, 2015. URL: `https://proceedings.neurips.cc/paper/2015/file/ba3866600c3540f67c1e9575e213be0a-Paper.pdf`.

[6] F.-X. Briol, C. J. Oates, M. Girolami, M. A. Osborne, and D. Sejdinovic. Probabilistic integration: a role in statistical computation? *Statistical Science*, 34(1):1–22, 2019.

[7] Gregory E. Fasshauer and Michael J. McCourt. Stable evaluation of Gaussian radial basis function interpolants. *SIAM Journal on Scientific Computing*, 34(2):A737–A762, 2012. URL: `https://doi.org/10.1137/110824784`.

[8] Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex Smola. A kernel method for the two-sample-problem. *Advances in neural information processing systems*, 19, 2006.

[9] S. Hayakawa, H. Oberhauser, and T. Lyons. Positively weighted kernel quadrature via subsampling. In *International Conference on Neural Information Processing Systems (NeurIPS)*, 2022. `doi:10.48550/arXiv.2107.09597`.

[10] F. Hutter, H. Hoos, and K. Leyton-Brown. An efficient approach for assessing hyperparameter importance. In *International Conference on Machine Learning (ICML)*, pages 754–762, 2014.

[11] S. G. Kim, W. T. Kim, and T. Suzuki. Phase-field model for binary alloys. *Physical review e*, 60(6):7186, 1999.

[12] Y. Matsuura, Y. Tsukada, and T. Koyama. Adjoint model for estimating material parameters based on microstructure evolution during spinodal decomposition. *Physical Review Materials*, 5(11):113801, 2021.

[13] C. J. Oates, M. Girolami, and N. Chopin. Control functionals for Monte Carlo integration. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(3):695–718, 2017.

[14] C. J. Oates, T. Papamarkou, and M. Girolami. The controlled thermodynamic integral for bayesian model evidence evaluation. *Journal of the American Statistical Association*, 111(514):634–645, 2016.

[15] M. A. Osborne, D. Duvenaud, R. Garnett, C. Rasmussen, S. Roberts, and Z. Ghahramani. Active learning of model evidence using Bayesian quadrature. In *International Conference on Neural Information Processing Systems (NeurIPS)*, 2012. URL: `https://proceedings.neurips.cc/paper/2012/file/6364d3f0f495b6ab9dcf8d3b5c6e0b01-Paper.pdf`.

[16] Bharath K Sriperumbudur, Arthur Gretton, Kenji Fukumizu, Bernhard Schölkopf, and Gert RG Lanckriet. Hilbert space embeddings and metrics on probability measures. *The Journal of Machine Learning Research*, 11:1517–1561, 2010.

[17] M. Tchernychova. *Carathéodory cubature measures*. PhD thesis, University of Oxford, 2015.

[18] G. Wahba. *Spline models for observational data*. SIAM, 1990.

B

Appendix of Chapter 4

Part I
Appendix

Table of Contents

A Proof of Proposition 1

Proof of Proposition 1. Note that the constraint $|\mathbf{w}|_0 \leq n$ is automatically satisfied when we use the simplex method or its variant. Without this constraint, we have a trivial feasible solution $\mathbf{w} = \mathbf{w}_{\mathrm{cand}}$, so, for the optimal solution $\mathbf{w}_*$, we have $\mathbf{w}_*^\top \left[g(\mathbf{X}_{\mathrm{cand}}) \odot q(\mathbf{X}_{\mathrm{cand}})\right] \geq \mathbf{w}_{\mathrm{cand}}^\top \left[g(\mathbf{X}_{\mathrm{cand}}) \odot q(\mathbf{X}_{\mathrm{cand}})\right]$. Since $\mathbb{E}[\tilde{\mathbf{w}}_{\mathrm{batch}}^\top g(\tilde{\mathbf{X}}_{\mathrm{batch}})] = \mathbf{w}_{\mathrm{batch}}^\top \left[g(\mathbf{X}_{\mathrm{batch}}) \odot q(\mathbf{X}_{\mathrm{batch}})\right] = \mathbf{w}_*^\top \left[g(\mathbf{X}_{\mathrm{cand}}) \odot q(\mathbf{X}_{\mathrm{cand}})\right]$, we obtain the first estimate Eq. (8).

For the latter estimate, we first decompose the error into two parts:

$$\begin{aligned}
&\mathbb{E}\left[\left|\tilde{\mathbf{w}}_{\mathrm{batch}}^\top f(\tilde{\mathbf{X}}_{\mathrm{batch}}) - \mathbf{w}_{\mathrm{cand}}^\top f(\mathbf{X}_{\mathrm{cand}})\right|\right] \\
&\leq \mathbb{E}\left[\left|\tilde{\mathbf{w}}_{\mathrm{batch}}^\top f(\tilde{\mathbf{X}}_{\mathrm{batch}}) - \mathbf{w}_{\mathrm{batch}}^\top f(\mathbf{X}_{\mathrm{batch}})\right|\right] + \left|\mathbf{w}_{\mathrm{batch}}^\top f(\mathbf{X}_{\mathrm{batch}}) - \mathbf{w}_{\mathrm{cand}}^\top f(\mathbf{X}_{\mathrm{cand}})\right|. \qquad (10)
\end{aligned}$$

For the first term, considering each $x \in \mathbf{X}_{\mathrm{batch}}$ on whether or not it gets included in $\tilde{\mathbf{X}}_{\mathrm{batch}}$, we have

$$\begin{aligned}
&\mathbb{E}\left[\left|\tilde{\mathbf{w}}_{\mathrm{batch}}^\top f(\tilde{\mathbf{X}}_{\mathrm{batch}}) - \mathbf{w}_{\mathrm{batch}}^\top f(\mathbf{X}_{\mathrm{batch}})\right|\right] \\
&\leq \mathbf{w}_{\mathrm{batch}}^\top \left[|f|(\mathbf{X}_{\mathrm{batch}}) \odot (1-q)(\mathbf{X}_{\mathrm{batch}})\right] \leq \mathbf{w}_{\mathrm{batch}}^\top (1-q)(\mathbf{X}_{\mathrm{batch}}) \max_{x \in \mathbf{X}_{\mathrm{batch}}} |f(x)| \\
&= \left[1 - \mathbf{w}_{\mathrm{batch}}^\top q(\mathbf{X}_{\mathrm{batch}})\right] \max_{x \in \mathbf{X}_{\mathrm{batch}}} |f(x)| \leq \left[1 - \mathbf{w}_{\mathrm{cand}}^\top q(\mathbf{X}_{\mathrm{cand}})\right] \max_{x \in \mathbf{X}_{\mathrm{batch}}} |f(x)|,
\end{aligned}$$

where the last inequality follows from the inequality constraint $(\mathbf{w} - \mathbf{w}_{\mathrm{cand}})^\top q(\mathbf{X}_{\mathrm{cand}}) \geq 0$ in the LP. Since $|f(x)| = |\langle f, K_{\mathrm{LP}}(\cdot, x)\rangle| \leq \|f\| K_{\mathrm{LP}}(x,x)^{1/2}$ from the reproducing property of RKHS, we obtain

$$\mathbb{E}\left[\left|\tilde{\mathbf{w}}_{\mathrm{batch}}^\top f(\tilde{\mathbf{X}}_{\mathrm{batch}}) - \mathbf{w}_{\mathrm{batch}}^\top f(\mathbf{X}_{\mathrm{batch}})\right|\right] \leq \epsilon_{\mathrm{rej}} K_{\max} \|f\|. \qquad (11)$$

Let us then bound the second term of the RHS of Eq. (10). Note that, from the formula of worst-case error of kernel quadrature (see, e.g., (Hayakawa et al., 2022, Eq. (14))), we can bound

$$\left|\mathbf{w}_{\mathrm{batch}}^\top f(\mathbf{X}_{\mathrm{batch}}) - \mathbf{w}_{\mathrm{cand}}^\top f(\mathbf{X}_{\mathrm{cand}})\right|^2 \leq \|f\|^2 (\mathbf{w}_* - \mathbf{w}_{\mathrm{cand}})^\top K_{\mathrm{LP}}(\mathbf{X}_{\mathrm{cand}}, \mathbf{X}_{\mathrm{cand}})(\mathbf{w}_* - \mathbf{w}_{\mathrm{cand}}) \qquad (12)$$

(recall $\mathbf{w}_*$ has the same dimension as $\mathbf{w}_{\mathrm{cand}}$). We now want to estimate

$$(\mathbf{w}_* - \mathbf{w}_{\mathrm{cand}})^\top K_{\mathrm{LP}}(\mathbf{X}_{\mathrm{cand}}, \mathbf{X}_{\mathrm{cand}})(\mathbf{w}_* - \mathbf{w}_{\mathrm{cand}}).$$

Consider approximating K_{LP} by K_{nys}. Since $K_{\mathrm{LP}} - K_{\mathrm{nys}}$ is positive semi-definite from the property of Nyström approximation (see, e.g., the proof of (Hayakawa et al., 2022, Corollary 4)), for any $x, y \in \mathbf{X}_{\mathrm{cand}}$, we have

$$|(K_{\mathrm{LP}} - K_{\mathrm{nys}})(x,y)| \leq |(K_{\mathrm{LP}} - K_{\mathrm{nys}})(x,x)|^{1/2} |(K_{\mathrm{LP}} - K_{\mathrm{nys}})(y,y)|^{1/2} \leq \epsilon_{\mathrm{nys}}^2.$$

Thus, we have

$$(\mathbf{w}_* - \mathbf{w}_{\text{cand}})^\top \Big[(K_{\text{LP}} - K_{\text{nys}})(\mathbf{X}_{\text{cand}}, \mathbf{X}_{\text{cand}})\Big](\mathbf{w}_* - \mathbf{w}_{\text{cand}})$$
$$\leq (\mathbf{w}_* + \mathbf{w}_{\text{cand}})^\top(\epsilon_{\text{nys}}^2 \mathbf{1}\mathbf{1}^\top)(\mathbf{w}_* + \mathbf{w}_{\text{cand}}) = 4\epsilon_{\text{nys}}^2. \tag{13}$$

Finally, we estimate

$$\begin{aligned}
&(\mathbf{w}_* - \mathbf{w}_{\text{cand}})^\top K_{\text{nys}}(\mathbf{X}_{\text{cand}}, \mathbf{X}_{\text{cand}})(\mathbf{w}_* - \mathbf{w}_{\text{cand}}) \\
&= (\mathbf{w}_* - \mathbf{w}_{\text{cand}})^\top \sum_{j=1}^{n-2} \mathbf{1}_{\{\lambda_j > 0\}} \lambda_j^{-1} \varphi_j(\mathbf{X}_{\text{cand}}) \varphi_j(\mathbf{X}_{\text{cand}})^\top (\mathbf{w}_* - \mathbf{w}_{\text{cand}}) \\
&= \sum_{j=1}^{n-2} \mathbf{1}_{\{\lambda_j > 0\}} \lambda_j^{-1} \left[(\mathbf{w}_* - \mathbf{w}_{\text{cand}})^\top \varphi_j(\mathbf{X}_{\text{cand}})\right]^2.
\end{aligned} \tag{14}$$

From the inequality constraint in the LP, we have $|(\mathbf{w}_* - \mathbf{w}_{\text{cand}})^\top \varphi_j(\mathbf{X}_{\text{cand}})| \leq \epsilon_{\text{LP}} \sqrt{\lambda_j/(n-2)}$, so that Eq. (14) is further bounded as

$$(\mathbf{w}_* - \mathbf{w}_{\text{cand}})^\top K_{\text{nys}}(\mathbf{X}_{\text{cand}}, \mathbf{X}_{\text{cand}})(\mathbf{w}_* - \mathbf{w}_{\text{cand}}) \leq \sum_{j=1}^{n-2} \mathbf{1}_{\{\lambda_j > 0\}} \lambda_j^{-1} \epsilon_{\text{LP}}^2 \frac{\lambda_j}{n-2} \leq \epsilon_{\text{LP}}^2. \tag{15}$$

By adding the both sides of Eqs. (13) and (15), we obtain

$$(\mathbf{w}_* - \mathbf{w}_{\text{cand}})^\top K_{\text{LP}}(\mathbf{X}_{\text{cand}}, \mathbf{X}_{\text{cand}})(\mathbf{w}_* - \mathbf{w}_{\text{cand}}) \leq 4\epsilon_{\text{nys}}^2 + \epsilon_{\text{LP}}^2 \leq (2\epsilon_{\text{nys}} + \epsilon_{\text{LP}})^2.$$

By applying this to Eq. (12), we have $\left|\mathbf{w}_{\text{batch}}^\top f(\mathbf{X}_{\text{batch}}) - \mathbf{w}_{\text{cand}}^\top f(\mathbf{X}_{\text{cand}})\right| \leq \|f\|(2\epsilon_{\text{nys}} + \epsilon_{\text{LP}})$. Combining this with Eqs. (10) and (11) yields the desired inequality Eq. (9). □

B Experimental Details

B.1 Training details

We have tested AdaBatAL for 7 synthetics and 7 real-world tasks for batch AL and BO tasks. Our experiments were repeated 10 times and took a mean and one standard error with different random seeds (the seeds are shared with baseline methods). We use FBGP for batch AL tasks, and simple GP with type-II maximum likelihood estimation for batch BO tasks. The kernel is different for each task but shared with baseline methods (see details in the dataset section). We randomly generated 10 samples as the initial dataset $\mathcal{D}_0$. We use different batch sizes for each task (see details in the dataset section). While the fixed batch size methods simply adopt this as the batch size, AdaBatAL sets this as the upper bound of batch sizes. This means the AdaBatAL tends to query a smaller number of samples than fixed batch size methods. We iterated this batch acquisition process for the fixed iteration times and compared the best-observed values at the last round. For the fair comparison with the adaptive batch size method, we employ the accumulated queries as the metric, which counts the total number of queries at the t-th iteration. As explained, AdaBatAL yields the smaller accumulated queries with the same iteration times. For constrained cases, we removed the violated samples. Thus, constrained tasks yield smaller accumulated queries than unconstrained cases even with the same batch sizes and the same iteration times. Surprisingly, non-adaptive batch baselines tend to have smaller batch sizes than adaptive AdaBatAL due to constraint violation (See Figure 5).

Our code is built upon PyTorch-based libraries (Paszke et al., 2019; Gardner et al., 2018; Balandat et al., 2020; Griffiths et al., 2022) and Gurobipy (Gurobi Optimization, LLC, 2024) is used to solve the linear programming. All baseline methods are official implementations in BoTorch or coded with BoTorch (Balandat et al., 2020).

Batch Bayesian Optimization We use a constant-mean GP with either RBF, Tanimto, or graph diffusion kernel for batch BO tasks. In each iteration of the active learning loop, the outputs are standardized to have zero mean and unit variance. We optimize the hyperparameter by maximizing the marginal likelihood (type-II maximum likelihood estimation) using L-BFGS-B optimizer (Liu & Nocedal, 1989) implemented with BoTorch

(Balandat et al., 2020). The initial data sets consist of ten data points drawn by Sobol sequence (Sobol', 1967), and in each iteration, multiple data points are queried as the batch acquisition (upper bound for AdaBatAL). We adopt log regret if the true global maxima are known, otherwise, the log of best-observed value is the evaluation metric using the test dataset. The models are implemented in GPyTorch (Gardner et al., 2018). All experiments are repeated ten times with different initial data sets via different random seeds.

Batch active learning We use a zero-mean GP with an RBF kernel for all batch AL tasks. In each iteration of the active learning loop, the inputs are rescaled to the unit cube $[0, 1]^d$, and the outputs are standardized to have zero mean and unit variance. Following Lalchand & Rasmussen (2020), we give all the hyperparameters relatively uninformative $\mathcal{N}(0, 3)$ lognormal priors. The initial data sets consist of ten data points drawn by Sobol sequence (Sobol', 1967), and in each iteration, 10 data points are queried as the batch acquisition (upper bound for AdaBatAL). The unlabeled pool consists of the 10,000 data points drawn by Sobol sequence all over the domain. We used this unlabelled pool and corresponding true values as the test dataset for the evaluation. We adopt negative log marginal likelihood (NLML) as the evaluation metric using the test dataset. The inference in FBGP is carried out using NUTS (Hoffman & Gelman, 2014) in Pyro (Bingham et al., 2019) with five chains and 500 samples, including a warm-up period with 200 samples. The remaining 1500 samples are all used for the acquisition functions. The models are implemented in GPyTorch (Gardner et al., 2018). All experiments are repeated ten times with different initial data sets via different random seeds.

For batch AL, we typically assume training a model is very expensive (e.g. deep learning). FBGP is expensive to train even with parallel chains. Thus, we exclude methods like hallucination that require the sequential update of the model to select multiple points. This assumption is widely shared with the AL community (e.g. Kirsch et al. (2019); Pinsler et al. (2019)). Moreover, all baseline batch AL methods do not consider probabilistic constraints. We simply follow the constrained BO approaches.

Extension to non-continuous input domain Almost all methods are not compatible with categorical and mixed input spaces due to the continuity assumption in these methods. To enable comparison against these methods, we adopt the nearest neighbor in discrete or mixed problems: namely, we optimise the discrete variables as bounded continuous variables, then the selected continuous locations are classified into the closest original discrete values. For the graph space, we deem the search space itself to be a graph and the objective is to find a subgraph. This is different from, for example, the drug discovery problem, whose input variables are graphs but the space itself is a non-Euclidean discrete set of drugs. In contrast, the graph space is over the large graph, and the graph example is only one. Thus, cTS is the only method applicable to graph space other than AdaBatAL.

Extension to constrained cases We simply follow the constrained BO approaches; Modelling the probabilistic constraints by GPs and multiplying the probability of constraint satisfaction to the acquisition function.

Training details of AdaBatAL For AdaBatAL, we have two hyperparameters; the number of Nyström samples M, and the tolerance ϵ_{LP}. The number of unlabeled pools N, the batch sizes n, and M need to satisfy the relationship $N \gg M \geq n$. We fixed $M = 500$. The larger M yields tighter error bounds for worst-case error but it slows down the computation. We find $M = 500$ works well over the tasks we have tested. For ϵ_{LP}, this is automatically determined for the constrained case via $\epsilon_{\text{LP}} = \epsilon_{\text{vio}}$. For unconstrained cases, we set $\epsilon_{\text{LP}} = 0.01$. For reward function g, we set B-QBC (Riis et al., 2022) for batch AL, and no reward function is set for batch BO. The probabilistic constraints q were modeled by GP as explained. For the intractable expectation of kernel means, we generate $N = 20,000$ data points from the probability distribution μ.

B.2 Baseline Implementations

Table 1 summarizes all baselines. Our method, AdaBatAL, is the only method that can offer adaptive batch size under probabilistic constraints for both AL and BO tasks.

B.2.1 Batch Bayesian Optimization

B3O Budgeted Batch Bayesian Optimization (B3O) (Nguyen et al., 2016) is the only baseline method that offers the adaptive batch size. B3O recasts batch construction as the approximation of acquisition function using a mixture of Gaussians. The adaptive batch size is determined through the marginal likelihood of Gaussian

Table 1: Summary of baseline method. cBO refers to constrained BO.

method	task	adaptive?	constraints?	discrete?	large batch?	any kernel?	any AF?
random	any			✓	✓	✓	✓
AdaBatAL (ours)	any	✓	✓	✓	✓	✓	✓
B3O	BO	✓				✓	✓
TS	BO			✓	✓	✓	
hallucination	BO			✓		✓	✓
LP	BO				✓		✓
TurBO	BO				✓		✓
SOBER	BO			✓	✓	✓	✓
MaxEnt	AL			✓	✓	✓	
BALD	AL			✓	✓	✓	
B-QBC	AL			✓	✓	✓	
ACS-FW	AL			✓	✓	✓	✓
cEI	cBO		✓		✓	✓	
cTS	cBO		✓	✓	✓	✓	
SCBO	cBO		✓		✓		
PropertyDAG	cBO		✓		✓		
PESC	cBO		✓	✓		✓	

mixture model; the number of Gaussians corresponds to the batch size, and select the batch sizes that yield the largest marginal likelihood, following the standard Bayesian model selection procedure. However, original B3O cannot apply to AL and constrained cases. Simple extension with constraining acquisition function or changing to AL acquisition function could apply to them but we do not investigate in this paper. B3O tends to select around 4-5 batch sizes regardless of the dimension, and is not applicable to large batch size. Moreover, Gaussian mixture model assumption is not always appropriate (e.g. Tanimoto kernel in drug discovery), whereas AdaBatAL naturally adopts these kernel via MMD.

Thompson sampling (TS) Thompson sampling (TS) (Hernández-Lobato et al., 2017) is a random sampling method of $P(x^* \mid \mathbf{D}_t)$ by maximising the function samples drawing from the predictive posterior. Due to its random sampling nature, exactly maximising the function samples is not strict when compared to others (e.g. hallucination). Thus, in practice, TS is typically done by taking argmax of function samples amongst the candidates of random samples over input space. This two-step sampling nature (random samples over input space $\rightarrow$ subsamples with argmax of random function samples) allows us for domain-agnostic BO. However, this scheme itself is a type of acquisition function, so other acquisition function is not naïvely supported. Moreover, due to the random sampling nature, the selected batch samples are not sparsified to efficiently explore uncertain regions.

Hallucination Hallucination (Azimi et al., 2010) tackled batch BO by simulating a sequential process by putting 'fantasy' oracles estimated by GP, translating batch selection into a sequential problem. Hallucination is successful in low batch size n, but not scalable. Even a single iteration of acquisition function maximisation is not trivial due to non-convexity, but they repeat this over n times and produce prohibitive overhead. For discrete and mixed space, maximizing the acquisition function requires enumerating all possible candidates. However, the higher the dimension and larger the number of categorical classes, the more infeasibly large the combination becomes (combinatorial explosion).

Local penalisation (LP) Local penalisation (González et al., 2016), simulates only acquisition function shape change, without fantasy oracles, by penalising acquisition function assuming Lipschitz continuity. This succeeds in speeding up the hallucination algorithm. However, the principled limitations are inherited (combinatorial explosion). Large batch sizes are also not applicable because maximising acquisition function still produces large overhead. This is because maximising acquisition function is typically computed by a multi-start optimiser, but the number of random seeds needs to increase dependent on the number of dimensions and multimodality of the true function. This optimiser also does not guarantee to be globally maximised, which contradicts the assumption

of acquisition function (only optimal if it is globally maximised.). Furthermore, Lipschitz continuity assumption limits its applicable range to be only for continuous space.

TurBO TurBO (Eriksson et al., 2019) introduced multiple local BO bounded with trust regions, and allocates batching budgets based on TS. This succeeded in scalable batching via maintaining local BOs that are compact, via shrinking trust regions, based on heuristics with many hyperparameters. Selecting hyperpameters is non-trivial and TurBO cannot apply to discrete and non-Euclidean space, for which kernels do not have lengthscale hyperparameters for the trust region update heuristic (e.g. Tanimoto kernel for drug discovery (Ralaivola et al., 2005)).

SOBER SOBER (Adachi et al., 2022) first introduced the idea of batch BO as a kernel quadrature. Our AdaBatAL is based on SOBER when applying to batch BO tasks. However, original SOBER is not capable of adaptive batch size or constrained cases.

B.2.2 Constrained Batch Bayesian Optimization

Constrained Expected Improvement (cEI) Constrained expected improvement (cEI) (Letham et al., 2019) is the method based on constrained expected improvement acquisition function (Jones et al., 1998). cEI simply multiplies the probability of constraint satisfaction q_ℓ to the acquisition function. We adopted the official implementation on BoTorch (Balandat et al., 2020). The batching algorithm is based on sample average approximation, a standard batching methid in BoTorch library (Balandat et al., 2020).

Predictive Entropy Search with Constraints (PESC) Predictive Entropy Seach with Constraints (PESC) (Hernández-Lobato et al., 2015) is the constrained version of the predictive entropy search acquisition function (Hernández-Lobato et al., 2014). The official implementation in Spearmint is dependent on Python 2 and is no longer supported in 2023. Thus, we adopted the implementation on BoTorch (Balandat et al., 2020). The batching algorithm is based on Monte Carlo sampling following the original code. However, this code is tremendously slow, which is repeatedly pointed out in BO literature (Eriksson & Poloczek, 2021). We set 7 days as the practical limit of execution time allowing for active learning, and PESC exceeds this limit for almost all tasks except for Hartmann synthetic function. Thus, we only compare PESC on Hartmann task but it was not the best performer.

Scalable Constrained Bayesian Optimization Scalable Constrained Bayesian Optimization (SCBO) is the constrained version of TurBO based on the TS acquistion function and trust region methods. We adopted the official implementation on BoTorch (Balandat et al., 2020) and the same hyperparameters in the original papers (Eriksson et al., 2019) for trust region update heuristics.

Constrained Thompson sampling (cTS) Constrained Thompson sampling (cTS) is the constrained TS method. cTS has not been considered in existing work but this is a simple modification of SCBO. We adopted the two-step sampling used in SCBO for TS and removed the trust region heuristics because this cannot apply to a non-Euclidean kernel (e.g. Tanimoto kernel does not have lengthscale hyperparameter). This is coded based on SCBO implementation on BoTorch (Balandat et al., 2020).

PropertyDAG PropertyDAG Park et al. (2022) is the method based on qNEHVI acquisition function and (Daulton et al., 2020, 2021) for multi-objective optimization. This method assumes (1) ordered constraints but the constraint function is given, (2) multi-objective BO. So it cannot simply apply to our setting as it is. This method is the only one considering ordered case, so we dismantle the components of PropertyDAG to compare in the blackbox ordered constraint case. PropertyDAG consists of three parts: (A) explicit modelling of DAG network in surrogate model (Astudillo & Frazier, 2021), (B) zero inflation model to encode ordered constraint information to qNEHVI acquisition function, and (C) resampling of posterior function samples using sample average approximation to be more likely to satisfy the constraint. We cannot apply (A) and (B) for black-box ordered constraint, because (A) is only for white-box ordered constraint (we cannot model of unknown DAG), and (B) is only for multi-objective BO and specific acquisition function. Thus, we extracted the last part, (C) resampling with sample average approximation, and combined this with cEI, which we refer to PropertyDAG in this paper. We can say this as just resampled version of cEI. The implementation is based on cEI implementation on BoTorch (Balandat et al., 2020) and added the resampling part.

B.2.3 Batch Active Learning

Maximum entropy (MaxEnt) Maximum entropy (MaxEnt) (MacKay, 1992) is the classic acquisition function to select the next query with the largest Shannon entropy. As Riis et al. (2022) pointed out, MaxEnt in FBGP is proportional to the posterior predictive variance. We adopted the following formulation (Riis et al., 2022):

$$\text{MaxEnt} := \mathbb{H}\left[\int \mathbb{P}(y \mid x, \theta) \mathrm{d}\mathbb{P}(\theta \mid \mathcal{D}_0)\right] \propto \mathbb{E}_{\mathbb{P}(\theta \mid \mathcal{D}_0)}\left[C(x, x \mid \theta)\right] \tag{16}$$

For the batch construction, we take the top n samples following the common practice in batch AL community (Kirsch et al., 2019).

