Report from Dagstuhl Seminar 25451

Bayesian Optimisation

Jürgen Branke[*1], Frank Hutter[*2], Giulia Pedrielli[*3], Matthias Poloczek[*4], and Leonard Papenmeier[†5]

1 **University of Warwick, GB.** juergen.branke@wbs.ac.uk
2 **Prior Labs – Freiburg, DE & ELLIS Institute Tübingen, DE & Universität Freiburg, DE.** fh@cs.uni-freiburg.de
3 **Arizona State University – Tempe, US.** giulia.pedrielli@asu.edu
4 **Amazon.com – San Francisco, US.** matthias.poloczek@gmx.de
5 **Universität Münster, DE.** Leonard.Papenmeier@uni-muenster.de

Abstract

This report documents the programme and outcomes of Dagstuhl Seminar 25451, "Bayesian Optimisation", held from November 2–7, 2025. The seminar brought together 39 international experts from machine learning, optimisation, statistics, and engineering to discuss recent advances, open challenges, and emerging research directions in Bayesian optimisation. The programme comprised plenary talks spanning foundational issues, benchmarking, and the growing interaction between Bayesian optimisation and generative AI, alongside focused working groups on key thematic areas. Beyond technical discussions, the seminar placed strong emphasis on community building and worked towards establishing best practices. This report summarises the plenary contributions, the outcomes of the working groups, and additional community-driven activities, including a collection of practical "tricks of the trade" and exploratory benchmarking exercises.

Seminar November 2–7, 2025 – https://www.dagstuhl.de/25451
2012 ACM Subject Classification Mathematics of computing → Bayesian computation; Computing methodologies → Gaussian processes; Computing methodologies → Machine learning approaches
Keywords and phrases AutoML, Bayesian optimization, Benchmarking, Gaussian processes
Digital Object Identifier 10.4230/DagRep.15.11.1

1 Executive Summary

Jürgen Branke (University of Warwick, GB)
Frank Hutter (Prior Labs – Freiburg, DE & ELLIS Institute Tübingen, DE & Universität Freiburg, DE)
Giulia Pedrielli (Arizona State University – Tempe, US)
Matthias Poloczek (Amazon.com – San Francisco, US)

The Dagstuhl Seminar "Bayesian Optimisation" (November 2–7, 2025) brought together 39 leading researchers from 11 countries to discuss the state of the art, emerging challenges, and future directions of Bayesian optimisation (BO). BO is a core methodology for optimizing expensive black-box functions and lies at the intersection of machine learning, optimisation,

[*] Editor / Organizer
[†] Editorial Assistant / Collector

Bayesian Optimisation, *Dagstuhl Reports*, Vol. 15, Issue 11, pp. 1–66
Editors: Jürgen Branke, Frank Hutter, Giulia Pedrielli, and Matthias Poloczek
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

and statistics, with strong contributions from engineering and applied sciences. It combines probabilistic surrogate modelling with principled decision-making to determine which evaluations are most informative under limited budgets.

Although foundational ideas in Bayesian optimisation date back to the 1950s, the field has experienced rapid growth only since around 2010, driven by advances in Gaussian process modelling, computational power, and high-impact applications such as hyperparameter tuning of machine learning methods, simulation-based optimisation, and experimental design. Despite this momentum, BO remains a fragmented research area: there is no dedicated flagship conference, and researchers are dispersed across multiple communities with differing assumptions, evaluation practices, and terminologies. A central aim of the seminar was therefore community building – bringing together researchers from machine learning, operations research, statistics, and engineering to develop a shared understanding of challenges, benchmarks, and research priorities.

The seminar programme featured 12 plenary talks designed to stimulate discussion across subfields, covering topics ranging from the historical development of BO and benchmarking practices to recent advances in high-dimensional optimisation, human-in-the-loop methods, and the integration of large language models and generative AI. In addition, participants worked in five focused working groups on themes identified as critical for the community: human-in-the-loop Bayesian optimisation; multi-fidelity Bayesian optimisation; acquisition strategies; surrogate modelling; and the intersection of Bayesian optimisation with generative AI. The outcomes of these discussions, together with abstracts of the plenary talks, are documented in this report.

Beyond structured discussions, the seminar also facilitated the exchange of practical insights. Participants jointly compiled a collection of "tricks of the trade" reflecting practical wisdom that is often absent from the formal literature. In addition, attendees were invited to apply their preferred BO tools to a small set of benchmark problems, providing a concrete basis for discussions on benchmarking, software libraries, and best practices. Summaries of these activities are also included in the report.

Schloss Dagstuhl once again proved to be an ideal environment for intensive scientific exchange and community formation. Both organisers and participants regarded the seminar as a significant success. Recognising the importance of sustained interaction for this rapidly evolving field, the participants established a Steering Group tasked with coordinating future meetings and related community-building activities. This seminar thus represents not only a snapshot of current research in Bayesian optimisation, but also a foundation for a more cohesive and collaborative BO community going forward.

2 Table of Contents

3 Seminar Program

Monday 3 November

9:00	Welcome and Introductions
10:45	Formation of Working Groups
12:15	Lunch
13:00	Roman Garnett: History of Bayesian Optimisation
13:30	Max Balandat: Bayesian Optimisation without a clear objective
14:00	Frank Hutter: Prior Fitted Networks
15:00	Coffee Break
16:00	Working Groups
18:00	Dinner

Tuesday 4 November

9:00	Carola Doerr: Benchmarking in the broader black-box optimization context: problems, performance criteria, and data sharing
9:30	Leonard Papenmeier: The Role of Benchmarks in (High-Dimensional) Bayesian Optimization
10:00	Mickaël Binois: Possible snags in benchmarking noisy or high-dimensional BO
10:30	Coffee Break
10:45	Discussion on Pre-Seminar Experimental Results
12:15	Lunch
13:30	Working Groups
15:30	Coffee Break
16:15	Aaron Klein: Towards Foundation Models for Blackbox Optimization
16:45	Elena Raponi: Automatic generation of Bayesian optimization algorithms using LLMs
17:15	Alexander Terenin: How are transformers and potential AGI affecting Bayesian optimization research?
18:00	Dinner

Wednesday 5 November

9:00	Working Group Reports
10:30	Coffee Break
10:45	Working Groups
12:15	Lunch
13:15	Group Photo
13:30	Hiking
15:30	Coffee Break
18:00	Dinner

Thursday 6 November

9:00	Working Groups
10:30	Coffee Break
10:45	Working Groups
12:15	Lunch
13:30	Bobby Gramacy: Triangulation Candidates for Bayesian Optimization
14:10	Antonio Candelieri: Convergence from any initial design: Is it utopia?
14:50	Gautam Dasarathy: Bayesian Optimization on and over graphs
15:30	Coffee Break
16:00	Working Groups
18:00	Dinner

Friday 7 November

9:15	Working Groups Reporting
10:45	Coffee Break
11:00	Tricks of the Trade
11:45	Next meetings
12:15	Lunch

4 Overview of Talks

4.1 Possible snags in benchmarking noisy or high-dimensional BO

Mickaël Binois (Inria Center at Université Côte d'Azur – Sophia Antipolis, FR)

In this short presentation, the goal is to highlight possible issues of common practices for assessing BO on noisy or high-dimensional benchmark problems. For noisy problems, it focuses on the challenge of estimating the current best solution. As for high-dimensional test problems, besides general issues for comparing different baselines, the usual practice of adding inactive variables is questioned.

4.2 Convergence from any initial design: is it an utopia?

Antonio Candelieri (University of Milano-Bicocca, IT)

Although its well-known sample efficiency and rate convergence proofs for some implementations, Bayesian Optimization's performance remains strongly affected from the random initial design. A common practice consists into running multiple experiments from different initial designs to mitigate the effect of randomness and analyse robustness of different algorithms. Although this procedure is reasonable for scientific research, it is quite often impractical in real-life applications, due to the high query cost. Moreover, it is not rare to start from

an initial design that has been collected for other – industrial-driven – purposes and that can therefore drastically affect the identification of the optimal solution given a strictly limited query budget. This talk is aimed at quickly revising some of the known issues preventing Bayesian Optimization from convergence to the global optimum, while facilitating the theoretical and methodological discussion about possible solutions to turn an utopia into reality.

References

1 A. Candelieri and E. Signori, *MLE-free Gaussian Process Based Bayesian Optimization*, in International Conference on Learning and Intelligent Optimization, pp. 81–95, Cham: Springer Nature Switzerland, June 2024.
2 A. Candelieri, *Mastering the Exploration-Exploitation Trade-off in Bayesian Optimization*, arXiv preprint arXiv:2305.08624, 2023.
3 C. Benjamins, E. Raponi, A. Jankovic, K. van der Blom, M. L. Santoni, M. Lindauer, and C. Doerr, *Pi Is Back! Switching Acquisition Functions in Bayesian Optimization*, arXiv preprint arXiv:2211.01455, 2022.
4 G. De Ath, R. M. Everson, A. A. Rahat, and J. E. Fieldsend, *Greed Is Good: Exploration and Exploitation Trade-offs in Bayesian Optimisation*, ACM Transactions on Evolutionary Learning and Optimization, 1(1), 1–22, 2021.
5 J. Berk, S. Gupta, S. Rana, and S. Venkatesh, *Randomised Gaussian Process Upper Confidence Bound for Bayesian Optimisation*, arXiv preprint arXiv:2006.04296, 2020.
6 F. Berkenkamp, A. P. Schoellig, and A. Krause, *No-regret Bayesian Optimization with Unknown Hyperparameters*, Journal of Machine Learning Research, 20(50), 1–24, 2019.
7 K. P. Wabersich and M. Toussaint, *Advancing Bayesian Optimization: The Mixed Global-Local (MGL) Kernel and Length-scale Cool Down*, arXiv preprint arXiv:1612.03117, 2016.
8 Z. Wang, F. Hutter, M. Zoghi, D. Matheson, and N. de Freitas, *Bayesian Optimization in a Billion Dimensions via Random Embeddings*, Journal of Artificial Intelligence Research, 55, 361–387, 2016.
9 Z. Wang and N. de Freitas, *Theoretical Analysis of Bayesian Optimisation with Unknown Gaussian Process Hyper-parameters*, arXiv preprint arXiv:1406.7758, 2014.
10 L. Papenmeier, M. Poloczek, and L. Nardi, *Understanding High-dimensional Bayesian Optimization*, arXiv preprint arXiv:2502.09198, 2025.
11 C. Hvarfner, E. O. Hellsten, and L. Nardi, *Vanilla Bayesian Optimization Performs Great in High Dimensions*, Proceedings of Machine Learning Research, 235, 20793–20817, 2024.

4.3 Bayesian Optimization On and Over Graphs

Gautam Dasarathy (Arizona State University – Tempe, US)

Main reference Parth Thaker, Mohit Malu, Nikhil Rao, Gautam Dasarathy: "Maximizing and Satisficing in Multi-armed Bandits with Graph Information", in Proc. of the Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 – December 9, 2022, 2022.
URL http://papers.nips.cc/paper_files/paper/2022/hash/0d561979f0f4bc6127cfcfe9c46ee205-Abstract-Conference.html

Graphs provide a natural language for representing complex, structured systems – from molecules and neural architectures to sensor networks and social systems. Bayesian Optimization (BO) offers a powerful framework for optimizing expensive black-box objectives in such domains, but incorporating graph structure raises new challenges.

This talk explores Bayesian Optimization on and over graphs. I will discuss recent advances that adapt Gaussian-process surrogates and acquisition strategies to graph domains, highlight key theoretical and computational challenges, and outline open questions in structure-aware optimization. A central theme is the role of graph topology – how spectral and combinatorial properties shape learning efficiency. I will also connect these ideas to some of our work on graph bandits, adaptive graph learning, and multifidelity learning, which may hint at a theory of Bayesian optimization that truly understands graphs.

4.4 Benchmarking BO – do we really want to learn from neighbouring disciplines?

Carola Doerr (Sorbonne University – Paris, FR)

Benchmarking is a crucial component in understanding advantages and limitations of Bayesian optimization algorithms. However, to date, researchers in the BO community did not agree on standardized way of benchmarking their algorithms; most paper use ad hoc experimental setups, data sharing and re-using is uncommon.

With the goal to spark a discussion on how to overcome this situation, I will share some benchmarking best practices developed in the broader black-box optimization community. Largely influenced by the COCO platform [2], available at `https://coco-platform.org/`, standardized benchmarking practices and data formats were adopted in the evolutionary computation community. Large data sets are available and explorable through the IOHanalyzer component [3] of the modular IOHprofiler environment [1], available at `https://iohanalyzer.liacs.nl/`.

My key take is that it would be much easier for the Bayesian optimization community to adopt similar benchmarking standards than what one might believe.

References

1 C. Doerr, H. Wang, F. Ye, S. van Rijn, and T. Bäck, *IOHprofiler: A Benchmarking and Profiling Tool for Iterative Optimization Heuristics*, CoRR, abs/1810.05281, 2018. Available at: `http://arxiv.org/abs/1810.05281`. More up-to-date documentation: `https://iohprofiler.github.io/`.

2 N. Hansen, A. Auger, R. Ros, O. Mersmann, T. Tušar, and D. Brockhoff, *COCO: A Platform for Comparing Continuous Optimizers in a Black-box Setting*, Optimization Methods and Software, pp. 1–31, 2020.

3 H. Wang, D. Vermetten, F. Ye, C. Doerr, and T. Bäck, *IOHanalyzer: Detailed Performance Analyses for Iterative Optimization Heuristics*, ACM Transactions on Evolutionary Learning and Optimization, 2(1), Article 3, pp. 3:1–3:29, 2022.

4.5 Some things I learned while writing the BayesOpt book

Roman Garnett (Washington University – St. Louis, US)

Main reference Roman Garnett: "Bayesian Optimization". Cambridge University Press, 2023.
URL https://doi.org/10.1017/9781108348973

I spoke about two somewhat niche topics that came up while working on the Bayesian optimization book:

- a brief history of Bayesian optimization, and some commentary on the origin of popular acquisition strategies (UCB, probability of improvement, expected improvement, etc.), which are almost universally misattributed in the literature
- methodology for marginalizing model hyperparameters through an acquisition strategy, and the pitfalls of the common approach of averaging the acquisition function wrt the hyperparameters rather than computing the expected utility with respect to the hyperparameter-marginal model directly

This led to a brief, but engaging discussion.

4.6 Triangulation candidates for Bayesian optimization

Robert Gramacy (Virginia Polytechnic Institute – Blacksburg, US)

Joint work of Robert Gramacy, Annie Booth, John Smith, Nathan Wycoff
Main reference Robert B. Gramacy, Annie Sauer, Nathan Wycoff: "Triangulation candidates for Bayesian optimization", in Proc. of the Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 – December 9, 2022, 2022.
URL http://papers.nips.cc/paper_files/paper/2022/hash/e9750610639c3e7a849cff746bf60dbd-Abstract-Conference.html

Bayesian optimization involves "inner optimization" over a new-data acquisition criterion which is non-convex/highly multi-modal, may be non-differentiable, or may otherwise thwart local numerical optimizers. In such cases it is common to replace continuous search with a discrete one over random candidates. Here we propose using candidates based on a Delaunay triangulation of the existing input design. We detail the construction of these "tricands" and demonstrate empirically how they outperform both numerically optimized acquisitions and random candidate-based alternatives, and are well-suited for hybrid schemes, on benchmark synthetic and real simulation experiments.

4.7 Prior-Data Fitted Networks

Frank Hutter (Prior Labs – Freiburg, DE & ELLIS Institute Tübingen, DE & Universität Freiburg, DE)

In this talk, I will discuss a new alternative to the predominant model class used in Bayesian optimization: prior-data fitted networks (PFNs). Like Gaussian processes, PFNs have strong uncertainty estimates, but unlike them they easily support many data complexities, such as mixed discrete/continuous inputs, heterogeneous noise, high dimensions, and good out-of-distribution generalization.

I will first discuss the general method for constructing PFNs and will then deepdive into one particular example: TabPFN, the first foundation model for tabular data. I hope that many colleagues in Bayesian optimization will start to explore PFNs and/or TabPFN for their applications and help develop them further.

4.8 Towards Foundation Models For Black-Box Optimization

Aaron Klein (ScaDS.AI – Leipzig, DE)

Joint work of Aaron Klein, Herilalaina Rakotoarison, Luca Thale-Bombien, David Salinas

Black-box optimization has so far relied on hand-crafted algorithms, carefully tuned for specific problem classes and rarely transferable beyond their original domains. Foundation models hold the promise of fundamentally changing this landscape: rather than manually designing optimizers, we can envision models that learn the process of optimization itself. Unlike representation learning in vision or language, this requires learning algorithms – arguably a substantially more ambitious goal – but one that recent advances in reasoning models suggest may be within reach.