Bayesian Active Learning by Disagreement (BALD) Bayesian active learning by disagreement (BALD) (Houlsby et al., 2011) is another popular objective in Bayesian active learning, is to maximize the expected decrease in posterior entropy (Guestrin et al., 2005). Houlsby et al. (2011) recast the objective from computing entropies in the parameter space to the output space by observing that it is equivalent to maximizing the conditional mutual information between the model's parameters θ and output $\mathbb{I}[\theta, y \mid x, \mathcal{D}_0]$:

$$\text{BALD} := \mathbb{H}\left[\mathbb{E}_{\mathbb{P}(\theta \mid \mathcal{D}_0)}\left[y \mid x, \mathcal{D}_0, \theta\right]\right] - \mathbb{E}_{\mathbb{P}(\theta \mid \mathcal{D}_0)}\left[\mathbb{H}\left[y \mid x, \theta\right]\right] \tag{17}$$

Kirsch et al. (2019) pointed out the original BALD criterion is independent selection of a batch of data points leads to data inefficiency as correlations between data points in an acquisition batch are not taken into account. Instead, BatchBALD is proposed whereby we jointly score points by estimating the mutual information between a joint of multiple data points and the model parameters:

$$\text{batchBALD} := \mathbb{H}\left[\mathbb{E}_{\mathbb{P}(\theta \mid \mathcal{D}_0)}\left[y_1, \ldots, y_n \mid x_1, \ldots, x_n, \mathcal{D}_0, \theta\right]\right] - \mathbb{E}_{\mathbb{P}(\theta \mid \mathcal{D}_0)}\left[\mathbb{H}\left[y_1, \ldots, y_n \mid x_1, \ldots, x_n, \theta\right]\right] \tag{18}$$

We adopted BatchBALD formulation for batch construction.

Bayesian Query-by-Committee (B-QBC) Richardson et al. (2017) propose a Bayesian version of the Query-by-Committee (Seung et al., 1992), using the MCMC samples of the hyperparameters' joint posterior. We query a new data point where the mean predictions $m(x \mid \theta)$ disagree the most. Each mean predictor $m(\cdot \mid \theta)$ drawn from the posterior is equivalent to a single model, and thus this criteria can be seen as a Bayesian variant of a Query-by-Committee, and thus denoted as Bayesian Query-by-Committee (B-QBC). Given that $\bar{m}(x)$ is the average mean function, B-QBC is given as:

$$\text{B-QBC} := \mathbb{V}_{\mathbb{P}(\theta \mid \mathcal{D}_0)}\left[m(x \mid \theta)\right] = \mathbb{E}_{\mathbb{P}(\theta \mid \mathcal{D}_0)}\left[\left(m(x \mid \theta) - \bar{m}(x)\right)^2\right] \tag{19}$$

For the batch construction, we take the top n samples following the common practice in batch AL community (Kirsch et al., 2019).

Active Bayesian CoreSets with Frank-Wolfe optimization (ACS-FW) Active Bayesian CoreSets with Frank-Wolfe optimization (ACS-FW) recasts the batch construction as the Bayesian coreset task. Our AdaBatAL is based on ACS-FW when applying to batch AL tasks. However, original ACS-FW is not capable of adaptive batch size nor constrained cases. Also, the Bayesian coreset formulation fails to incorporate the predictive uncertainty for batch construction unlike the kernel quadrature formulation. We implemented ACS-FW via following the official code https://github.com/rpinsler/active-bayesian-coresets.

B.3 Dataset

All datasets and tasks are summarized in Table 2.

Table 2: Summary of tasks.

task	method	real/synthetic	space $\mathcal{X}$	dimension	constraints	batch size	kernel
Hartmann	BO	synthetic	continuous	6	-	5-90	RBF
Branin	cBO	synthetic	continuous	2	2	20	RBF
Hartmann	AL	synthetic	continuous	6	-	10	RBF
Ishigami	AL	synthetic	continuous	3	-	10	RBF
Friedman	AL	synthetic	continuous	5	-	10	RBF
Electrolyte	cAL	real-world	continuous	3	2	10	RBF
Cantilever	cAL	real-world	continuous	4	2	10	RBF
Steel	cAL	real-world	continuous	9	1	10	RBF
Ackley	cBO	synthetic	mixed	23	2	200	RBF
Hartmann	cBO	synthetic	continuous	6	2	5	RBF
PestControl	cBO	real-world	discrete	15	2	200	RBF
Malaria	cBO	real-world	discrete	molecule	4	100	Tanimoto
FindFixer	cBO	real-world	graph	node	3	100	graph diffusion
TeamOpt	cBO	real-world	graph	subgraph	3	100	graph diffusion

B.3.1 Synthetic Functions

Hartmann Hartmann 6-dimensional function is defined as:

$$f(x) := -\sum_{i=1}^{4} \alpha_i \exp\left(-\sum_{j=1}^{6} A_{ij}(x_j - P_{ij})^2\right), \tag{20}$$

$$\alpha = (1.0, 1.2, 3.0, 3.2)^\top, \tag{21}$$

$$\mathbf{A} = \begin{pmatrix} 10 & 3 & 17 & 3.5 & 1.7 & 8 \\ 0.05 & 10 & 17 & 0.1 & 8 & 14 \\ 3 & 3.5 & 1.7 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 0.1 & 14 \end{pmatrix}, \tag{22}$$

$$\mathbf{P} = \begin{pmatrix} 1312 & 1696 & 5569 & 124 & 8283 & 5886 \\ 2329 & 4135 & 8307 & 3736 & 1004 & 9991 \\ 2348 & 1451 & 3522 & 2883 & 3047 & 6650 \\ 4047 & 8828 & 8732 & 5743 & 1091 & 381 \end{pmatrix} \tag{23}$$

We take the negative Hartmann function as the objective of BO to make this optimisation problem maximisation. All input variables are continuous with bounds $[0, 1]^6$. The batch size n is 100. The continuous prior is the uniform distribution ranging from [0, 1], following Adachi et al. (2023a). The noisy output is generated by adding i.i.d. zero-mean Gaussian noise with the 0.0192^2 variance to the noiseless $f(x)$.

For constrained BO, we added two constraints; (1) $\sum_{i=1}^{d} x_i \geq 0.15$ and (2) $\sum_{i=1}^{d} x_i \leq 3$.

Branin Branin function is defined as:

$$f(x) := \prod_{i=1}^{d} \frac{\sqrt{\sin(x) + 0.5\cos(3x)}}{\sqrt{0.5x} + 0.3}, \tag{24}$$

where the dimension $d = 2$. All input variables are continuous with bounds $x \in [-2, 3]^d$. The batch size n is 20. The continuous prior is the uniform distribution. The noisy output is generated by adding i.i.d. zero-mean Gaussian noise with the 0.0192^2 variance to the noiseless $f(x)$.

For constrained BO, we added two constraints; (1) $\sum_{i=1}^{d} x_i^2 \leq 4$ and (2) $\sum_{i=1}^{d} x_i \leq 0$.

Ishigami Ishigami function is defined as:

$$f(x) := \sin(x_1) + 7\sin^2(x_2) + 0.1x_3^4 \sin(x_1), \tag{25}$$

where x_i is the i-th dimensional input and the dimension $d = 3$. All input variables are continuous with bounds $x \in [-\pi, \pi]^d$. The batch size n is 10. The continuous prior is the uniform distribution. The noisy output is generated by adding i.i.d. zero-mean Gaussian noise with the 0.187^2 variance to the noiseless $f(x)$.

Friedman Friedman function is defined as:

$$f(x) := 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5, \tag{26}$$

where x_i is the i-th dimensional input and the dimension $d = 5$. All input variables are continuous with bounds $x \in [0, 1]^d$. The batch size n is 10. The continuous prior is the uniform distribution. The noisy output is generated by adding i.i.d. zero-mean Gaussian noise with the 0.05^2 variance to the noiseless $f(x)$.

Ackely Ackley funciton is defined as:

$$f(x) := -a\exp\left[-b\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right] - \exp\left[\frac{1}{d}\sum_{i=1}^{d}\cos(cx_i)\right] + a + \exp(1) \tag{27}$$

where $a = 20, c = 2\pi, d = 23$. We take the negative Ackley function as the objective of BO to make this optimisation problem maximisation. We modified the original Ackley function into a 23-dimensional function with the mixed spaces of 3 continuous and 20 binary inputs from $[0, 1]^{20}$, following Adachi et al. (2023a). The batch size n is 200. The continuous prior is the uniform distribution ranging from [-1, 1]. The binary prior is the Bernoulli distribution with unbiased weights of 0.5. We assume each of the continuous and binary priors at each dimension is independent.

For constrained BO, we added the two constraints; (1) $x_1 \geq 0$ and (2) $x_2 \geq 0$, where x_1 and x_2 are the first and second dimensions of continuous inputs.

B.3.2 Real-World Functions

Electrolyte Electrolyte is the new problem for the AL task. This is the task of creating the model that predicts the ionic conductivity for the given composition of liquid electrolyte material for the next generation of lithium-ion batteries. This ionic-conductivity function is used for the control model of batteries and plays a crucial role in the control accuracy. However, common practice is to use the lookup table with massive data or pairwise linear function fitting. Collecting the ionic conductivity data requires costly laboratory experiments and fewer data points can accelerate this process while minimizing the cost. GP and AL are powerful frameworks to offer more accurate models with fewer data sizes and cheap models allowing them to be implemented in the control chip. Still, this data collection is under an unknown constraint; the freezing point. While low-temperature operation performance is the key performance indicator of batteries, it causes freezing electrolytes and cannot measure ionic conductivity. The freezing point is dependent on both lithium salt molarity and the cosolvent composition. They show the complex non-linear relationship due to the solvation effect and cannot predict even with the state-of-the-art quantum chemistry simulator. Thus, it is natural to assume this freezing point is an unknown constraint. We create the true function by fitting the experimental data of MA-DMC-EMC-$LiPF_6$ (Logan et al., 2018) system using the Casteel-Amis equation (Casteel & Amis, 1972). Note that Casteel-Amis equation is just for the interpolation of experimental data to be continuous, and is not capable of predicting different cosolvent nor freezing points.

Electrolyte is a three-dimensional continuous input function with two constraints. The input features are (1) the lithium salt ($LiPF_6$) molarity, (2) DMC/EMC cosolvent ratio, and (3) MA/carbonates cosolvent ratio, respectively. The inputs are bounded with $x_1 \in [0, 2]$, $x_2 \in [0, 1]$, and $x_3 \in [0, 0.3]$. The constraints are $x_1 > 0.3$ and $x_2 < 0.9$. The noisy output is generated by adding i.i.d. zero-mean Gaussian noise with the 3^2 variance to the noiseless $f(x)$.

Cantilever Cantilever (Wu et al., 2001) has been proposed for a task to develop a probability-based design optimization framework for ensuring high reliability and safety. This task is to design a cantilever beam under the two failure modes as safety constraints. The objective function to model with GP is the tip displacement,

modelled as:

$$f(x) := \frac{4 \times 100^3}{E} \sqrt{X^2 + Y^2}, \tag{28}$$

$$\text{subject to:} \tag{29}$$

$$\frac{f(x) - 4400}{3100} < 2.2535, \tag{30}$$

$$0.8(X + Y) < R. \tag{31}$$

Cantilever is a four-dimensional continuous input function with two constraints. The input features are (1) the yield stress R, (2) the Young's modulus of beam material E, (3) the horizontal load X, and (4) the vertical load Y, respectively. The inputs are bounded with $R \in [3E + 4, 5E + 4]$, $E \in [1E + 7, 5E + 7]$, $X \in [1E + 2, 1E + 3]$, and $Y \in [5E + 3, 5E + 4]$. The noisy output is generated by adding i.i.d. zero-mean Gaussian noise with the 1^2 variance to the noiseless $f(x)$.

Steel Steel (Kuschel & Rackwitz, 1997) has been proposed for design optimization to balance the reliability and cost. This task is to design a steel column under cost constraints. The objective function to model with GP is the limit state function, modelled as:

$$f(x) := F_s - P \left[\frac{1}{2BD} + \frac{F_0 E_b}{BDH(E_b - P)} \right], \tag{32}$$

$$\text{subject to:} \tag{33}$$

$$BD + 5H < 9000, \tag{34}$$

where

$$P := P_1 + P_2 + P_3 \tag{35}$$

$$E_b := \frac{\pi^2 EBDH^2}{2L^2} \tag{36}$$

Steel is the nine-dimensional continuous input function with one constraint. The input features are (1) the yield stress F_s, (2) the dead weight load P_1, (3) the variable load P_2, (4) the variable load P_3, (5) the flange breadth B, (6) the flange thickness D, (7) the profile height H, (8) the initial deflection F_0, and (9) Young's modulus E, respectively. The inputs are bounded with $F_s \in [300, 500]$, $P_1 \in [1E + 4, 1E + 5]$, $P_2 \in [4E + 5, 1E + 6]$, $P_3 \in [4E + 5, 1E + 6]$, $B \in [290, 310]$, $D \in [14, 26]$, $H \in [290, 310]$, $F_0 \in 209800, 210100]$. The noisy output is generated by adding i.i.d. zero-mean Gaussian noise with the 1^2 variance to the noiseless $f(x)$.

PestControl Pest Control (PestControl in the main) is proposed in Oh et al. (2019), which is a multi-categorical optimisation problem (15 dimensions, 5 categories for each dimension). We wish to optimise the effectiveness of pesticides by choosing the 5 actions (selection of pesticides from 4 different firms, or not using any of them), but penalised by their prices. This choice is a sequential decision of 15 stages, and the objective function is expressed as the cumulative loss function with the total of both cost and the portion having pest. The batch size n is 200. We set the categorical prior with equal weights for each choice (discrete uniform distribution). Code is used in https://github.com/xingchenwan/Casmopolitan (Wan et al., 2021).

We added 2 constraints for a more realistic situation. The first constraint is ecosystem change, which assumes exterminating pests too much causes other harmful pests/animals to increase when they reach the hidden threshold. The portion of the product having pests follows the dynamics below:

$$z_i = \alpha_i(1 - x_i)(1 - z_{i-1}) + (1 - \Gamma_i x_i) z_{i-1}, \tag{37}$$

$$z_i \geq z_{\text{limit}}, \tag{38}$$

where i is the number of pest control cycles (15 in total), z is the portion of the product having pest, x is the effectiveness of pesticide that follows a beta distribution with the parameters, which has been adjusted according to the sequence of actions taken in previous control points, α is the action taken (selection of pesticides from 4 different firms, or not using any of it), and z_{limit} is the threshold for ecosystem change (we set $1e - 3$). Eq. 38 is the constraint of ecosystem change, and we assume the latent variable z_i is observable.

The second constraint is neighbour disputes, which assume some of the pesticides have unfavourable smells. Neighbours objection follows the Bernoulli distribution and its weights based on the proportion of certain pesticide types and random Gaussian noise. Thus, the feedback to this constraint is in the noisy binary value. If the neighbours' objection is larger than supportive opinion $\theta_{\text{pest}} > 0.5$, a decision maker stops spraying pesticides, thus, objective value cannot be evaluated.

Malaria The objective is to discover an anti-malarial drug exhibiting the smallest EC50 value, which is defined as the concentration of the drug that gives half the maximal response. The lower the concentration, the more effective (better) the drug. The dataset consists of 20,746 small molecules taken from the P. falciparum whole-cell screening derived by the Novatis-GNF Malaria Box (Spangenberg et al., 2013). The molecules are represented as SMILES string and are converted into 2048-dimensional binary features for the Tanimoto kernel. We set four safety constraints, all of which are rules of thumb for judging molecules likely to be oral drugs, shared in drug discovery community (Lipinski et al., 1997; Veber et al., 2002; Butler, 2004; Mochizuki et al., 2019).

The first is Lipinski's rule of five (Lipinski et al., 1997), (A) no more than 5 hydrogen bond donors, (B) no more than 10 hydrogen bond acceptors, (C) A molecular mass less than 500 daltons, (D) A calculated octanol-water partition coefficient that does not exceed 5, (E) no more than 5 rotatable bonds. The second is the Veber filter (Veber et al., 2002), (A) no more than 10 rotatable bonds, (B) a polar surface area that does not exceed 140. The third is the REOS filter (Butler, 2004), (A) A molecular mass more than 200 daltons and less than 500 daltons, (B) A calculated octanol-water partition coefficient that exceeds -5 but does not exceed 5, (C) no more than 5 hydrogen bond donors, (D) no more than 10 hydrogen bond acceptors, (E) no more than 8 roratable bonds, (F) more than 15 but less than 50 heavy atoms, (G) more than -2 but less than 2 formal charge. The fourth is the drug likeliness filter, (A) A molecular mass less than 400 daltons, (B) at least one ring structure, (C) no more than 5 roratable bonds, (D) no more than 5 hydrogen bond donors, (E) no more than 10 hydrogen bond acceptors, (F) A calculated octanol-water partition coefficient that does not exceed 5.

FindFixer This task is to find the fixer connecting influencers rather than finding the most popular influencer on the social networks graph. A job seeker who wishes to be a celebrity explores the fixer to ask introductions based on graph data using the centrality analysis. Finding a node requires searching on a website or meeting in person, both of which are expensive to evaluate. Fixer can be interpreted as a node with maximum eigenvector centrality under constraints on the degree centrality that does not exceed the threshold (Kiss & Bichler, 2008). In other words, finding the node that is connected to the largest number of nodes with many edges but does not have many edges itself. A job seeker wishes to find the fixer who connects influencers with similar popularity (degree centrality). Thus, the node is constrained based on the degree centrality, and other hidden preference factors. A job seeker judges constraints as a binary value, and the judgment is possibly shaky. We assume the domain is defined as a social network graph synthesized by the Barábsi–Albert model (BA) (Barabási & Albert, 1999).

TeamOpt This task is to organise a team consisting of the most diverse skill sets of members (Wan et al., 2023). The objective is measured by the entropy of the skills of members, assuming the optimal team is when each member is specialised in one skill, and the whole skill distribution is close to uniform. Such teams are positioned on the node of the supergraph, of which edge is the similarity between teams defined as the Jaccord index. The constraints are interpersonal relationships. Every combination of two individuals from N candidates has unobservable hidden continuous likability from 0 to 1. The first is the mean likability constraint, which is the mean of likeability between all possible combinations of members that should be larger than equal-chance. The second is the tragedy-avoidance constraint, which is a binary judge that none of them has a likability lower than a threshold. The third is a flat-relationship constraint, which assumes an entropy of likability must be higher than a threshold. As likability is unobservable, a decision-maker needs to seek advice from many colleagues who partially know each constraint but are noisy estimations.

B.4 Complexity Analysis

As explained in the section 3.8, the time complexity of the AdaBatAL is lower than $\mathcal{O}(NM + M^2 \log n + Mn^2 \log(N/n))$ (Hayakawa et al., 2022), where N is the number of unlabelled pool, M is the number of Nyström samples, and n is the upper bound of the batch size. The space complexity is $\mathcal{O}(NM)$.

We empirically compare the time complexity against the baselines using the Hartmann function with unconstrained batch BO tasks. Figure 6 shows the log overhead to generate the batch samples with different batch sizes that

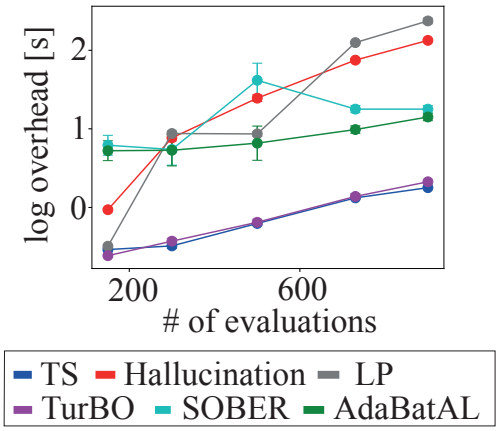


Figure 6: Overhead on Batch Bayesian optimization task for Hartmann ($d = 6$)

are the same with Figure 3 setting. While TurBO and TS were faster than others, our AdaBatAL was relatively faster than other baselines (SOBER, hallucination, and LP).

C

Appendix of Chapter 5

Appendix A. DERIVATION OF CANONICAL FORM

The impedance of RC pair ECM is typically expressed as:

$$\begin{aligned} Z &= R_0 + \sum_{i=1}^{N} \frac{R_i}{1 + j\omega C_i R_i}, \\ &= R_0 + \sum_{i=1}^{N} \frac{R_i}{1 + (\omega C_i R_i)^2} - j \sum_{i=1}^{N} \frac{\omega C_i R_i^2}{1 + (\omega C_i R_i)^2}. \end{aligned}$$

This can be written as:

$$\begin{aligned} \mathrm{Re}[Z] &= R_0 + \sum_{i=1}^{N} \frac{R_i}{1 + (\omega C_i R_i)^2}, \\ \mathrm{Im}[Z] &= \sum_{i=1}^{N} R_i \frac{\omega C_i R_i}{1 + (\omega C_i R_i)^2}. \end{aligned}$$

For the real part, we can rewrite as:

$$\begin{aligned} \mathrm{Re}[Z] &= R_0 + \sum_{i=1}^{N} \frac{R_i}{1 + (\omega\tau_i)^2}, \\ &= R_0 + \sum_{i=1}^{N} \frac{R_i}{2} \frac{2}{1 + (\omega\tau_i)^2}, \\ &= R_0 + \sum_{i=1}^{N} \frac{R_i}{2} \left[1 - \frac{(\omega\tau_i)^2 - 1}{(\omega\tau_i)^2 + 1} \right], \\ &= R_0 + \sum_{i=1}^{N} \frac{R_i}{2} \left[1 - \tanh(\ln \omega\tau_i) \right], \\ &= R_{\mathrm{total}} \left[r_0 + \sum_{i=1}^{N} \frac{r_i}{2} \left[1 - \tanh(\ln \omega\tau_i) \right] \right], \\ &= R_{\mathrm{re}} \left[r_0 + \sum_{i=1}^{N} \frac{r_i}{2} \left[1 - \tanh(\ln \omega\tau_i) \right] \right]. \end{aligned}$$

Similarly, the imaginary part can be rewritten as:

$$\begin{aligned} \mathrm{Im}[Z] &= \sum_{i=1}^{N} R_i \frac{\omega\tau_i}{1 + (\omega\tau_i)^2}, \\ &= \sum_{i=1}^{N} \frac{R_i}{2} \frac{2\omega\tau_i}{1 + (\omega\tau_i)^2}, \\ &= \sum_{i=1}^{N} \frac{R_i}{2} \mathrm{sech}(\ln \omega\tau_i), \\ &= \frac{\pi \left\{ \sum_{i=1}^{N} R_i \right\}}{2} \sum_{i=1}^{N} \frac{R_i}{\pi \left\{ \sum_{i=1}^{N} R_i \right\}} \mathrm{sech}(\ln \omega\tau_i), \\ &= R_{\mathrm{im}} \sum_{i=1}^{N} \frac{w_i}{\pi} \mathrm{sech}(\ln \omega\tau_i), \end{aligned}$$

where w_i is introduced to be $\sum_{i=1}^{N} w_i = 1$, $1/\pi$ is introduced to be standardised as $\int_{-\infty}^{\infty} \frac{w_i}{\pi} \mathrm{sech}(\ln \omega\tau_i) \mathrm{d} \ln \omega\tau_i = 1$.

Appendix B. BAYESIAN QUADRATURE TRAINING PROCEDURE

B.1 Four-Layered BASQ Formulation

The likelihood surrogate model $\ell(\Theta)$ is defined as:

$$\ell(\Theta) \sim \mathcal{N}(\ell; \mu_\ell(\Theta), \sigma_\ell(\Theta)), \tag{B.1}$$

$$\mu_\ell(\Theta) = K(\Theta, \boldsymbol{\Theta}) K(\boldsymbol{\Theta}, \boldsymbol{\Theta})^{-1} \ell_{\mathrm{true}}(\boldsymbol{\Theta}), \tag{B.2}$$

$$\sigma_\ell(\Theta, \Theta') = K(\Theta, \Theta') - K(\Theta, \boldsymbol{\Theta}) K(\boldsymbol{\Theta}, \boldsymbol{\Theta})^{-1} K(\boldsymbol{\Theta}, \Theta), \tag{B.3}$$

where $\ell(\Theta)$ is the surrogate likelihood function modelled by GP, $\boldsymbol{\Theta}$ is the 'observed parameter sets', and K is the kernel.

GP is a non-parametric probabilistic model, typically applied to regression tasks in machine learning. GP can flexibly increase the model complexity in accordance with the number of data, thwarting under/over-confidence. GP model shape is determined by the data points $\boldsymbol{\Theta}$ and the kernel $K(\Theta, \Theta')$. The kernel maps the correlation between data points into a covariance matrix. Gaussianity of GP provides analytical predictive distribution $\ell(\Theta)$, with predictive mean $\mu_\ell(\Theta)$ and covariance $\sigma_\ell(\Theta, \Theta')$, as shown in Eqs (B.2) - (B.3). While the predictive mean $\mu_\ell(\Theta)$ predicts the likelihood $\ell_{\mathrm{true}}(\Theta)$, predictive covariance $\sigma_\ell(\Theta, \Theta')$ predicts the uncertainty of the prediction at given Θ. That is, training GP means minimising the predictive covariance over all possible parameters $\pi(\Theta)$, namely, minimising $\iint_\chi \sigma_\ell(\Theta, \Theta') \mathrm{d}\pi(\Theta) \mathrm{d}\pi(\Theta')$. Such training can be done via querying more observations from the true likelihood $\mathbf{D}_{\boldsymbol{\Theta}} = \{\boldsymbol{\Theta}, \ell_{\mathrm{true}}(\boldsymbol{\Theta})\}$. Hence, the most straightforward training is to sample from the prior $\pi(\Theta)$ until the integral variance becomes smaller than a convergence threshold. However, the prior often barely overlaps over the likelihood, resulting in observing unhelpful tiny likelihood values over most samples.

To overcome this problem, we consider sample-efficient training that fully exploits the information from GP. Osborne et al. (2012) showed that active learning sampling could efficiently reduce the number of samples. The active learning scheme guides the next query point to minimise the integral variance, exploiting the GP surrogate model information. A function called acquisition function formulated by predictive mean $\mu_\ell(\Theta)$ and covariance $\sigma_\ell(\Theta, \Theta')$ can evaluate where to sample, and optimising it can locate where to sample next. Still, the overhead of the next query guidance is not negligible, and it is an inevitably sequential procedure. Adachi et al. (2022) proposed batch Bayesian quadrature, termed Bayesian Alternately Subsampled Quadrature (BASQ), permitting a lightweight active learning scheme and parallelisation of querying. They adopted the discretised sampling method (Hayakawa et al., 2022) for probability measure rather than an acquisition function. This allows us to query the true function in parallel. As the modern computational environment exploits an efficient parallel computation via a graphical processing unit or a computer cluster in the cloud, such computing power can accelerate inference computation. They demonstrated that BASQ could accelerate Bayesian inference over various synthetic and real-world datasets, including SPMe model inference.

The evidence can be calculated via kernel recombination. Kernel recombination is a discrete approximation of continuous kernel integral into weighted summation so as to minimise the integral variance, as such:

Table B.1. Four-layered GPs and warped functions at each layer

Layers	e space	f space	g space	h space
Correspondence	likelihood	normalised likelihood	square-root norm. likelihood	sqrt. norm. log likelihood
Warp	scaling	square-root	log	base GP
Forward	e	$e/\exp\beta$	$\sqrt{2(f-\alpha)}$	$\log(g+1)$
Backward	$f\exp\beta$	$\alpha+\frac{1}{2}g^2$	$\exp(h)-1$	h
GP	$e\sim\mathcal{GP}(\mu_e,\sigma_e)$	$f\sim\mathcal{GP}(\mu_f,\sigma_f)$	$g\sim\mathcal{GP}(\mu_g,\sigma_g)$	$h\sim\mathcal{GP}(\mu_h,\sigma_h)$
Mean	$\mu_f(x)\exp\beta$	$\alpha+\frac{1}{2}\left[\mu_g(x)^2+\sigma_g(x,x)\right]$	$\exp\left[\mu_h(x)+\frac{1}{2}\sigma_h(x,x)\right]$	$\mu_h(x)$
Covariance	$\sigma_f(x,y)\exp(2\beta)$	$\frac{1}{2}\sigma_g(x,y)^2+\mu_g(x)\sigma_g(x,y)\mu_g(y)$	$\mu_g(x)\mu_g(y)[\exp\{\sigma_h(x,y)-1\}]$	$\sigma_h(x,y)$

$$\int_Q \varphi(x)\mathrm{d}q(x)\approx\sum_p^P w_p\varphi(X_p),$$
$$X_p\in\mathbf{X},w_p\in\mathbf{W},$$

$\mathbf{X}$ is the discretised samples over the probability measure, $\mathbf{W}$ is the positive weights to approximate integration. When we recall our training objective is to minimise the predictive covariance over the probability measure $\pi(\Theta)$, this can be formulated as kernel recombination. Hence, we pass the predictive covariance $\sigma_\ell(\Theta,\Theta')$ as kernel to the kernel recombination algorithm (Hayakawa et al., 2022), which yields the following approximation:

$$\mathbf{X},\mathbf{W}=\mathrm{recombination}[\sigma_\ell(\Theta,\Theta'),\pi(\Theta)],$$
$$\mathbb{E}_\pi[\ell(\Theta)]=\int_\chi\mu_\ell(\Theta)\mathrm{d}\pi(\Theta),\approx\sum_k^L W_k\mu_\ell(X_k),$$
$$\mathbb{V}\mathrm{ar}_\pi[\ell(\Theta)]=\iint_\chi\sigma_\ell(\Theta,\Theta')\mathrm{d}\pi(\Theta)\mathrm{d}\pi(\Theta'),\approx\sum_{k,l}^L W_kW_l\sigma_\ell(X_k,X_l),$$

where $X_k,X_l\in\mathbf{X}$, $W_k,W_l\in\mathbf{W}$. However, they adopted square-root warping for fast computation, which assumed a narrow dynamic range in likelihood. Battery models' likelihood turns out to be very sharp, as the number of data points over the frequency range is typically over a hundred.

Therefore, we adopted four-layered GPs to accommodate the dynamic range, permitting solving Bayesian inference even in this wide dynamic range case. Functions at each layer are summarised in Table B.1, where $\mathbf{Y}_{\log}$ is the observed log-likelihood values, $\alpha=\min[\exp(\mathbf{Y}_{\log}-\beta)]$, $\beta=\max[\mathbf{Y}_{\log}]$. e space corresponds to the original likelihood space. Square-root warping and log-warping layers are approximated via the moment-matching method (Gunter et al., 2014; Chai et al., 2019). To accommodate the wide dynamic range, log transformation is widely applied in the BQ community. However, log-warped GP inevitably results in sampling from log space, leading to ineffective exploration. As meaningful samples from a very sharp likelihood are localised in only the vicinity of the maximum values, log space exploration is too blunt to explore the original space. The combination of square-root warping and log-warping can overcome this issue using the following relationship:

$$f=\alpha+\frac{1}{2}g^2\approx\alpha+\frac{1}{2}\exp(h)\exp(h),$$
$$\mathbb{E}_\pi[\mu_f(\Theta)]=\alpha+\frac{1}{2}\int_\Xi\mu_g(\Theta)\mathrm{d}\pi'(\Theta),$$
$$\pi'(\Theta):=\mu_g(\Theta)\pi(\Theta).$$

As such, this doubly warping structure enables us to copy exponentiated function information to both likelihood and prior. Thus, this double structure can sample from sharp exponentiated distribution $\pi'(\Theta)$ as well as keep the surrogate model exponentiated $\mu_g(\Theta)$.

The last layer, e, exists to avoid overflow in computation by scaling the whole dynamic range via maximum value. This warping layer can be avoided as such:

$$\ln\mathbb{E}_\pi[\mu_e(\Theta)]=\ln\mathbb{E}_\pi[\mu_f(\Theta)]+\beta,\approx\ln\sum_k^L W_k\mu_f(X_k)+\beta,$$
$$\ln\mathbb{V}\mathrm{ar}_\pi[\sigma_e(\Theta)]=\ln\mathbb{V}\mathrm{ar}_\pi[\sigma_f(\Theta)]+2\beta,\approx\ln\sum_{k,l}^L W_kW_l\sigma_f(X_k,X_l)+2\beta,$$
$$p(\Theta|\mathbf{D},M)=\frac{\mu_e(\Theta)\pi(\Theta)}{\mathbb{E}_\pi[\mu_e(\Theta)]}=\frac{\mu_f(\Theta)\pi(\Theta)}{\mathbb{E}_\pi[\mu_f(\Theta)]}.$$

B.2 Training Procedures

Training consists of four processes:

(1) Subsampling from the exponentiated distribution
(2) Kernel recombination for batch sampling
(3) GP hyperparameter optimisation
(4) Evidence estimation

We iterate the above four procedures until the evidence variance reaches plateau. Only the first training procedure is different from the original BASQ (Adachi et al., 2022).

The subsampling is to sample from the prior distribution to construct the empirical measure. As the kernel recombination is to select the sparse sample set from subsamples that can minimise the integral variance, subsamples should be sampled from prior but well overlapped from the higher predictive variance of GP $\ell(x)$. Adachi et al.

(2022) adopted uncertainty sampling for faster convergence, which samples from predictive variance $\sigma_\ell(x)$ and corrected to prior distribution via importance sampling, as such:

$$\begin{aligned} g_{\text{prop}}(\Theta) &:= (1-r)\mu_g(\Theta) + r\tilde{A}(\Theta), \quad 0 \le r \le 1 \\ \mathrm{w}_{\text{IS}}(\Theta) &:= \mu_g(\Theta)/g_{\text{prop}}(\Theta), \\ \tilde{A}(\Theta) &:= \sigma_g(\Theta)\pi'(\Theta)/Z_{\tilde{A}}, \\ Z_{\tilde{A}} &:= \int_\Xi \sigma_g(\Theta) d\pi'(\Theta), \\ \sigma_g(\Theta) &:= \operatorname{diag}\left[\sigma_g(\Theta,\Theta)\right]. \end{aligned}$$

We wish to adopt the same strategy for a four-layered GP, but the log-warp layer hinders the application. The predictive variance of the original BASQ can be analytically translated into the mixture of Gaussian with Gaussian kernel because the squared Gaussian distribution is still Gaussian. However, the exponentiated Gaussian is no more Gaussian, which becomes a log-normal distribution. As such, we cannot take the same strategy which exploits the Gaussianity. Hence, we employ the heuristical method. The predictive variance is expected to be larger at the midpoints between the observed data points. Thus, sampling from the midpoints with half lengthscale of GP is expected to be good proposal distribution of sampling the uncertainty region, as such:

$$\begin{aligned} g_{\text{heur}}(\Theta) &:= \sum_{r,s}^{N_{\text{heur}}} w_{r,s}^{\text{heur}} \mathcal{N}\left(\Theta; \Theta_{r,s}^{\text{mid}}, \frac{\mathbf{W}_{\text{length}}}{2}\right), \\ \Theta_{r,s}^{\text{mid}} &:= \frac{\Theta_r + \Theta_s}{2}, \\ w_{r,s}^{\text{heur}} &:= \frac{\sigma_g(\Theta_{r,s}^{\text{mid}})\pi'(\Theta_{r,s}^{\text{mid}})}{\sum_{r,s}^{N_{\text{heur}}} \sigma_g(\Theta_{r,s}^{\text{mid}})\pi'(\Theta_{r,s}^{\text{mid}})}, \end{aligned}$$

where $\Theta_r, \Theta_s \in \boldsymbol{\Theta}$ are the observed parameters, $\mathbf{W}_{\text{length}}$ is the diagonal covariance matrix whose diagonal elements are the lengthscales of each dimension. Supersampling from this offers the uncertainty sampling, as such:

$$\begin{aligned} \Theta_t^{\text{super}} &\sim g_{\text{heur}}(\Theta) \in \mathbb{R}^{N_{\text{super}}}, \\ Z_{\tilde{A}} &= \int \sigma_g(\Theta) \frac{\pi'(\Theta)}{g_{\text{heur}}(\Theta)} dg_{\text{heur}}(\Theta), \\ &\approx \frac{1}{N_{\text{super}}} \sum_t^{N_{\text{super}}} \sigma_g(\Theta_t^{\text{super}}) \frac{\pi'(\Theta_t^{\text{super}})}{g_{\text{heur}}(\Theta_t^{\text{super}})}, \\ w^{\text{super}} &:= \tilde{A}(\Theta_t^{\text{super}})/g_{\text{heur}}(\Theta_t^{\text{super}}). \end{aligned}$$

Sequential Monte Carlo (Kitagawa, 1993) permits to sample from $\tilde{A}(\Theta)$.

B.3 Ablation study of layered GPs

We discuss the efficacy of four-layered GP by comparing the results of evidence inference for the easy case introduced in Table 1. We compared the following six configurations in Table B.2. The ground truth of LEM is estimated via exhaustive nested sampling with millions of samples until convergence, which yields 703.7285. The ablation study shows that the four-layered GPs can estimate the most accurate LEV of all compared configurations. GPs without the scaling layer reached the overflow limit, which returned a positive infinite value. GPs without the logarithmic layer scored the lower log evidence mean because the surrogate model cannot accommodate the wide dynamic range. Scaled GP with only log warp results was the second best. However, the non-exponentiated prior struggled to find the MAP location. As such, the four-layered GP, employing all features, was the performant.

Table B.2. Ablation study of warped layers

log	square-root	scaling	LEM	LEV
✔			overflow	overflow
	✔		overflow	overflow
		✔	361.8172	-11.90735
✔		✔	677.8633	-21.86860
	✔	✔	449.6425	-13.13063
✔	✔	✔	**703.6569**	-27.31169

Appendix C. IDENTIFIABILITY DERIVATION

C.1 Hyperbolic Secant Distribution Identities

$$\int_{-\infty}^{\infty} \operatorname{sech}(x)\, dx = \pi, \tag{C.1}$$

$$\int_{-\infty}^{\infty} \operatorname{sech}\left(\frac{x-a}{b}\right) dx = \frac{\pi}{b}, \tag{C.2}$$

$$\int_{-\infty}^{\infty} \operatorname{sech}(x) \ln \operatorname{sech}(x)\, dx = -\pi \ln 2, \tag{C.3}$$

$$\int_{-\infty}^{\infty} \operatorname{sech}(x) \operatorname{sech}(x-a)\, dx = 2a\operatorname{csch}(a), \tag{C.4}$$

$$\int_{-\infty}^{\infty} \operatorname{sech}(x)^2\, dx = 2. \tag{C.5}$$

C.2 SNR Derivation

$$\begin{aligned} \mathrm{SNR} &:= \ln \frac{\mathbb{V}\mathrm{ar}_{P(\ln\omega)}[\mathrm{Im}[Z]]}{\sigma_{\text{noise}}^2}, \\ \mathbb{V}\mathrm{ar}_{P(\ln\omega)}[\mathrm{Im}[Z]] &= \mathbb{E}_{P(\ln\omega)}[\mathrm{Im}[Z]^2] - \mathbb{E}_{P(\ln\omega)}[\mathrm{Im}[Z]]^2, \\ \mathbb{E}_{P(\ln\omega)}[\mathrm{Im}[Z]] &= \int_\Omega \mathrm{Im}[Z](\ln\omega)\mathrm{d}P(\ln\omega), \\ &= \frac{\exp(r_{\text{total}})\pi(1-r_0)}{2(b-a)}, \\ \mathbb{E}_{P(\ln\omega)}[\mathrm{Im}[Z]^2] &= \frac{\exp(2r_{\text{total}})(1-r_0)^2}{2(b-a)} A, \end{aligned}$$

where

$$\begin{aligned} P(\ln\omega) &:= \mathcal{U}(\ln\omega; a, b), \\ a, b &:= \min[\ln\omega], \max[\ln\omega], \\ A &:= \sum_i^N w_i^2 + \sum_{i,j}^N 2 w_i w_j \Delta\tau_{ij} \operatorname{csch}(\Delta\tau_{ij}), \\ \Delta\tau_{ij} &:= \sigma_\omega(\tau_i^{\text{std}} - \tau_j^{\text{std}}). \end{aligned}$$

Eq. (C.2) yields the analytical solution of the first expectation:

$$\begin{aligned}\mathbb{E}_{P(\ln\omega)}[\mathrm{Im}[Z]] &= \int_\Omega P(\mathrm{Im}[Z]|\omega)dP(\omega),\\ &= \frac{\exp(r_{\mathrm{total}})\pi(1-r_0)}{2(b-a)}\sum_{i=1}^{N}\frac{w_i}{\pi}\\ &\quad\int_{-\infty}^{\infty}\mathrm{sech}\left(\omega+\Delta\tau_{ij}\right)d\omega,\\ &= \frac{\exp(r_{\mathrm{total}})\pi(1-r_0)}{2(b-a)}\sum_{i=1}^{N}w_i,\\ &= \frac{\exp(r_{\mathrm{total}})\pi(1-r_0)}{2(b-a)}.\end{aligned}$$

Eqs. (C.4) - (C.5) yield the analytical solution of the second expectation:

$$\begin{aligned}\mathbb{E}_{P(\ln\omega)}[\mathrm{Im}[Z]^2] &= \int_\Omega P(\mathrm{Im}[Z]|\omega)^2dP(\omega),\\ &= \frac{1}{b-a}\left[\frac{\exp(r_{\mathrm{total}})\pi(1-r_0)}{2}\right]^2\\ &\quad\int_{-\infty}^{\infty}\left[\sum_{i=1}^{N}\frac{w_i}{\pi}\mathrm{sech}\left(\omega+\Delta\tau_{ij}\right)\right]^2d\omega,\\ &= \frac{1}{b-a}\left[\frac{\exp(r_{\mathrm{total}})\pi(1-r_0)}{2}\right]^2\\ &\quad\int_{-\infty}^{\infty}\left[\sum_{i}^{N}\frac{w_i^2}{\pi^2}\mathrm{sech}\left(\omega+\Delta\tau_{ij}\right)^2+\right.\\ &\quad\left.\sum_{i,j}^{N}\frac{2w_iw_j}{\pi^2}\mathrm{sech}\left(\omega\right)\mathrm{sech}\left(\omega+\Delta\tau_{ij}\right)\right]d\omega,\\ &= \frac{1}{b-a}\left[\frac{\exp(r_{\mathrm{total}})\pi(1-r_0)}{2}\right]^2\\ &\quad\left\{\sum_{i}^{N}\frac{2w_i^2}{\pi^2}+\sum_{i,j}^{N}\frac{4w_iw_j}{\pi^2}\Delta\tau_{ij}\mathrm{csch}\left(\Delta\tau_{ij}\right)\right\},\\ &= \frac{\exp(2r_{\mathrm{total}})(1-r_0)^2}{2(b-a)}\\ &\quad\left\{\sum_{i}^{N}w_i^2+\sum_{i,j}^{N}2w_iw_j\Delta\tau_{ij}\mathrm{csch}\left(\Delta\tau_{ij}\right)\right\}.\end{aligned}$$

C.3 JS derivation

The asymmetric $\mathrm{JS}_{i|j}$ and symmetric JS definitions are as follows:

$$\begin{aligned}\mathrm{JS}_{i|j} &:= \int_P \ln\left(\frac{P_i(x)}{P_j(x)}\right)dP_i(x),\\ \mathrm{JS} &:= \mathrm{JS}_{i|j}+\mathrm{JS}_{j|i}.\end{aligned}$$

To incorporate the information of weights, we adopt the following scaled hyperbolic secant distributions:

$$\begin{aligned}P_i(\ln\omega) &:= \frac{w_i}{\pi}\mathrm{sech}\left[w_i(\ln\omega+\sigma_\omega\tau_i^{\mathrm{std}})\right],\\ P_j'(\ln\omega) &:= \frac{w_j}{\pi}\mathrm{sech}\left[w_j(\ln\omega+\sigma_\omega\tau_j^{\mathrm{std}})\right],\end{aligned}$$

where $\tau_j^{\mathrm{std}} > \tau_i^{\mathrm{std}}$. For efficient computation of the integrals, we can adopt the importance sampling, as such:

$$\begin{aligned}\mathrm{JS}_{i|j} &= \int_P \frac{P_i(x)}{g_{\mathrm{JS}}(x)}\ln\frac{P_i(x)}{P_j(x)}dg_{\mathrm{JS}}(x),\\ X_q^{\mathrm{IS}} &\sim g_{\mathrm{JS}}(x)\in\mathbb{R}^{N_{\mathrm{IS}}},\\ \mathrm{JS}_{i|j} &\approx \frac{1}{N_{\mathrm{IS}}}\sum_{q}^{N_{\mathrm{IS}}}\frac{P_i(X_q^{\mathrm{IS}})}{g_{\mathrm{JS}}(X_q^{\mathrm{IS}})}\ln\frac{P_i(X_q^{\mathrm{IS}})}{P_j(X_q^{\mathrm{IS}})},\end{aligned}$$

where

$$\begin{aligned}g_{\mathrm{JS}}(x) &:= \frac{1}{2N}\sum_{i}^{N}\frac{w_i}{\pi}\mathrm{sech}\left[w_i(x+\sigma_\omega\tau_i^{\mathrm{std}})\right]\\ &\quad+\frac{1}{4\pi}\mathrm{sech}\left[0.5(x+\sigma_\omega w_i\tau_i^{\mathrm{std}}+0.5\Delta_{ij})\right],\\ \Delta_{ij} &:= \sigma_\omega|w_j\tau_j^{\mathrm{std}}-w_i\tau_i^{\mathrm{std}}|,\end{aligned}$$

$g_{\mathrm{JS}}(x)$ is a proposal distribution. As the logarithmic term is a subtraction of two hyperbolic secant distributions, the peak is estimated around the overlapped area, namely the midpoint of the two peaks $x+\sigma_\omega w_i\tau_i^{\mathrm{std}}+0.5\Delta_{ij}$. We can solve this integral via Monte Carlo integration. As sampling and evaluation of the probability density function of hyperbolic secant distribution are done within a millisecond order, computation with millions of samples for accuracy is not demanding.

D
Appendix of Chapter 6

Part I
Appendix

Table of Contents

A Proof of theorem

A.1 Regret analysis of normal UCB policy

We begin with the finite case, $|D| < \infty$. We recall two lemmas from Srinivas et al. (2010).

Lemma 5. *Pick $\delta \in (0,1)$ and set $\beta_t = 2\log(|D|p_t/\delta)$, where $\sum_{t\geq 1} p_t^{-1} = 1$, $p_t > 0$. Then,*

$$|f(\boldsymbol{x}) - \mu_{f_{t-1}}(\boldsymbol{x}))| \leq \beta_t^{1/2}\sigma_{f_{t-1}}(\boldsymbol{x}) \quad \forall \boldsymbol{x} \in D,\, \forall t \geq 1 \tag{13}$$

holds with probability $\geq 1-\delta$.

Proof Fix $t \geq 1$ and $\boldsymbol{x} \in D$. Conditioned on $\boldsymbol{y}_{t-1} = (y_1, \ldots, y_{t-1})$, $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{t-1}\}$ are deterministic, and $f(\boldsymbol{x}) \sim \mathcal{N}(\mu_{f_{t-1}}(\boldsymbol{x}), \sigma^2_{f_{t-1}}(\boldsymbol{x})$. Now, if $r \sim \mathcal{N}(0,1)$, then

$$\begin{aligned}
\mathbb{P}(r > c) &= \exp\left(-\frac{c^2}{2}\right)(2p)^{-1/2}\int \exp\left(-\frac{(r-c)^2}{2} - c(r-c)\right)\mathrm{d}r, &(14)\\
&\leq \exp\left(-\frac{c^2}{2}\right)\mathbb{P}(r > 0), &(15)\\
&= \frac{1}{2}\exp\left(-\frac{c^2}{2}\right) &(16)
\end{aligned}$$

for $c > 0$, since $\exp(-c(r-c)) \leq 1$ for $r > c$. Therefore, $\mathbb{P}\left(|f(\boldsymbol{x}) - \mu_{f_{t-1}}(\boldsymbol{x}))| \leq \beta_t^{1/2}\sigma_{f_{t-1}}(\boldsymbol{x})\right) \leq \exp\left(-\frac{\beta_t}{2}\right)$, using $r = \frac{f(\boldsymbol{x})-\mu_{f_{t-1}}(\boldsymbol{x}))}{\sigma_{f_{t-1}}(\boldsymbol{x})}$ and $c = \beta_t^{1/2}$. Applying the union bound,

$$|f(\boldsymbol{x}) - \mu_{f_{t-1}}(\boldsymbol{x}))| \leq \beta_t^{1/2}\sigma_{f_{t-1}}(\boldsymbol{x}) \quad \forall \boldsymbol{x} \in D \tag{17}$$

holds with probability $\geq 1 - |D|\exp\left(-\frac{\beta_t}{2}\right)$. Choosing $|D|\exp\left(-\frac{\beta_t}{2}\right) = \frac{\delta}{p_t}$ and using the union bound for $t \in \mathbb{N}$, the statement holds.

Lemma 6. *Fix $t \geq 1$. If $|f(\boldsymbol{x}) - \mu_{f_{t-1}}(\boldsymbol{x}))| \leq \beta_t^{1/2}\sigma_{f_{t-1}}(\boldsymbol{x})$ for all $\boldsymbol{x} \in D$, then the regret r_t is bounded by $2\beta_t^{1/2}\sigma_{f_{t-1}}(\boldsymbol{x})$.*

Proof By definition of $\boldsymbol{x}_{\text{bo}} := \text{argmax}_{x\in\mathcal{X}}\, \mu_{f_{t-1}}(\boldsymbol{x}) + \beta_t^{1/2}\sigma_{f_{t-1}}(\boldsymbol{x})$, we have $\mu_{f_{t-1}}(\boldsymbol{x}_{\text{bo}}) + \beta_t^{1/2}\sigma_{f_{t-1}}(\boldsymbol{x}_{\text{bo}}) \geq \mu_{f_{t-1}}(\boldsymbol{x}^*_{\text{true}}) + \beta_t^{1/2}\sigma_{f_{t-1}}(\boldsymbol{x}^*_{\text{true}}) \geq f_{\text{true}}(\boldsymbol{x}^*_{\text{true}})$. Therefore

$$r_t = f(\boldsymbol{x}^*_{\text{true}}) - f(\boldsymbol{x}_{\text{bo}}) \leq \beta_t^{1/2}\sigma_{f_{t-1}}(\boldsymbol{x}_{\text{bo}}) + \mu_{f_{t-1}}(\boldsymbol{x}_{\text{bo}}) - f(\boldsymbol{x}_{\text{bo}}) \tag{18}$$

$$\leq 2\beta_t^{1/2}\sigma_{f_{t-1}}(\boldsymbol{x}_{\text{bo}}) \tag{19}$$

A.2 Proof of Regrets

Recall the definition of the good user belief is $|f(\boldsymbol{x}) - \mu_{f_{t-1},\hat{\pi}_{t-1}}(\boldsymbol{x}))| \leq \beta_t^{1/2}\sigma_{f_{t-1},\hat{\pi}_{t-1}}(\boldsymbol{x})$.

Proof of good user belief regrets

$$\frac{r_{\hat{\pi}_t}}{r_t} = \frac{f(x^*_{\text{true}}) - f(x_2)}{f(x^*_{\text{true}}) - f(x_1)} \tag{20}$$

$$\leq \frac{2\beta_t^{1/2}\sigma_{f_{t-1},\hat{\pi}_{t-1}}(x_2)}{2\beta_t^{1/2}\sigma_{f_{t-1}}(x_1)} \qquad \text{(Lemma 5)} \tag{21}$$

$$= \frac{\sigma_{f_{t-1},\hat{\pi}_{t-1}}(x_2)}{\sigma_{f_{t-1}}(x_1)} \tag{22}$$

$$= \frac{\sigma_{\hat{\pi}_{t-1}}(x_2)}{\sqrt{\sigma^2_{\hat{\pi}_{t-1}}(x_2) + \sigma^2_{f_{t-1}}(x_1)}} \qquad \text{(Proposition 1)} \tag{23}$$

$$= \sqrt{\frac{\rho^2(\mathbb{V}_{g_{t-1}}[\hat{\pi}_{g_{t-1}}(x_2)]) + \gamma(t-1)^2\sigma^2_{f_{t-1}}(x_2)}{\rho^2(\mathbb{V}_{g_{t-1}}[\hat{\pi}_{g_{t-1}}(x_2)]) + \gamma(t-1)^2\sigma^2_{f_{t-1}}(x_2) + \sigma^2_{f_{t-1}}(x_1)}} \qquad \text{(Proposition 1)} \tag{24}$$

$$= R_{\hat{\pi}_t} \tag{25}$$

$$< 1 \qquad (\sigma^2_{f_{t-1}}(x_1) > 0) \tag{26}$$

Proof of bad user belief regrets The bad user belief case trivially follows the same steps with the additional term:

$$\frac{r_{\hat{\pi}_t}}{r_t} = \frac{f(x^*_{\text{true}}) - f(x_2)}{f(x^*_{\text{true}}) - f(x_1)} \tag{27}$$

$$\leq \frac{2\beta_t^{1/2}\sigma_{f_{t-1},\hat{\pi}_{t-1}}(x_2)}{2\beta_t^{1/2}\sigma_{f_{t-1}}(x_1)} + \frac{|\mu_{f_{t-1}}(x_1) - \mu_{f_{t-1},\hat{\pi}_{t-1}}(x_2)|}{2\beta_t^{1/2}\sigma_{f_{t-1}}(x_1)} \qquad \text{(Bad user belief definition)} \tag{28}$$

$$= \Delta\mu_t + R_{\hat{\pi}_t} \tag{29}$$

These proofs are for finite decision sets. We can extend this proof to a general decision set by following Srinivas et al. (2010) steps. We omit this procedure, but it essentially boils down to the same procedure and similar results with slight coefficient differences.