In this talk, we will provide an overview of recent trends in Bayesian optimization for building foundation models and highlight current gaps in the literature.

4.9 Bayesian Optimization Without a Clear Objective: Leveraging Interactive Natural Language Feedback

Maximilian Balandat (Meta – Menlo Park, US)

Joint work of Katarzyna Kobalczyk, Zhiyuan Jerry Lin, Benjamin Letham, Zhuokai Zhao, Maximilian Balandat, Eytan Bakshy
Main reference Katarzyna Kobalczyk, Zhiyuan Jerry Lin, Benjamin Letham, Zhuokai Zhao, Maximilian Balandat, Eytan Bakshy: "LILO: Bayesian Optimization with Interactive Natural Language Feedback", CoRR, Vol. abs/2510.17671, 2025.
URL https://doi.org//10.48550/ARXIV.2510.17671

For many real-world applications, feedback is essential in translating complex, nuanced, or subjective goals into quantifiable optimization objectives. We propose a language-in-the-loop framework that uses a large language model (LLM) to convert unstructured feedback in

the form of natural language into scalar utilities to conduct BO over a numeric search space. Unlike preferential BO, which only accepts restricted feedback formats and requires customized models for each domain-specific problem, our approach leverages LLMs to turn varied types of textual feedback into consistent utility signals and to easily include flexible user priors without manual kernel design. At the same time, our method maintains the sample efficiency and principled uncertainty quantification of BO. We show that this hybrid method not only provides a more natural interface to the decision maker but also outperforms conventional BO baselines and LLM-only optimizers, particularly in feedback-limited regimes.

4.10 The Role of Benchmarks in High-Dimensional Bayesian Optimization

Leonard Papenmeier (Universität Münster, DE)

Joint work of Leonard Papenmeier, Matthias Poloczek, Luigi Nardi
Main reference Leonard Papenmeier, Matthias Poloczek, Luigi Nardi: "Understanding High-Dimensional Bayesian Optimization", in Proc. of the Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025, Proceedings of Machine Learning Research, Vol. 267, PMLR / OpenReview.net, 2025.
URL https://proceedings.mlr.press/v267/papenmeier25a.html

High-dimensional Bayesian optimization (HDBO) is a notoriously difficult challenge. This is due to the curse of dimensionality that, with linearly increasing dimensionality, requires exponentially more observations to maintain the same level of surrogate model accuracy. Recent advances have pushed the envelope to hundreds or even thousands of dimensions on various benchmarks that reflect controlled environments and real-world scenarios. This talk gives insight into specific structural properties of several of these benchmarks, how these properties interplay with design decisions of BO and high-dimensional BO algorithms, and what implicit assumptions several BO algorithms make on the objective function. The goal of this talk is to spark a discussion around best practices for HDBO benchmarking.

4.11 Automated Generation of Bayesian Optimization Algorithms assisted by LLMs

Elena Raponi (Leiden University, NL)

Joint work of Wenhu Li, Niki van Stein, Thomas Bäck, Elena Raponi
Main reference Wenhu Li, Niki van Stein, Thomas Bäck, Elena Raponi: "LLaMEA-BO: A Large Language Model Evolutionary Algorithm for Automatically Generating Bayesian Optimization Algorithms", CoRR, Vol. abs/2505.21034, 2025.
URL https://doi.org//10.48550/ARXIV.2505.21034

Large Language Models (LLMs) are rapidly becoming creative co-pilots for AI, Optimization and AutoML, moving beyond tuning to generating and evolving entire algorithms on demand. This talk introduces LLaMEA-BO, an evolutionary computation framework that uses LLMs to iteratively mutate and improve full codebases, with specific focus on Bayesian Optimization. Our framework uses an evolution strategy to guide an LLM in generating Python code that preserves the key components of BO algorithms: An initial design, a surrogate model, and an

acquisition function. The LLM is prompted to produce multiple candidate algorithms, which are evaluated on the established Black-Box Optimization Benchmarking (BBOB) test suite from the COmparing Continuous Optimizers (COCO) platform. Based on their performance, top candidates are selected, combined, and mutated via controlled prompt variations, enabling iterative refinement. Despite no additional fine-tuning, the LLM-generated algorithms outperform state-of-the-art BO baselines in 19 (out of 24) BBOB functions in dimension 5 and generalize well to higher dimensions, and different tasks (from the Bayesmark framework).

4.12 How are transformers and potential AGI affecting Bayesian optimization research?

Alexander Terenin (Cornell University – Ithaca, US)

Since the release of ChatGPT, a non-trivial segment of the machine learning community believes that human and/or superhuman artificial general intelligence will arrive in the next half-decade. While one could certainly argue that people with this view are wrong, I would instead like to ask: What if they're right? What would the implications for our research community be? Will Bayesian optimization become a valuable tool used internally as part of every AI company's infrastructure? Or, will it become possible to train transformers to do the same tasks Bayesian optimization does, thereby rendering our methods obsolete? I will not answer these questions: instead, I will introduce concepts that allow them to be asked in a precise form. To do so, I will briefly illustrate recent theory which describes what transformer models can in principle do, before moving to an open-ended discussion and question-answer period focused on showcasing different views from within the community.

4.13 The Gittins Index: A Design Principle for Decision-Making Under Uncertainty

Alexander Terenin (Cornell University – Ithaca, US)

Joint work of Ziv Scully, Alexander Terenin
Main reference Ziv Scully, Alexander Terenin: "The Gittins Index: A Design Principle for Decision Making Under Uncertainty", pp. 28–70, .
URL https://doi.org//10.1287/educ.2025.0290

The Gittins index is a tool that optimally solves a variety of decision-making problems involving uncertainty, including multi-armed bandit problems, minimizing mean latency in queues, and search problems like the Pandora's box model. However, despite the above examples and later extensions thereof, the space of problems that the Gittins index can solve perfectly optimally is limited, and its definition is rather subtle compared to those of other multi-armed bandit algorithms. As a result, the Gittins index is often regarded as being primarily a concept of theoretical importance, rather than a practical tool for solving decision-making problems. The aim of this tutorial is to demonstrate that the Gittins index can be fruitfully applied to practical problems. We start by giving an example-driven introduction to the Gittins index, then walk through several examples of problems it solves

– some optimally, some suboptimally but still with excellent performance. Two practical highlights in the latter category are applying the Gittins index to Bayesian optimization, and applying the Gittins index to minimizing tail latency in queues.

5 Working Groups

5.1 Acquisition Strategies

Thomas Bartz-Beielstein (TH Köln, DE)
Jack Buckingham (University of Warwick – Coventry, GB)
Carola Doerr (Sorbonne University – Paris, FR)
Matthias Feurer (TU Dortmund, DE)
Roman Garnett (Washington University – St. Louis, US)
Leonard Papenmeier (Universität Münster, DE)
Alexander Terenin (Cornell University – Ithaca, US)

5.1.1 Introduction

A fundamental question in Bayesian optimization research is how to acquire the next data point. The key challenge comes from the presence of *explore-exploit tradeoffs*: should the algorithm pick a location about which little information is available, in case it might perform better, or a location that one can already expect to perform well? Different acquisition functions resolve these and related tradeoffs in different ways, and the underlying design principles describing these differences are only beginning to be understood.

As a simple example, consider Bayesian optimization under a budget constraint. How should the types of points acquired depend on the remaining budget? In an informal poll of the working group, participants overwhelmingly agreed that different kinds of points should be preferred under large vs. small budgets. If the budget is large, it makes sense to prefer points with high uncertainty more, compared to if the budget is small: intuitively, this is because there is more opportunity left to benefit from taking a risk. In contrast, if the budget is small, a risky acquisition point is unlikely to pay off, so it may be better to stick with points that have a better mean. The question becomes: *how should this intuition be reflected concretely via acquisition function design?*

The default expected improvement acquisition function, in its simplest incarnation, is hyper-parameter-free and therefore cannot be adjusted depending on the budget. On the other hand, UCB can, through its confidence parameter β_T. The exploratory behavior of both probability of improvement and expected improvement can also be tuned by introducing a hyperparmeter controlling the improvement target; in the case of probability of improvement, the point maximizing the PI acquisition function for a given target also maximizes the UCB acqusition function for a corresponding confidence parameter and vice versa [12]. Some authors have also considered modifying the expected improvement acquisition strategy to account for the remaining budget via relatively simple, yet effective mechanisms [18]. Recent cost-aware acquisition functions, such as those based on Gittins index theory [25], can be adapted to the budget-constrained setting using Lagrangian relaxation: rather than working with a budget constraint, one can introduce Lagrange multipliers and add a term involving the sum of the costs to the usual simple regret objective, in some cases with appropriate optimality guarantees [30].

The need for different acquisition strategies is also reflected in other differences, beyond behavior with respect to budgets. For example, in high-dimensional Bayesian optimization, it is thought that finding the true global optimum may often be impossible in practice, and therefore be a poor goal for the algorithm to try to achieve. Instead, the algorithm should seek a *local optimum*. The question becomes: in situations with many local optima, how should the algorithm balance the tradeoff between the quality of a potential optimum and its difficulty to find? Comprehensively understanding tradeoffs like these – both from a comparative perspective focused on the behavior, popularity, and performance of existing methods, and from a fundamental theoretical perspective – was a recurring theme among the working group.

In the following, we discuss two main topics that were thoroughly explored throughout the seminar: the initial design and the optimization of the acquisition function. However, there are also other topics in the realm of acquisition strategies that we discussed in less breadth, but that still deserve further discussion, which is why we only mention them briefly. First, the topic can be extended to batch and asynchronous Bayesian optimization. However, recent work suggests that this problem might be easier than initially assumed [24]. Second, instead of using fully global Bayesian optimization, local Bayesian optimization methods could be more beneficial [27]. Third, as an intermediate strategy, it is possible to build kernels that balance the global fit of the function with a local fit around the optimum, potentially alleviating the issue that the global and local lengthscales differ [17].

5.1.2 Initial design

Bayesian optimization algorithms and other surrogate-based search techniques are often initialized with a set of non-adaptively queried points, the so-called *initial design* or *design of experiments (DoE)*. These points are often sampled uniformly at random, following some Latin Hypercube construction, or using quasi-random constructions such as subsets of Sobol' sequences or similar. Some state-of-the-art surrogate-based optimization algorithms, such as SMAC [10], also use non-adaptive throughout the optimization process, to enforce exploratory behavior. However, later versions of SMAC, which were configured for hyperparameter optimization and blackbox optimization, reduced the fraction of random samples from 50% to 20% and 8%, respectively [16, 26] We discussed the role of such non-adaptive sampling for Bayesian optimization and whether alternative sampling techniques that take into account the location but not the quality of already sampled points could improve the search behavior. In particular, we discussed the impact of replacing non-adaptive sampling by strategies that (i) maximize some form of "uniformity of distribution" of the collection of already evaluated and newly suggested points or (ii) maximize some minimal distance criterion.

As an example for (ii), we can mention the Morris-Mitchell criterion, a widely used metric for evaluating the space-filling properties of a sampling plan (also known as a design of experiments) [19]. It combines the concept of maximizing the minimum distance between points with minimizing the number of pairs of points separated by that distance. The standard Morris-Mitchell criterion, Φ_q, is defined as:

$$]\Phi_q = \left(\sum_{i=1}^{m} J_i d_i^{-q} \right)^{1/q},$$

where $d_1 < d_2 < \cdots < d_m$ are the distinct distances between all pairs of points in the sampling plan. J_i is the number of pairs of points separated by the distance d_i, and q is a positive integer (e.g., $q = 2, 5, 10, \dots$). As $q \to \infty$, minimizing Φ_q is equivalent to maximizing

the minimum distance d_1, and then minimizing the number of pairs J_1 at that distance, and so on. One limitation of the standard Φ_q is that its value depends on the number of points in the sampling plan. This makes it difficult to compare the quality of sampling plans with different sample sizes. To address this, [3] proposed a size-invariant version of the criterion, Φ_q^*. Multi-objective optimization can also use Φ_q^* as an additional criterion [3].

Initial investigations such as Bossek et al. [5] suggest a correlation between the so-called L_∞ star discrepancy of the initial design and the quality of the best found solution. However, the size and structure of the best initial design depend to a large extent on the problem's structure, and we have only a very weak understanding of which problem characteristics are most decisive for the most suitable design.

An alternative approach identifies the role of the initial design in learning good model hyperparameters and observes that typical space-filling designs may not be optimal for this purpose. For example, it will not be possible to learn short length scales if all points in the initial design are far apart, and a traditional algorithm will have to discover this "by accident" during the sequential phase. This is not to suggest that the space-filling properties are to be completely discarded,d though, since in the presence of large model-mismatch between the black-box function and its GP surrogate, points from the initial design can provide good local corrections to what might otherwise be an overconfident model. One work in this area is the development of a hybrid Latin Hypercube design for which the pairwise distances between points approximately follow a beta distribution [31]. Another recent proposal [11] uses a specific linear combination of two acquisition functions from active learning: BALD [9, 14], which maximizes mutual information between the proposed sample and the GP hyperparameters, and EPIG [4], which minimizes the conditional entropy of the GP given the proposed sample in expectation over the input space.

5.1.3 Optimizing the acquisition function

Most Bayesian optimization strategies require solving an inner optimization problem – the optimization of an *acquisition function* α:

$$\boldsymbol{x}_{\text{next}} \in \arg\max_{\boldsymbol{x}\in\mathcal{X}} \alpha(\boldsymbol{x})$$

Although most acquisition functions have a known form and are amenable to gradient-based optimizers such as L-BFGS-B, the acquisition landscape often becomes highly multimodal as the number of observations grows. While evaluating α on a discrete set of points is effective in lower dimensions, the acquisition function optimization becomes non-trivial in higher dimensions and requires, at a minimum, a multi-start strategy to address the multimodality of the function surface [29]. Another issue that can already arise for problems of moderate dimensionality is vanishing gradients of the acquisition function. Numerically more stable variants of popular acquisition functions help reduce the severity of this issue [1]. Still, for sufficiently high-dimensional problems, even these numerically more stable acquisition functions fall short in providing sufficient gradient signal, and vanishing gradients remain a challenge when maximizing α [21].

To alleviate the difficulties arising in high-dimensional Bayesian optimization, several works focus the search on regions around promising points [23, 6, 28, 22]. Since most acquisition functions are not flat around previously observed points, focusing the search to regions around previously well-performing points reduces the risk of vanishing gradients and is likely to yield points of high acquisition value [1]. The `TuRBO` algorithm [6] implements such a strategy for Thompson sampling (TS). Interestingly, TS does not suffer from vanishing

gradients as it simply searches for the maximum or minimum of a Gaussian process (GP) sample path. However, TS is overly explorative in high dimensions [20], a behavior [6] counterbalance by enforcing locality: Starting from the incumbent observation, `TuRBO` creates a dense grid of candidate points obtained from a GP sample path. To enforce locality, `TuRBO` keeps the values of the incumbent point for $1 - \min\left(1, \frac{20}{d}\right)$ randomly chosen dimensions, essentially only perturbing a subset of the dimensions for problems with more than 20 dimensions. Here, d is the dimensionality of the search space $\mathcal{X}$. By perturbing only a subset of the dimensions, candidate points remain close to the incumbent observation. Following up on that idea, [22] drop the requirement of the axis-alignment of the perturbation and propose a perturbation strategy that ensures candidate points remain within a certain radius of incumbent points.

The `spotoptim` package [2] implements a multi-objective infill strategy that uses desirability [15] to determine the next point [3]. Multiple criteria can be combined; for example, space-fillingness can be computed using the Morris-Mitchell criterion [19]. However, preliminary results indicate that space-fillingness should be used with caution, as it may push infill points too far from promising regions that have already been evaluated.

The aforementioned strategies focus on subregions of the space, mostly sacrificing global optimization unless a high number of function observations can be afforded. However, for a multi-start acquisition function optimization approach, one may use these strategies to augment the set of *starting points* with local candidates [21]. Combining global and local samples is a seemingly safe strategy, as the starting points are chosen among all samples. In fact, `BoTorch` implements such a strategy with the `sample_around_best` option, but does not enable it by default. Furthermore, `BoTorch` does not even choose the points of maximum initial acquisition value to start the gradient-based optimization, but performs Boltzmann sampling of starting points according to their initial acquisition value to ensure asymptotically space-filling behavior [1]. Arguably, the motivation for this design choice is that *strictly maximizing the acquisition function may not be the best idea*, an idea that has also been stated explicitly within the community [7]. Although, to the best of our knowledge, no works explicitly outline the conditions for this to be fulfilled, various works follow inexact acquisition function maximization schemes – often motivated by reduced computational effort. Delaunay triangulation and Voronoi tessellation leverage empirical observations of the behavior of the acquisition function to select points of high expected acquisition value [8, 8]. Additionally, [13] argue that random grid search may be sufficient for acquisition function optimization.

Despite the numerous strategies to optimize the acquisition function, no clear best strategy has been identified yet, and further research is required.

5.1.4 Outlook

Despite widespread use of Bayesian optimization, the underlying design principles for acquisition strategies are only beginning to be understood, particularly regarding how they should adapt to problem structure, dimensionality, and resource constraints. This is both from an empirical and a theoretical perspective, following other, more basic advances in Bayesian optimization, which have made tackling harder problem classes with structure possible.

References

1 Sebastian Ament, Samuel Daulton, David Eriksson, Maximilian Balandat, and Eytan Bakshy. Unexpected improvements to expected improvement for Bayesian optimization. *Advances in Neural Information Processing Systems*, 36:20577–20612, 2023.

2 Thomas Bartz-Beielstein. Sequential parameter optimization cookbook.

3 Thomas Bartz-Beielstein. Multi-objective optimization and hyperparameter tuning with desirability functions, 2025.

4 Freddie Bickford Smith, Andreas Kirsch, Sebastian Farquhar, Yarin Gal, Adam Foster, and Tom Rainforth. Prediction-oriented Bayesian active learning. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, pages 7331–7348. PMLR.

5 Jakob Bossek, Carola Doerr, and Pascal Kerschke. Initial design strategies and their effects on sequential model-based optimization: an exploratory case study based on BBOB. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, GECCO '20, page 778–786, New York, NY, USA, 2020. Association for Computing Machinery.

6 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.

7 Roman Garnett. *Bayesian optimization*. Cambridge University Press, 2023.

8 Robert B Gramacy, Annie Sauer, and Nathan Wycoff. Triangulation candidates for Bayesian optimization. *Advances in Neural Information Processing Systems*, 35:35933–35945, 2022.

9 Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning.

10 Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In Carlos A. Coello Coello, editor, *Learning and Intelligent Optimization*, pages 507–523, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

11 Carl Hvarfner, David Eriksson, Eytan Bakshy, and Max Balandat. Informed initialization for Bayesian optimization and active learning. In *Advances in Neural Information Processing Systems*.

12 Donald R. Jones. A Taxonomy of Global Optimization Methods Based on Response Surfaces. *Journal of Global Optimization*, 21(4):345–383, 2001.

13 Hwanwoo Kim, Chong Liu, and Yuxin Chen. Bayesian optimization with inexact acquisition: Is random grid search sufficient? *arXiv preprint arXiv:2506.11831*, 2025.

14 Andreas Kirsch, prefix=van useprefix=true family=Amersfoort, given=Joost, and Yarin Gal. BatchBALD: Efficient and diverse batch acquisition for deep Bayesian active learning. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

15 Max Kuhn. desirability: Function optimization and ranking via desirability functions., 9 2016.

16 Marius Lindauer, Matthias Feurer, Katharina Eggensperger, André Biedenkapp, and Frank Hutter. Towards assessing the impact of bayesian optimization's own hyperparameters, 2019.

17 Ruben Martinez-Cantin. Local nonstationarity for efficient bayesian optimization, 2015.

18 Jonas Mockus. *Bayesian Approach to Global Optimization: Theory and Applications*. Mathematics and Its Applications. Kluwer Academic Publishers, 1989.

19 Max D. Morris and Toby J. Mitchell. Exploratory designs for computational experiments. *Journal of Statistical Planning and Inference*, 43(3):381–402, 1995.

20 Leonard Papenmeier, Nuojin Cheng, Stephen Becker, and Luigi Nardi. Exploring exploration in Bayesian optimization. In *The 41st Conference on Uncertainty in Artificial Intelligence*, 2024.

21 Leonard Papenmeier, Matthias Poloczek, and Luigi Nardi. Understanding high-dimensional Bayesian optimization. In *Forty-second International Conference on Machine Learning*, 2024.

22 Bahador Rashidi, Kerrick Johnstonbaugh, and Chao Gao. Cylindrical Thompson sampling for high-dimensional Bayesian optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 3502–3510. PMLR, 2024.

23 Rommel G Regis and Christine A Shoemaker. Combining radial basis function surrogates and dynamic coordinate search in high-dimensional expensive black-box optimization. *Engineering Optimization*, 45(5):529–555, 2013.

24 Ben Riegler, James A C Odgers, and Vincent Fortuin. Rethinking exploration in asynchronous bayesian optimization: Standard acquisition is all you need. In *The Exploration in AI Today Workshop at ICML 2025*, 2025.

25 Ziv Scully and Alexander Terenin. The Gittins index: A design principle for decision making under uncertainty. In *Tutorials in Operations Research: Advances in Analytics and Operations Research: Improving Decisions to Secure the Future*, pages 28–70. INFORMS, 2025.

26 Kim Peter Wabersich and Marc Toussaint. Advancing bayesian optimization: The mixed-global-local (mgl) kernel and length-scale cool down, 2016.

27 Kaiwen Wu, Kyurae Kim, Roman Garnett, and Jacob Gardner. The behavior and convergence of local bayesian optimization. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 73497–73523. Curran Associates, Inc., 2023.

28 Kaiwen Wu, Kyurae Kim, Roman Garnett, and Jacob Gardner. The behavior and convergence of local Bayesian optimization. *Advances in neural information processing systems*, 36:73497–73523, 2023.

29 Nathan Wycoff, John W. Smith, Annie S. Booth, and Robert B. Gramacy. Voronoi candidates for Bayesian optimization. *CoRR*, abs/2402.04922, 2024.

30 Qian Xie, Raul Astudillo, Peter Frazier, Ziv Scully, and Alexander Terenin. Cost-aware Bayesian optimization via the Pandora's box Gittins index. *Advances in Neural Information Processing Systems*, 37:115523–115562, 2024.

31 Boya Zhang, D. Austin Cole, and Robert B. Gramacy. Distance-distributed design for Gaussian process surrogates. 63(1):40–52.

5.2 Multifidelity

Mauricio Alvarez (University of Manchester, GB)
Cédric Archambeau (Helsing – Berlin, DE)
Maximilian Balandat (Meta – Menlo Park, US)
Ivo Couckuyt (Ghent University, BE)
Frank Hutter (Prior Labs – Freiburg, DE & ELLIS Institute Tübingen, DE & Universität Freiburg, DE)
Aaron Klein (ScaDS.AI – Leipzig, DE)
Neeratyoy Mallik (Universität Freiburg, DE)
Ruth Misener (Imperial College London, GB)
Juan Ungredda (ESTECO SpA – Trieste, IT)

5.2.1 Introduction

Many scientific, engineering, and machine learning problems require making critical decisions under strict resource constraints. In such settings, directly evaluating the performance metric of interest – through large-scale model training, high-resolution simulations, or physical experiments – is often prohibitively expensive. As a result, practitioners rely on a collection of cheaper approximations, such as simplified models, coarse simulations, or proxy experiments, which differ in accuracy, uncertainty, and cost. These approximations are commonly referred to as fidelities.

More recently, a growing body of work has demonstrated that performance in complex systems often follows regular, predictable patterns as a function of key resource-controlling variables, such as model size, data size, resolution, or experimental scale. These relationships, commonly referred to as scaling laws, provide models that extrapolate performance beyond observed regimes and enable informed decision making about how to allocate resources. While scaling laws have been most prominently studied in the context of large language model pretraining, similar ideas arise naturally in other domains like computational fluid dynamics.

Multi-fidelity methods exploit the availability of multiple information sources with varying querying costs, while scaling laws provide structured models that describe how performance evolves as fidelity or resource levels change. When combined within a Bayesian framework, scaling laws can serve as informative priors or mean functions, while multi-fidelity surrogate models capture deviations, uncertainty, and correlations across fidelities. Crucially, this combination enables the design of more efficient acquisition functions that explicitly trade off cost, uncertainty reduction, and extrapolation to high-fidelity regimes. Arguably, scaling laws are indispensable when high fidelity evaluations are extremely time-consuming and/or costly, and available computing budget can typically support 1 high fidelity evaluation (up to 5 evaluations).

The central goal of this working group is to explore scaling-laws and multi-fidelity Bayesian optimization to predict performance and guide data collection. This includes developing probabilistic models for scaling behavior, quantifying uncertainty in extrapolation, and designing acquisition strategies that actively select which designs, fidelities, and resource levels to evaluate next. Such strategies move beyond naive grid searches or uniform sampling, enabling principled exploration of large design spaces under tight budgets.

We begin by formalizing a general multi-fidelity optimization problem in which design variables and fidelity levels are treated as joint decision variables under a global budget constraint. We then examine several application domains, including scaling laws for large language model pretraining, computational fluid dynamics simulations, and chemical engineering. These case studies illustrate how scaling laws and multi-fidelity structure manifest differently across domains, yet can be unified under a common Bayesian optimization perspective.

5.2.2 Problem Statement

Let $\mathcal{X}$ denote the *design space*, which may be continuous ($\mathcal{X} \subseteq \mathbb{R}^d$) or discrete/finite. Let $\mathcal{W}$ denote the *fidelity space*, which may likewise be continuous ($\mathcal{W} \subseteq \mathbb{R}^p$) or discrete (e.g., $\{1, \dots, M\}$). We consider an unknown response function

$$h : \mathcal{X} \times \mathcal{W} \to \mathbb{R}.$$

where an evaluation at a design–fidelity pair $(x, w) \in \mathcal{X} \times \mathcal{W}$ produces an observation

$$y = h(x, w),$$

In some settings, such as deterministic simulations, evaluations may be effectively noise-free. In others, including stochastic simulations or physical experiments, it is common to assume additive noise, $y = h(x, w) + \varepsilon_w$, where ε_w is a random noise term whose distribution and variance may depend on the fidelity level w.

Given a *target fidelity* $w^\star \in \mathcal{W}$ is specified – corresponding to the most accurate and most expensive evaluation – we aim to identify a design that maximizes the performance,

$$x^\star \in \arg\max_{x \in \mathcal{X}} h(x, w^\star).$$

We assume access to a total budget $B > 0$, representing a limited resource such as time, monetary cost, computational effort, or experimental capacity. This budget may be allocated sequentially over rounds $t = 1, 2, \dots, T$. At each round, the algorithm selects a design–fidelity pair (x_t, w_t), observes y_t, and incurs a cost $c(x_t, w_t)$, subject to the constraint

$$\sum_{t=1}^{T} c(x_t, w_t) \leq B.$$

After consuming the budget, the algorithm returns a recommended design x_r, intended to achieve high performance at the target fidelity, i.e., a large value of $h(x_r, w^\star)$.

5.2.3 Application: Scaling Laws for LLM pretraining

Scaling laws are used widely in pretraining generative model, such as LLM, to make high-stakes decisions, for example, which model size, which data size, which hyperparameters, etc.. Current practice is to run large grid searches across model and data scales as well as hyperparameters for each of these, and then fit a parametric law. This is likely highly inefficient since large amounts of the data collected may not be very relevant to improving the fit of the parametric law. Furthermore, the scaling law approaches in the literature (and in practice) typically do not take into account uncertainty. We believe that there is a large opportunity to improve the estimation of scaling laws by using Bayesian techniques (modeling as well as optimization / active learning).

There are multiple aspects to this problem, the primary ones are surrogate modeling for the scaling behavior, and the acquisition strategy for adaptively choosing which models to train on which data under which hyperparameter settings. Both modeling and acquisition strategies will depend on the specific goal we are trying to achieve downstream. In the following, we consider a particular, popular scaling law and propose a GP-based modeling approach for it. We plan to evaluate this proposed model on publicly available data sets.

5.2.3.1 An Example Scaling Law

The well-known Chinchilla Scaling Law [1] models the validation loss of a transformer model by $\mathcal{L}(D,N) = E + \frac{A}{N^\alpha} + \frac{B}{D^\beta}$ where N specifies the number of parameter and D the number of token. To estimate the parameters E, A, B, α, β, it is common to use a predefined grid of different N, D with optimal learning rates LR and batch sizes BS. To estimate the optimal hyperparameters (incl. LR and BS), they need to evaluate another gird search for each N, D pair. This is potentially quite wasteful, and we suggest to instead employ Bayesian modeling together with an active learning strategy to explore the joint space of model size, data size, and hyperparameters.

Specifically, we aim to model the validation loss $\mathcal{L}(\mathbf{x})$ with $\mathbf{x} = (D, N, LR, BS)$ to also include hyperparameters LR and BS in addition to token (D) and parameter (N) count. This has two key advantages:

1. We hypothesize that leveraging additional data points (all parameter settings rather than just the optimal one) improves the predictive quality of the model.
2. The ability to predict the loss not only as a function of the number of tokens D and parameters N but also for the hyperparameters will enable us to predict the optimal hyperparameter settings for large D and N – a key decision one needs to make for training the full model, and often shifts with increasing D and N.

Using a probabilistic model of this behavior will enable uncertainty quantification, which generally is very useful for decision making (esp. given the costs of the training jobs involved) and, also be leveraged in a principled active learning approach to efficiently estimate the parameters of the scaling law.

We assume:

$$\mathcal{L}(\mathbf{x})|f(\mathbf{x}),\theta \sim \mathcal{N}(f(\mathbf{x}),\sigma^2) \tag{1}$$

$$f(\mathbf{x})|\theta \sim \mathcal{GP}\left(m_{\theta_m}(\mathbf{x}), k_{\theta_k}(\mathbf{x},\cdot)\right) \tag{2}$$

where:

$$m_{\theta_m}(\mathbf{x}) = m_{\theta_m}(N,D) = E + \frac{A}{N^\alpha} + \frac{B}{D^\beta} \tag{3}$$

is the prior mean function with parameters $\theta_m = (E, A, B, \alpha, \beta)$ and k_{θ_k} is the kernel function with θ_k the kernel hyperparameters (incl. lengthscales, etc.).

Given a data set $\mathcal{X}$, we can now fit the parameters $\theta = (\theta_m, \theta_k)$ using maximum likelihood optimization (we could also use other inference approaches such as fully Bayesian / MCMC).

5.2.3.2 Selective selection of hyperparameter configurations

We use Bayesian active learning by disagreement (BALD) [2], which is equivalent to the expected information gain, to sample the hyperparameter configurations to evaluate:

$$U_{\text{BALD}}(\mathbf{x}) = \mathbb{H}\left[\mathcal{L}(\mathbf{x})|\mathcal{D}\right] - \mathbb{E}_{p(\theta|\mathcal{D})}\mathbb{H}\left[\mathcal{L}(\mathbf{x})|\theta)\right] \tag{4}$$

$$= \mathbb{H}\left[\mathbb{E}_{p(\theta|\mathcal{D})}\mathcal{L}(\mathbf{x})|\theta,\mathcal{D}\right] - \mathbb{E}_{p(\theta|\mathcal{D})}\mathbb{H}\left[\mathcal{L}(\mathbf{x})|\theta\right] \tag{5}$$

where $\mathcal{D} = \{(\mathbf{x}_i, \mathcal{L}_i\}_{i=1}^n$. Note that $p(\mathcal{L}(\mathbf{x})|\theta, \mathcal{D})$ is the predictive posterior Gaussian process and $p(\mathcal{L}(\mathbf{x})|\theta)$ is the marginal likelihood, both of which are analytically tractable (and have a Gaussian form).

A somewhat simpler approach would be to adopt a Bayesian query by committee approach.

$$U_{\text{BQC}}(\mathbf{x}) = \mathbb{V}_{p(\theta|\mathcal{D})}\left[\mu_\theta(\mathbf{x}) - \bar{\mu}_\theta(\mathbf{x})|\theta\right] = \mathbb{E}_{p(\theta|\mathcal{D})}\left[\left(\mu_\theta(\mathbf{x}) - \bar{\mu}_\theta(\mathbf{x})\right)^2 |\theta\right] \tag{6}$$

where $\mu_\theta(\mathbf{x})$ is the predictive posterior Gaussian process mean and $\bar{\mu}_\theta(\mathbf{x}) = \mathbb{E}_{p(\theta|\mathcal{D})}\left[\mu_\theta(\mathbf{x})\right]$.

In addition, we want to penalize cost. We adopt the approximate cost $c(N, D) = 6ND$ [3], such that

$$U_c(\mathbf{x}) = \frac{U(\mathbf{x})}{c(\mathbf{x})}. \tag{7}$$

How to approximate the marginal posterior $p(\theta|\mathcal{D})$? The challenge with the active selection of the hyperparameter configurations is that the posterior $p(\theta|\mathcal{D})$ is analytically untractable. One strategy is to approximate it via variational inference: $q(\theta) \approx p(\theta|\mathcal{D})$. The expectations appearing in the acquisition functions can then be computed by Monte Carlo integration by sampling from $q(\theta)$.