Proof of No Harm Guarantee We first review the general large t limit properties of π-augmented GP.

Lemma 7. *At the $t \to \infty$ limit, the posterior GP is asymptotically equal to the original GP.*

$$\lim_{t\to\infty} \sigma^2_{f_t,\hat{\pi}_t}(x) = \sigma^2_{f_t}(x), \tag{30}$$

$$\lim_{t\to\infty} \mu_{f_t,\hat{\pi}_t}(x) = \mu_{f_t}(x), \tag{31}$$

$$\lim_{t\to\infty} \alpha_{f_t,\hat{\pi}_t}(x) = \alpha_{f_t}(x), \tag{32}$$

$$\lim_{t\to\infty} x_2 = x_1, \tag{33}$$

Proof of Lemma 6

$$\lim_{t\to\infty} \sigma^2_{f_t,\hat{\pi}_t}(x) = \lim_{t\to\infty} \frac{\sigma^2_{\hat{\pi}_t}(x)\sigma^2_{f_t}(x)}{\sigma^2_{\hat{\pi}_t}(x) + \sigma^2_{f_t}(x)} \tag{34}$$

$$= \lim_{t\to\infty} \frac{(\rho^2(\mathbb{V}_{g_t}[\hat{\pi}_{g_t}(x)]) + \gamma t^2 \sigma^2_{f_t}(x))\sigma^2_{f_t}(x)}{\rho^2(\mathbb{V}_{g_t}[\hat{\pi}_{g_t}(x)]) + \gamma t^2 \sigma^2_{f_t}(x) + \sigma^2_{f_t}(x)} \tag{35}$$

$$= \lim_{t\to\infty} \frac{(\rho^2(\mathbb{V}_{g_t}[\hat{\pi}_{g_t}(x)])/t^2 + \gamma \sigma^2_{f_t}(x))\sigma^2_{f_t}(x)}{\rho^2(\mathbb{V}_{g_t}[\hat{\pi}_{g_t}(x)])/t^2 + \gamma \sigma^2_{f_t}(x) + \sigma^2_{f_t}(x)/t^2} \tag{36}$$

$$= \frac{\gamma\sigma^2_{f_t}(x)\sigma^2_{f_t}(x)}{\gamma\sigma^2_{f_t}(x)} \tag{37}$$

$$= \sigma^2_{f_t}(x) \tag{38}$$

$$\lim_{t\to\infty} \mu_{f_t,\hat{\pi}_t}(x) = \lim_{t\to\infty} \frac{\sigma^2_{f_t,\hat{\pi}_t}(x)}{\sigma^2_{\hat{\pi}_t}(x)}\mu_{\hat{\pi}_t}(x) + \lim_{t\to\infty} \frac{\sigma^2_{f_t,\hat{\pi}_t}(x)}{\sigma^2_{f_t}(x)}\mu_{f_t}(x) \tag{39}$$

$$= \lim_{t\to\infty} \frac{\sigma^2_{f_t}(x)}{\sigma^2_{\hat{\pi}_t}(x)}\mu_{\hat{\pi}_t}(x) + \frac{\sigma^2_{f_t}(x)}{\sigma^2_{f_t}(x)}\mu_{f_t}(x) \tag{40}$$

$$= \lim_{t\to\infty} \frac{\sigma^2_{f_t}(x)}{\rho^2(\mathbb{V}_{g_t}[\hat{\pi}_{g_t}(x)]) + \gamma t^2 \sigma^2_{f_t}(x)}\mu_{\hat{\pi}_t}(x) + \mu_{f_t}(x) \tag{41}$$

$$= \mu_{f_t}(x) \tag{42}$$

$$\lim_{t\to\infty} \alpha_{f_t,\hat{\pi}_t}(x) = \lim_{t\to\infty} \mu_{f_t,\hat{\pi}_t}(x) + \beta_t^{1/2} \lim_{t\to\infty} \sigma^2_{f_t,\hat{\pi}_t}(x) \tag{43}$$

$$= \mu_{f_t}(x) + \beta_t^{1/2}\sigma^2_{f_t}(x) \tag{44}$$

$$= \alpha_{f_t}(x) \tag{45}$$

$$\lim_{t\to\infty} x_2 = \operatorname*{argmax}_{x\in\mathcal{X}} \lim_{t\to\infty} \alpha_{f_t,\hat{\pi}_t}(x) \tag{46}$$

$$= \operatorname*{argmax}_{x\in\mathcal{X}} \alpha_{f_t}(x) \tag{47}$$

$$= x_1 \tag{48}$$

Proof of Lemma 2 By definition and lemma 6,

$$\lim_{t\to\infty} r_t^{\pi} = \lim_{t\to\infty} |f(x^*_{\text{true}}) - f(x_2)| \tag{49}$$

$$= |f(x^*_{\text{true}}) - f(x_1)| \tag{50}$$

$$= r_t \tag{51}$$

$$\lim_{t\to\infty} \frac{r_{\hat{\pi}_t}}{r_t} = 1. \tag{52}$$

B Modelling Human Preference via Gaussian Process

Preferential Gaussian process modelling While many algorithms, rooted in either probabilistic methods (Bradley & Terry, 1952; Luce, 1959) or spectral approaches (Cucuringu, 2016; Chau et al., 2022a), can model human preferences, we opted for Gaussian Processes (GP). This choice enables us to consider epistemic uncertainty more effectively during modeling. However, the classical preferential GP (PGP)(Chu & Ghahramani, 2005) has several limitations; it is computationally expensive and unable to learn preferences that might be inconsistent and with heteroscedastic noise. We combined two existing simple approaches instead of PGP; Dirichlet-based GP (DGP) (Milios et al., 2018), and skew-symmetric data-augmentation (Chau et al., 2022b). DGP translates classification problem as regression one via transforming classification labels to the coefficients of a degenerate Dirichlet distribution. This offers a fast and heteroscedastic GP classifier that has essentially the same accuracy and uncertainty quantification as the original GP classifier. Skew-symmetric data augmentation is to add the

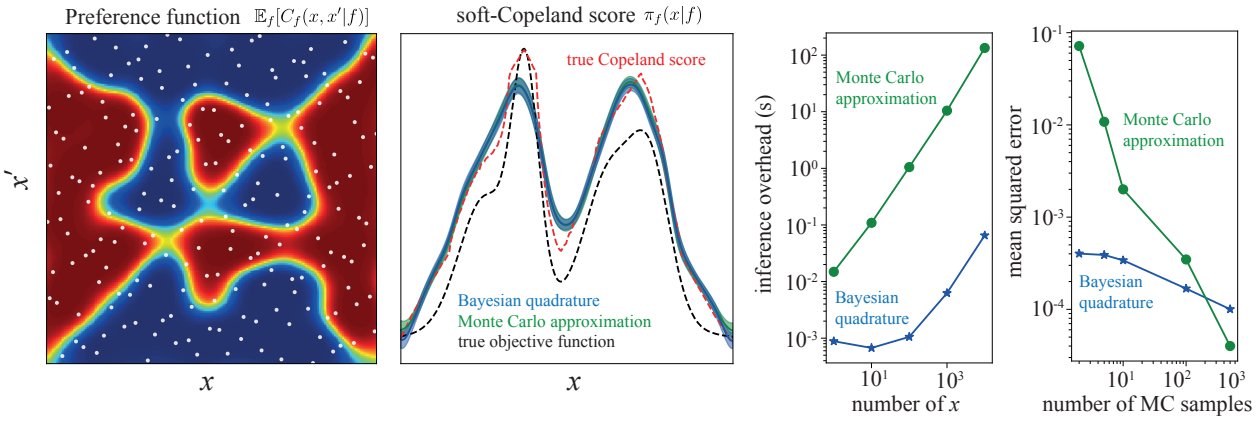


Figure 9: Comparison of Monte Carlo (MC) approximation and Bayesian quadrature approximation on marginalisation for soft-Copeland score. Overhead refers to the wall-clock time to compute the soft-Copeland score at x with the 100 MC samples. The mean squared error is the error between the estimated soft-Copeland score and the ground truth computed with massive MC samples.

symmetric data of original data $Y_{j,i} = 1 - Y_{i,j}$ for the duel (x_i, x_j). Additionally, we used Bayesian quadrature for fast approximation of integral in Eq. (2). However, our setting is not limited to this GP and Bayesian quadrature approximation.

Bayesian quadrature modelling As the integration Eq. (2) is intractable, we have to approximate it. The original preferential BO (González et al., 2017) adopts Monte Carlo (MC) approximation, but it is slow as the authors admit there should be a better way. We adopted Bayesian quadrature (BQ) O'Hagan (1991), particularly BASQ (Adachi et al., 2022, 2023b) for fast approximation. Figure 9 compares the overhead and accuracy of MC integration and BQ. BQ yields faster computation than MC samples at inference (BQ: $\mathcal{O}((n_{\mathrm{MC}} + n_{\mathrm{duel}})^2)$ vs. MC: $\mathcal{O}(n_{\mathrm{MC}} n_{\mathrm{func}} n_{\mathrm{duel}}^2)$), which repeats the computation in the acquisition function optimization loop. Of course, training the BQ model takes additional cost ($\mathcal{O}(n_{\mathrm{mBCG}} n_{\mathrm{duel}}^2)$ for mBCG algorithm (Gardner et al., 2018)), but this can be understood as "pretraining" to avoid the quartic cost of MC integration at each inference point. As RBF kernel assumption of BQ is misspecified against true Bernoulli distribution, the convergence over sample size is limiting; still, BQ works well as it is robust against misspecification (Kanagawa et al., 2016). In larger MC sample sizes, MC integration outperforms BQ, but it produces non-negligible overhead. It should be noted that BQ can derive the closed form of the denominator of Eq. (2), whereas MC approximation requires another higher-level MC approximation of quintic cost, which is prohibitively slow. So, we adopt BQ in this paper thanks to its good balance of computational accuracy and cost; however, our setting is not limited to BQ.

C Bayesian Quadrature for Fast Soft-Copeland Score Approximation

We have GP classifier $g \sim \mathcal{GP}(\mu_g, \kappa_g)$, and Monte Carlo integration via transforming sampled function with the link function can estimate the expectation of binary probability as Bernoulli distribution:

$$\mathbb{E}_g[g(x, x'|g, \mathbf{D}_{\mathrm{pref}})] = \int \frac{\exp(g_i)}{\sum \exp(g_i)} \mathbb{P}(g_i|x, x', \mathbf{D}_{\mathrm{pref}}) \mathrm{d}g \tag{53}$$

As this is intractable, the forthcoming Condorset winner $\hat{\pi}_g(x)$ marginalisation is also intractable.

Therefore, we apply Bayesian quadrature (BQ). Bayesian quadrature is the model-based approximation technique for intractable integration; typically, GP is applied for the surrogate model for the integrand. We apply the simple GP with RBF kernel to the integrand. Namely, we place the surrogate model GP with the pair of dataset:

$$\mathbf{x}_{\mathrm{bq}} := \mathbf{x}_{\mathrm{pref}} = (\mathbf{x}_1, \mathbf{x}_2), \tag{54}$$

$$\mathbf{y}_{\mathrm{bq}} := \mathbb{E}_g[g(\mathbf{x}_{\mathrm{pref}} \mid \mathbf{D}_{\mathrm{pref}})], \tag{55}$$

$$\mathbf{D}_{\mathrm{bq}} := \left(\mathbf{x}_{\mathrm{bq}}, \mathbf{y}_{\mathrm{bq}}\right), \tag{56}$$

Then, all integration in equation 2 becomes analytical:

$$
\begin{aligned}
f_{\mathrm{bq}} &\sim \mathcal{GP}(\mu_{\mathrm{bq}}, \kappa_{\mathrm{bq}} \mid \mathbf{D}_{\mathrm{bq}}), && (57)\\
\mu_{\mathrm{bq}}(X) &= k(X, \mathbf{x}_{\mathrm{bq}}) \left[k(\mathbf{x}_{\mathrm{bq}}, \mathbf{x}_{\mathrm{bq}}) + \lambda \mathbf{I}\right]^{-1} \mathbf{y}_{\mathrm{bq}}, && (58)\\
&= k(X, \mathbf{x}_{\mathrm{bq}})^{\top} \boldsymbol{\omega}, && (59)\\
&= v' \mathcal{N}(X; \mathbf{x}_{\mathrm{bq}}, \mathbf{W})^{\top} \boldsymbol{\omega}, && (60)\\
\kappa_{\mathrm{bq}}(X, X') &= k(X, X') - k(X, \mathbf{x}_{\mathrm{bq}}) \left[k(\mathbf{x}_{\mathrm{bq}}, \mathbf{x}_{\mathrm{bq}}) + \lambda \mathbf{I}\right]^{-1} k(\mathbf{x}_{\mathrm{bq}}, X), && (61)\\
&= v' \mathcal{N}(X; x', \mathbf{W}) - v'^{2} \mathcal{N}(X; \mathbf{x}_{\mathrm{bq}}, \mathbf{W}) \boldsymbol{\Omega} \mathcal{N}(X; \mathbf{x}_{\mathrm{bq}}, \mathbf{W})^{T}, && (62)\\
&= v' \mathcal{N}(X; x', \mathbf{W}) - v'^{2} \sum_{i,j}^{N} \Omega_{i,j} \mathcal{N}(X; x_{bq,i}, \mathbf{W}) \mathcal{N}(X; x_{bq,j}, \mathbf{W}), && (63)\\
&= v' \mathcal{N}(X; X', \mathbf{W}) - v'^{2} \mathcal{N}(x_{bq,i}; x_{bq,j}, 2\mathbf{W}) \sum_{i,j}^{N} \Omega_{i,j} \mathcal{N}\left(X; \frac{x_{bq,i} + x_{bq,j}}{2}, \frac{\mathbf{W}}{2}\right), && (64)
\end{aligned}
$$

where

$$
\begin{aligned}
X &:= (x, x'), && (65)\\
k(X, X') &:= v' \mathcal{N}(X; X', \mathbf{W}), && (66)\\
v' &= v \sqrt{|2\pi \mathbf{W}|}, && (67)\\
\mathbf{W} &:= \ell^2 \mathbf{I}, && (68)\\
\boldsymbol{\omega} &:= \left[k(\mathbf{x}_{\mathrm{bq}}, \mathbf{x}_{\mathrm{bq}}) + \lambda \mathbf{I}\right]^{-1} \mathbf{y}_{\mathrm{bq}}, && (69)\\
\boldsymbol{\Omega} &:= \left[k(\mathbf{x}_{\mathrm{bq}}, \mathbf{x}_{\mathrm{bq}}) + \lambda \mathbf{I}\right]^{-1}, && (70)
\end{aligned}
$$

v and ℓ are the output scale and length scale of the RBF kernel, and λ is the Gaussian likelihood variance. Posterior predictive mean and variance are just a mixture of Gaussians; thus integration is tractable.

Then, we consider the soft-Copeland score, which is the marginalisation of x' from $x = (x, x')$. Marginalisation of Gaussian is just extracting the corresponding elements. We have

$$
X = \begin{bmatrix} x \\ x' \end{bmatrix} \qquad \mathbf{x}_{\mathrm{bq}} = \begin{bmatrix} x_{\mathrm{bq}} \\ x'_{\mathrm{bq}} \end{bmatrix} \qquad \mathbf{W} = \begin{bmatrix} \mathbf{W}' & \mathbf{0} \\ \mathbf{0} & \mathbf{W}'' \end{bmatrix} \tag{71}
$$

Consequently, the soft-Copeland score is reduced to be:

$$
\begin{aligned}
\hat{\pi}_g(x) &:= \int_{\mathcal{X}} g(x, x') \mathrm{d}x', && (72)\\
&= \mathrm{V}_{\mathcal{X}}^{-1} \int g(x, x') \mathrm{d}x', && (73)
\end{aligned}
$$

where $\mathrm{V}_{\mathcal{X}} = \int \mathbb{E}_g[\hat{\pi}_g(x)] \mathrm{d}x$ is the normalizing constant to make $\hat{\pi}_g(x)$ become a probability density function.

$$
\begin{aligned}
\mathbb{E}_g\left[\hat{\pi}_g(x)\right] &= \mathrm{V}_{\mathcal{X}}^{-1} \int \mathbb{E}_g\left[g(x, x')\right] \mathrm{d}x', && (74)\\
&= \mathrm{V}_{\mathcal{X}}^{-1} \int \mu_{\mathrm{bq}}(x, x') \mathrm{d}x', && (75)\\
&= v' \mathrm{V}_{\mathcal{X}}^{-1} \int \mathcal{N}\left(X; \mathbf{x}_{\mathrm{bq}}, \mathbf{W}\right)^{\top} \mathrm{d}x' \boldsymbol{\omega}, && (76)\\
&= v' \mathrm{V}_{\mathcal{X}}^{-1} \int \mathcal{N}\left(\begin{bmatrix} x \\ x' \end{bmatrix}; \begin{bmatrix} x_{\mathrm{bq}} \\ x'_{\mathrm{bq}} \end{bmatrix}, \begin{bmatrix} \mathbf{W}' & \mathbf{0} \\ \mathbf{0} & \mathbf{W}'' \end{bmatrix}\right)^{\top} \mathrm{d}x' \boldsymbol{\omega}, && (77)\\
&= v' \mathrm{V}_{\mathcal{X}}^{-1} \mathcal{N}(x; x_{\mathrm{bq}}, \mathbf{W}')^{\top} \boldsymbol{\omega}, && (78)
\end{aligned}
$$

Similarly, we place another GP on the variance, $\mathbb{V}_g\left[g(x,x')\right]$, then the procedure is the same:

$$\mathbb{V}_g\left[\hat{\pi}_g(x)\right] = \mathrm{V}_{\mathcal{X}}^{-1}\int \mathbb{E}_g\left[g(x,x')(1-g(x,x'))\right]\mathrm{d}x', \tag{79}$$

$$\approx \mathrm{V}_{\mathcal{X}}^{-1}\int \mu'_{\mathrm{bq}}(x,x')\mathrm{d}x', \tag{80}$$

$$= v'\mathrm{V}_{\mathcal{X}}^{-1}\mathcal{N}(x;x''_{\mathrm{bq}},\mathbf{W}''')^\top\boldsymbol{\omega}, \tag{81}$$

Then, the soft-Copeland score can be approximated by moment-matching the original Bernoulli distribution with the Gaussian distribution.

$$\pi_g(x) \approx \mathcal{N}(\hat{\pi}_g(x);\mathbb{E}_g\left[\hat{\pi}_g(x)\right],\mathbb{V}_g\left[\hat{\pi}_g(x)\right]). \tag{82}$$

This is a coarse approximation of Condorcet winner probability $\mathbb{P}(y=1|x)$ and is probabilistically wrong (Gaussian is not bounded in [0, 1]). Still, recall our original motivation is to model the probability of global optimal location, which is not bounded in [0, 1]. In this sense, precisely computing Bernoulli distribution is not required. We further use this soft-Copeland score function to model the prior distribution on the continuous value y, which is not the Bernoulli distribution and rather assumes Gaussian distribution. Thus we adopt Gaussian moment-matching approximation.

However, this soft-Copeland score is not normalised over the domain, so we need to take further integral over x domain. Bayesian quadrature makes this integral analytical:

$$\mathrm{V}_{\mathcal{X}} = \int_{\mathcal{X}}\mathbb{E}_g\left[\hat{\pi}_g(x)\right]\mathrm{d}x \tag{83}$$

$$= v'\mathrm{V}_{\mathcal{X}}^{-1}\int\mathcal{N}(x;x_{\mathrm{bq}},\mathbf{W}')^\top\mathrm{d}x\boldsymbol{\omega}, \tag{84}$$

$$= v'\mathrm{V}_{\mathcal{X}}^{-1}\mathbf{1}^\top\boldsymbol{\omega}, \tag{85}$$

$$= \sqrt{v'\mathbf{1}^\top\boldsymbol{\omega}}, \tag{86}$$

$$\pi(x) := \frac{\mathbb{E}_g[\pi_g(x]}{\int_{\mathcal{X}}\mathbb{E}_g[\hat{\pi}_g(x)]\mathrm{d}x}, \tag{87}$$

$$= \frac{1}{\mathbf{1}^\top\boldsymbol{\omega}}\mathcal{N}(x;x_{\mathrm{bq}},\mathbf{W}')^\top\boldsymbol{\omega} \tag{88}$$

Now we have the closed-form soft-Copeland score approximation. It should be noted that the original $g(x,x')$ distribution is the Bernoulli distribution, whereas this BQ-GP is the Gaussian distribution, which is a crude approximation. So only predictive mean is reliable. To estimate the variance of $\mathbb{V}_g[\hat{\pi}_g(x)]$ at the same time, we need to have another GP for variance estimation. Even the predictive mean of BQ-GP can be a crude approximation. One simple solution to boost the accuracy is to increase the number of data $\mathbf{D}_{\mathrm{bq}}$, as this can be augmented cheaply by f_{pref}. However, increasing the number can lead to large computational overhead as training GP costs cubic complexity $\mathcal{O}(n^3)$. We wish to minimise the number of augmented data. So we adopt BASQ Adachi et al. (2022) for selecting the next query point. This allows provably small predictive uncertainty (see Theorem 1 in Adachi et al. (2022) and also Hayakawa et al. (2022)).

D Dueling Acquisition Function

Figure 10 visualizes the dueling acquisition function. In the first cycle, the distance between preference-based and standard UCB-based recommendations is large. But it gradually decreases over iterations, and it is almost the same in the last (fourth) iteration. Furthermore, this figure also shows that the human preference successfully avoids climbing up the wrong left peak by placing the belief over the left peak, which accelerates early-stage exploration.

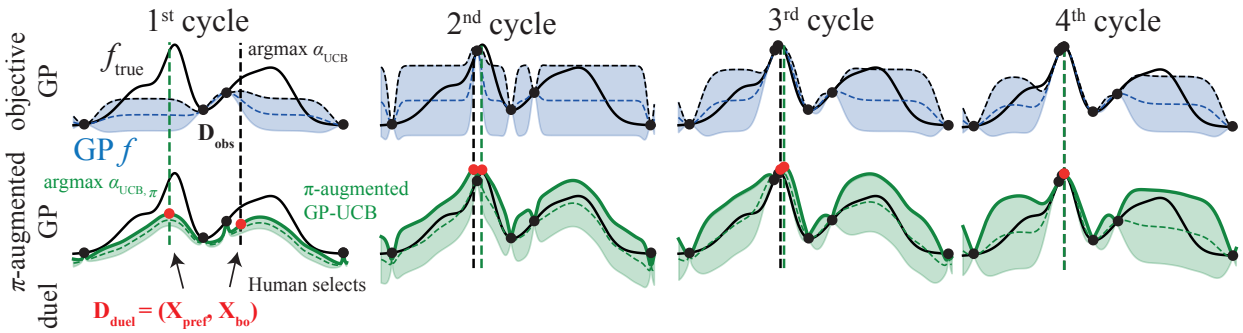


Figure 10: Dueling candidate generation algorithm. While the optimistic sample is selected by maximizing the π-augmented GP, the pessimistic sample is selected by the original objective GP. The distance between pairwise samples asymptotically decreases over iterations. (We set $\gamma = 0.1$).

E Explaining Bayesian Optimization

We have three explanation features explained in Figure 2; Spatial relation, feature importance, and selection accuracy feedback. For the spatial relation, we select the two primary dimensions using Shapley values. As Shapley values are conditioned on x, we need to select x. For simplicity, we take average of Shapley values at the pairwise candidates, x_1 and x_2, then take two dimensions with top 2 mean Shapley values. For the drawing range, we first compute the rectangle which bounds the following three points; x_1, x_2, and the current best observed point $x_{\text{current}} := \arg\max \mathbf{D}_{obj_t}$. Then, we expand this bounded rectangle 2 times as large as the original for visibility. We set this expanded rectangle as the visualisation range. This procedure is shared with the preference model visualisation. For feature importance, we simply visualise the Shapley values at x_1 and x_2 as a bar plot.

For selection accuracy feedback, we first update the GP surrogate function with the queried point $x_t \in (x_1, x_2)$ and the queried value $y_t = f(x_t)$. For the sake of argument, we assume $x_t = x_1$. Then, we compute the probability of correct selection, given by:

$$\mathbb{P}(f(x_1) \geq f(x_2)) \sim \mathcal{N}(\mathbb{E}_f[\ell(x_1, x_2)], \mathbb{V}_f[\ell(x_1, x_2)]), \tag{89}$$

$$\ell(x_1, x_2) := \Phi\left(\frac{f(x_1) - f(x_2)}{\sqrt{\lambda}}\right), \tag{90}$$

where Φ is the cumulative density function of standard normal distribution $\mathcal{N}(0, 1)$. We compute the expectation and variance over f space by Monte Carlo integration.

F Experimental details

We have tested CoExBO for 5 synthetic functions against 6 baselines. We use a constant-mean GP with an RBF kernel. In each iteration of the active learning loop, the outputs are standardized to have zero mean and unit variance. We optimize the hyperparameter by maximizing the marginal likelihood (type-II maximum likelihood estimation) using L-BFGS-B optimizer (Liu & Nocedal, 1989) implemented with BoTorch (Balandat et al., 2020). We also maximized the acquisition function using the same optimizer. The initial data sets consist of ten data points drawn by Sobol sequence (Sobol', 1967) and corresponding noisy observations. We generate 10 samples for objective dataset and 100 samples for preferential dataset construction. We adopt the log regret as the evaluation metric using the test dataset. The models are implemented in GPyTorch (Gardner et al., 2018). All experiments are repeated ten times with different initial data sets via different random seeds (the seeds are shared with baseline methods).