5.2.3.3 Data

We identified the following published data sources as potential candidates for evaluating our modeling approach:

1. Porian et al. [4] provides the validation loss for various pre-trained language models across an increasing N and D trained with a fixed grid of learning rates and batch sizes.
2. Nezhurina et al. [5] provides data to derive scaling laws for language-vision models.

In addition, ongoing work by some of the authors on training dense transformers on language data using µP parameterization will provide additional data points.

A preliminary investigation of the data sets can be found in the following Colab notebook: `https://colab.research.google.com/drive/1XPEaJ3az8DhwEa6grkKYy0pUjfi0XjqX`

Initial observations from literature and experiments on large language/vision models show consistent scaling behavior with model size, data size, and compute costs. The novelty and challenge of this problem stem from the need for strict extrapolation: at larger scales, both optimal hyperparameters and loss values fall outside the ranges observed in training data. Additionally, the parameter space is sparsely sampled across all dimensions – data (D), model size (N), and hyperparameters (LR, BS, etc.) – due to high computational costs and non-systematic exploration strategies used in data collection.

5.2.4 Applications

5.2.4.1 LLMs

Scaling laws are an important tool for (pre-)training LLMs, as they allow performance to be extrapolated and enable informed decisions about the hyperparameters of the final model. However, accumulating the data needed to fit these scaling laws is a manual, multi-step process that requires human expertise.

The approach presented here would automate this process and reduce the overall computational cost. This would be particularly appealing to smaller or academic labs that do not have access to large amounts of computational resources, as it would enable them to more thoroughly compare different architectures or datasets and ensure sensible scaling.

5.2.4.2 CFD simulations

Computational Fluid Dynamics (CFD) is a domain in which scaling-law multifidelity modeling may be applicable. In CFD applications, the fidelity can often be controlled continuously through physically meaningful parameters such as mesh resolution, time-step size, or solver accuracy. As these fidelity controls are refined, such as refining the mesh or tightening solver tolerances, the simulation fidelity tends to produce more reliable information, but at an increasing computational cost. Because this accuracy–cost trade-off follows regular patterns, coarse simulations can often be used to guide how high-fidelity simulations should be performed. In this setting, scaling laws (whether derived analytically, empirically, or learned probabilistically) can play a central role in guiding multifidelity Bayesian optimization by predicting how the solution error and uncertainty decay with increasing fidelity.

In engineering (e.g., CFD), scaling laws can be constructed for multi-fidelity objective functions (i.e., end-to-end or black-box approaches, or, alternatively, considering multi-fidelity mesh sizes (indirectly defining the objective function, i.e., a grey-box approach).

We propose to study these relationships using the MegaFlow2D dataset and reactor benchmark[1]. This dataset provides CFD simulations at multiple resolutions but does not include a specific downstream objective. It is well-suited for exploring super-resolution and multifidelity modeling approaches.

5.2.4.3 Chemistry & Chemical Engineering

Scale-up is a consistent challenge in industrial chemistry and chemical engineering: we want to move from (i) cheap computer simulations to (ii) expensive computer simulations to (iii) cheap experiments on a lab bench to (iv) more expensive experiments on a lab bench to (v) experiments in a pilot plant to (vi) experiments at scale. At each of these levels, we have data on all previous levels and wish to move to the next level. The number of experiments at the coarser level(s) will be large compared to the number of experiments at the finer levels.

Unless there are computer simulations in the chemical engineering examples that obey known and exploitable scaling laws, scaling-law approaches are less applicable to this class of applications. Instead, fidelity levels are best viewed as categorical experimental or computational modalities, such as laboratory-scale experiments versus pilot-scale studies.

Therefore, while scaling laws are less relevant to the chemistry and chemical engineering examples, there are additional considerations that need to be incorporated:

- The input space is highly constrained because not every molecule is synthesizable,
- Each of the fidelities in the output space are subject to uncertainty, these are typically notated in chemistry experiments as the different results for different replicas.

The three examples we have already integrated into our Google Colab document consist on:

- Battery design [6] represents experiments aimed at optimizing electrode materials for lithium-ion batteries by leveraging measurements obtained at different experimental fidelities. The variables dop1–dop4 correspond to controllable electrode design parameters, that represent dopant concentrations in high-nickel cathode materials, which define a specific candidate electrode formulation. Each experiment is conducted at a fidelity level indicated by W, where lower fidelities represent cheaper and faster proxy experiments, most notably coin-cell tests or shortened testing protocols, while higher fidelities correspond to

[1] `https://github.com/trsav/reactor_benchmark` – The reactor benchmark is provided by Imperial college. Currently, contacts have been established to investigate possible collaboration.

more expensive, time-consuming, and industrially relevant experiments, such as pouch-cell testing that better reflects real battery performance. The output variable Y measures the specific capacity (electrochemically accessible charge per unit mass) at the chosen fidelity and is the variable to be maximized.

- Covalent organic frameworks [7, 8] represents a materials discovery problem over a fixed, finite set of 609 experimentally reported covalent organic frameworks (COFs), where each candidate material is represented by a 14-dimensional continuous feature vector derived from its crystal structure and composition. These 14 features encode physically meaningful properties of each COF that are known to influence gas adsorption behavior. The variable W denotes the simulation fidelity level used to evaluate a given COF: a low-fidelity Monte Carlo integration to compute Xe and Kr Henry coefficients under the dilute approximation (15 min runtime), or a high-fidelity Markov-chain Monte Carlo simulation of the binary grand-canonical ensemble that explicitly models competitive Xe/Kr adsorption (230 min runtime). The response variable Y is the simulated xenon/krypton adsorptive selectivity at room temperature, which quantifies how preferentially a COF adsorbs Xe over Kr.
- Molecule polarizability [8, 9] consists of 1,134 organic molecules, each represented by a vector of molecular descriptors that capture structural and chemical features such as atom types, bonding patterns, and functional groups. Following the authors' methodology, these high-dimensional descriptors are reduced to a 10-dimensional continuous representation using principal component analysis. The variable W denotes the fidelity level of the polarizability measurement, where the low-fidelity data corresponds to polarizability computed at the Hartree–Fock 6-31G+ level of theory, which is relatively inexpensive, while the high-fidelity data corresponds to experimentally measured polarizability. The response variable Y is the molecular polarizability itself. The dataset is structured to support multi-fidelity Bayesian optimization, with the objective of identifying molecules that maximize experimental polarizability.

References

1 Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training Compute-Optimal Large Language Models. *arXiv e-prints*, page arXiv:2203.15556, March 2022.

2 D. A. Cohn, Z. Ghahramani, and M. I. Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 1996.

3 J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei. Scaling laws for neural language models. *arXiv:2001.08361 [cs.LG]*, 2020.

4 Tomer Porian, Mitchell Wortsman, Jenia Jitsev, Ludwig Schmidt, and Yair Carmon. Resolving Discrepancies in Compute-Optimal Scaling of Language Models. *arXiv e-prints*, page arXiv:2406.19146, 2024.

5 Marianna Nezhurina, Tomer Porian, Giovanni Pucceti, Tommie Kerssies, Romain Beaumont, Mehdi Cherti, and Jenia Jitsev. Scaling Laws for Robust Comparison of Open Foundation Language-Vision Models and Datasets. *arXiv e-prints*, page arXiv:2506.04598, June 2025.

6 Jose Pablo Folch, Robert M. Lee, Behrang Shafei, David Walz, Calvin Tsay, Mark van der Wilk, and Ruth Misener. Combining multi-fidelity modelling and asynchronous batch bayesian optimization. *Computers & Chemical Engineering*, 172:108194, 2023.

7 Nickolas Gantzler, Aryan Deshwal, Janardhan Rao Doppa, and Cory M. Simon. Multi-fidelity bayesian optimization of covalent organic frameworks for xenon/krypton separations. *Digital Discovery*, 2:1937–1956, 2023.
8 Víctor Sabanza-Gil, Riccardo Barbano, Daniel Pacheco Gutiérrez, Jeremy S Luterbacher, José Miguel Hernández-Lobato, Philippe Schwaller, and Loïc Roch. Best practices for multi-fidelity bayesian optimization in materials and molecular research. *Nature Computational Science*, pages 1–10, 2025.
9 Clyde Fare, Peter Fenner, Matthew Benatan, Alessandro Varsi, and Edward O Pyzer-Knapp. A multi-fidelity machine learning approach to high throughput materials screening. *npj Computational Materials*, 8(1):257, 2022.

5.3 Human-in-the-loop Bayesian Optimization: Explainability and Interactivity

Jürgen Branke (University of Warwick, GB)
Katharina Eggensperger (TU Dortmund, DE)
Jonathan Fieldsend (University of Exeter, GB)
Daniel Hernández-Lobato (Autonomous University of Madrid, ES)
Mike McCourt (Distributional – Toronto, CA)
and Sebastian Rojas Gonzalez (Ghent University, BE)

5.3.1 Introduction

This is the report of the working group on Human-In-the-Loop (HITL) Bayesian Optimization which was part of the Dagstuhl Seminar on Bayesian Optimization. We started by collecting and categorizing different ways in which humans can interact with Bayesian Optimization (BO). These modes of interaction include: providing prior knowledge, by interacting with the algorithm during optimization, or after optimization, when picking or fine-tuning solutions or asking for explanations. Section 5.3.2 summarizes what we found. Later, we focused on two particular aspects: explainability, which is discussed in Section 5.3.3 and bounding future improvements or deciding when to stop – initial results of which are set out in Section 5.3.4.

5.3.2 Human-in-the-loop Optimization

In most optimization scenarios, the HITL provides a priori information about the problem in the form of a proper problem formulation, including objective function(s), decision variables and constraints. However, there can also be additional information that the HITL may provide, such as expert or domain knowledge about possible future scenarios, the location within the search space where good solutions are believed to be, or some already evaluated initial solutions, e.g., from historical observations.

Interactively, the HITL may propose promising solutions (i.e., taking the role of the acquisition function), add or remove constraints, modify the objective, modify the explore/exploit balance, or decide when they are satisfied with the solution and want to stop. For example, in [39], the HITL can provide advice on the next query point through binary accept/reject recommendations.

Preferential BO [15, 31] assumes that it is not possible to formulate a quantifiable objective function; instead, one must rely entirely on the HITL to guide the optimization. Pairwise BO interacts with the HITL by repeatedly asking the HITL which of two solutions is preferred. The goal of the acquisition function is then to pick the most informative pair of solutions to show to the HITL. Preferences can also be stated as rankings when the HITL is capable of simultaneously considering more than two solutions. A very interesting recent development in this area is the use of LLMs to interact with the HITL; this may allow the BO to elicit, in a very natural way, more complex preference information than just which of two solutions is better [24, 23].

After optimization, the HITL may ask for a better understanding of the solution, such as how far the solution might be from the optimum, how robust the solution is, which decision variables are the most influential, or under what circumstances another solution would be better. Section 5.3.3 will look more closely into this latter aspect.

5.3.2.1 Multiobjective Bayesian Optimization

A special case of HITL BO is the case of multi-objective optimization. In the case of multiple objectives, there is usually not a single solution best in all the objectives, but there are several Pareto-optimal solutions with different trade-offs. Therefore, in multi-objective BO, a HITL has to be involved at some point in order to identify the most preferred of the Pareto-optimal solutions. The definition of the *selected* "optimal" solution is subjective and depends on the goals of the HITL. Preference elicitation can be done *a priori*, i.e., before the optimization, interactively during the optimization, or *a posteriori*, after optimization by the HITL picking the most preferred solution from the set of Pareto-optimal solutions.

Progress in this area has evolved from multiple relatively independent research streams. Despite the shared goal of aiding decision-making for the HITL, these research directions have historically evolved in isolation, with contributions spanning multiple fields. For example:

- Analytical methods, including (multi-attribute) utility theory (e.g., [28, 34, 1]);
- Heuristic optimization approaches, especially evolutionary algorithms (e.g., [7, 8, 38]); and
- Bayesian optimization frameworks (e.g., [32, 15, 6, 2]).

Recent efforts to bridge the research through hybrid or integrative approaches are promising, but remain scarce and face several unresolved challenges:

Type of Preference Information. Human–algorithm interaction can yield diverse forms of preference data, including:

- weights or priority levels assigned to objectives,
- partial or complete rankings of candidate solutions,
- pairwise comparisons between alternatives,
- specification of aspiration levels or desired improvement directions (which may e.g. relate to the performance of existing designs),
- upper or lower bounds on objective values.

However, the amount and quality of preference information that can be reliably extracted from the Decision Maker (DM) are difficult, especially considering that the ultimate objective is to *select the most preferred solution from a relatively small "menu" of diverse options that are all high-performing*, independently of the characteristics of the problem (e.g., dimensionality, convexity, stochasticity, etc.). Excessive interaction or cognitive demand may induce biases, inconsistencies, and decision fatigue [3]. Moreover, the *informativeness* and *interaction cost* of different preference types may vary across optimization stages, yet

systematic guidance on this aspect is lacking. Most existing preference models are parametric, requiring parameter estimation from limited human feedback [9], which introduces additional sources of computational complexity (and uncertainty).

Frequency and Timing of Interaction. Determining when and how often to engage the HITL remains an open problem. Frequent interactions can lead to inconsistency or transitivity violations (e.g., preferring A over B, B over C, but C over A), whereas infrequent interactions may yield insufficient information and inefficient search behavior [29]. Adaptive interaction strategies that balance informativeness and cognitive load are therefore an important research frontier.

Imperfect and Noisy Feedback. Many existing studies assume that human-provided information is accurate and consistent. In reality, human responses are subject to uncertainty, hesitation, and contextual variability. Modeling and propagating this uncertainty within the optimization process – rather than neglecting it – is crucial for realistic HITL systems [5].

Heterogeneity of Decision Makers. Decision makers differ in their attitudes toward risk, prior knowledge, and cognitive strategies. For instance, risk-averse, risk-neutral, and risk-seeking individuals may exhibit fundamentally different preference structures [16]. Capturing such heterogeneity in a generalizable manner remains challenging, as fully characterizing individual decision models is infeasible without imposing strong assumptions.

Decisions under Uncertainty. The integration of preference learning with stochastic or uncertain optimization remains underexplored relative to deterministic formulations. While utility theory approaches have modeled exogenous uncertainty (arising from environmental variability) [16], Bayesian optimization methods often address endogenous uncertainty (stemming from model or observation noise) [6]. Developing frameworks that jointly account for both uncertainty sources represents a promising but demanding direction for future research [20, 35].

Performance Evaluation and Benchmarking. The evaluation of human-in-the-loop optimization algorithms remains a nascent research area. Traditional performance metrics – typically based on the quantitative assessment of the Pareto front in deterministic problems (e.g., regret or hypervolume) – are inadequate for interactive or stochastic contexts. Emerging studies propose benchmark problems with known solutions [3, 4], but standardized methodologies for evaluating interactive and preference-based optimization remain largely undeveloped.

5.3.3 Problem formulation: Explainability in HITL-BO

A critical component of a successful HITL system is **explainability**. A lack of transparency can cause the user to lose trust in the system. For example: the system solves an optimization problem and finds the best solution x^* based on the current model parameters, θ_0:

$$x^* = \arg\max_x f(x, \theta_0). \tag{8}$$

The HITL is unsatisfied and asks: "*Why did you not select x'?*". Similar to counterfactual explanations in machine learning and following [13], we aim to explain the decision by answering the question: "*In what alternative context θ' would the solution expected by the HITL be optimal, or better than the solution found by BO under θ_0?*".

The proposed solution is to find the smallest change to the preference parameters θ_0 that would make x' a "good" choice. We have explored several mathematical formulations for the problem in the context of BO. We aim to find a new set of parameters θ' that is minimally distant from the original parameters θ_0, as measured by a distance function $d(\theta', \theta_0)$.

Dividing the decision variables into two sub-groups $x = (a, b)$ where $a \in \mathbb{R}^{m_1}, b \in \mathbb{R}^{m_2}$ and $x \in \mathbb{R}^{m_1+m_2}$, we distinguish the following two cases.

Case 1: x' is a complete alternative solution.

Case 2: The HITL only specifies a partial solution a', essentially asking the question "*Why does the optimal solution not use a'?*".

5.3.3.1 Explaining why the HITL proposal is inferior

In Case 1, one way to convince the HITL that the identified optimal solution x^* is superior, is to simply compute $f(x', \theta_0)$ and show that $f(x', \theta_0) < f(x^*, \theta_0)$. If the HITL has only specified a partial solution a' as in Case 2, then one would have to demonstrate that even the best $x = (a', b)$ would be worse, i.e., $\max_b f(a', b, \theta_0) < f(x^*, \theta_0)$. This is straightforward to do with BO.

5.3.3.2 Finding a context where the HITL proposal is better

In Case 1, this approach finds the closest possible context (the θ' with the minimum distance $d(\theta', \theta_0)$) where the user's choice x' would score higher than the system's original choice x^*.

$$\theta' = \arg\min_{\theta} d(\theta, \theta_0) \text{ s.t. } f(x', \theta) \geq f(x^*, \theta). \tag{9}$$

Figure 1 illustrates the situation. At θ_0, the function for x^* (blue line) is higher than for x' (black line). The algorithm finds the closest θ' where the blue line is higher than or equal to the black line.

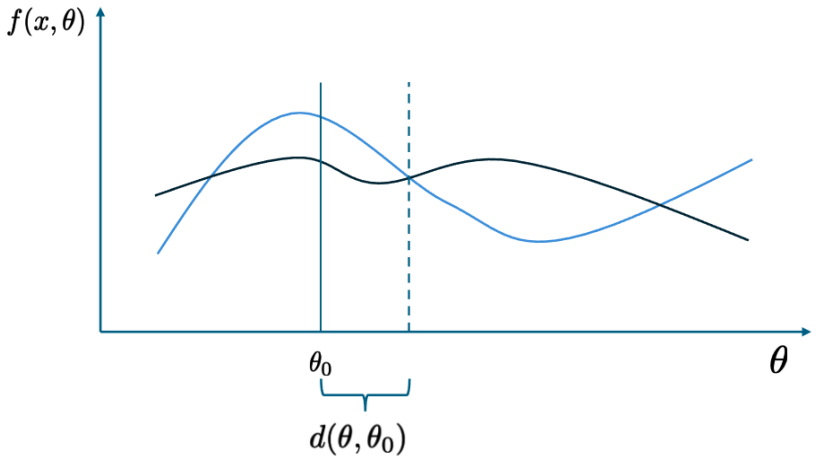


Figure 1 Blue line illustrates $f(x^*, \theta)$, black line illustrates $f(x', \theta)$.

In BO, this can be achieved as follows. First, the goal would be to find a solution where $f(x', \theta) > f(x^*, \theta)$ (if we cannot find one, we can inform the HITL that the proposed x' is worse under any possible context θ). This can simply be achieved by maximizing $f(x', \theta) - f(x^*, \theta)$ over θ, e.g., with expected improvement. Note that every sample for this problem requires the evaluation of two solutions, (x', θ) and (x^*, θ). Using Knowledge Gradient or Entropy Search, this could also be tackled by allowing only one of the solutions x' or x^* to be evaluated in each iteration.

Once a θ has been found, the goal changes to finding the closest θ, i.e., we search for a reduction in distance subject to still ensuring $f(x', \theta) > f(x^*, \theta)$. A possible acquisition function for this could be an adapted form of constrained EI:

$$\alpha(\theta) = (d(\theta', \theta^*) - d(\theta, \theta^*))\mathbb{P}(f(x', \theta) > f(x^*, \theta)) \tag{10}$$

with θ' being the best θ found so far and $\mathbb{P}(.)$ being the probability that the suggested solution is better under model parameters θ.

Case 2 is a bit more complicated, as it would only fix a partial solution a', while allowing other variables, namely b, to vary. So, the problem becomes

$$\min d(\theta, \theta_0) \text{ s.t. } \max_b f((a', b), \theta) \geq f(x^*, \theta). \tag{11}$$

This can still be solved using a similar two-phase approach as above, first identifying any θ that satisfies the constraint by optimising over both, θ and b, then aiming to find a better θ. Note, however, that when varying b for a previously evaluated θ, only $f(a', b, \theta)$ needs to be evaluated. This could be taken into account by multiplying the EI value for those points by two.

5.3.3.3 Find a context where the HITL proposal is (close to) optimal

For Case 1, this approach seeks the closest context θ' where x' is not just better than x^*, but is "close to optimal". This means its value is within a small margin ϵ of the *true maximum* value achievable in that new scenario:

$$\theta' = \arg\min_\theta d(\theta, \theta_0) \quad | \quad f(x', \theta) \geq (1 - \epsilon) \max_x f(x, \theta). \tag{12}$$

This answers the question, "*What is the smallest change in input parameters that would make x' a genuinely good solution (e.g., within 1% of the best possible function value)?*", see Figure 2.

This is a bi-level optimisation problem – optimising θ subject to x being optimal, and is thus difficult to solve. For some BO approaches to bi-level problems see [12, 10].

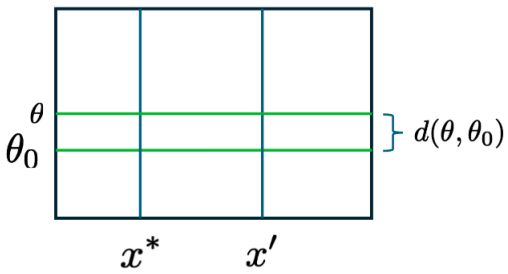


Figure 2

Case 2 is even more complicate, involving a maximisation on each side of the constraint:

$$\theta' = \arg\min_\theta d(\theta, \theta_0) \text{ s.t. } \max_b f((a', b), \theta) \geq (1 - \epsilon) \max_x f(x, \theta). \tag{13}$$

5.3.3.4 An Entropy Search Approach for Case 1

We observe that if we had access to the blackbox, i.e., $f(\cdot, \cdot)$, we could directly answer all questions regarding the problem of interest. However, our knowledge about $f(\cdot, \cdot)$ is only partial, as a result of solving the optimization problem in (8). Ideally, what we would like

to do is to collect data efficiently, to provide an answer to the question, given Case 1. This involves evaluating the black box at new candidate locations $f(x_{\text{new}}, \theta_{\text{new}})$. However, since evaluating f is expensive, we would like to choose carefully these new evaluations. This is challenging, since providing an answer to the question corresponding to Case 1 is not exactly optimizing f.

In this section we describe how entropy search (ES) [33, 17, 19] can be used to address Case 1, described above. A nice feature of ES is that it can be easily adapted to aim at reducing the uncertainty about some particular feature of the objective, which need not be related to the optimum. This contrasts with most acquisition functions from the BO literature, which are particular designed to find the global optimum and may require complicated tweaking to address other goals (e.g, EI may require to perform two evaluations at each iteration, as described above). To our knowledge, ES has not been used to counterfactual analysis in the context of BO.

To address Case 1, we may define a random variable that represents the answer to the question posed. Namely,

$$\omega \equiv \begin{cases} 1 \text{ if there is a } \theta \text{ for which } x' \text{ is optimal.} \\ 0 \text{ otherwise.} \end{cases} \tag{14}$$

Since there is a probabilistic model for f, the black-box function, this induces a probability distribution for ω, as illustrated in Figure 3.

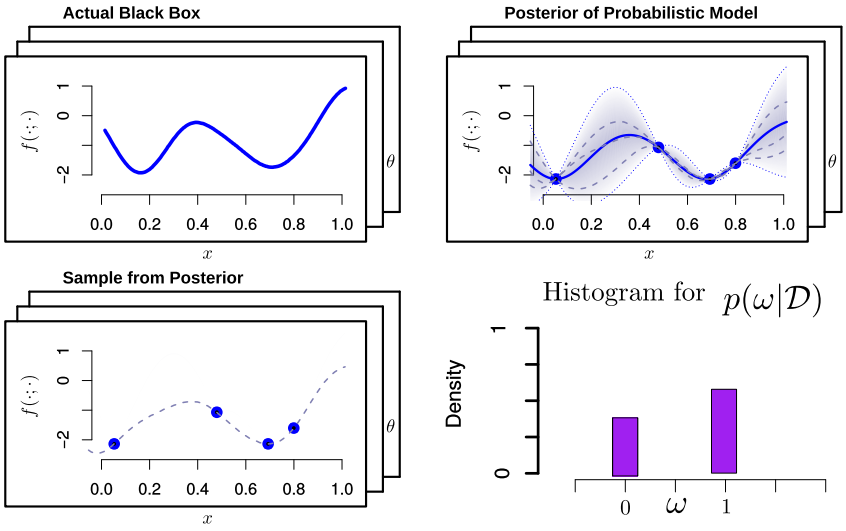


Figure 3 Graphical illustration of the process of generating samples of ω. Note that we have illustrated the dependence of f with respect to θ as an extra axis.

Ideally, we would like to reduce our uncertainty about the potential values of ω, which directly translates into reducing or uncertainty about the potential answer, under Case 1. The idea is that if the entropy of ω is large, we are far from answering the question, and if the entropy of ω is small, we are close to answering the question. This is illustrated in Figure 4.

Therefore, with the goal of reducing the entropy of ω, we can define the following acquisition function:

$$\alpha(x, \theta) = (H[\omega|\mathcal{D}] - \mathbb{E}_y[H[\omega|\mathcal{D} \cup \{x, \theta, y\}]])/d(\theta, \theta_0)\,, \tag{15}$$

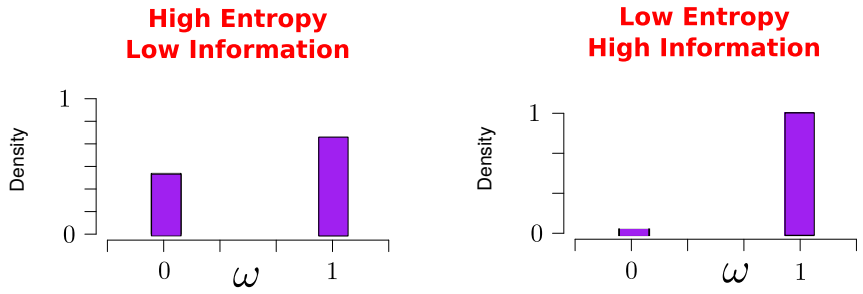


Figure 4 Graphical illustration of two distributions for ω. One with high and one with low entropy.

where $H[\cdot]$ denotes differential entropy, $\mathcal{D}$ are the data collected so far, the expectation is with respect to the predictive distribution of the probabilistic model, and $d(\theta, \theta_0)$ is the distance to θ_0. Thus, we simply favor choosing points that reduce the most the entropy of ω in expectation, while preferring evaluations that are close to θ_0.

Of course, evaluating (15) is very challenging. However, we can use use that such an expression is simply the mutual information between y and ω [19]. Namely, $I(y;\omega)$. Since the mutual information is symmetric, we can swap the variables y and ω, resulting in the following equivalent expression to (15):

$$\alpha(x,\theta) = (H[y|\mathcal{D}] - \mathbb{E}_\omega[H[y|\mathcal{D},\omega]])/d(\theta,\theta_0)\,, \tag{16}$$

which now involves the entropy of the unconditional predictive distribution, $H[y|\mathcal{D}]$, an expectation with respect to ω, and the entropy of the conditional predictive distribution, $H[y|\mathcal{D},\omega]$, given ω. The first term is easy to compute, the expectation can be approximated via Monte Carlo, simply by sampling ω, and $H[y|\mathcal{D},\omega]$ must be approximated.

Computing $H[y|\mathcal{D}]$ is straight-forward. To sample ω, we can generate samples of the black-box function, and check whether ω takes values 1 or 0 under that sample. The conditional predictive distribution whose entropy, $H[y|\mathcal{D},\omega]$, needs to be computed, requires approximation. Note that this depends on the predictive distribution given the sampled value for ω. Thus, we have to constrain the predictive distribution of the probabilistic model to be compatible with ω. This may be, in principle, done simply by incorporating extra likelihood factors that enforce that:

- there is a θ, for which x' is optimal, if $\omega = 1$,
- there is not a θ for which x' is optimal, if $\omega = 0$,

as it is done in standard predictive entropy search [19]. Depending on the probabilistic model, it can be the case that these factors must be approximated with Gaussians. Similar factors have been used in the context of multi-objective optimization [18].

Of course, there are some challenges to address, like a large number of samples have to be generated to ensure that ω takes values 1. Thus, we may relax that $\omega \in \{0,1\}$ and simply let $\omega \in [0,1]$ [11]. In this case, ω will simply give an intuitive idea about how close x' is to being optimal. Ideally, we should also try to use the already observed data $\mathcal{D}$ collected when solving problem (8). Therefore, we may want to impose strong correlations with respect to θ.

5.3.4 A Metric to Quantify the Expected Improvement of the Recommended Solution w.r.t. the Preferences of the HITL

In Bayesian optimization, determining when to stop the algorithm is crucial because the process is inherently expensive, often involving costly evaluations of the objective. Continuing indefinitely can lead to diminishing returns, where additional evaluations provide negligible

improvement in the solution while consuming significant resources. A well-defined stopping criterion ensures efficiency and prevents over-exploration. Moreover, stopping at the right time balances exploration and exploitation, avoiding unnecessary sampling in regions unlikely to yield better results, and thus optimizes resource allocation.

Stopping has been a concern in Bayesian optimization for a considerable time, though mainly in a non-HITL context. As e.g. set out in the recent textbook in the area by [14], from decision theory the time to stop is when the expected gain from the optimal observation exceeds the cost of acquisition (note that practically this requires an accurate utility function representing the decision maker preferences, a well-calibrated predictive model of $f()$ and an accurate model of the mapping from each solution to its evaluation cost). Early works such as by [22] discuss stopping once EI yields a value lower than a certain threshold in terms of the best value seen so far (1%, or 0.01 on the log scale). More recent work has explored this in terms of regret for EI over the best-observed value so far [30]. Using combinations of acquisition functions have also been explored: [26] proposed using EI to determine query points, but applies a threshold on the probability of improvement of the proposed sample to decide when to stop. [27] derives an upper bound for the simple regret (assuming a well-calibrated Gaussian process model), and utilizes this with a threshold for stopping hyperparameter optimization (recommending the threshold is of a similar magnitude as the statistical error of the empirical estimator in order for it to be useful in practice). Recent work by [25] relies on tracking information in a historical window of queried data and optimization is stopped if (i) these data all lie within a convex hull and (ii) local regret in this region falls below some preset threshold. All these approaches do require the user to determine one (or more) termination thresholds *a priori*.[2] As discussed (along with many other approaches) in [14], stopping may also be prompted by achieving some desired target function value (as distinct from seeking the best possible performing solution under $f()$). Garnett also notes that "*In many cases, the time of termination may in fact not be under the control of the optimization routine at all but instead decided by an external agent.*".

We intuitively defined a metric that measures what is the expected gain in the objective if we keep evaluating the objective. For this, one only has to consider the knowledge about the objective provided by the probabilistic model. For simplicity, consider the noiseless evaluation setting, and let γ be the best observation so far. We can define a global expected improvement (GlobalEI) metric that estimates the expected improvement in the objective by performing as many evaluations as needed to obtain the objective's global optimum:

$$\text{GlobalEI} = \mathbb{E}_{f(x^\star)}[f(x^\star) - \gamma]\,, \tag{17}$$

where the expectation is with respect to the distribution of the global optimum, assuming maximization. Of course, the evaluation of (17) is intractable. However, the expectation can be approximated by Monte Carlo, simply by generating sample functions from the probabilistic model, which can then be optimized very cheaply. Figure 5 and 6 compare this metric with expected improvement and regret, showing that GlobalEI provides more reasonable estimates than EI of how close we are to solving the optimization problem. In Figure 6, the model hyper-parameters (the model is a GP) are fixed to the optimal ones. Thus, there should be no model bias. We speculate that the differences between the GlobalEI and regret are due to sub-optimal optimization when computing GlobalEI using samples from the GP.

[2] In a number of these works a naive baseline heuristic of stopping once the best found has not improved for a certain (user defined) number of queries is also compared.

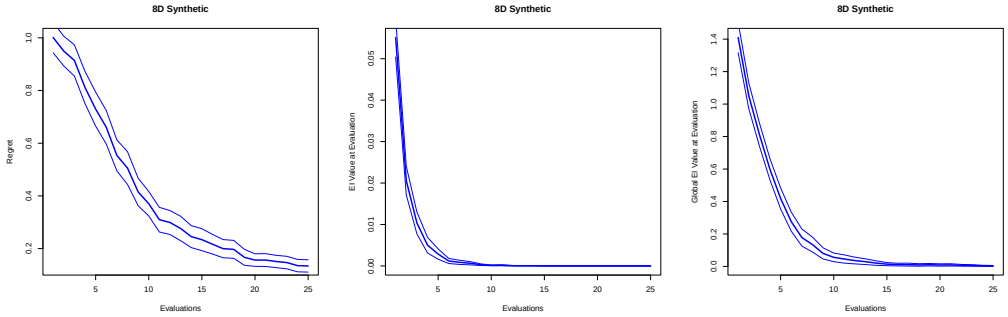


Figure 5 Comparison, on an 8-dimensional synthetic problem, where the objective is sampled from a GP, of (left) regret, expected improvement at the candidate location (center), and the proposed Global EI metric (right). We observe that Global EI does a better job than EI at estimating how far we are from solving the optimization problem. However, both metrics seem to be too optimistic in this case. Model hyper-parameters are estimated from the observed data and not set to optimal values. We report average results over 25 repetitions.