F.1 Synthetic functions with Synthetic Human Selection

F.1.1 Synthetic Functions

Ackely Ackley funciton is defined as:

$$f(x) := -a \exp\left[-b\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right] - \exp\left[\frac{1}{d}\sum_{i=1}^{d}\cos(cx_i)\right] + a + \exp(1) \tag{91}$$

where $a = 20, c = 2\pi, d = 4$. We take the negative Ackley function as the objective of BO to make this optimisation problem maximisation. This is a 4-dimensional function bounded by $x \in [-1, 1]^d$. The global optimum is $x^*_{\text{true}} = [0, 0, 0, 0]$ and $f(x^*_{\text{true}}) = 0$.

Hölder Table Hölder Table funciton is defined as:

$$f(x) := \left|\sin(x_1)\cos(x_2)\exp\left(\left|1 - \frac{\sqrt{x_1^2 + x_2^2}}{\pi}\right|\right)\right| \tag{92}$$

where x_i is the i-th dimensional input. This is a 2-dimensional function bounded by $x \in [0, 10]^d$. The global optimum is $x^*_{\text{true}} = [8.05502, 9.66459]$ and $f(x^*_{\text{true}}) = 19.2085$.

Styblinski-Tang Styblinski-Tang funciton is defined as:

$$f(x) := \frac{1}{2}\sum_{i=1}^{d}(x_i^4 - 16x_i^2 + 5x_i) \tag{93}$$

where x_i is the i-th dimensional input. This is a 3-dimensional function bounded by $x \in [-5, 5]^d$. The global optimum is $x^*_{\text{true}} = [-2.903534]^d$ and $f(x^*_{\text{true}}) = 39.166166d$.

Michalewicz Michalewicz funciton is defined as:

$$f(x) := \sum_{i=1}^{d}\sin(x_i)\sin^{2m}\left(\frac{ix_i^2}{\pi}\right) \tag{94}$$

where x_i is the i-th dimensional input and $m = 10$. This is a 5-dimensional function bounded by $x \in [0, \pi]^d$. The global optimum is $f(x^*_{\text{true}}) = 4.687658$.

Rosenbrock Rosenbrock funciton is defined as:

$$f(x) := \sum_{i=1}^{d-1}\left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right] \tag{95}$$

where x_i is the i-th dimensional input. This is a 3-dimensional function bounded by $x \in [-5, 10]^d$. The global optimum is $x^*_{\text{true}} = [1]^d$ and $f(x^*_{\text{true}}) = 0$.

F.1.2 Robustness evaluation

In the adversarial selection, both CoExBO and batchUCB superficially surpass the original UCB. We first point out that this is within the standard error, but there may be possible reasons why these two are better than others even in adversarial settings. For CoExBO, this may come from randomised effect. Recent work shows randomizing β parameter of UCB yields faster convergence than original (Berk et al., 2020; Takeno et al., 2023). Our CoExBO can be understood as randomising β, which may effect positively. Still, we can confirm the trend that confident and correct human feedback can accelerate convergence. For batchUCB, this may come from a nonmyopic effect. González et al. (2016) pointed out the similarity between hallucination and one-step lookahead BO, which empirically yields better convergence than the original UCB.

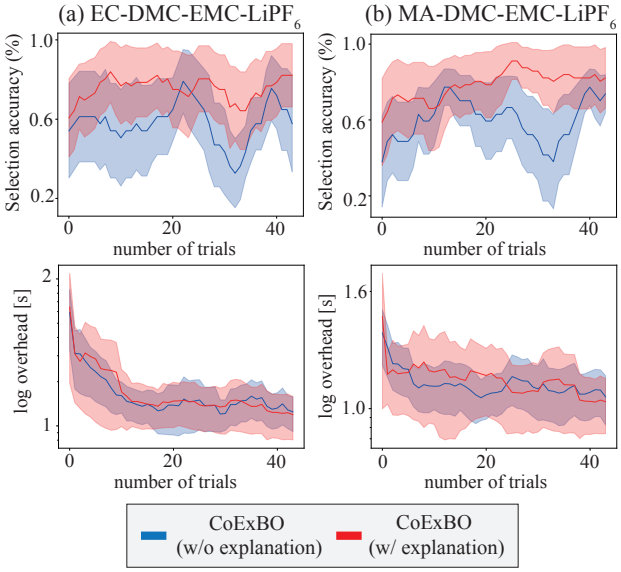


Figure 11: Selection accuracy and overhead analysis over iterations for the tasks (a) EC-DMC-EMC-LiPF$_6$ and (b) MA-DMC-EMC-LiPF$_6$. The solid lines and shaded areas refer to the mean and 1 standard error of the results of four participants. To smooth out the noisy results, we take the moving average with the window size of 3 trials.

F.2 Real-world tasks

F.2.1 Designing battery

The problem involves finding the best electrolyte material combination to maximize ionic conductivity. Ionic conductivity is dependent on both lithium salt molarity and the cosolvent composition. They show the complex non-linear relationship due to the solvation effect and cannot predict even with the state-of-the-art quantum chemistry simulator. We create the true functions by fitting the experimental data of EC-DMC-EMC-LiPF$_6$ (Dave et al., 2022) and MA-DMC-EMC-LiPF$_6$ (Logan et al., 2018) systems using the Casteel-Amis equation (Casteel & Amis, 1972). Both tasks are the three-dimensional continuous input function. The input features are (1) the lithium salt (LiPF$_6$) molarity, (2) DMC/EMC cosolvent ratio, and (3) (EC or MA)/carbonates cosolvent ratio, respectively. The inputs are bounded with $x_1 \in [0, 2]$, $x_2 \in [0, 1]$, and $x_3 \in [0, 1]$. The noisy output is generated by adding i.i.d. zero-mean Gaussian noise with the 3^2 variance to the noiseless $f(x)$.

F.2.2 Selection Accuracy and Complexity Analysis

We further analyzed the real-world task results based on selection accuracy and overhead over iterations. Figure 11 illustrates the results. For selection accuracy, while results with explanation remain stable over iterations, the ones without explanation fluctuate largely, particularly in the later rounds. Over iterations, the pairwise candidates become closer due to the no-harm guarantee. Hence, the later iterations are more difficult to select the correct one. The explanation feature can help users distinguish the slight differences by the quantitative Shapley values, leading to accurate selection even for the later iterations. The bottom row of Figure 11 shows the overhead of the candidate selection process, including pairwise candidate generation, explanation feature, and human selection time. We can observe the general decrease trend over iterations regardless of the explanation feature, and the difference in overhead between with and without explanation features is negligible. This is because the most time-consuming part is the human selection process. In the early stage, human users are also uncertain and need more time to decide which to select. Over time, it becomes more confident and smoother to select, resulting in quicker selection. This suggests the algorithmic overhead is negligible when compared to the human selection.

E

Appendix of Chapter 7

Part I
Appendix

Table of Contents

A Proof of Lem. 3.2

To prepare for the proof of the lemma, we first prove several preliminary lemmas.

Lemma A.1. *For any fixed $\hat{g} \in \mathcal{B}_g$, we have,*

$$\mathbb{P}\left(\log \mathbb{P}_{\hat{g}}((x_\tau, \mathbf{1}_\tau)_{\tau \in \mathcal{Q}_t^g}) - \log \mathbb{P}_g((x_\tau, \mathbf{1}_\tau)_{\tau \in \mathcal{Q}_t^g}) \leq \sqrt{8|\mathcal{Q}_t^g| B_f^2 \log \frac{1}{\delta_t}}\right) \geq 1 - \delta_t. \tag{8}$$

Proof. We use u_τ to denote $g(x_\tau)$, z_τ to denote $\hat{g}(x_\tau)$, and p_τ to denote $S(g(x_\tau))$.

$$\mathbb{P}\left(\log \mathbb{P}_{\hat{g}}((x_\tau, \mathbf{1}_\tau)_{\tau\in\mathcal{Q}_t^g}) - \log \mathbb{P}_g((x_\tau, \mathbf{1}_\tau)_{\tau\in\mathcal{Q}_t^g}) \le \xi\right) \tag{9}$$

$$=\mathbb{P}\left(\sum_{\tau\in\mathcal{Q}_t^g}((z_\tau - u_\tau)\mathbf{1}_\tau - \log(1+e^{z_\tau}) + \log(1+e^{u_\tau})) \le \xi\right) \tag{10}$$

$$=\mathbb{P}\left(\sum_{\tau\in\mathcal{Q}_t^g}(z_\tau - u_\tau)\mathbf{1}_\tau - \sum_{\tau\in\mathcal{Q}_t^g}(z_\tau - u_\tau)p_\tau \le \xi'\right) \tag{11}$$

where the probability $\mathbb{P}$ is taken over the randomness from the feedback expert/oracle and the randomness from the algorithm, and $\xi' = \xi + \sum_{\tau\in\mathcal{Q}_t^g}\log(1+e^{z_\tau}) - \sum_{\tau\in\mathcal{Q}_t^g}\log(1+e^{u_\tau}) - \sum_{\tau\in\mathcal{Q}_t^g}(z_\tau - u_\tau)p_\tau$. Let the function $\psi_\tau(z_\tau) := \log(1+e^{z_\tau}) - \log(1+e^{u_\tau}) - (z_\tau - u_\tau)p_\tau$. It can be checked that $\psi_\tau^{''}(z_\tau) = {}^{e^{z_\tau}}/_{(1+e^{z_\tau})^2} \ge 0, \forall z_\tau \in \mathbb{R}$ and $\psi_\tau^{'}(u_\tau) = 0$. Therefore, ψ_τ is a convex function and achieves the optimal value at the point u_τ. Hence, $\psi_\tau(z_\tau) \ge \psi_\tau(u_\tau) = 0$, which implies $\xi' \ge \xi$. Therefore,

$$\mathbb{P}\left(\sum_{\tau\in\mathcal{Q}_t^g}(z_\tau - u_\tau)\mathbf{1}_\tau - \sum_{\tau\in\mathcal{Q}_t^g}(z_\tau - u_\tau)p_\tau \le \xi'\right) \ge \mathbb{P}\left(\sum_{\tau\in\mathcal{Q}_t^g}(z_\tau - u_\tau)\mathbf{1}_\tau - \sum_{\tau\in\mathcal{Q}_t^g}(z_\tau - u_\tau)p_\tau \le \xi\right). \tag{12}$$

Furthermore, it is easy to see that $(z_\tau - u_\tau)\mathbf{1}_\tau \in [-2B_g, 2B_g]$, and thus, by applying Azuma-Hoeffding inequality, we have,

$$\mathbb{P}\left(\sum_{\tau\in\mathcal{Q}_t^g}(z_\tau - u_\tau)\mathbf{1}_\tau - \sum_{\tau\in\mathcal{Q}_t^g}(z_\tau - u_\tau)p_\tau \le \xi\right) \ge 1 - \exp\left\{-\frac{\xi^2}{8|\mathcal{Q}_t^g|B_g^2}\right\} \tag{13}$$

Let $\exp\left\{-\frac{\xi^2}{8|\mathcal{Q}_t^g|B_g^2}\right\} \le \delta_t$, we need,

$$\xi \ge \sqrt{8|\mathcal{Q}_t^g|B_g^2 \log\frac{1}{\delta_t}}. \tag{14}$$

It is sufficient to pick $\xi = \sqrt{8|\mathcal{Q}_t^g|B_g^2 \log\frac{1}{\delta_t}}$. Therefore,

$$\begin{aligned}
&\mathbb{P}\left(\log \mathbb{P}_{\hat{g}}((x_\tau, \mathbf{1}_\tau)_{\tau\in\mathcal{Q}_t^g}) - \log \mathbb{P}_g((x_\tau, \mathbf{1}_\tau)_{\tau\in\mathcal{Q}_t^g}) \le \sqrt{8|\mathcal{Q}_t^g|B_g^2 \log\frac{1}{\delta_t}}\right) \\
\ge&\mathbb{P}\left(\sum_{\tau\in\mathcal{Q}_t^g}(z_\tau - u_\tau)\mathbf{1}_\tau - \sum_{\tau\in\mathcal{Q}_t^g}(z_\tau - u_\tau)p_\tau \le \sqrt{8|\mathcal{Q}_t^g|B_g^2 \log\frac{1}{\delta_t}}\right) \\
\ge&1 - \delta_t,
\end{aligned}$$

where the first inequality follows by combining Eq. (11) and Eq. (12).

□

We then have the following high probability confidence set lemma.

Lemma A.2. *For any fixed $\hat{g}$ that is independent of $((x_\tau, \mathbf{1}_\tau)_{\tau\in\mathcal{Q}_t^g})$, we have, with probability at least $1-\delta$, $\forall t \ge 1$,*

$$\log \mathbb{P}_{\hat{g}}((x_\tau, \mathbf{1}_\tau)_{\tau\in\mathcal{Q}_t^g}) - \log \mathbb{P}_g((x_\tau, \mathbf{1}_\tau)_{\tau\in\mathcal{Q}_t^g}) \le \sqrt{8|\mathcal{Q}_t^g|B_g^2 \log\frac{\pi^2 t^2}{6\delta}}. \tag{15}$$

Proof. We use $\mathcal{E}_t$ to denote the event $\log \mathbb{P}_{\hat{g}}((x_\tau, \mathbf{1}_\tau)_{\tau \in \mathcal{Q}_t^g}) - \log \mathbb{P}_g((x_\tau, \mathbf{1}_\tau)_{\tau \in \mathcal{Q}_t^g}) \leq \sqrt{8|\mathcal{Q}_t^g| B_g^2 \log \frac{1}{\delta_t}}$. We pick $\delta_t = {}^{(6\delta)}/_{(\pi^2 t^2)}$ and have,

$$
\begin{aligned}
&\mathbb{P}\left(\log \mathbb{P}_{\hat{g}}((x_\tau, \mathbf{1}_\tau)_{\tau \in \mathcal{Q}_t^g}) - \log \mathbb{P}_g((x_\tau, \mathbf{1}_\tau)_{\tau \in \mathcal{Q}_t^g}) \leq \sqrt{8|\mathcal{Q}_t^g| B_g^2 \log \frac{1}{\delta_t}}, \forall t \geq 1\right) \\
=&1 - \mathbb{P}\left(\overline{\cap_{t=1}^{\infty} \mathcal{E}_t}\right) \\
=&1 - \mathbb{P}\left(\cup_{t=1}^{\infty} \overline{\mathcal{E}_t}\right) \\
\geq&1 - \sum_{t=1}^{\infty} \mathbb{P}\left(\overline{\mathcal{E}_t}\right) \\
=&1 - \sum_{t=1}^{\infty} \left(1 - \mathbb{P}\left(\mathcal{E}_t\right)\right) \\
=&1 - \sum_{t=1}^{\infty} \left(1 - \mathbb{P}\left(\log \mathbb{P}_{\hat{g}}((x_\tau, \mathbf{1}_\tau)_{\tau \in \mathcal{Q}_t^g}) - \log \mathbb{P}_g((x_\tau, \mathbf{1}_\tau)_{\tau \in \mathcal{Q}_t^g}) \leq \sqrt{8|\mathcal{Q}_t^g| B_g^2 \log \frac{1}{\delta_t}}\right)\right) \\
\geq&1 - \sum_{t=1}^{\infty} \delta_t \\
=&1 - \frac{6\delta}{\pi^2} \sum_{t=1}^{\infty} \frac{1}{t^2} \\
=&1 - \delta.
\end{aligned}
$$

□

We then have a lemma to bound the difference of log likelihood when two functions are close in infinity-norm sense.

Lemma A.3. *$\forall \epsilon > 0$, $\forall g_1, g_2 \in \mathcal{B}_g$ that satisfies $\|g_1 - g_2\|_\infty \leq \epsilon$, we have,*

$$
\log \mathbb{P}_{g_1}((x_\tau, \mathbf{1}_\tau)_{\tau \in \mathcal{Q}_t^g}) - \log \mathbb{P}_{g_2}((x_\tau, \mathbf{1}_\tau)_{\tau \in \mathcal{Q}_t^g}) \leq 2\epsilon |\mathcal{Q}_t^g|. \tag{16}
$$

Proof.

$$
\begin{aligned}
&\log \mathbb{P}_{g_1}((x_\tau, \mathbf{1}_\tau)_{\tau \in \mathcal{Q}_t^g}) - \log \mathbb{P}_{g_2}((x_\tau, \mathbf{1}_\tau)_{\tau \in \mathcal{Q}_t^g}) \\
\leq& \sum_{\tau \in \mathcal{Q}_t^g} \left((z_{1,\tau} - z_{2,\tau})\mathbf{1}_\tau - \log(1 + e^{z_{1,\tau}}) + \log(1 + e^{z_{2,\tau}})\right) \\
\leq& \epsilon |\mathcal{Q}_t^g| + \sum_{\tau \in \mathcal{Q}_t^g} \max_{z \in [-B_g, B_g]} \left|\nabla_z \log\left(1 + e^z\right)\right| |z_{1,\tau} - z_{2,\tau}| \\
\leq& \epsilon |\mathcal{Q}_t^g| + \sum_{\tau \in \mathcal{Q}_t^g} \epsilon \\
\leq& 2\epsilon |\mathcal{Q}_t^g|,
\end{aligned}
$$

where $z_{1,\tau} = g_1(x_\tau)$ and $z_{2,\tau} = g_2(x_\tau)$. □

We use $\mathcal{N}(\mathcal{B}_g, \epsilon, \|\cdot\|_\infty)$ to denote the covering number of the set $\mathcal{B}_g$, with $(g_i^\epsilon)_{i=1}^{\mathcal{N}(\mathcal{B}_g, \epsilon, \|\cdot\|_\infty)}$ be a set of ϵ-covering for the set $\mathcal{B}_g$. Set the 'δ' in Lem. A.2 as ${}^{\delta}/_{\mathcal{N}(\mathcal{B}_g, \epsilon, \|\cdot\|_\infty)}$ and applying the probability union bound, we have, with probability at least $1 - \delta$, $\forall g_i^\epsilon$,

$$
\log \mathbb{P}_{g_i^\epsilon}((x_\tau, \mathbf{1}_\tau)_{\tau \in \mathcal{Q}_t^g}) - \log \mathbb{P}_g((x_\tau, \mathbf{1}_\tau)_{\tau \in \mathcal{Q}_t^g}) \leq \sqrt{8|\mathcal{Q}_t^g| B_g^2 \log \frac{\pi^2 t^2 \mathcal{N}(\mathcal{B}_g, \epsilon, \|\cdot\|_\infty)}{6\delta}}. \tag{17}
$$

By the definition of ϵ-covering, there exists $j \in [\mathcal{N}(\mathcal{B}_g, \epsilon, \|\cdot\|_\infty)]$, such that,

$$
\|\hat{g}_{t+1}^{\mathrm{MLE}} - g_j^\epsilon\|_\infty \leq \epsilon. \tag{18}
$$

Hence, with probability at least $1-\delta$,

$$\begin{aligned}
&\log \mathbb{P}_{\hat{g}_{t+1}^{\mathrm{MLE}}}((x_\tau, \mathbf{1}_\tau)_{\tau\in\mathcal{Q}_t^g}) - \log \mathbb{P}_g((x_\tau, \mathbf{1}_\tau)_{\tau\in\mathcal{Q}_t^g}) \\
=&\log \mathbb{P}_{\hat{g}_{t+1}^{\mathrm{MLE}}}((x_\tau, \mathbf{1}_\tau)_{\tau\in\mathcal{Q}_t^g}) - \log \mathbb{P}_{g_j^\epsilon}((x_\tau, \mathbf{1}_\tau)_{\tau\in\mathcal{Q}_t^g}) + \log \mathbb{P}_{g_j^\epsilon}((x_\tau, \mathbf{1}_\tau)_{\tau\in\mathcal{Q}_t^g}) - \log \mathbb{P}_g((x_\tau, \mathbf{1}_\tau)_{\tau\in\mathcal{Q}_t^g}) \\
\leq& 2\epsilon|\mathcal{Q}_t^g| + \sqrt{8|\mathcal{Q}_t^g|B_g^2 \log\frac{\pi^2 t^2 \mathcal{N}(\mathcal{B}_g, \epsilon, \|\cdot\|_\infty)}{6\delta}},
\end{aligned}$$

where the inequality follows by Lem. A.3 and Lem. A.2.

B Proof of Thm. 4.1

B.1 Bound Error over Historical Evaluations

Lem. 3.2 gives a high confidence set based on the likelihood function. The following Lem. B.1 further gives error bound over the historical sample points. Lem. B.1 highlights that with high probability, all the functions in the confidence set have values over the historical sample points that lie in a ball with the ground-truth function value as the center and $\sqrt{\alpha(\epsilon, \delta/2, |\mathcal{Q}_t^g|, t)}$ as the radius. Before we proceed, we first introduce several constants that we will use,

$$\bar{S} := \max_{u\in[-B_g, B_g]} S(u) = \frac{1}{1+e^{-B_g}}, \underline{S} := \min_{u\in[-B_g, B_g]} S(u) = \frac{1}{1+e^{B_g}}. \tag{19}$$

$$\underline{S'} := \min_{u\in[-B_g, B_g]} S'(u) = \frac{1}{e^{B_g}+e^{-B_g}+2}, \bar{S'} := \max_{u\in[-B_g, B_g]} S'(u) = \frac{1}{4}. \tag{20}$$

$$H_S := \frac{1}{2\bar{S}^2}, B_p = \frac{S(B_g)}{S(-B_g)} - \frac{S(-B_g)}{S(B_g)}. \tag{21}$$

Lemma B.1. *For any estimate $\hat{g}_{t+1} \in \mathcal{B}_g^{t+1}$ that is measurable with respect to the filtration $\mathcal{F}_t$, we have, with probability at least $1-\delta/2$, $\forall t \geq 1$,*

$$\sum_{\tau\in\mathcal{Q}_t^g} \left(\hat{g}_{t+1}(x_\tau) - g(x_\tau)\right)^2 \leq \alpha(\epsilon, \delta/2, |\mathcal{Q}_t^g|, t), \tag{22}$$

and

$$g \in \mathcal{B}_g^{t+1}, \tag{23}$$

where $\alpha(\epsilon, \delta/2, |\mathcal{Q}_t^g|, t) = \frac{\bar{S'}^2}{\underline{S_S}}\left(\alpha_2(\epsilon, \delta/2, |\mathcal{Q}_t^g|, t) + 2\alpha_1(\epsilon, \delta/2, |\mathcal{Q}_t^g|, t)\right) = \mathcal{O}\left(\sqrt{|\mathcal{Q}_t^g| \log\frac{t\mathcal{N}(\mathcal{B}_g, \epsilon, \|\cdot\|_\infty)}{\delta}} + \epsilon t + \epsilon^2 t\right)$, *with* $\alpha_2(\epsilon, \delta, |\mathcal{Q}_t^g|, t) = 8H_S\bar{S'}^2\epsilon^2 t + 4\epsilon t + \sqrt{8|\mathcal{Q}_t^g|B_p^2 \log\frac{\pi^2 t^2 \mathcal{N}(\mathcal{B}_g, \epsilon, \|\cdot\|_\infty)}{3\delta}}$.