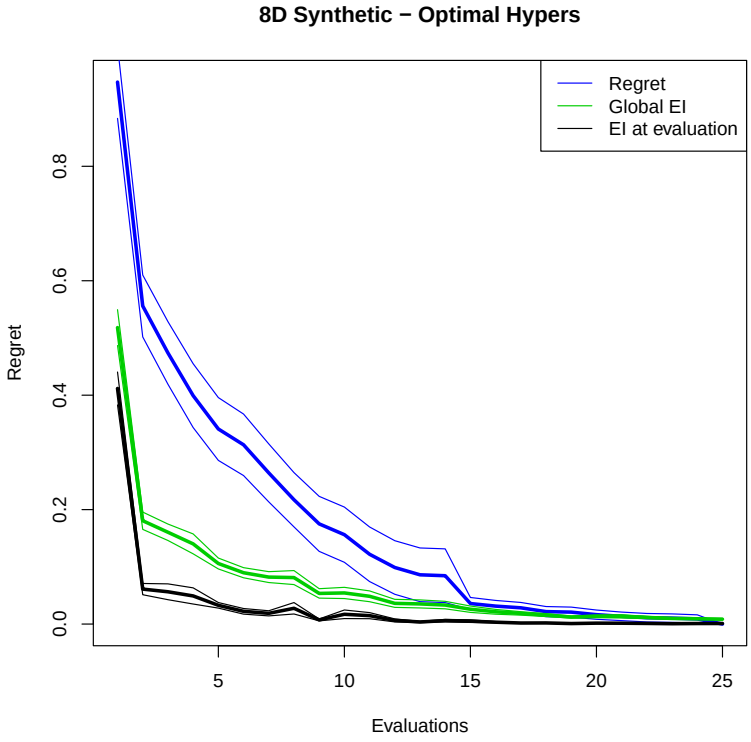


Figure 6 Comparison, on an 8-dimensional synthetic problem, where the objective is sampled from a GP, of (left) regret, expected improvement at the candidate location (center), and the proposed Global EI metric (right). We observe that Global EI does a better job than EI at estimating how far we are from solving the optimization problem. However, both metrics seem to be too optimistic in this case. Model hyper-parameters are fixed to the optimal ones. We report average results over 25 repetitions.

The proposed metric is related to the one described by [36], who, instead of considering the expected value in (17), considers the probability that $f(x^\star) - \gamma$ is above a particular threshold value. GlobalEI is equivalent to the criterion described by [21], for minimization instead of maximization. Finally, in the literature, there are also more complicated rules for deciding when to stop when the objective evaluations may have different costs, as described by [37].

References

1 A. E. Abbas. *Foundations of Multi-attribute Utility*. Cambridge University Press, Cambridge, UK, 2018.

2 M. Adachi, B. Planden, D. A. Howey, M. A. Osborne, S. Orbell, N. Ares, and S. L. ... Chau. Looping in the Human Collaborative and Explainable Bayesian Optimization. arXiv preprint arXiv:2310.17273, 2023.

3 B. Afsar, K. Miettinen, and F. Ruiz. Assessing the performance of interactive multiobjective optimization methods: A survey. *ACM Computing Surveys (CSUR)*, 54(4):1–27, 2021.

4 B. Afsar, A. B. Ruiz, and K. Miettinen. Comparing interactive evolutionary multiobjective optimization methods with an artificial decision maker. *Complex & Intelligent Systems*, 9 (2):1165–1181, 2023.

5 B. Armbruster and E. Delage. Decision making under uncertainty when preference information is incomplete. *Management science*, 61(1):111–128, 2015.

6 R. Astudillo, Z. J. Lin, E. Bakshy, and P. I. Frazier. qEUBO: A decision-theoretic acquisition function for preferential Bayesian optimization. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS '23)*, pages 1093–1114, 2023.

7 J. Branke. Consideration of partial user preferences in evolutionary multiobjective optimization. In J. Branke, K. Deb, K. Miettinen, and R. Słowiński, editors, *Multiobjective Optimization: Interactive and Evolutionary Approaches*, volume 5252 of *Lecture Notes in Computer Science*, pages 157–178. Springer, Berlin / Heidelberg, Germany, 2008. doi:10.1007/978-3-540-88908-3_6.

8 J. Branke. MCDA and Multiobjective Evolutionary Algorithms. In S. Greco, M. Ehrgott, and J. R. Figueira, editors, *Multiple Criteria Decision Analysis: State of the Art Surveys*, volume 148 of *International Series in Operations Research and Management Science*, pages 977–1008. Springer, Cham, Switzerland, 2016. doi:10.1007/978-1-4939-3094-4_23.

9 J. Branke, S. Corrente, S. Greco, R. Słowiński, and P. Zielniewicz. Using Choquet integral as preference model in interactive evolutionary multiobjective optimization. *European Journal of Operational Research*, 250(3):884–901, 2016.

10 R. W. T. Chew, Q. P. Nguyen, and B. K. H. Low. BILBO: BILevel Bayesian Optimization. *arXiv preprint arXiv:2502.02121*, 2025.

11 M. De Santis and F. Rinaldi. Continuous reformulations for zero–one programming problems. *Journal of Optimization Theory and Applications*, 153:75–84, 2012.

12 O. Ekmekcioglu, N. Aydin, and J. Branke. Bayesian optimization of bilevel problems. *arXiv preprint arXiv:2412.18518*, 2024.

13 A. Forel, A. Parmentier, and T. Vidal. Explainable data-driven optimization: from context to decision and back again. In *International Conference on Machine Learning*, pages 10170–10187. PMLR, 2023.

14 R. Garnett. *Bayesian Optimization*. Cambridge University Press, 2023.

15 J. González, Z. Dai, A. Damianou, and N. D. Lawrence. Preferential Bayesian optimization. In *International Conference on Machine Learning*, pages 1282–1291. PMLR, 2017.

16 W. B. Haskell, W. Huang, and H. Xu. Preference elicitation and robust optimization with multi-attribute quasi-concave choice functions. *arXiv preprint arXiv:1805.06632*, 2018.

17 P. Hennig and C. J. Schuler. Entropy Search for Information-Efficient Global Optimization. *Journal of Machine Learning Research*, 13(6):1809–1837, 2012.

18 D. Hernández-Lobato, J. Hernandez-Lobato, A. Shah, and R. Adams. Predictive entropy search for multi-objective bayesian optimization. In *International conference on machine learning*, pages 1492–1501. PMLR, 2016.

19 J. M. Hernández-Lobato, M. W. Hoffman, and Z. Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. *Advances in neural information processing systems*, 27, 2014.

20 F. Huber, S. Rojas Gonzalez, and R. Astudillo. Bayesian preference elicitation for decision support in multi-objective optimization. *Journal of Multi-Criteria Decision Analysis*, 32(3): e70019, 2025.

21 H. Ishibashi, M. Karasuyama, I. Takeuchi, and H. Hino. A stopping criterion for Bayesian optimization by the gap of expected minimum simple regrets. In *International Conference on Artificial Intelligence and Statistics*, pages 6463–6497. PMLR, 2023.

22 D. Jones, M. Schonlau, and W. Welch. Efficient Global Optimization of Expensive Black-Box Functions. *Journal of Global Optimization*, 13:455–492, 1998.

23 A. S. Jovine, T. Ye, F. Bahk, J. Wang, D. B. Shmoys, and P. I. Frazier. LISTEN to Your Preferences: An LLM Framework for Multi-Objective Selection. *arXiv preprint arXiv:2510.25799*, 2025.

24 K. Kobalczyk, Z. J. Lin, B. Letham, Z. Zhao, M. Balandat, and E. Bakshy. LILO: Bayesian Optimization with Interactive Natural Language Feedback. *arXiv preprint arXiv:2510.17671*, 2025.

25 S. Li, K. Li, and W. Li. "Why Not Looking backward?"A Robust Two-Step Method to Automatically Terminate Bayesian Optimization. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 43435–43446. Curran Associates, Inc., 2023. URL `https://proceedings.neurips.cc/paper_files/paper/2023/file/870c1e0589822bf37590b84984c345c4-Paper-Conference.pdf`.

26 R. Lorenz, R. P. Monti, I. R. Violante, A. A. Faisal, C. Anagnostopoulos, R. Leech, and G. Montana. Stopping criteria for boosting automatic experimental design using real-time fMRI with Bayesian optimization. In *5th NIPS Workshop on Machine Learning and Interpretation in Neuroimaging: Beyond the Scanner*. arXiv preprint arXiv:1511.07827, 2015.

27 A. Makarova, H. Shen, V. Perrone, A. Klein, J. B. Faddoul, A. Krause, M. Seeger, and C. Archambeau. Automatic Termination for Hyperparameter Optimization. In I. Guyon, M. Lindauer, M. van der Schaar, F. Hutter, and R. Garnett, editors, *Proceedings of the First International Conference on Automated Machine Learning*, volume 188 of *Proceedings of Machine Learning Research*, pages 7/1–21. PMLR, 25–27 Jul 2022. URL `https://proceedings.mlr.press/v188/makarova22a.html`.

28 K. Miettinen. *Nonlinear Multiobjective Optimization*, volume 12 of *International Series in Operations Research and Management Science*. Kluwer Academic Publishers, Boston, MA, USA, 1999.

29 G. Misitano, B. Afsar, G. Lárraga, and K. Miettinen. Towards explainable interactive multiobjective optimization: R-XIMO. *Autonomous Agents and Multi-Agent Systems*, (2), 2022.

30 V. Nguyen, S. Gupta, S. Rana, C. Li, and S. Venkatesh. Regret for Expected Improvement over the Best-Observed Value and Stopping Condition. In M.-L. Zhang and Y.-K. Noh, editors, *Proceedings of the Ninth Asian Conference on Machine Learning*, volume 77 of *Proceedings of Machine Learning Research*, pages 279–294, Yonsei University, Seoul, Republic of Korea, 15–17 Nov 2017. PMLR. URL `https://proceedings.mlr.press/v77/nguyen17a.html`.

31 S. Takeno, M. Nomura, and M. Karasuyama. Towards practical preferential Bayesian optimization with skew Gaussian processes. In *International Conference on Machine Learning*, pages 33516–33533. PMLR, 2023.

32 P. Viappiani and C. Boutilier. Optimal Bayesian recommendation sets and myopically optimal choice query sets. In *Advances in Neural Information Processing Systems (NIPS)*, volume 23, 2010.

33 J. Villemonteix, E. Vazquez, and E. Walter. An informational approach to the global optimization of expensive-to-evaluate functions. *Journal of Global Optimization*, 44(4): 509–534, 2009.

34 J. Wallenius, J. S. Dyer, P. C. Fishburn, R. E. Steuer, S. Zionts, and K. Deb. Multiple Criteria Decision Making and Multiattribute Utility Theory: Recent Accomplishments and What Lies Ahead. *Management Science*, 54(7):1336–1349, 2008. doi:10.1287/mnsc.1070.0838.

35 H. Wang, J. Branke, and M. Poloczek. Bayesian Optimization with Preference Exploration by Monotonic Neural Network Ensemble. *arXiv preprint arXiv:2501.18792*, 2025.

36 J. Wilson. Stopping Bayesian optimization with probabilistic regret bounds. *Advances in Neural Information Processing Systems*, 37:98264–98296, 2024.

37 Q. Xie, L. Cai, A. Terenin, P. I. Frazier, and Z. Scully. Cost-aware stopping for Bayesian optimization. *arXiv preprint arXiv:2507.12453*, 2025.

38 B. Xin, L. Chen, J. Chen, H. Ishibuchi, K. Hirota, and B. Liu. Interactive Multiobjective Optimization: A Review of the State-of-the-Art. *IEEE Access*, 6:41256–41279, 2018. doi:10.1109/ACCESS.2018.2852976.

39 W. Xu, M. Adachi, C. N. Jones, and M. A. Osborne. Principled Bayesian optimization in collaboration with human experts. *Advances in Neural Information Processing Systems*, 37: 104091–104137, 2024.

5.4 Surrogate Models for Bayesian Optimization

Steven Adriaensen (University of Freiburg, DE)
Nathalie Bartoli (DTIS, ONERA, Université de Toulouse, FR)
Mickaël Binois (Inria, Université Côte d'Azur, CNRS, LJAD, Sophia Antipolis, FR)
Robert Gramacy (Virginia Polytechnic Institute – Blacksburg, US)
Rodolphe Le Riche (CNRS LIMOS at Mines Saint-Etienne, FR)
Szu Hui Ng (National University of Singapore, SG)
Inneke Van Nieuwenhuyse (Computational Mathematics, Hasselt University, BE)
Jixiang Qing (Mathematical Sciences, Lancaster University, UK)
Antonio Candelieri (Economics Management and Statistics, University of Milano-Bicocca, IT)

5.4.1 Overview

Surrogates are models of the objective function learned from observed data. This working group focused on surrogate models for Bayesian Optimization (BO), surveying the range of surrogate modeling approaches currently in use and emerging in the literature. We discussed how different surrogate models address key modeling challenges in BO, compared their respective strengths and limitations, and reflected on what fundamentally qualifies as a surrogate model in the BO loop and why/when surrogate-based approaches are preferred over model-free alternatives.

5.4.1.1 What are Surrogate Models?

This report is concerned with **surrogate models** [35, 29] that appear as components of the Bayesian Optimization (BO) algorithm. Recall that the overall goal of such algorithm is to estimate the solutions to the optimization problem

$$\min_{x \in \mathcal{X}} f(x) \qquad \text{where } f \ : \ \mathcal{X} \to \mathbb{R} \tag{18}$$

within a limited number of calls to f, a budget denoted as T. A pseudo-code for BO is given in Algorithm 1.

The surrogate, denoted $Y(x)$, is a model of the function f that is learned from past observations $\mathbf{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, where $y_i = f(x_i)$. The surrogate is a key ingredient to make BO cost effective by guiding the search towards high potential parts of the search space $\mathcal{X}$ without calling the true function f. In its most general form, $Y(x)$ is a random

Algorithm 1 Sketch of a Bayesian Optimization algorithm that produces batches of q points at each iteration.

Require: Search space $\mathcal{X}$, Objective function $f : \mathcal{X} \to \mathbb{R}$
Require: A surrogate model $Y(x)$ (e.g., a Gaussian Process), an acquisition function $\alpha_Y(.)$ (e.g., Expected Improvement, Upper Confidence Bound) that depends on $Y(x)$
Require: Max number of calls to f, T, Initial design points $\mathbf{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, where $y_i = f(x_i)$, $t \leftarrow n$
1: Initialize surrogate model with data $\mathbf{D}$
2: **while** $t \leq T$ **do**
3: Fit the surrogate model to the observed data $\mathcal{D}$
4: Optimize the acquisition function to find the next query points:

$$\{x_{t+1}, \ldots, x_{t+q}\} \in \arg \max_{x'_1, \ldots x'_q \in \mathcal{X}^q} \alpha_Y(x'_1, \ldots x'_q)$$

5: Evaluate the objective functions:

$$y_{t+1} = f(x_{t+1}) \ , \ldots , y_{t+q} = f(x_{t+q}) \ , \ t \leftarrow t + q$$

6: Augment the data:

$$\mathbf{D} = \mathbf{D} \cup \{(x_{t+1}, y_{t+1}), \ldots, (x_{t+q}, y_{t+q})\}$$

7: **end while**
8: **return**
9: - the best observed point $x^* \in \arg\min_{x_i \in \mathbf{D}} y_i$ (in noiseless context) or the best expected function at the observed points (in a noisy context) $x^* \in \arg\min_{x_i \in \mathbf{D}} \mathbb{Y}(x_i)$
10: - the last surrogate and $\mathbf{D}$

process where each instance $y(x, \omega) : \mathcal{X} \times \Omega \to \mathbb{R}$, with Ω the sample space, can be seen as a possible f. An illustration of surrogate in 1-D is given in Figure 7. To define a BO algorithm, a complete random process is not needed: the minimal amount of information needed to construct a global optimization algorithm such as a BO is *i)* a prediction of the function f at a new point x and *ii)* a measure of how the value of f is uncertain at this new x. For the illustration in Figure 7, these would be the expectation of $Y(x)$ and its variance, or any monotonous transformations of them. These two pieces of information are required in any acquisition criterion $\alpha_Y()$ that trades off intensification of the search in regions of $\mathcal{X}$ that are likely to have high-performance with exploration of unknown regions.

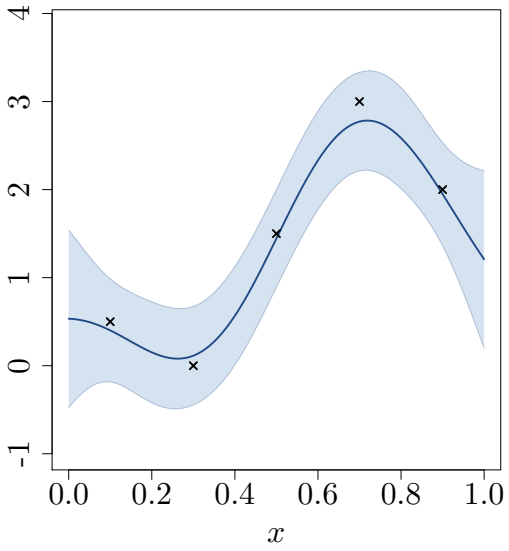


Figure 7 Illustration of a surrogate $Y(x)$. Here, it is a Gaussian process regression in the presence of noise. The solid line is the function prediction, the interval around has size plus/minus 2 standard deviations, the filled tube is $\mathbb{E}Y(x) \pm 2\sqrt{\mathbb{V}Y(x)}$, from [50].

In brief, in this report, surrogates are models of the objective function that provide, at least, both a prediction and a measure of uncertainty. We do not consider models directly outputting activation functions, or distribution models directly proposing points in $\mathcal{X}$ to evaluate (like in evolutionary optimization, e.g., CMA-ES [42, 41]).

5.4.1.2 Why Surrogate Models?

Probabilistic surrogate models enable principled, uncertainty-aware guidance, which is central to achieving high sample efficiency in settings where evaluations are expensive. Beyond guiding where to evaluate next, surrogate models can also inform which solution to return, for example by selecting the model-predicted optimum rather than the best observed evaluation, a distinction that is particularly relevant in settings with observation noise. Moreover, this enables non-myopic strategies that select evaluation points primarily to improve model fidelity rather than for their immediate objective value. Finally, surrogate models naturally support warm-starting by incorporating previously collected data, and provide an additional mechanism for injecting prior knowledge through the model prior. A recent framework for comparing surrogates is proposed by [72].

Different surrogate models are considered here: Gaussian Process in Section 5.4.2, tree-based models in Section 5.4.3, and neural models in Section 5.4.4.

5.4.2 Gaussian Process Models

Gaussian Processes (GPs) constitute the canonical surrogate model in Bayesian Optimization, due to their probabilistic formulation, analytical tractability, and flexibility. A GP defines a distribution over functions such that, for any finite set of inputs, the corresponding outputs follow a multivariate Gaussian distribution,

$$Y(x) \sim \mathcal{N}\big(m(x), c(x, x')\big),$$

where the mean function $m(x) \equiv \mathbb{E}Y(x)$ encodes prior beliefs about the function values and the covariance function or kernel $c(x, x') \equiv \mathrm{Cov}(Y(x), Y(x'))$ specifies assumptions about smoothness, correlation structure, and uncertainty.

In practice, most modeling expressiveness arises from the covariance function, which depends only on the inputs and must be positive definite. Its diagonal elements $c(x, x)$ yield uncertainty estimates on the predictions, which are central to acquisition-driven exploration. **Kernel design** [87, 25], therefore plays a fundamental role in GP-based BO, allowing the incorporation of structural assumptions such as stationarity, smoothness, periodicity [24], or compositional structure [23, 32]. Despite the flexibility offered by kernel design, the current practice still largely relies on transformations that reduce modeling to stationary kernels.

A key strength of Gaussian Processes lies in their ability to **accommodate complex input spaces** through kernel construction. This is particularly relevant when optimizing over design spaces where the classical Euclidean distances are not meaningful, e.g., when tuning hyperparameters of Machine Learning algorithms.

Kernels have been specifically defined for **categorical variables** [38], whether qualitative or quantitative [94]. Among the principles used to design these kernels, it seems important to learn the underlying group structure [71, 67]. Embedding in latent (continuous) spaces are, of course, also a key to handling categorical variables [93, 18].

Hierarchical/conditional input spaces [4], which may also be referred to as a variable-size spaces [66], or as hierarchical domains [40] have also their own kernels. This is an important class of search space $\mathcal{X}$ because it describes the numerous, practical situations where specific sub-systems can be added or removed, each coming with its own parameters. The algebraic distance introduced in [75] and the resulting kernel known as the Alg-Kernel, is symmetric positive definite and available in the opensource SMT framework[3]. This distance is a particular case of the more general hierarchical distance introduced in [40].

In fact, kernels over all types of possible discrete mathematical objects seem to have been defined: **kernels over sets** [31, 80], **over combinatorial variables** [12, 3], **over graphs** [68], over combinatorial variables seen as graphs [63], over object or label preferences [5] ...

Mixed continuous/discrete spaces can be handled by combining continuous/discrete kernels through any kernel combination technique [74]. What is important is that the combinations yields a valid, positive-definite, kernel. Most often, this is achieved by kernel product [18]. Adding kernels is also possible [22].

The performance of GPs with standard kernels deteriorates as **input dimensionality** increases. In high-dimensional spaces, the distances between points become nearly identical (a phenomenon referred to as distance concentration), leading to posterior means and variances that are approximately the same throughout the search space. consequently, the surrogate can no longer meaningfully guide the acquisition function during the search. Moreover, identifiability issues arise, as different kernel parameterizations can produce practically equal likelihoods or posteriors. Recent work addresses this challenge through projection-based kernels [9], active subspaces, and dimension-adaptive modeling strategies [7, 14, 34]. In the context of BO, these approaches are often tightly coupled with trust-region or local modeling assumptions.

Classical GP inference scales cubically with the number of observations, which used to limit the applicability of GPs to **small-data** regimes. In recent research, a wide range of scalable approximations has been developed to accommodate **large data** regimes, including the Vecchia approximation, sparse inducing-point methods, variational formulations, and

[3] `https://smt.readthedocs.io/en/latest/_src_docs/applications/Mixed_Hier_usage.html`

local approximations [45, 85, 13]. Local and partitioned GP models, such as treed or neighborhood-based approaches, further improve scalability while retaining uncertainty estimates [36].

Many real-world optimization problems exhibit **nonstationary output** behavior, both in function values and noise characteristics. Nonstationary kernels, input warping, and locally adaptive GPs provide mechanisms to relax stationarity assumptions [64, 79, 62, 73]. Gaussian processes can also model **heteroscedastic noise** by allowing the observation variance to vary across the input space [33, 82, 6]; yet, this usually requires more data. It also increases inference complexity and often requires a more careful prior specification.

An important advantage of GPs is their ability to incorporate **expert knowledge** through constraints on the output space. Inequality constraints, boundedness, monotonicity, and convexity can be enforced either approximately or exactly through kernel design or posterior conditioning [19, 55]. Physics-informed Gaussian Processes explicitly encode governing equations, such as partial differential equations, as soft or hard constraints on the function space [69, 15, 54].

GPs benefit from their strong theoretical link with Reproducible Kernel Hilbert Spaces (RKHS). Once the GP's parameters have been estimated, the results (e.g., variance, lengthscales) have a **straightforward interpretation**. In the case of **extrapolation** (away from observed data), predictions will naturally revert to the prior (and are thus strongly influenced by the chosen kernel). Recent work has explored kernels that induce structured extrapolation, such as spectral and Brownian motion–based constructions [88, 92, 52].

GPs also allow the analyst to exploit **derivative information**, by analytically differentiating the posterior. Gradient-enhanced GP models can significantly improve sample efficiency in BO, especially in smooth optimization landscapes [83, 89, 44, 21, 26], and when the function is deterministic. When the output data are noisy, or very sparse, though, the estimated value of the derivative may exhibit high uncertainty. Many BO problems involve **multiple correlated outputs** (e.g., in the case of multi-fidelity evaluations, multi-task objectives, or vector-valued responses). Multi-output Gaussian Processes provide a principled framework for modeling such dependencies through structured covariance functions [76, 16, 57], which can substantially improve the sample efficiency during the optimization. The downside is an increase in modeling/computational complexity. Designing interpretable cross-covariance structures and scaling inference to realistic problem sizes remain open challenges, particularly when combined with other extensions such as nonstationarity or constraints.

5.4.3 Tree-based surrogates and other local surrogates

The scope of these models includes hierarchical treed-models, tree ensembles such as Random Forests (RF) and Bayesian Additive Regression Trees (BART), and other local models [36, 27]. In what follows, we zoom in mainly on tree-based surrogates. These can be split up in two subtypes: **Treed-Models** and **Tree Ensembles**. Treed-Models use trees for recursive search space partitioning. This often leads to simple tree structures, with possibly complex (local) models (e.g., local GPs) fit to the data in each leaf [38, 37]. This approach is highly effective for handling mixed and conditional search spaces. Tree Ensembles, by contrast, represent the objective function as a weighted sum or average of trees with simple leaves. Examples include Random Forests (which are similar to bagging) and BART (similar to boosting) [17].

The primary benefits of tree-based surrogates stem from their "divide and conquer" nature. They are computationally efficient even in the case of **high dimensions** and a **large data**, as complexity adaptively scales with training set size [48]. They are fit for modeling **non-stationary and heteroscedastic output** functions (though, in the case of standard

RFs, heteroscedasticity requires requires estimating separate trees for the noise variance [17]). They are highly flexible w.r.t. the **input types** and the **structure of the input space**: they do not require any particular assumptions w.r.t. the nature of the input variables, can naturally accomodate hierarchical search spaces, and constraints on the inputs can be easily included. Moreover, ensembles like BART automatically provide **regularization** through an implicit application of Occam's razor [17]. The practical implementation of the models is supported by various **toolboxes** across Python and R for BART, RF, Treed-GP, and local GP methods (including trust region approaches like TuRBO, [27]).

Nevertheless, tree-based models also present a number of downsides. They are known for their tendency to **overfit**, in the case of **small data** the resulting trees may be overly simple. They also **lack gradient information**, and do not leverage any spatial information across the leaves in the uncertainty estimates and predictions.

The performance of treed-models, when used in the BO loop, fully depends on the type of surrogate used to fit the data in the leaves. Modeling **multiple outputs** may require multiple trees, unless you have MO surrogate models in the leaves. **Acquisition** is computationally expensive, as it is performed per leaf, after which the highest value among the leaves is chosen. For ensembles, inference is often performed via Monte Carlo, and acquisition is usually done using candidate sets. Ensembles do not easily allow for the inclusion of **expert knowledge**.

5.4.4 Neural Models

We discussed a broad class of surrogate models based on neural networks for predicting unseen function values in Bayesian optimization. This category encompasses a wide range of approaches with very different characteristics. Classical methods rely on training Bayesian neural networks [1, 65] (BNNs) directly on the observed data within the BO loop, while more recent approaches leverage pretrained models, such as prior-data fitted networks [59] (PFNs) or large language models (LLMs), that perform Bayesian-style prediction via in-context learning, i.e., observations are an input used to condition the pretrained model.

While diverse in their realizations, neural surrogate models share several common traits. Compared to Gaussian processes, they offer **richer functional priors**, **non-stationary output scales**, and **heteroscedastic noise modeling** without explicit kernel design. Relative to tree-based models, they provide **smooth predictive surfaces** and **efficient input gradients**, which are for example useful for gradient-based acquisition function optimization. However, these methods also share notable drawbacks, including **training instabilities**, **limited interpretability**, and **limited integration is existing toolkits**, compared to more classical surrogates.

Bayesian Neural Networks (BNNs)

BNNs [1, 65] represent the integration of Bayesian inference with neural networks, where prior distributions are placed over network components (typically the parameters) and Bayesian inference is performed to quantify epistemic uncertainty [60, 56]. Given a prior $p(\mathbf{w})$ over weights and observed data $\mathcal{D}$, the posterior $p(\mathbf{w}|\mathcal{D})$ induces a predictive distribution $p(y|\mathbf{x}, \mathcal{D}) = \int p(y|\mathbf{x}, \mathbf{w})p(\mathbf{w}|\mathcal{D})\, d\mathbf{w}$ that naturally provides uncertainty quantification. Since exact inference is intractable, various approximation schemes have been proposed, including Markov Chain Monte Carlo (MCMC) methods [60], variational inference [39, 8], Laplace approximations [56, 20], and partial Bayesian methods that place priors only on a subset of parameters [77], such as last-layer approaches [78, 43, 11].

Compared to Gaussian Processes, BNNs offer the ability to **model non-stationary functions**, learn **problem-specific similarity metrics**, and **scale to large observation sets** where GP cubic complexity becomes prohibitive. BNNs have been explored as surrogate models in BO [81, 78], with recent work demonstrating competitive performance with GPs on standard benchmarks [53, 11]. However, BNNs present their own caveats typically stemming from **approximate inference**, and empirically their performance depends on architectural choices and prior specification, **requiring careful tuning** to achieve well-calibrated uncertainty quantification, and to avoid **overfitting in low data regimes** [86, 28, 53].

Prior-data Fitted Networks (PFNs)

PFNs [59] are models pretrained to do in-context Bayesian prediction for a specific function prior $\Pi = p(f)$. Note that PFNs are closely related (can be seen as a special instance of) conditional Neural processes [30, 61]. Typically, a PFN is a transformer neural network [84], pretrained on synthetic data by repeatedly sampling a function $f' \sim \Pi(f)$, picking "train" and "test" inputs (X_{train}, X_{test}) (often uniform random noise), generating labels $Y_{train} = f'(X_{train})$ (optionally with observation noise), and minimizing expected cross-entropy of the predictive distribution over queries $x_{test} \in X_{test}$ conditioned on context $C =(X_{train}, Y_{train})$. Note that a PFNs output a bar distribution over finely discretized output range, where the PFN has an output per bar, corresponding to the probability that y_{test} falls in the associated interval. Under risk minimization, the solution matches the Bayesian posterior predictive $p(y_{test} \mid x_{test}, C)$ induced by Π, so PFNs learn Bayesian one-shot prediction in a single forward pass. For example, a PFN pretrained on samples of a GP prior, provably approximates GP prediction in a single forward pass. The training and usage of PFNs are illustrated in Figure 8. While TabPFN [47, 46], a tabular foundation model, arguably remains the most prominent application of this paradigm, PFNs have already been explored in the context of BO. Müller *et al.* [58] first proposed PFNs as a surrogate for black box BO in 2023. Recent work has explored PFN surrogates in the context of Freeze-thaw BO [70, 51] and high-dimensional BO [91]. Recently, PFNs have been integrated in the popular BoTorch framework [2].

In contrast to BNNs, PFNs are **not retrained on observed function values**, but amortize Bayesian inference through pretraining on synthetic tasks, yielding **strong performance with very few observations**. This makes them attractive for Bayesian optimization, where **posterior updates reduce to a single forward pass**, avoiding retraining, hyperparameter optimization, and associated instabilities. PFNs also naturally support **mixed input spaces** including continuous, discrete, and categorical variables, as demonstrated by TabPFN on heterogeneous tabular data, which is a practical advantage for BO with structured or conditional search spaces. At the same time, current PFN transformer architectures rely on dense attention, leading to **quadratic compute and memory growth** and making **very large contexts computationally demanding** compared to tree-based surrogates. Another trade-off is that PFNs offer **explicit control over the functional prior**, but **changing this prior requires pretraining from scratch**. As with any learned surrogate, **strong out-of-distribution mismatch may lead to instability**, although empirical performance is **often reasonable**. Overall, PFN-based Bayesian optimization remains **an emerging and experimental paradigm**, with several modeling and design choices still under active investigation.

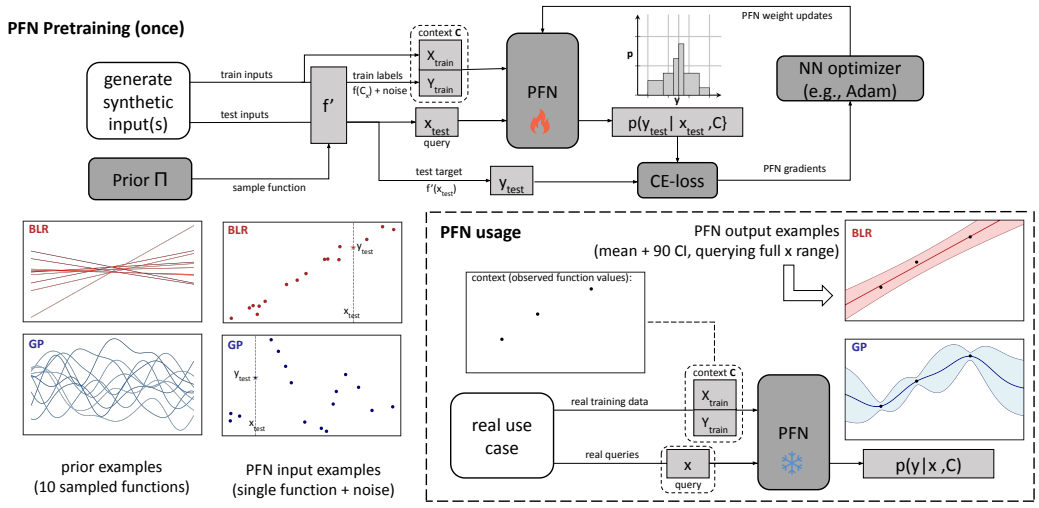


Figure 8 Illustration of the training and usage of prior-data fitted networks (PFNs), with specific in-/output examples fitting a Bayesian Linear Regression (BLR) and Gaussian Process (GP) prior.