Proof. For any fixed function $\hat{g}$, we have,

$$\begin{aligned}
&\log \mathbb{P}_{\hat{g}}((x_\tau, \mathbf{1}_\tau)_{\tau\in\mathcal{Q}_t^g}) - \log \mathbb{P}_g((x_\tau, \mathbf{1}_\tau)_{\tau\in\mathcal{Q}_t^g}) \\
=& \sum_{\tau\in\mathcal{Q}_t^g} \left(\log \mathbb{P}_{\hat{g}}((x_\tau, \mathbf{1}_\tau)) - \log \mathbb{P}_g((x_\tau, \mathbf{1}_\tau))\right) \\
=& \sum_{\tau\in\mathcal{Q}_t^g} \left(\mathbf{1}_\tau\left(\log\hat{p}_\tau - \log p_\tau\right) + (1-\mathbf{1}_\tau)\left(\log\left(1-\hat{p}_\tau\right) - \log(1-p_\tau)\right)\right),
\end{aligned}$$

where $\hat{p}_\tau = S(\hat{g}(x_\tau))$ and $p_\tau = S(g(x_\tau))$. We have,

$$\log y \leq \log x + \frac{1}{x}(y-x) - H_S(y-x)^2, \forall x, y \in [\underline{S}, \bar{S}], \tag{24}$$

where $H_S = \frac{1}{2S^2}$. Hence,

$$\begin{aligned}
&\log \mathbb{P}_{\hat{g}}((x_\tau, \mathbf{1}_\tau)_{\tau \in \mathcal{Q}_t^g}) - \log \mathbb{P}_g((x_\tau, \mathbf{1}_\tau)_{\tau \in \mathcal{Q}_t^g}) \\
=& \sum_{\tau \in \mathcal{Q}_t^g} \left(\mathbf{1}_\tau \left(\log \hat{p}_\tau - \log p_\tau\right) + (1 - \mathbf{1}_\tau) \left(\log\left(1 - \hat{p}_\tau\right) - \log(1 - p_\tau)\right)\right) \\
\leq& \sum_{\tau \in \mathcal{Q}_t^g} \left(\mathbf{1}_\tau \left(\frac{\hat{p}_\tau - p_\tau}{p_\tau} - H_S \left(\hat{p}_\tau - p_\tau\right)^2\right) + (1 - \mathbf{1}_\tau) \left(\frac{p_\tau - \hat{p}_\tau}{1 - p_\tau} - H_S \left(\hat{p}_\tau - p_\tau\right)^2\right)\right)
\end{aligned}$$

Rearrangement gives,

$$\begin{aligned}
&H_S \sum_{\tau \in \mathcal{Q}_t^g} \left(\hat{p}_\tau - p_\tau\right)^2 + \log \mathbb{P}_{\hat{g}}((x_\tau, \mathbf{1}_\tau)_{\tau \in \mathcal{Q}_t^g}) - \log \mathbb{P}_g((x_\tau, \mathbf{1}_\tau)_{\tau \in \mathcal{Q}_t^g}) \\
\leq& \sum_{\tau \in \mathcal{Q}_t^g} \left(\mathbf{1}_\tau \frac{\hat{p}_\tau - p_\tau}{p_\tau} + (1 - \mathbf{1}_\tau) \frac{p_\tau - \hat{p}_\tau}{1 - p_\tau}\right).
\end{aligned}$$

Since $\mathbb{E}\left[\mathbf{1}_\tau \frac{\hat{p}_\tau - p_\tau}{p_\tau} + (1 - \mathbf{1}_\tau) \frac{p_\tau - \hat{p}_\tau}{1 - p_\tau} | \mathcal{F}_{\tau-1}\right] = \mathbb{E}\left[p_\tau \frac{\hat{p}_\tau - p_\tau}{p_\tau} + (1 - p_\tau) \frac{p_\tau - \hat{p}_\tau}{1 - p_\tau} | \mathcal{F}_{\tau-1}\right] = 0$ and with probability one,

$$\begin{aligned}
\left|\mathbf{1}_\tau \frac{\hat{p}_\tau - p_\tau}{p_\tau} + (1 - \mathbf{1}_\tau) \frac{p_\tau - \hat{p}_\tau}{1 - p_\tau}\right| &\leq \mathbf{1}_\tau \left|\frac{\hat{p}_\tau - p_\tau}{p_\tau}\right| + (1 - \mathbf{1}_\tau) \left|\frac{p_\tau - \hat{p}_\tau}{1 - p_\tau}\right| && (25) \\
&= \mathbf{1}_\tau \left|\frac{\hat{p}_\tau}{p_\tau} - 1\right| + (1 - \mathbf{1}_\tau) \left|\frac{1 - \hat{p}_\tau}{1 - p_\tau} - 1\right| && (26) \\
&\leq \frac{S(B_g)}{S(-B_g)} - \frac{S(-B_g)}{S(B_g)} = B_p. && (27)
\end{aligned}$$

By Azuma–Hoeffding inequality, we have, $\forall \xi > 0$,

$$\mathbb{P}\left(\sum_{\tau \in \mathcal{Q}_t^g} \left(\mathbf{1}_\tau \frac{\hat{p}_\tau - p_\tau}{p_\tau} + (1 - \mathbf{1}_\tau) \frac{p_\tau - \hat{p}_\tau}{1 - p_\tau}\right) \leq \xi\right) \geq 1 - \exp\left\{-\frac{2\xi^2}{|\mathcal{Q}_t^g| B_p^2}\right\}. \tag{28}$$

We set $\exp\left\{-\frac{2\xi^2}{|\mathcal{Q}_t^g| B_p^2}\right\} = \delta_t > 0$, and derive

$$\mathbb{P}\left(H_S \sum_{\tau \in \mathcal{Q}_t^g} \left(\hat{p}_\tau - p_\tau\right)^2 + \log \mathbb{P}_{\hat{g}}((x_\tau, \mathbf{1}_\tau)_{\tau \in \mathcal{Q}_t^g}) - \log \mathbb{P}_g((x_\tau, \mathbf{1}_\tau)_{\tau \in \mathcal{Q}_t^g}) \leq \sqrt{\frac{|\mathcal{Q}_t^g| B_p^2 \log \frac{1}{\delta_t}}{2}}\right) \tag{29}$$

$$\geq \mathbb{P}\left(\sum_{\tau \in \mathcal{Q}_t^g} \left(\mathbf{1}_\tau \frac{\hat{p}_\tau - p_\tau}{p_\tau} + (1 - \mathbf{1}_\tau) \frac{p_\tau - \hat{p}_\tau}{1 - p_\tau}\right) \leq \sqrt{\frac{|\mathcal{Q}_t^g| B_p^2 \log \frac{1}{\delta_t}}{2}}\right) \tag{30}$$

$$\geq 1 - \delta_t. \tag{31}$$

We use $\mathcal{E}_t$ to denote the event $H_S \sum_{\tau\in\mathcal{Q}_t^g}(\hat{p}_\tau - p_\tau)^2 \leq \log \mathbb{P}_g((x_\tau, \mathbf{1}_\tau)_{\tau\in\mathcal{Q}_t^g}) - \log \mathbb{P}_{\hat{g}}((x_\tau, \mathbf{1}_\tau)_{\tau\in\mathcal{Q}_t^g}) + \sqrt{\frac{|\mathcal{Q}_t^g| B_p^2 \log\frac{1}{\delta_t}}{2}}$. We pick $\delta_t = {}^{(6\delta)}/_{(\pi^2 t^2)}$. We have,

$$
\begin{aligned}
&\mathbb{P}\left(H_S \sum_{\tau\in\mathcal{Q}_t^g}(\hat{p}_\tau - p_\tau)^2 \leq \log \mathbb{P}_g((x_\tau, \mathbf{1}_\tau)_{\tau\in\mathcal{Q}_t^g}) - \log \mathbb{P}_{\hat{g}}((x_\tau, \mathbf{1}_\tau)_{\tau\in\mathcal{Q}_t^g}) + \sqrt{\frac{|\mathcal{Q}_t^g| B_p^2 \log\frac{1}{\delta_t}}{2}}, \forall t \geq 1\right)\\
=&1 - \mathbb{P}\left(\overline{\cap_{t=1}^{\infty}\mathcal{E}_t}\right)\\
=&1 - \mathbb{P}\left(\cup_{t=1}^{\infty}\overline{\mathcal{E}_t}\right)\\
\geq&1 - \sum_{t=1}^{\infty}\mathbb{P}\left(\overline{\mathcal{E}_t}\right)\\
=&1 - \sum_{t=1}^{\infty}\left(1 - \mathbb{P}\left(\mathcal{E}_t\right)\right)\\
=&1 - \sum_{t=1}^{\infty}\left(1 - \mathbb{P}\left(H_S \sum_{\tau\in\mathcal{Q}_t^g}(\hat{p}_\tau - p_\tau)^2 \leq \log \mathbb{P}_g((x_\tau, \mathbf{1}_\tau)_{\tau\in\mathcal{Q}_t^g}) - \log \mathbb{P}_{\hat{g}}((x_\tau, \mathbf{1}_\tau)_{\tau\in\mathcal{Q}_t^g}) + \sqrt{\frac{|\mathcal{Q}_t^g| B_p^2 \log\frac{1}{\delta_t}}{2}}\right)\right)\\
\geq&1 - \sum_{t=1}^{\infty}\delta_t\\
=&1 - \frac{6\delta}{\pi^2}\sum_{t=1}^{\infty}\frac{1}{t^2}\\
=&1 - \delta.
\end{aligned}
$$

Resetting the 'δ' to be ${}^{\delta}/_{\mathcal{N}(\mathcal{B}_g, \epsilon, \|\cdot\|_\infty)}$, we can guarantee the inequality (32) holds for all the functions in an ϵ-covering of $\mathcal{B}_g$.

For any $\hat{g}_{t+1} \in \mathcal{B}_g^{t+1}$, there exists $\hat{g}$ in the ϵ-covering of $\mathcal{B}_g$, such that $\|\hat{g}^{t+1} - \hat{g}\|_\infty \leq \epsilon$. We use the notations $\hat{p}_\tau^{t+1} = \hat{g}_{t+1}(x_\tau)$, and $\hat{p}_\tau = \hat{g}(x_\tau)$. Thus, we have,

$$
\begin{aligned}
&H_S \sum_{\tau\in\mathcal{Q}_t^g}\left(\hat{p}_\tau^{t+1} - p_\tau\right)^2\\
=&2H_S \sum_{\tau\in\mathcal{Q}_t^g}\left(\hat{p}_\tau^{t+1} - \hat{p}_\tau\right)^2 + 2H_S \sum_{\tau\in\mathcal{Q}_t^g}(\hat{p}_\tau - p_\tau)^2\\
=&2H_S\bar{S'}^2 \sum_{\tau\in\mathcal{Q}_t^g}\left(\hat{g}^{t+1}(x_\tau) - \hat{g}(x_\tau)\right)^2 + 2H_S \sum_{\tau\in\mathcal{Q}_t^g}(\hat{p}_\tau - p_\tau)^2\\
\leq&8H_S\bar{S'}^2 \sum_{\tau\in\mathcal{Q}_t^g}\epsilon^2 + 2H_S \sum_{\tau\in\mathcal{Q}_t^g}(\hat{p}_\tau - p_\tau)^2\\
\leq&8H_S\bar{S'}^2 \sum_{\tau\in\mathcal{Q}_t^g}\epsilon^2 + 2H_S \sum_{\tau\in\mathcal{Q}_t^g}(\hat{p}_\tau - p_\tau)^2\\
\leq&8H_S\bar{S'}^2\epsilon^2|\mathcal{Q}_t^g| + \sqrt{2|\mathcal{Q}_t^g| B_p^2 \log\frac{\pi^2 t^2 \mathcal{N}(\mathcal{B}_g, \epsilon, \|\cdot\|_\infty)}{6\delta}} + 2\left(\log \mathbb{P}_g((x_\tau, \mathbf{1}_\tau)_{\tau\in\mathcal{Q}_t^g}) - \log \mathbb{P}_{\hat{g}}((x_\tau, \mathbf{1}_\tau)_{\tau\in\mathcal{Q}_t^g})\right)\\
\leq&C(\epsilon, \delta, |\mathcal{Q}_t^g|, t) + 2\left(\log \mathbb{P}_{\hat{g}_{t+1}^{\mathrm{MLE}}}((x_\tau, \mathbf{1}_\tau)_{\tau\in\mathcal{Q}_t^g}) - \log \mathbb{P}_{\hat{g}_{t+1}}((x_\tau, \mathbf{1}_\tau)_{\tau\in\mathcal{Q}_t^g})\right)\\
&+ 2\left(\log \mathbb{P}_{\hat{g}_{t+1}}((x_\tau, \mathbf{1}_\tau)_{\tau\in\mathcal{Q}_t^g}) - \log \mathbb{P}_{\hat{g}}((x_\tau, \mathbf{1}_\tau)_{\tau\in\mathcal{Q}_t^g})\right)\\
\leq&C(\epsilon, \delta, |\mathcal{Q}_t^g|, t) + 4\epsilon t + 2\alpha_1(\epsilon, \delta, |\mathcal{Q}_t^g|, t)\\
=&\alpha_2(\epsilon, \delta, |\mathcal{Q}_t^g|, t) + 2\alpha_1(\epsilon, \delta, |\mathcal{Q}_t^g|, t),
\end{aligned}
$$

where $C(\epsilon, \delta, |\mathcal{Q}_t^g|, t) = 8H_S\bar{S'}^2\epsilon^2 t + \sqrt{2|\mathcal{Q}_t^g| B_p^2 \log\frac{\pi^2 t^2 \mathcal{N}(\mathcal{B}_g, \epsilon, \|\cdot\|_\infty)}{6\delta}}$ and $\alpha_2(\epsilon, \delta, |\mathcal{Q}_t^g|, t) = C(\epsilon, \delta, |\mathcal{Q}_t^g|, t) + 4\epsilon t$.

Furthermore,

$$\sum_{\tau=1}^{t} \left(\hat{p}_{\tau}^{t+1} - p_{\tau}\right)^2 \geq \sum_{\tau=1}^{t} \left(\underline{S}'\right)^2 \left(\hat{g}^{t+1}(x_{\tau}) - g(x_{\tau})\right)^2 .$$

The conclusion then follows.

□

B.2 Bound Point-Wise Error

Lemma B.2 (Point-wise Error Bound). *For any estimate $\tilde{g} \in \mathcal{B}_g^{t+1}$ measurable with respect to $\mathcal{F}_t$, we have, with probability at least $1-\delta$, $\forall t \geq 1, x \in \mathcal{X}$,*

$$\left|\tilde{g}(x) - g(x)\right| \leq \; 2\left(2B_g + r^{-1/2}\sqrt{\alpha(\epsilon, \delta/2, |\mathcal{Q}_t^g|, t)}\right)\sigma_{g_{t+1}}(x). \tag{32}$$

where $\sigma_{g_{t+1}}(x) = \sqrt{k_g(x,x) - k_g(X_{\mathcal{Q}_t^g}, x)^\top (K_{\mathcal{Q}_t^g} + rI)^{-1} k_g(X_{\mathcal{Q}_t^g}, x)}$.

Proof. We use $\phi(x)$ to denote the function $k_g(x, \cdot)$, where $\phi : \mathbb{R}^d \to \mathcal{H}_{k_g}$ maps a finite dimensional point $x \in \mathbb{R}^d$ to the RKHS $\mathcal{H}_{k_g}$. For notation simplicity, we set $k(\cdot,\cdot) = k_g(\cdot,\cdot)$ in this proof. For simplicity, we use $h_1^\top h_2$ to denote the inner product of two functions h_1, h_2 from the RKHS $\mathcal{H}_{k_g}$. Therefore, $h(x) = \langle h, k(x,\cdot)\rangle_{k_g} = h^\top \phi(x)$ and $k_g(x,x') = \langle k_g(x,\cdot), k_g(x',\cdot)\rangle = \phi(x)^\top \phi(x')$, $\forall x, x' \in \mathcal{X}$. We can introduce the feature map

$$\Phi_t := \left[\phi(x_\tau)^\top\right]_{\tau \in \mathcal{Q}_t^g}^\top ,$$

we then get the kernel matrix $K_t = \Phi_t \Phi_t^\top$, $k_t(x) = \Phi_t \phi(x)$ for all $x \in \mathcal{X}$ and $h_{\mathcal{Q}_t^g} = \Phi_t h$.

Note that when the Hilbert space $\mathcal{H}_{k_g}$ is a finite-dimensional Euclidean space, Φ_t is interpreted as the normal finite-dimensional matrix. In the more general setting where $\mathcal{H}_{k_g}$ can be an infinite-dimensional space, Φ_t is the evaluation operator $\mathcal{H}_{k_g} \to \mathbb{R}^{|\mathcal{Q}_t^g|}$ defined as $\Phi_t h = [h(x_\tau)]_{\tau \in \mathcal{Q}_t^g}^\top, \forall h \in \mathcal{H}$, with $\Phi_t^\top$ as its adjoint operator.

Since the matrices $(\Phi_t^\top \Phi_t + rI) : \mathcal{H}_{k_g} \to \mathcal{H}_{k_g}$ and $(\Phi_t \Phi_t^\top + rI) : \mathbb{R}^{|\mathcal{Q}_t^g|} \to \mathbb{R}^{|\mathcal{Q}_t^g|}$ are strictly positive definite and

$$(\Phi_t^\top \Phi_t + rI)\Phi_t^\top = \Phi_t^\top (\Phi_t \Phi_t^\top + rI),$$

we have

$$\Phi_t^\top (\Phi_t \Phi_t^\top + rI)^{-1} = (\Phi_t^\top \Phi_t + rI)^{-1} \Phi_t^\top . \tag{33}$$

Also from the definitions above $(\Phi_t^\top \Phi_t + rI)\phi(x) = \Phi_t^\top k_t(x) + r\phi(x)$, and thus from Eq. (33) we deduce that

$$\phi(x) = \Phi_t^\top (\Phi_t \Phi_t^T + rI)^{-1} k_t(x) + r(\Phi_t^\top \Phi_t + rI)^{-1}\phi(x), \tag{34}$$

which gives

$$\phi(x)^\top \phi(x) = k_t(x)^\top (\Phi_t \Phi_t^\top + rI)^{-1} k_t(x) + r\phi(x)^\top (\Phi_t^\top \Phi_t + rI)^{-1}\phi(x). \tag{35}$$

This implies

$$r\phi(x)^\top (\Phi_t^\top \Phi_t + rI)^{-1}\phi(x) = k(x,x) - k_t(x)^\top (K_t + rI)^{-1} k_t(x), \tag{36}$$

which is by definition the posterior variance $(\sigma_{g_{t+1}}(x))^2$. Now we can observe that

$$
\begin{aligned}
&|g(x) - k_t(x)^\top (K_t + rI)^{-1} g_{\mathcal{Q}_t^g}| \\
=&|\phi(x)^\top g - \phi(x)^\top \Phi_t^\top (\Phi_t \Phi_t^\top + rI)^{-1} \Phi_t g| \\
=&|\phi(x)^\top g - \phi(x)^\top (\Phi_t^\top \Phi_t + rI)^{-1} \Phi_t^\top \Phi_t g| \\
=&|\phi(x)^\top (\Phi_t^\top \Phi_t + rI)^{-1} (\Phi_t^\top \Phi_t + rI) g - \phi(x)^\top (\Phi_t^\top \Phi_t + rI)^{-1} \Phi_t^\top \Phi_t g| \\
=&|r\phi(x)^\top (\Phi_t^\top \Phi_t + rI)^{-1} g| \\
\leq& \left\| r(\Phi_t^\top \Phi_t + rI)^{-1} \phi(x) \right\|_{k_g} \|g\|_{k_g} \\
=& \|g\|_{k_g} \sqrt{r\phi(x)^\top (\Phi_t^\top \Phi_t + rI)^{-1} rI (\Phi_t^\top \Phi_t + rI)^{-1} \phi(x)} \\
\leq& B_g \sqrt{r\phi(x)^\top (\Phi_t^\top \Phi_t + rI)^{-1} (\Phi_t^\top \Phi_t + rI) (\Phi_t^\top \Phi_t + rI)^{-1} \phi(x)} \\
=& B_g \, \sigma_{g_{t+1}}(x),
\end{aligned}
$$

where the second equality uses Eq. (33), the first inequality is by Cauchy-Schwartz and the final equality is from Eq. (36). We define $\epsilon_{\mathcal{Q}_t^g} = \tilde{g}_{\mathcal{Q}_t^g} - g_{\mathcal{Q}_t^g}$, where $\tilde{g}_\tau = \tilde{g}(x_\tau)$. We have,

$$
\begin{aligned}
&|k_t(x)^\top (K_t + rI)^{-1} \epsilon_{\mathcal{Q}_t^g}| \\
=&|\phi(x)^\top \Phi_t^\top (\Phi_t \Phi_t^\top + rI)^{-1} \epsilon_{\mathcal{Q}_t^g}| \\
=&|\phi(x)^\top (\Phi_t^\top \Phi_t + rI)^{-1} \Phi_t^\top \epsilon_{\mathcal{Q}_t^g}| \\
\leq& \left\| (\Phi_t^\top \Phi_t + rI)^{-1/2} \phi(x) \right\|_{k_g} \left\| (\Phi_t^\top \Phi_t + rI)^{-1/2} \Phi_t^\top \epsilon_{\mathcal{Q}_t^g} \right\|_{k_g} \\
=& \sqrt{\phi(x)^\top (\Phi_t^\top \Phi_t + rI)^{-1} \phi(x)} \sqrt{(\Phi_t^\top \epsilon_{\mathcal{Q}_t^g})^\top (\Phi_t^\top \Phi_t + rI)^{-1} \Phi_t^\top \epsilon_{\mathcal{Q}_t^g}} \\
=& r^{-1/2} \sigma_{g_{t+1}}(x) \sqrt{\epsilon_{\mathcal{Q}_t^g}^\top \Phi_t \Phi_t^\top (\Phi_t \Phi_t^\top + rI)^{-1} \epsilon_{\mathcal{Q}_t^g}} \\
=& r^{-1/2} \sigma_{g_{t+1}}(x) \sqrt{\epsilon_{\mathcal{Q}_t^g}^\top K_t (K_t + rI)^{-1} \epsilon_{\mathcal{Q}_t^g}} \\
\leq& r^{-1/2} \sigma_{g_{t+1}}(x) \sqrt{\epsilon_{\mathcal{Q}_t^g}^\top \epsilon_{\mathcal{Q}_t^g}} \\
\leq& r^{-1/2} \alpha_t^{1/2} \sigma_{g_{t+1}}(x),
\end{aligned}
$$

where the second equality is from Eq. (33), the first inequality is by Cauchy-Schwartz and the last inequality follows by Eq. (22) and $\alpha_t = \alpha(\epsilon, \delta/2, |\mathcal{Q}_t^g|, t)$.

$$
\begin{aligned}
&|\tilde{g}(x) - g(x)| \\
\leq& \left| \left(k_t(x)^\top (K_t + rI)^{-1} (\tilde{g}_{\mathcal{Q}_t^g} - g_{\mathcal{Q}_t^g})\right) - \left(g(x) - k_t(x)^\top (K_t + rI)^{-1} g_{\mathcal{Q}_t^g}\right) + \left(\tilde{g}(x) - k_t(x)^\top (K_t + rI)^{-1} \tilde{g}_{\mathcal{Q}_t^g}\right) \right| \\
\leq& |k_t(x)^\top (K_t + rI)^{-1} (\tilde{g}_{\mathcal{Q}_t^g} - g_{\mathcal{Q}_t^g})| + |g(x) - k_t(x)^\top (K_t + rI)^{-1} g_{\mathcal{Q}_t^g}| + |\tilde{g}(x) - k_t(x)^\top (K_t + rI)^{-1} \tilde{g}_{\mathcal{Q}_t^g}| \\
\leq& \sigma_{g_{t+1}}(x) \Big(2B_g + r^{-1/2} \alpha_t^{1/2} \Big).
\end{aligned}
$$

□

B.3 Efficient Computations of Confidence Range for the Latent Expert function g

Leveraging the representer theorem [77, 107] thanks to the RKHS property, the MLE problem and confidence range computation problem can be reduced to an $\mathcal{O}(|\mathcal{Q}_t^g|)$-dimensional, tractable optimisation problem (37), problem (38) and problem (39).

$$
\begin{aligned}
\ell_t(\hat{g}_t^{\mathrm{MLE}}) = \min_{Z_{\mathcal{Q}_t^g} \in \mathbb{R}^{|\mathcal{Q}_t^g|}} \quad & \sum_{\tau \in \mathcal{Q}_t^g} Z_\tau \mathbf{1}_\tau - \sum_{\tau \in \mathcal{Q}_t^g} \log\left(1 + e^{Z_\tau}\right) \\
\text{subject to} \quad & Z_{\mathcal{Q}_t^g} K_{\mathcal{Q}_t^g}^{-1} Z_{\mathcal{Q}_t^g} \leq B_g^2,
\end{aligned}
\tag{37}
$$

where $K_{\mathcal{Q}_t^g} := (k_g(x_{\tau_1}, x_{\tau_2}))_{\tau_1, \tau_2 \in \mathcal{Q}_t^g}$.

$$\bar{g}_t(x) = \max_{Z_{\mathcal{Q}_t^g} \in \mathbb{R}^{|\mathcal{Q}_t^g|},\ z \in \mathbb{R},\ x \in \mathcal{X}} \quad z$$
$$\text{subject to} \quad \begin{bmatrix} Z_{\mathcal{Q}_t^g} \\ z \end{bmatrix}^\top K_{\mathcal{Q}_t^g, x}^{-1} \begin{bmatrix} Z_{\mathcal{Q}_t^g} \\ z \end{bmatrix} \leq B_g^2, \tag{38}$$
$$\ell(Z_{\mathcal{Q}_t^g} \mid \mathcal{D}_t^g) \geq \ell_t(\hat{g}_t^{\mathrm{MLE}}) - \alpha_1(\epsilon, \delta, |\mathcal{Q}_t^g|, t),$$

where $K_{\mathcal{Q}_t^g, x} := (k_g(\tilde{x}, \tilde{x}'))_{\tilde{x}, \tilde{x}' \in X_{\mathcal{Q}_t^g} \cup \{x\}}$, and $\ell(Z_{\mathcal{Q}_t^g} \mid \mathcal{D}_t^g) = \sum_{\tau \in \mathcal{Q}_t^g} Z_\tau \mathbf{1}_\tau - \sum_{\tau \in \mathcal{Q}_t^g} \log\left(1 + e^{Z_\tau}\right)$ is the LL value when the function value at x_τ is Z_τ.

$$\underline{g}_t(x) = \min_{Z_{\mathcal{Q}_t^g} \in \mathbb{R}^{|\mathcal{Q}_t^g|},\ z \in \mathbb{R},\ x \in \mathcal{X}} \quad z$$
$$\text{subject to} \quad \begin{bmatrix} Z_{\mathcal{Q}_t^g} \\ z \end{bmatrix}^\top K_{\mathcal{Q}_t^g, x}^{-1} \begin{bmatrix} Z_{\mathcal{Q}_t^g} \\ z \end{bmatrix} \leq B_g^2, \tag{39}$$
$$\ell(Z_{\mathcal{Q}_t^g} \mid \mathcal{D}_t^g) \geq \ell_t(\hat{g}_t^{\mathrm{MLE}}) - \alpha_1(\epsilon, \delta, |\mathcal{Q}_t^g|, t),$$

where $K_{\mathcal{Q}_t^g, x} := (k_g(\tilde{x}, \tilde{x}'))_{\tilde{x}, \tilde{x}' \in X_{\mathcal{Q}_t^g} \cup \{x\}}$, and $\ell(Z_{\mathcal{Q}_t^g} \mid \mathcal{D}_t^g) = \sum_{\tau \in \mathcal{Q}_t^g} Z_\tau \mathbf{1}_\tau - \sum_{\tau \in \mathcal{Q}_t^g} \log\left(1 + e^{Z_\tau}\right)$ is the LL value when the function value at x_τ is Z_τ.

B.4 Bound Cumulative Standard Deviation over Sample Trajectory

Lemma B.3 (Lemma 4, [22][8])**.**

$$\sum_{t \in \mathcal{Q}_T^f} \sigma_{f_t}(x_t) \leq \sqrt{4(|\mathcal{Q}_T^f| + 2)\gamma_{|\mathcal{Q}_T^f|}^f} = \mathcal{O}\left(\sqrt{|\mathcal{Q}_T^f| \gamma_{|\mathcal{Q}_T^f|}^f}\right). \tag{40}$$

Similarly, we have,

$$\sum_{t \in \mathcal{Q}_T^g} \sigma_{g_t}(x_t) \leq \sqrt{4(|\mathcal{Q}_T^g| + 2)\gamma_{|\mathcal{Q}_T^g|}^g} = \mathcal{O}\left(\sqrt{|\mathcal{Q}_T^g| \gamma_{|\mathcal{Q}_T^g|}^g}\right). \tag{41}$$

B.5 Bound Cumulative Regret

We can then analyze the regret of our algorithm. We use $\mathcal{C}_T$ to denote the set $\{t \in [T] | x_t = x_t^c\}$.

$$\begin{aligned} R_{\mathcal{Q}_T^f} &= \sum_{t \in \mathcal{Q}_T^f} [f(x_t) - f(x^\star)] \\ &= \sum_{t \in \mathcal{Q}_T^f \cap \mathcal{C}_T} [f(x_t) - f(x^\star)] + \sum_{t \in \mathcal{Q}_T^f \setminus \mathcal{C}_T} [f(x_t) - f(x^\star)] \\ &= \sum_{t \in \mathcal{Q}_T^f \cap \mathcal{C}_T} [f(x_t^c) - f(x^\star)] + \sum_{t \in \mathcal{Q}_T^f \setminus \mathcal{C}_T} [f(x_t^u) - f(x^\star)] \end{aligned}$$

[8] Appears in the arXiv version: `https://arxiv.org/pdf/1704.00445`.

For the first part, we have,

$$
\begin{aligned}
&\sum_{t\in\mathcal{Q}_T^f\cap\mathcal{C}_T}[f(x_t^c)-f(x^\star)]\\
=&\sum_{t\in\mathcal{Q}_T^f\cap\mathcal{C}_T}[f(x_t^c)-\underline{f}_t(x_t^c)+\underline{f}_t(x_t^c)-f(x^\star)]\\
\leq&\sum_{t\in\mathcal{Q}_T^f\cap\mathcal{C}_T}[f(x_t^c)-\underline{f}_t(x_t^c)+\underline{f}_t(x_t^c)-\underline{f}_t(x^\star)]\\
=&\sum_{t\in\mathcal{Q}_T^f\cap\mathcal{C}_T}[f(x_t^c)-\underline{f}_t(x_t^c)+\underline{f}_t(x_t^c)-\underline{f}_t(x_t^u)+\underline{f}_t(x_t^u)-\underline{f}_t(x^\star)]\\
\leq&\sum_{t\in\mathcal{Q}_T^f\cap\mathcal{C}_T}2\beta_{f_t}\sigma_{f_t}(x_t)+\sum_{t\in\mathcal{Q}_T^f\cap\mathcal{C}_T}[\underline{f}_t(x_t^c)-\underline{f}_t(x_t^u)]\\
\leq&2\beta_{f_T}\sum_{t\in\mathcal{Q}_T^f\cap\mathcal{C}_T}\sigma_{f_t}(x_t)+\sum_{t\in\mathcal{Q}_T^f\cap\mathcal{C}_T}[\underline{f}_t(x_t^c)-\underline{f}_t(x_t^u)],
\end{aligned}
$$

where σ_{f_t} is as given in Eq. (2b), the first inequality follows by Lem. 3.1, the second inequality follows by Lem. 3.1 and the line 5 of Alg. 1.

Furthermore, we have,

$$
\begin{aligned}
&\sum_{t\in\mathcal{Q}_T^f\cap\mathcal{C}_T}[\underline{f}_t(x_t^c)-\underline{f}_t(x_t^u)] &(42)\\
\leq&\sum_{t\in\mathcal{Q}_T^f\cap\mathcal{C}_T}[\bar{f}_t(x_t^u)-\underline{f}_t(x_t^u)] &(43)\\
\leq&\sum_{t\in\mathcal{Q}_T^f\cap\mathcal{C}_T}2\beta_{f_t}\sigma_{f_t}(x_t^u) &(44)\\
\leq&\sum_{t\in\mathcal{Q}_T^f\cap\mathcal{C}_T}2\beta_{f_t}\eta\sigma_{f_t}(x_t^c) &(45)\\
=&\sum_{t\in\mathcal{Q}_T^f\cap\mathcal{C}_T}2\beta_{f_t}\eta\sigma_{f_t}(x_t) &(46)
\end{aligned}
$$

where the first inequality follows by the condition in line 6 of the Alg. 1, the second inequality follows by the Lem. 3.1, and the third inequality follows by the condition in line 6 of the Alg. 1.

For the second part, we have,

$$
\begin{aligned}
&\sum_{t\in\mathcal{Q}_T^f\setminus\mathcal{C}_T}[f(x_t^u)-f(x^\star)] &(47)\\
=&\sum_{t\in\mathcal{Q}_T^f\setminus\mathcal{C}_T}[f(x_t^u)-\underline{f}_t(x_t^u)+\underline{f}_t(x_t^u)-f(x^\star)] &(48)\\
\leq&\sum_{t\in\mathcal{Q}_T^f\setminus\mathcal{C}_T}[f(x_t^u)-\underline{f}_t(x_t^u)+\underline{f}_t(x_t^u)-\underline{f}_t(x^\star)] &(49)\\
\leq&\sum_{t\in\mathcal{Q}_T^f\setminus\mathcal{C}_T}2\beta_{f_t}\sigma_{f_t}(x_t) &(50)\\
\leq&2\beta_{f_T}\sum_{t\in\mathcal{Q}_T^f\setminus\mathcal{C}_T}\sigma_{f_t}(x_t), &(51)
\end{aligned}
$$

where the first inequality follows by that $f(x^\star)\geq\underline{f}_t(x^\star)$, the second inequality follows by the optimality of x_t^u for the problem in line 5 and the Lem. 3.1, and the third inequality follows by the monotonicity of β_{f_t} in t.

Hence,

$$\begin{aligned} R_{\mathcal{Q}_T^f} \leq & 2(2+\eta)\beta_{f_T} \sum_{t \in \mathcal{Q}_T^f} \sigma_{f_t}(x_t) \\ \leq & 2(2+\eta)\beta_{f_T} \sqrt{4(|\mathcal{Q}_T^f|+2)\gamma^f_{|\mathcal{Q}_T^f|}} \\ = & \mathcal{O}\left(\gamma^f_{|\mathcal{Q}_T^f|}\sqrt{|\mathcal{Q}_T^f|}\right). \end{aligned}$$

B.6 Bound Cumulative Queries to Labeler

We can then analyze the cumulative queries to the expert. We notice that, $\forall t \in \mathcal{Q}_T^g$,

$$\bar{g}_t(x_t) - \underline{g}_t(x_t) \geq g_{\text{thr}} \tag{52}$$

Meanwhile, by Lem. B.2,

$$\bar{g}_t(x_t) - \underline{g}_t(x_t) \leq 4\left(2B_g + r^{-1/2}\sqrt{\alpha_t}\right)\sigma_{g_t}(x). \tag{53}$$

Hence,

$$g_{\text{thr}} \leq 4\left(2B_g + r^{-1/2}\sqrt{\alpha_t}\right)\sigma_{g_t}(x). \tag{54}$$

Therefore,

$$\begin{aligned} Q_T^g = & |\mathcal{Q}_T^g| && (55)\\ = & \sum_{t \in \mathcal{Q}_T^g} 1 && (56)\\ \leq & \frac{1}{g_{\text{thr}}} \sum_{t \in \mathcal{Q}_T^g} g_{\text{thr}} && (57)\\ \leq & \frac{1}{g_{\text{thr}}} \sum_{t \in \mathcal{Q}_T^g} 4\left(2B_g + r^{-1/2}\sqrt{\alpha_t}\right)\sigma_{g_t}(x_t) && (58)\\ \leq & \frac{4}{g_{\text{thr}}}\left(2B_g + r^{-1/2}\sqrt{\alpha_T}\right) \sum_{t \in \mathcal{Q}_T^g} \sigma_{g_t}(x_t) && (59)\\ = & \mathcal{O}\left(\sqrt{\alpha_T \gamma^g_{|\mathcal{Q}_T^g|}|\mathcal{Q}_T^g|}\right). && (60) \end{aligned}$$

Dividing by $\sqrt{|\mathcal{Q}_T^g|}$, we obtain,

$$\sqrt{|\mathcal{Q}_T^g|} = \mathcal{O}(\sqrt{\alpha_T \gamma^g_{|\mathcal{Q}_T^g|}}). \tag{61}$$

Hence,

$$Q_T^g = |\mathcal{Q}_T^g| = \mathcal{O}(\alpha_T \gamma^g_{|\mathcal{Q}_T^g|}). \tag{62}$$

By setting $\epsilon = \frac{1}{T}$, we have

$$\alpha_T = \mathcal{O}\left(\sqrt{|\mathcal{Q}_T^g| \log \frac{T\mathcal{N}(\mathcal{B}_g, 1/T, \|\cdot\|_\infty)}{\delta}}\right). \tag{63}$$

Hence, dividing by $\sqrt{|\mathcal{Q}_T^g|}$ on Eq. (62) again, we obtain,

$$Q_T^g = |\mathcal{Q}_T^g| = \mathcal{O}\left(\left(\gamma^g_{|\mathcal{Q}_T^g|}\right)^2 \log \frac{T\mathcal{N}(\mathcal{B}_g, 1/T, \|\cdot\|_\infty)}{\delta}\right) \leq \mathcal{O}\left(\left(\gamma_T^g\right)^2 \log \frac{T\mathcal{N}(\mathcal{B}_g, 1/T, \|\cdot\|_\infty)}{\delta}\right). \tag{64}$$

C Detailed Discussions on The Significance of Thm 4.1

Order-wise improvement can not be attained under current mild assumption. g may contain no information (e.g., $g = 0$) or even adversarial. Even if human expertise is helpful, we can not guarantee an *order-wise* improvement either. For example, consider the following g,

$$g(x) = \begin{cases} f(x^\star) + c, & \text{if } f(x) - f(x^\star) \leq c, \\ f(x) & \text{otherwise,} \end{cases}$$

where $c > 0$ is a positive constant. In practice, such a scenario means the human expert has some rough idea in a near-optimal region but not exactly sure where the exact optimum is. This is common in practice. In this case, human expert is helpful in identifying the region with $f(x) \leq f(x^\star) + c$ but no longer helpful for further optimization inside the region $\{x \in \mathcal{X} | f(x) \leq f(x^\star) + c\}$. However, convergence rate is defined in the asymptotic sense. Hence, an order-wise improvement can not be guaranteed.

Assumption becomes unrealistic if we really want it. Some papers that show theoretical superiority [2, 6], yet the assumptions are unrealistic. For example, [6] assumed that the human knows the true kernel hyperparameters while GP is misspecified, and [2] assumed the human belief function g has better and tighter confidence intervals over the entire domain. We can derive the better convergence rate of our algorithm than AI-only ones if we use [2] assumption, but this is unlikely to be true in reality. In fact, our method outperforms these method empirically (see Figure 5). This supports the superiority based on unrealistic conditions is not meaningful in practice.

Empirical success can be achieved without order-wise improvement on worst-case convergence. Our assumption is more natural; following [37], we posit humans have better prior knowledge than GP and are only useful at the beginning as a warm starter. This assumption is widely accepted by the community and practitioners, which leads to real-world impact (e.g. Nature [42]). The warm-starting-based papers [36, 37, 44] have been published in reputable venues without such a theory. In our manuscript, real-world applications also empirically demonstrate that our method not only improves the convergence of BO, but also maintains robustness despite varying labelling accuracy.

Worst-case convergence and hand-over guarantees matter. We believe that the value of theory is the worst-case guarantee. To be clear, starting point of human-AI collaborative BO is that *the experts are not currently using BO*. The scientific experts do very expensive tasks, which often cost millions of dollars and weeks to months to test one design (e.g. battery design). They are reluctant to employ BO due to its opaque and untrustworthy nature. The experts want to be involved in the AI decision-making process, otherwise they are forced to work as a robot feeding experimental results to the AI. But, they are also in the middle of trial and error, so their advice is not always reliable. Our worst-case guarantee assures that at least their involvement does not harm the AI-only results, and also assures the automation in the later round. Thus, we believe our approach can extend the applicable range of BO to high-stakes optimisation tasks. Furthermore, our handover guarantee assures that only limited human labeling effort is needed, which is also meaningful because the motivation to use BO is to alleviate the tedious human effort in the first place.

D Proof of the Kernel-Specific Bounds in Tab. 1

For the cumulative regret part, we have,

- If the kernel function is linear, $\gamma^f_{|\mathcal{Q}^f_T|} = \mathcal{O}(\log|\mathcal{Q}^f_T|)$, and thus $R_{|\mathcal{Q}^f_T|} = \mathcal{O}\left(\sqrt{|\mathcal{Q}^f_T|}\log|\mathcal{Q}^f_T|\right)$.
- If the kernel function is squared exponential, $\gamma^f_{|\mathcal{Q}^f_T|} = \mathcal{O}((\log|\mathcal{Q}^f_T|)^{d+1})$, $R_{\mathcal{Q}^f_T} = \mathcal{O}(\sqrt{|\mathcal{Q}^f_T|}(\log|\mathcal{Q}^f_T|)^{d+1})$.
- If the kernel function is Mátern, $\gamma^f_{|\mathcal{Q}^f_T|} = \mathcal{O}\left(|\mathcal{Q}^f_T|^{\frac{d}{2\nu+d}}\log^{\frac{2\nu}{2\nu+d}}(|\mathcal{Q}^f_T|)\right)((\nu > \frac{d}{2}))$, $R_{\mathcal{Q}^f_T} = \mathcal{O}\left(|\mathcal{Q}^f_T|^{\frac{2\nu+3d}{4\nu+2d}}\log^{\frac{2\nu}{2\nu+d}}(|\mathcal{Q}^f_T|)\right)$.

To bound the cumulative queries, we have,

1. k_g is a linear kernel, then $\log \mathcal{N}(\mathcal{B}_g, T^{-1}, \|\cdot\|_\infty) = \mathcal{O}\left(\log \frac{1}{\epsilon}\right) = \mathcal{O}\left(\log T\right)$. By Thm. 5 in [84],
$$\gamma_T^g = \mathcal{O}(\log T).$$
Hence,
$$Q_T^g = \mathcal{O}\left((\log T)^2 \log T\right) = \mathcal{O}\left((\log T)^3\right).$$
2. k_g is a squared exponential kernel, then $\log \mathcal{N}(\mathcal{B}_g, T^{-1}, \|\cdot\|_\infty) = \mathcal{O}\left((\log \frac{1}{\epsilon})^{d+1}\right) = \mathcal{O}\left((\log T)^{d+1}\right)$ (Example 4, [109]). By Thm. 4 in [49], we have,
$$\gamma_T^g = \mathcal{O}((\log T)^{d+1}).$$
Hence,
$$Q_T^g = \mathcal{O}\left((\log T)^{2(d+1)} (\log T)^{d+1}\right) = \mathcal{O}\left((\log T)^{3(d+1)}\right).$$
3. k_g is a Matern kernel, then $\log \mathcal{N}(\mathcal{B}_g, T^{-1}, \|\cdot\|_\infty) = \mathcal{O}\left((\frac{1}{\epsilon})^{d/\nu} \log \frac{1}{\epsilon}\right) = \mathcal{O}\left(T^{d/\nu} \log T\right)$ (by Thm. 5.1 and Thm. 5.3 in [105]). By Thm. 4 in [49], we have,
$$\gamma_T^g = \mathcal{O}\left(T^{\frac{d(d+1)}{2\nu + d(d+1)}} \log T\right).$$
Hence,
$$Q_T^g = \mathcal{O}\left(T^{\frac{2d(d+1)}{2\nu + d(d+1)}} (\log T)^2 T^{\frac{d}{\nu}} \log T\right) = \mathcal{O}(T^{\frac{2d(d+1)}{2\nu + d(d+1)}} T^{\frac{d}{\nu}} (\log T)^3),$$
where $\nu > \frac{d\left(d+3+\sqrt{d^2+14d+17}\right)}{4}$.

E Theoretical improvement of convergence rate

Algorithm 2 COllaborative **B**ayesian **O**ptimization with **H**elpful **L**abelling Experts (**COBOHL**).

Input and Initialization: function space ball $\mathcal{B}_g$, and uncertainty threshold g_{thr}.
Set $\mathcal{B}_g^1 = \mathcal{B}_g$, $\mathcal{Q}_0^f = \emptyset$, and $\mathcal{Q}_0^g = \emptyset$.
for $t \in [T]$ **do**
  Generate x_t by solving the constrained auxiliary optimization problem $\min_{x \in \mathcal{X}} \underline{f}_t(x)$ subject to $\underline{g}_t(x) \leq 0$. ▷ Expert-constrained LCB
  if $\bar{g}_t(x_t) - \underline{g}_t(x_t) > g_{\text{thr}}$ **then** ▷ Handover guarantee
    Query the expert's label to get the feedback $\mathbf{1}_t$.
    Update $\mathcal{Q}_t^g = \mathcal{Q}_{t-1}^g \cup \{t\}$ and the posterior confidence set $\mathcal{B}_g^{t+1}$. Set $\mathcal{Q}_t^f = \mathcal{Q}_{t-1}^f$.
  else
    Evaluate the black-box function at the point x_t, and set $\mathcal{Q}_t^f = \mathcal{Q}_{t-1}^f \cup \{t\}$. Set $\mathcal{Q}_t^g = \mathcal{Q}_{t-1}^g$.
    Update the posterior mean/variance of the objective f.

Here, we give the analysis on the regret of COBOHL,

$$\sum_{t \in \mathcal{Q}_T^f} (f(x_t) - f(x^\star)) = \sum_{t \in \mathcal{Q}_T^f} (f(x_t) - \underline{f}_t(x_t) + \underline{f}_t(x_t) - \underline{f}_t(x^\star) + \underline{f}_t(x^\star) - f(x^\star)) \tag{65}$$
$$\leq \sum_{t \in \mathcal{Q}_T^f} (f(x_t) - \underline{f}_t(x_t)) \tag{66}$$
$$\leq \sum_{t \in \mathcal{Q}_T^f} 2\beta_{f_t} \sigma_{f_t}(x_t) \tag{67}$$
$$\leq 2\beta_{f_T} \sum_{t \in \mathcal{Q}_T^f} \sigma_{f_t}(x_t) \tag{68}$$
$$= \mathcal{O}\left(\gamma^{f, \mathcal{X}^g}_{|\mathcal{Q}_T^f|} \sqrt{|\mathcal{Q}_T^f|}\right), \tag{69}$$

Table 2: Comparisons between our algorithm with the existing baseline methods.

baselines	blackbox human model?	no-rankability assumption?	continuous guarantee?	no-harm guarantee?	data-driven trust?	handpver guarantee?
AV et al. (2022) [11]	✓	✗	✗	✗	✗	✗
Hvarfner et al. (2022) [43]	✗	✗	✓	✓	✗	✗
Gupta et al. (2023) [39]	✓	✗	✓	✗	✗	✗
Khoshvishkaie et al. (2023) [50]	✓	✗	✗	✗	✗	✗
Cisse et al. (2023) [24]	✗	✗	✗	✗	✗	✗
Adachi et al. (2023) [7]	✓	✗	✗	✓	✗	✗
Rodemann et al. (2024) [70]	✓	✗	✗	✗	✗	✗
AV et al. (2024) [12]	✓	✗	✗	✗	✗	✗
Hvarfner et al. (2024) [42]	✗	✗	✗	✗	✗	✗
Ours	✓	✓	✓	✓	✓	✓

where the first inequality follows by the feasibility of x_t in the expert-constrained LCB problem and $\underline{f}_t(x^\star) \leq f(x^\star)$, the maximum information gain is defined over the set $\mathcal{X}^g := \{x \in \mathcal{X} | g(x) \leq g_{\text{thr}}\}$. Meanwhile, the regret bound of vanilla LCB has a similar form of $\mathcal{O}\left(\gamma^{f,\mathcal{X}}_{|\mathcal{Q}^f_T|}\sqrt{|\mathcal{Q}^f_T|}\right)$. Notably, the regret bound for vanilla LCB has a maximum information gain defined over the region $\mathcal{X}$. For commonly used kernel functions, the maximum information gain is proportional to the volume of the set. Since $\mathcal{X}^g \subset \mathcal{X}$, $\text{vol}(\mathcal{X}^g) \leq \text{vol}(\mathcal{X})$ and the maximum information gain gets reduced by a ratio of $\frac{\text{vol}(\mathcal{X}^g)}{\text{vol}(\mathcal{X})}$. Therefore, the regret bound gets improved by a ratio of $\frac{\text{vol}(\mathcal{X}^g)}{\text{vol}(\mathcal{X})}$.

F Estimating norm bound online

By Assumption 2.4, there exists a large enough constant B_g that upper bounds the norm of the ground-truth latent black-box function g. However, a tight estimate of this upper bound may be unknown to us in practice, while the execution of our algorithm explicitly relies on knowing a bound B_g (in Prob. (6), B_g is a key parameter).

So it is necessary to estimate the norm bound B_g using the online data. Suppose our guess is $\hat{B}$. It is possible that $\hat{B}$ is even smaller than the ground-truth function norm $\|g\|$. To detect this underestimate, we observe that, with the correct setting of B_g such that $B_g \geq \|g\|$, we have that by Lemma 3.2 and the definition of maximum likelihood estimate,

$$\ell_t(\hat{g}^{\text{MLE}}_{t|\hat{B}}) \geq \ell_t(g) \geq \ell_t(\hat{g}^{\text{MLE}}_{t|B}) - \alpha_1(\epsilon, \delta, |\mathcal{Q}^g_t|, t|\hat{B}),$$

where $\hat{g}^{\text{MLE}}_{t|\hat{B}}$ is the maximum likelihood estimate function with function norm bound $\hat{B}$ and $\alpha_1(\epsilon, \delta, |\mathcal{Q}^g_t|, t|\hat{B})$ is the corresponding parameter as defined in Lemma 3.2 with norm bound $\hat{B}$. We also have $2\hat{B}$ is a valid upper bound on $\|g\|$ and thus,

$$\ell_t(\hat{g}^{\text{MLE}}_{t|2\hat{B}}) \geq \ell_t(g) \geq \ell_t(\hat{g}^{\text{MLE}}_{t|2\hat{B}}) - \alpha_1(\epsilon, \delta, |\mathcal{Q}^g_t|, t|2\hat{B}).$$

Therefore,

$$\ell_t(\hat{g}^{\text{MLE}}_{t|\hat{B}}) \geq \ell_t(g) \geq \ell_t(\hat{g}^{\text{MLE}}_{t|2\hat{B}}) - \alpha_1(\epsilon, \delta, |\mathcal{Q}^g_t|, t|2\hat{B}).$$

That is to say, $\ell_t(\hat{g}^{\text{MLE}}_{t|\hat{B}})$ needs to be greater than or equal to $\ell_t(\hat{g}^{\text{MLE}}_{t|2\hat{B}}) - \alpha_1(\epsilon, \delta, |\mathcal{Q}^g_t|, t|2\hat{B})$ when $\hat{B}$ is a valid upper bound on $\|g\|$.

Therefore, we can use the heuristic: every time we find that

$$\ell_t(\hat{g}^{\text{MLE}}_{t|\hat{B}}) < \ell_t(\hat{g}^{\text{MLE}}_{t|2\hat{B}}) - \alpha_1(\epsilon, \delta, |\mathcal{Q}^g_t|, t|2\hat{B}),$$

we double the upper bound guess $\hat{B}$.

G Related Work

We summarized the baseline comparison in terms of five factors in Table 2. Our algorithm is the first to offer a data-driven trust level no-harm guarantee and a handover guarantee under no rankability assumption.

We briefly introduce the baseline methods used in the real-world experiments::

1. AV. et al., NeurIPS 2022 [11]: This algorithm initially proposed the human-AI collaborative setting. The approach is straightforward: human experts can intervene in the optimization process if they find the next query location suggested by the vanilla LCB BO to be unpromising. This method can be described as a 'human as constraint' approach, where the BO must adhere to the experts' recommendations regardless of the quality of their advice. This approach assumes that human experts are at least better than the vanilla LCB, thus requiring a high level of trust in the experts. As shown in Figure 5, experts' input is not always reliable.
2. Khoshvishkaie et al., ECML 2023 [50]: This setting assumes that the querying budget is equally divided between human experts and the vanilla LCB BO. This means that once a point is selected by human experts, the BO will alternately select the next query. This method can select the vanilla LCB regardless of what the human expert selected, making it likely to achieve a no-harm guarantee, although no theoretical proof is provided. The trust level in experts in this method is low, as all expert inputs are treated equally regardless of their quality. Therefore, while this method performs well in unreliable settings, it is not as effective when experts are good advisors. To be fair, their work focuses more on imperfect cases and does not consider scenarios with effective experts.
3. Adachi et al., AISTATS 2024 [7]: This setting assumes that the BO provides two possible candidates, from which the human selects one. Both candidates have convergence guarantees, thus ensuring a no-harm guarantee, although their proof is limited to discrete settings. However, the human must ultimately choose one of the candidates, maintaining a high level of trust in human experts. They introduced a discounting function that hand-tunes the decaying rate of trust, gradually generating the same candidates. Although their work initiated the no-harm guarantee concept, the trust level adjustment is not data-driven and the proof is limited to discrete cases. To be fair, their main focus is on the explainability of black-box optimizers, which we did not consider in this work. Their method can be integrated into the GP surrogate model as a plug-and-play feature, making it easy to extend our work.

We did not compare against the following papers due to difficulty in aligning assumptions and similarity.

1. [43, 42, 24]: These works assume that humans can explicitly express their beliefs as a probability distribution, such as a Gaussian distribution centered at the most promising location. This assumption is too strong and incompatible with our black-box assumption of human belief.
2. [39, 12]: These methods are nearly identical to [50]. Therefore, we selected [50] as a representative work for this pessimistic approach.
3. [70]: This method is almost identical to [11]. Thus, we selected [11] as a representative work for this pessimistic approach.

H Comparison and Generalization to Other Feedback Forms.

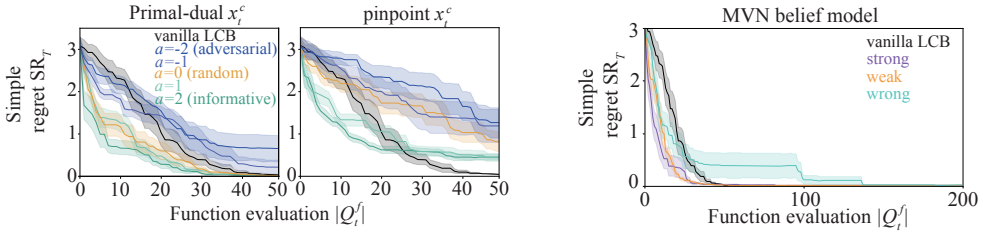
H.1 Other feedback forms

(a) **Pinpoint form:** [11, 39, 50] adopt this form that the algorithm asks the humans to directly pinpoint the next query location.

(b) **Pairwise comparison:** [7] adopts this form that the algorithm presents paired candidates, and the human selects the preferred one.

(c) **Ranking:** [12] adopts this form that the algorithm proposes a list of candidates, and the human provides a preferential ranking.

(d) **Belief function:** [43, 42] adopt a Gaussian distribution as expert input. Unlike the others, this form assumes an offline setting where the input is defined at the beginning and remains unchanged during the optimization. Human experts must specify the mean and variance of the Gaussian, which represent their belief in the location of the global optimum and their confidence in this estimation, respectively.



(a) (a) Primal-dual vs pinpointing

(b) (b) MVN human belief model

Figure 6: Different forms of human feedback

H.2 Adaptation

Slight modification can adapt these forms to our method.

(a) **Pinpoint form:** We can simply replace the expert-augmented LCB in line 4 of Algorithm 1 with the pinpointed candidate.

(b) **Pairwise comparison:** By adopting the Bradley-Terry-Luce (BTL) model [17], we can extend our likelihood ratio model to incorporate preferential feedback. This allows us to obtain the surrogate , while the other parts of our algorithm remain unchanged.

(c) **Ranking:** Ranking feedback can be decomposed into multiple pairwise comparisons. Therefore, we can apply the same method as in the pairwise comparison.

(d) **Belief function:** We can use this Gaussian distribution model as the surrogate.

H.3 Comparison

We demonstrate the adaptation of (a) pinpoint and (d) belief function forms in Fig. 6. The pinpoint strategy employs a sample from the expert belief function as x_c on line 4 in Algorithm 1, while keeping the remaining lines the same as the original. It performs worse than the original primal-dual approaches, particularly in later iterations. This is because expert sampling does not incorporate GP information. Generally, humans excel at exploration in the beginning, while GP excels at finding precise locations in the later stages. This finding is supported by other literature, such as [48], involving human expert studies.

In Fig. 6(b), we employed the multivariate normal distribution (MVN) belief model proposed by [43]. This model represents the human belief function as $\tilde{p} = \mathcal{N}(x; \mu, \Sigma)$, where μ is the mean vector representing the estimated location of the global optimum x^*, and Σ is the covariance matrix, representing the confidence of the estimation. We use $\Sigma = \mathbf{I}$, the identity matrix $\mathbf{I}$, as suggested by [43]. We transform: $[0, |2\pi\Sigma|^{-1/2}] \to [0, 1]$, and we use this normalised belief function as the acceptance probability of a Bernoulli distribution $1 - p$ at given location x (note that $p = 0$ is acceptance). Following [43], we set three levels of beliefs: strong, weak, and wrong. These levels are established by adjusting the mean vector to be offset from x^*. 'Strong' aligns with x^*, 'wrong' is the furthest possible location from x^*, and 'weak' is an intermediate location. Our algorithm robustly converges for any level of trust.

As such, the primary reason we adopted binary labelling is due to its empirical success, as demonstrated in Fig. 5 and Fig. 6. None of the other formats, including (a) pinpoint form [11, 50] and (b) pairwise comparison [7], outperforms our method. In the experiments by [7], the authors showed that (a) pairwise comparison outperforms both (d) belief form [43]. Therefore, it logically follows that our binary labeling format yields the best performance.

The main reasons why the binary format works better are as follows:

(a) **Pinpoint form:** The accuracy of pinpointing is generally lower than that of kernel-based models. Humans excel at qualitative comparison rather than estimating absolute quantities [47]. Numerous studies [11, 48, 50, 70] have confirmed that manual search (pinpointing) by human experts only outperforms in the initial stages, with standard BO with GP performing

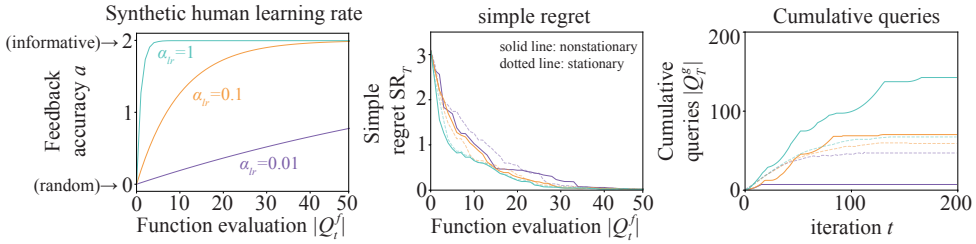


Figure 7: Non-stationary human accuracy.

better in later rounds. [39] shows that this type of feedback only outperforms when the expert's manual sampling is consistently superior to the standard BO. However, such cases are rare in our examples (e.g., Rosenbrock), and [11, 70] corroborate this conclusion.

(b) **Pairwise comparison:** This format relies on two critical assumptions: transitivity and completeness. Transitivity assumes no inconsistencies, which are often referred to as a "rock-paper-scissors" relationship. However, real-world human preferences frequently exhibit this issue [20]. Completeness assumes that humans can always rank their preferences at any given points. In practice, when a user is unsure which option is better, this assumption does not hold. Our imprecise probability approach avoids these issues by not relying on an absolute ranking structure [10, 41].

(c) **Ranking:** Ranking is an extension of pairwise comparison and has been classically researched as the Borda count, which is known not to satisfy all rational axioms. Theoretically, the Condorcet winner in pairwise comparison is the only method that is known to identify the global maximum of ordinal utility.

(d) **Belief function:** This is another form of absolute quantity, which humans are generally not proficient at estimating. Additionally, the offline nature of this method does not allow for knowledge updates.

I Potential Extensions for Future Work

I.1 Extension to Time-varying Human Feedback Model

In practice, human's belief in the black-box function may be influenced by the online evaluation results of the ground-truth black-box function. To further incorporate such online influence, we need to model the change of human feedback model.

Simple extension, yet not promising performance gain. The most naïve approach for non-stationary model is windowing, i.e., forgetting the previous queried dataset. This can be very easy to apply to our setting, as it simply removes the old data outside the predefined iteration window.

Fig. 7 shows the scenario where the accuracy of human experts' labelling improves over time, represented by $a = 2(1 - \exp(-\alpha_{lr}/|Q_t^f|))$, where α_{lr} controls the learning rate. The non-stationary model employs windowing, retaining only the most recent w-th data points, with $w = 5$. The stationary model does not use windowing, thereby retaining all labelled datasets. The plots represent the average of 10 runs without standard error for improved visibility. While simple regret showed slight improvement initially, the performance gain varied depending on α_{lr}. In contrast, the cumulative number of queries $|Q_t^g|$ significantly increased due to the increased uncertainty introduced by windowing.

More sophisticated extension. Another more sophisticated approach is modelling the dynamics of behavioural change. A potential idea is modelling the human behaviour change as an implicit online learning process of the latent function g. That is, $g_{t+1} = F(g_t, x_t, y_t)$, where g_t is the human latent function at step t. The forward dynamics F captures the update of human latent function g when observing the new data point. One potential F is gradient ascent of log-likelihood as shown in $g_{t+1} = g_t + \lambda \nabla_g \log p_g(x_t, y_t)$, where $p_g(x_t, y_t)$ is the probability of observing y_t at the input x_t given the black-box objective function is g. We can then combine this dynamic with our likelihood

Table 3: The complete list of hyperparameters and their settings.

hyperparameters	initial value	data-driven optimisation?	tuning method
f kernel hyperparamters	BoTorch default	✓	maximising the marginal likelihood
g kernel hyperparamters	BoTorch default	✓	copying f kernel values
r in Eq.2b	1e-4	fixed	–
$\gamma^f_{\lvert\mathcal{Q}^f_t\rvert}$ in Eq.3	–	✓	algorithm using [40]
B_f in Lemma 3.1	standardised (=1)	fixed	–
σ in Lemma 3.1	$\sigma = r$	fixed	–
δ in Lemma 3.1	0.01	fixed	–
β_{f_t} in Lemma 3.1	1	✓	using the equation in Lemma 3.1
λ_t in Eq. 6	1	✓	using dual update in Eq. 5
ξ in Eq. 5	0.02	fixed	–
B_g in Eq. 6	1	✓	the method in Appendix F
α_1 in Eq. 6	0.01	✓	the method in Appendix F
η in line. 6 in Alg. 1	3	fixed	–
g_{thr} in line. 8 in Alg. 1	1e-5	fixed	–

ratio model. Since this part requires significantly different analysis and experiments, we leave it as future work.

I.2 Extension to Adaptive Trust Weight η

In line 6 of Alg. 2, the weight η is fixed. An adaptive η could offer better resilience to adversity. However, even without such a scheme, our no-harm guarantee holds, both theoretically and empirically.

Adaptation through the posterior standard deviation. Although η is set to be a constant in our current design of the algorithm, there is still adaptation on trusting human or the vanilla BO algorithm through the time-varying posterior standard deviation. Intuitively, if originally the expert-augmented solution x_t^c is trusted more, more samples are allocated to human-preferred region and $\sigma_t(x_t^c)$ drops quickly. Intuitively, if we keep sampling x_t^c and $x_t^u \neq x_t^c$, $\sigma_t(x_t^u)$ would finally be larger than $\eta\sigma_t(x_t^c)$ and we switch to sampling x_t^u.

Choice of η does not need to be very large in practice. Intuitively, η captures the belief on the expertise level of the human. The more trust we have on the expertise of the human, the larger η we can choose. But larger η increases the risk of higher regret due to potential over-trust in adversarial human labeler. In our experience, η does not need to be very large. Indeed, $\eta = 3$ already achieves superior performance in our experiment (see Fig. 3).

I.3 Extension to Different Acquisition Function

Our algorithm can be easily extended to other acquisition functions. For example, we can indeed use similar idea to extend expected improvement (EI) acquisition function to human constrained expected improvement (HCEI) to generate x_t^c.

$$x_t^c \in \arg\max_{x\in\mathcal{X}} \mathbb{P}(x \text{ is accepted by human})\mathrm{EI}(x). \tag{70}$$

J Experiments

J.1 Hyperparameters

We summarized the comprehensive list of hyperparameters used in this work and their settings in Table 3. Most of these are standard in typical GP-UCB approaches. The newly introduced hyperparameters are primarily tunable in a data-driven manner, and we provided a sensitivity analysis in the experiment section for those that are not.

J.2 Synthetic Function Details

J.2.1 Task Definitions

Ackely Ackley funciton is defined as:

$$f(x) := -a \exp\left[-b\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right] - \exp\left[\frac{1}{d}\sum_{i=1}^{d}\cos(cx_i)\right] + a + \exp(1) \tag{71}$$

where $a = 20, c = 2\pi, d = 4$. We take the negative Ackley function as the objective of BO to make this optimisation problem maximisation. This is a 4-dimensional function bounded by $x \in [-1, 1]^d$. The global optimum is $x^* = [0, 0, 0, 0]$ and $f(x^*) = 0$.

Hölder Table Hölder Table funciton is defined as:

$$f(x) := \left|\sin(x_1)\cos(x_2)\exp\left(\left|1 - \frac{\sqrt{x_1^2 + x_2^2}}{\pi}\right|\right)\right| \tag{72}$$

where x_i is the i-th dimensional input. This is a 2-dimensional function bounded by $x \in [0, 10]^d$. The global optimum is $x^* = [8.05502, 9.66459]$ and $f(x^*) = 19.2085$.

Rastringin Rastringin function is defined as:

$$f(x) := 10d\sum_{i=1}^{d}\left[x_i^2 - 10\cos(2\pi x_i)\right] \tag{73}$$

where x_i is the i-th dimensional input. This is a 2-dimensional function bounded by $x \in [-5.12, 5.12]^d$. The global optimum is $x^* = [0, 0]$ and $f(x^*) = 0$.

Michalewicz Michalewicz funciton is defined as:

$$f(x) := \sum_{i=1}^{d}\sin(x_i)\sin^{2m}\left(\frac{ix_i^2}{\pi}\right) \tag{74}$$

where x_i is the i-th dimensional input and $m = 10$. This is a 5-dimensional function bounded by $x \in [0, \pi]^d$. The global optimum is $f(x^*) = -4.687658$.

Rosenbrock Rosenbrock funciton is defined as:

$$f(x) := \sum_{i=1}^{d-1}\left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right] \tag{75}$$

where x_i is the i-th dimensional input. This is a 3-dimensional function bounded by $x \in [-5, 10]^d$. The global optimum is $x^* = [1]^d$ and $f(x^*) = 0$.

J.2.2 Computational time and elicitation efficiency

Figure 8 presents the comprehensive experimental results, including overhead and cumulative queries. Overhead refers to the wall-clock time in seconds required to generate the next query location. While the time taken to query the objective function is excluded, the time to query human (or synthetic) experts is included. Our overhead is the largest among the simple baselines; however, an average of around 10 seconds per query is reasonable when compared to more computationally expensive algorithms, such as information-theoretic acquisition functions, which typically require several hours per query. In most experiments, we observe a plateau in cumulative queries, indicating a handover guarantee. In the case of the Michalewicz function, a plateau has not yet been reached due to its high-dimensional nature. Nevertheless, we observe convergence acceleration in both simple and cumulative regrets.

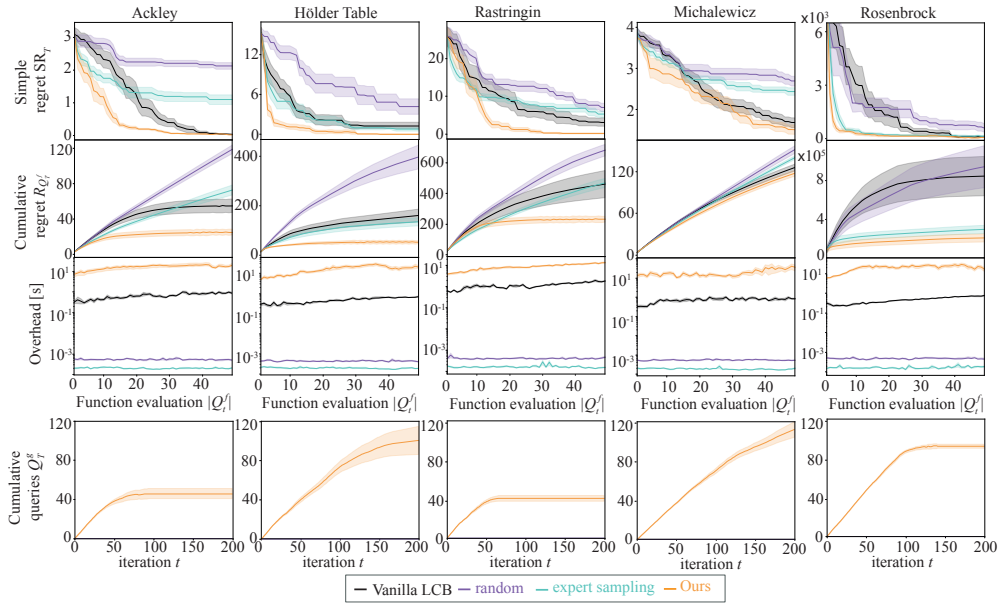


Figure 8: Simple and cumulative regrets, overhead, and cumulative queries for synthetic experiments.

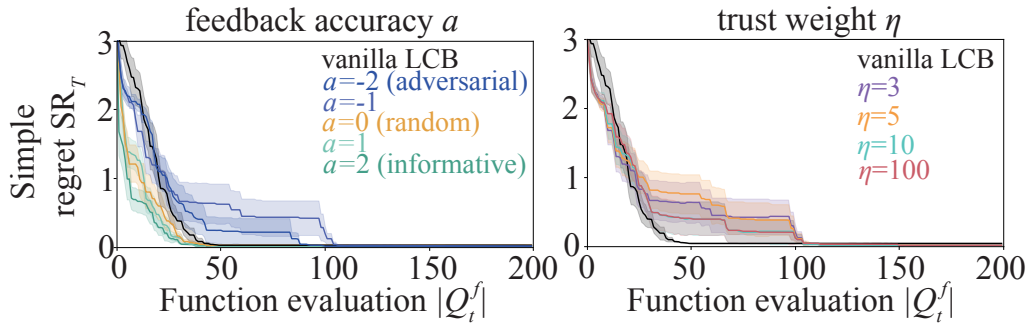


Figure 9: Confirming no-harm guarantee.

J.2.3 Comprehensive check for no-harm guarantee

We examine the no-harm guarantee by extending the iterations to confirm that our algorithm can converge at a rate comparable to the vanilla LCB. We tested with the two adversarial cases; (1) varying feedback accuracy $a \in \{-2, -1, 0, 1, 2\}$ for the fixed trust weight $\eta = 3$ and (2) varying trust weights $\eta \in \{3, 5, 10, 100\}$ for the fixed accuracy $a = -2$. Our algorithm converges to the same regret as the vanilla LCB over multiple iterations in both cases. We observed saturation behavior, where the convergence drop starts at similar locations among larger η, indicating that the no-harm guarantee is assured regardless of how large η becomes. Particularly, the convergence curves of $\eta = 10$ and $\eta = 100$ are almost identical, supporting the saturation perspective.

J.3 Human experiment details

J.3.1 Task definitions

The task involves identifying the optimal electrolyte material combination to maximize ionic conductivity in lithium-ion batteries. Ionic conductivity is crucial for reducing internal resistance, which is essential for fast charging. Slow charging remains one of the biggest challenges for the widespread adoption of electric vehicles. Therefore, finding the best electrolyte combination is crucial to advancing electric vehicle development and realizing a sustainable society.

In our study, we considered four types of electrolyte materials. For demonstration purposes, we did not conduct physical experiments. Instead, we utilized an open dataset and fitted functions to interpolate between data points, creating a continuous search space. Experiments were then performed on this synthetic data using software and four human experts. In real-world development, researchers and engineers synthesize these materials, which is expensive, making the expert's labeling process significantly cheaper than objective queries.

Li^+ standard design The first task involves the EC-DMC-EMC-$LiPF_6$ system [26, 7], where EC, DMC, and EMC are ethylene carbonate, dimethyl carbonate, and ethyl methyl carbonate, respectively, and $LiPF_6$ is lithium hexafluorophosphate. Ionic conductivity depends on both lithium salt molarity and cosolvent composition. Using the dataset from [26], we fitted the Casteel-Amis equation [19] and extended it to a continuous space. The input features are (1) $LiPF_6$ molarity, (2) DMC vs. EMC cosolvent ratio, and (3) EC vs. carbonates cosolvent ratio, with inputs bounded as $x_1 \in [0, 2]$, $x_2 \in [0, 1]$, and $x_3 \in [0, 1]$. The output is generated by adding i.i.d. zero-mean Gaussian noise with a variance of 1 to the noiseless function. We take the negative of the ionic conductivity in log mS/cm as the minimization objective.

Li^+ methyl-acetate The second task involves the MA-DMC-EMC-$LiPF_6$ system [56, 7], with MA being methyl acetate. Using the dataset from [56], we fitted the Casteel-Amis equation and extended it to continuous space. The input features are (1) $LiPF_6$ molarity, (2) DMC vs. EMC cosolvent ratio, and (3) MA vs. carbonates cosolvent ratio, with inputs bounded as $x_1 \in [0, 2]$, $x_2 \in [0, 1]$, and $x_3 \in [0, 1]$. The output is generated by adding i.i.d. zero-mean Gaussian noise with a variance of 1 to the noiseless function. We take the negative of the ionic conductivity in log mS/cm as the minimization objective.

Li^+ polymer-nanocomposite The third task involves the PEO-LLZTO nanocomposite electrolyte system [108], where PEO is polyethylene oxide, and LLZTO is lithium garnet ($Li_6.4La_3Zr_1.4Ta_0.6O_12$) nanoparticles. Using the dataset from[108], we fitted a GP model and extended it to continuous space. The input features are (1) PEO volume %, (2) LLZTO volume %, and (3) LLZTO particle size in micrometers, with inputs bounded as $x_1 \in [70, 95]$, $x_2 \in [5, 30]$, and $x_3 \in [0.04, 10]$. The output is generated by adding i.i.d. zero-mean Gaussian noise with a variance of 1 to the noiseless function. We take the negative of the ionic conductivity in log mS/cm as the minimization objective.

Li^+ Ionic liquid The fourth task involves the bmimSCN-$LiClO_4$-LiTFSI ionic liquid [72], where bmimSCN is 1-butyl-3-methylimidazolium thiocyanate, $LiClO_4$ is lithium perchlorate, and LiTFSI is lithium bis(trifluoromethanesulfonyl)imide. Using the dataset from [72], we fitted a GP model and extended it to continuous space. The input features are (1) $LiClO_4$ molarity, (2) LiTFSI molarity, and (3) bmimSCN molarity, with inputs bounded as $x_1 \in [0, 4]$, $x_2 \in [0, 1.5]$, and $x_3 \in [3, 5]$. The output is generated by adding i.i.d. zero-mean Gaussian noise with a variance of 1 to the noiseless function. We take the negative of the ionic conductivity in log mS/cm as the minimization objective.

J.4 How Do Human Experts Reason?

We explore how experts reason through these optimization tasks. Ionic conductivity is roughly estimated by the product of movable ion density and diffusivity, as described by the Nernst-Einstein equation. Experts base their evaluations on this relationship.

Li^+ standard design In this system, EC plays a crucial role in both factors. $LiPF_6$ provides movable ions (Li^+ and PF_6^-), but these ions are not mobile in their raw state due to strong electrostatic forces. EC, a highly polarized but non-charged solvent, dissolves $LiPF_6$ through solvation. Increasing EC concentration can raise movable ion density, but EC's high viscosity slows diffusivity, creating a convex curve. Experts generally agree that the global maximum is around 30% EC and 1 M $LiPF_6$, but the optimal EMC/DMC ratio remains uncertain. EMC and DMC are similar, with EMC being larger and asymmetric, and DMC being smaller and symmetric. Smaller molecules tend to be more diffusive, so a higher DMC ratio is expected to be better, although the asymmetric structure of EMC could disrupt higher-order solvation networks, contributing to diffusivity.

In summary, experts vaguely know the whole function shape and possible global optimum location for two variables, yet others are unknown.

Li^+ methyl-acetate his task involves replacing EC with MA from the first task, making the overall system similar. However, MA is an unusual material, and none of the participants are familiar with it. We will explain how experts reasoned this change in the optimization task.

EC plays a central role in dissolving $LiPF_6$, increasing movable ion density, although it is viscous. While no one knows methyl acetate, it can be inferred that it also dissolves $LiPF_6$. The challenge lies in determining its polarization ability and viscosity. EC is a planar molecule with a five-membered ring, resembling a 'small sheet magnet' with strong magnetic power but easy stacking. Conversely, MA is a small, non-ring-structured, asymmetric molecule. This asymmetry prevents MA molecules from stacking, enhancing diffusivity. However, the asymmetry also reduces polarization, leading to a weaker solvation effect and lower movable ion density.

Thus, MA has a mix of positive and negative effects, making it difficult for experts to predict the exact shape of the convex curve. Nonetheless, in most "less viscous" solvent systems, the peak typically occurs around 1.5 M of $LiPF_6$. Experts can roughly estimate this position, and this estimation is fairly accurate, as the true position is at 1.35 M.

Li^+ polymer-nanocomposite This task is completely different from the previous two tasks. Our electrolyte is now solid-state rather than liquid, so the Nernst-Einstein equation may not be applicable. However, the core idea remains the same. PEO is a framework material without ionic conductivity, whereas LLZTO has ionic conductivity. Generally, a higher LLZTO content should result in greater conductivity. Other factors are less certain.

We can anticipate the effects in both directions. Smaller particle sizes might be better because they distribute more evenly within the PEO, increasing ionic conductive paths. However, smaller particles might also be worse due to increased grain boundaries and aggregation caused by electric forces. Thus, most experts expected a convex relationship with particle size and a monotonic increase with LLZTO ratio.

In reality, experimental results showed that conductivity improved monotonically with smaller particle sizes and displayed a convex relationship with LLZTO volume. Therefore, the experts' advice was somewhat inaccurate.

Looking back, experts were partially correct. Aggregation did create the convex shape in LLZTO volume ratio, indicating their understanding of the phenomenon. However, they did not identify the correct input dimension where aggregation mattered. For particle size, the thorough mixing procedure with ball milling used in the dataset prevented aggregation, leading to misconceptions about the function shape.

Na^+ Ionic liquid This task is completely different from the previous tasks. Although our electrolyte is liquid, all materials are ionically conductive. As the name suggests, ionic liquids are special materials that can dissolve themselves without the need for a cosolvent. Consequently, the movable ion density factor remains almost unchanged, as all components are conductive regardless of composition. Therefore, diffusivity becomes the dominant factor. Diffusivity primarily depends on two factors: molecule size and electric interaction. Smaller molecules are generally more mobile, but they also have stronger electric interactions when the charge is the same (all ions in this system are monovalent).

This dual dependence leads to different expectations: if size is the dominant factor, smaller molecules (like $LiCl_4$) are expected to perform best. Conversely, if electric interaction is dominant, the results will differ.

Most experts anticipated a monotonic change in all dimensions, expecting both $LiCl_4$ and LiTFSI to show increased performance due to their smaller size compared to bmimSCN. However, experimental results showed a double peak shape for LiTFSI vs. bmimSCN and a convex shape for $LiCl_4$ and bmimSCN. Thus, the experts' advice was inaccurate. The real physical phenomena were more complex than initially thought, with electric interactions playing a more dominant role.

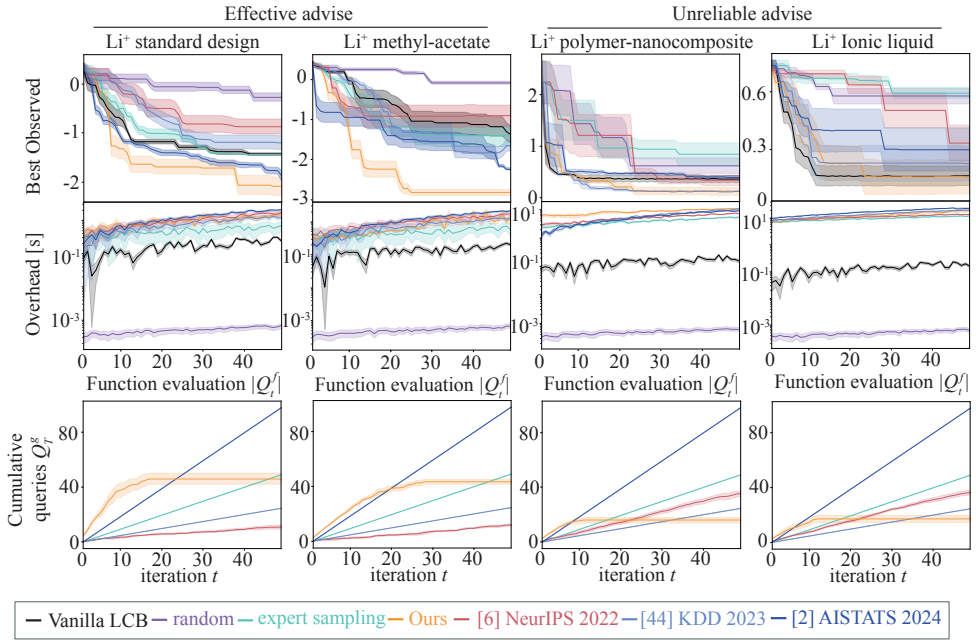


Figure 10: Simple and cumulative regrets, overhead, and cumulative queries for real-world experiments.

J.4.1 Computational time and elicitation efficiency

Figure 10 illustrates the full experimental results, including the best observed values $\max Y_{|\mathcal{Q}_T^f|}$, overhead, and cumulative queries $\mathcal{Q}_t^g$. The overhead definition remains consistent with that in the synthetic experiments. Note that these experiments include only four human trials, resulting in noisier data compared to the synthetic experiments, which used 10 random seeds. The overhead for our method and the baselines is approximately the same, around 10 seconds per query. This is manageable compared to the significantly slower methods, such as information-theoretic acquisition functions, which take several hours per query.

Regarding cumulative queries, only our method demonstrates a handover guarantee. While baseline methods continue to request human intervention even as the experiments conclude, our method stops requesting input midway through the experiments, thereby freeing the human expert from the task. Our approach allows for more effective input from experts in cases where their advice is beneficial and reduces input in unreliable cases. In contrast, the baselines request input regardless of the quality of the advice. Notably, the method described in [11] increases the frequency of requests when experts provide incorrect information. This occurs because disagreements between the surrogate f and human beliefs prompt human experts to intervene, aiming to prevent the BO from proceeding in the wrong direction. Unfortunately, this intervention can act as an adversarial response. In contrast, our algorithm avoids such scenarios through active learning constraints (as highlighted in line 6), thus achieving a no-harm guarantee in unreliable cases.