Large Language Models (LLMs)

LLMs are general-purpose neural networks trained on massive text corpora to model and generate natural language, and have emerged as a versatile foundation for a wide range of reasoning, prediction, and decision-making tasks [10]. They can be employed for Bayesian optimization in many different ways, ranging from proposing candidate points [90] to guiding search through natural-language feedback [49]; these broader uses are discussed in the GenAI for BO working group (Section 5.5). In our discussions, we focused on a deliberately narrow and directly comparable setting, where the LLM is used *as a surrogate model* via in-context learning: previously observed input–output pairs (x, y) are provided in the prompt, a new candidate x_{test} is queried, and the model predicts the corresponding outcome y_{test}. While this setup is natural, it may not the most practical way to deploy LLMs in BO, and we are not aware of prior work considering this exact formulation, it enables a clean conceptual contrast with PFNs by isolating in-context prediction with frozen weights.

In this setting, LLMs share some appealing properties with PFNs, most notably **data efficiency via in-context learning**, allowing them to adapt from very few observations without retraining. Beyond this, LLMs offer distinctive advantages: they can leverage **broad world knowledge**, support flexible textual input and output, and are **easy to condition on auxiliary information**, enabling predictions to be augmented with explanations, constraints, or meta-information that go beyond scalar regression targets. At the same time, these strengths come with substantial drawbacks when used as BO surrogates. LLMs are typically **large and computationally expensive**, making repeated surrogate queries costly. Their inductive biases are **implicit and hard to control**, limiting the ability to align model behavior with a desired functional prior. Moreover, predictions can be **sensitive to prompt formulation and syntax**, introducing brittleness and requiring careful prompt tuning. As a result, while LLMs provide a flexible and expressive baseline for in-context surrogate modeling, their use in this direct surrogate role is best viewed as **experimental** and primarily valuable for comparison and exploration rather than as a default practical choice.

5.4.4.1 Conclusions

The discussions in this working group underscored that surrogate modeling remains a central and evolving design choice in Bayesian optimization rather than a settled component. Gaussian processes continue to be the default surrogate model due to their strong theoretical foundations, flexibility, and principled uncertainty estimates, but they also exhibit limitations in scalability, representational capacity, and prior specification, which are often addressed through increasingly complex extensions. This has motivated interest in alternative surrogate models, including neural approaches. While traditional in-weight learning methods such as Bayesian neural networks have so far had limited impact in BO, possibly due to the structured, tabular nature of BO data, the low-data regime, and the need for frequent model updates, settings in which deep learning has historically been less competitive. More recent approaches based on in-context learning, including prior-data fitted networks and large language models, offer a qualitatively different perspective by shifting weight learning to a pretraining phase and avoiding retraining during optimization. These models have shown strong performance in low-data and tabular settings and introduce new capabilities, such as explicit control over prior assumptions (PFNs) or access to broad world knowledge and flexible input–output interfaces (LLMs). Finally, we emphasized that surrogate models should not be considered in isolation, as their effectiveness depends on interactions with other components of the Bayesian optimization pipeline, and that model fit and prior mismatch remain underexplored issues where improved evaluation practices could yield deeper insight.

References

1 Julyan Arbel, Konstantinos Pitas, Mariia Vladimirova, and Vincent Fortuin. A primer on bayesian neural networks: Review and debates. *arXiv preprint arXiv:2309.16314*, 2023.

2 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.

3 Ricardo Baptista and Matthias Poloczek. Bayesian optimization of combinatorial structures. In *International conference on machine learning*, pages 462–471. PMLR, 2018.

4 Lucas Baraton, Annafederica Urbano, Loïc Brevault, and Mathieu Balesdent. Bayesian quality-diversity optimization for conditional search-space problems: L. baraton et al. *Optimization and Engineering*, pages 1–46, 2025.

5 Alessio Benavoli and Dario Azzimonti. A tutorial on learning from preferences and choices with gaussian processes. *arXiv preprint arXiv:2403.11782*, 2024.

6 Mickael Binois, Robert B Gramacy, and Mike Ludkovski. Practical heteroscedastic gaussian process modeling for large simulation experiments. *Journal of Computational and Graphical Statistics*, 27(4):808–821, 2018.

7 Mickael Binois and Nathan Wycoff. A survey on high-dimensional Gaussian process modeling with application to Bayesian optimization. *ACM Transactions on Evolutionary Learning and Optimization*, 2(2):1–26, 2022.

8 Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR, 2015.

9 Mohamed Amine Bouhlel, Nathalie Bartoli, Abdelkader Otsmane, and Joseph Morlier. Improving kriging surrogates of high-dimensional design models by partial least squares dimension reduction. *Structural and Multidisciplinary Optimization*, 53(5):935–952, 2016.

10 Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

11 Paul Brunzema, Mikkel Jordahn, John Willes, Sebastian Trimpe, Jasper Snoek, and James Harrison. Bayesian optimization via continual variational last layer training. In *The Thirteenth International Conference on Learning Representations*, 2025.

12 Poompol Buathong, David Ginsbourger, and Tipaluck Krityakierne. Kernels over sets of finite sets using rkhs embeddings, with application to bayesian (combinatorial) optimization. In *International conference on artificial intelligence and statistics*, pages 2731–2741. PMLR, 2020.

13 Jian Cao, Joseph Guinness, Marc G Genton, and Matthias Katzfuss. Scalable gaussian-process regression and variable selection using vecchia approximations. *The Journal of Machine Learning Research*, 23(1):15799–15828, 2022.

14 Jialei Chen, Zhehui Chen, Chuck Zhang, and CF Jeff Wu. Apik: Active physics-informed kriging model with partial differential equations. *SIAM/ASA Journal on Uncertainty Quantification*, 10(1):481–506, 2022.

15 Yifan Chen, Bamdad Hosseini, Houman Owhadi, and Andrew M Stuart. Solving and learning nonlinear pdes with gaussian processes. *Journal of Computational Physics*, 447:110668, 2021.

16 Zexun Chen, Bo Wang, and Alexander N Gorban. Multivariate gaussian and student-t process regression for multi-output prediction. *Neural Computing and Applications*, 32(8):3005–3028, 2020.

17 Hugh A Chipman, Edward I George, and Robert E McCulloch. Bart: Bayesian additive regression trees. *The Annals of Applied Statistics*, pages 266–298, 2010.

18 Jhouben Cuesta Ramirez, Rodolphe Le Riche, Olivier Roustant, Guillaume Perrin, Cedric Durantin, and Alain Gliere. A comparison of mixed-variables bayesian optimization approaches. *Advanced Modeling and Simulation in Engineering Sciences*, 9(1):6, 2022.

19 Sébastien Da Veiga and Amandine Marrel. Gaussian process modeling with inequality constraints. In *Annales de la faculté des sciences de Toulouse*, volume 21, pages 529–555, 2012.

20 Erik Daxberger, Agustinus Kristiadi, Alexander Immer, Runa Eschenhagen, Matthias Bauer, and Philipp Hennig. Laplace redux-effortless bayesian deep learning. *Advances in neural information processing systems*, 34:20089–20103, 2021.

21 Filip de Roos, Alexandra Gessner, and Philipp Hennig. High-dimensional gaussian process inference with derivatives. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 2535–2545. PMLR, 18–24 Jul 2021.

22 Nicolas Durrande, David Ginsbourger, and Olivier Roustant. Additive covariance kernels for high-dimensional gaussian process modeling. In *Annales de la Faculté des sciences de Toulouse: Mathématiques*, volume 21, pages 481–499, 2012.

23 Nicolas Durrande, David Ginsbourger, Olivier Roustant, and Laurent Carraro. Anova kernels and rkhs of zero mean functions for model-based sensitivity analysis. *Journal of Multivariate Analysis*, 115:57–67, 2013.

24 Nicolas Durrande, James Hensman, Magnus Rattray, and Neil D Lawrence. Detecting periodicities with gaussian processes. *PeerJ Computer Science*, 2:e50, 2016.

25 David Duvenaud. *Automatic model construction with Gaussian processes*. PhD thesis, University of Cambridge, 2014.

26 David Eriksson, Kun Dong, Eric Lee, David Bindel, and Andrew G Wilson. Scaling gaussian process regression with derivatives. In *Advances in Neural Information Processing Systems*, pages 6866–6876, 2018.

27 David Eriksson, Michael Pearce, Jacob Gardner, Ryan P Turner, and Matthias Poloczek. Scalable global optimization via local bayesian optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

28 Jonathan Foldager, Mikkel Jordahn, Lars K Hansen, and Michael R Andersen. On the role of model uncertainties in bayesian optimisation. In *Uncertainty in Artificial Intelligence*, pages 592–601. PMLR, 2023.

29 Alexander Forrester, Andras Sobester, and Andy Keane. *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons, 2008.

30 Marta Garnelo, Dan Rosenbaum, Christopher Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo Rezende, and SM Ali Eslami. Conditional neural processes. In *International conference on machine learning*, pages 1704–1713. PMLR, 2018.

31 Roman Garnett, Michael A Osborne, and Stephen J Roberts. Bayesian optimization for sensor set selection. In *Proceedings of the 9th ACM/IEEE international conference on information processing in sensor networks*, pages 209–219, 2010.

32 David Ginsbourger, Xavier Bay, Olivier Roustant, and Laurent Carraro. Argumentwise invariant kernels for the approximation of invariant functions. In *Annales de la Faculté des sciences de Toulouse: Mathématiques*, volume 21, pages 501–527, 2012.

33 Paul W Goldberg, Christopher KI Williams, and Christopher M Bishop. Regression with input-dependent noise: A gaussian process treatment. *Advances in neural information processing systems*, pages 493–499, 1998.

34 Miguel González-Duque, Richard Michael, Simon Bartels, Yevgen Zainchkovskyy, Søren Hauberg, and Wouter Boomsma. A survey and benchmark of high-dimensional bayesian optimization of discrete sequences. *Advances in Neural Information Processing Systems*, 37:140478–140508, 2024.

35 Robert B Gramacy. *Surrogates: Gaussian process modeling, design, and optimization for the applied sciences*. Chapman and Hall/CRC, 2020.

36 Robert B Gramacy and Daniel W Apley. Local gaussian process approximation for large computer experiments. *Journal of Computational and Graphical Statistics*, 24(2):561–578, 2015.

37 Robert B Gramacy and Herbert KH Lee. Bayesian treed gaussian process models with optimization applications. *Journal of Computational and Graphical Statistics*, 17(4):964–980, 2008.

38 Robert B Gramacy and Matthew Taddy. Categorical inputs, sensitivity analysis, optimization and importance tempering with tgp version 2, an R package for treed gaussian process models. *Journal of Statistical Software*, 33(1):1–48, 2010.

39 Alex Graves. Practical variational inference for neural networks. *Advances in neural information processing systems*, 24, 2011.

40 Edward Hallé-Hannan, Charles Audet, Youssef Diouane, Sébastien Le Digabel, and Paul Saves. A distance for mixed-variable and hierarchical domains with meta variables. *Neurocomputing*, 653:131208, 2025.

41 Nikolaus Hansen. The CMA evolution strategy: A tutorial. *arXiv preprint arXiv:1604.00772*, 2016.

42 Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evol. Comput.*, 9(2):159–195, June 2001.

43 James Harrison, John Willes, and Jasper Snoek. Variational bayesian last layers. In *The Twelfth International Conference on Learning Representations*, 2024.

44 Xu He and Peter Chien. On the instability issue of gradient-enhanced gaussian process emulators for computer experiments. *SIAM/ASA Journal on Uncertainty Quantification*, 6(2):627–644, 2018.

45 James Hensman, Alexander G Matthews, Maurizio Filippone, and Zoubin Ghahramani. Mcmc for variationally sparse gaussian processes. *Advances in neural information processing systems*, 28, 2015.

46 Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025.

47 Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. Tabpfn: A transformer that solves small tabular classification problems in a second. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023.

48 Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *LION*, 2011.

49 Katarzyna Kobalczyk, Zhiyuan Jerry Lin, Benjamin Letham, Zhuokai Zhao, Maximilian Balandat, and Eytan Bakshy. Lilo: Bayesian optimization with interactive natural language feedback. *arXiv preprint arXiv:2510.17671*, 2025.

50 Rodolphe Le Riche and Nicolas Durrande. An overview of kriging for researchers. Summer school on Modeling and Numerical Methods for Uncertainty Quantification, slides as HAL course cel-02285439, `https://hal.science/cel-02285439`, September 2019.

51 Dong Bok Lee, Aoxuan Silvia Zhang, Byungjoo Kim, Junhyeon Park, Steven Adriaensen, Juho Lee, Sung Ju Hwang, and Hae Beom Lee. Cost-sensitive freeze-thaw bayesian optimization for efficient hyperparameter tuning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

52 Minyong R Lee and Art B Owen. Single nugget kriging. *Statistica Sinica*, pages 649–669, 2018.

53 Yucen Lily Li, Tim G. J. Rudner, and Andrew Gordon Wilson. A study of bayesian neural network surrogates for bayesian optimization. In *The Twelfth International Conference on Learning Representations*, 2024.

54 Da Long, Zheng Wang, Aditi Krishnapriyan, Robert Kirby, Shandian Zhe, and Michael Mahoney. Autoip: A united framework to integrate physics into gaussian processes. In *International Conference on Machine Learning*, pages 14210–14222. PMLR, 2022.

55 Andrés F López-Lopera, François Bachoc, Nicolas Durrande, and Olivier Roustant. Finite-dimensional gaussian approximation with linear inequality constraints. *SIAM/ASA Journal on Uncertainty Quantification*, 6(3):1224–1255, 2018.

56 David J. C. MacKay. A practical bayesian framework for backpropagation networks. *Neural Computation*, 4(3):448–472, 05 1992.

57 Wesley J Maddox, Maximilian Balandat, Andrew G Wilson, and Eytan Bakshy. Bayesian optimization with high-dimensional outputs. *Advances in Neural Information Processing Systems*, 34:19274–19287, 2021.

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

59 Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do bayesian inference. In *International Conference on Learning Representations*, 2022.

60 Radford M. Neal. *Bayesian Learning for Neural Networks*. Springer-Verlag, Berlin, Heidelberg, 1996.

61 Tung Nguyen and Aditya Grover. Transformer neural processes: Uncertainty-aware meta learning via sequence modeling. In *International Conference on Machine Learning*, pages 16569–16594. PMLR, 2022.

62 Marcus M Noack and James A Sethian. Advanced stationary and nonstationary kernel designs for domain-aware gaussian processes. *Communications in Applied Mathematics and Computational Science*, 17(1):131–156, 2022.

63 Changyong Oh, J Tomczak, Efstratios Gavves, and Max Welling. Combo: Combinatorial bayesian optimization using graph representations. In *ICML Workshop on Learning and Reasoning with Graph-Structured Data*, 2019.

64 Christopher J Paciorek and Mark J Schervish. Spatial modelling using a new class of nonstationary covariance functions. *Environmetrics: The official journal of the International Environmetrics Society*, 17(5):483–506, 2006.

65 Theodore Papamarkou, Maria Skoularidou, Konstantina Palla, Laurence Aitchison, Julyan Arbel, David Dunson, Maurizio Filippone, Vincent Fortuin, Philipp Hennig, José Miguel Hernández-Lobato, et al. Position: Bayesian deep learning is needed in the age of large-scale ai. In *International Conference on Machine Learning*, 2024.

66 Julien Pelamatti, Loic Brevault, Mathieu Balesdent, El-Ghazali Talbi, and Yannick Guerin. Bayesian optimization of variable-size design space problems. *Optimization and Engineering*, 22:387–447, 2021.

67 Raphaël Carpintero Perez, Sébastien Da Veiga, and Josselin Garnier. A reproducible comparative study of categorical kernels for gaussian process regression, with new clustering-based nested kernels. *arXiv preprint arXiv:2510.01840*, 2025.

68 Raphaël Carpintero Perez, Sébastien Da Veiga, Josselin Garnier, and Brian Staber. Gaussian process regression with sliced wasserstein weisfeiler-lehman graph kernels. In *International Conference on Artificial Intelligence and Statistics*, pages 1297–1305. PMLR, 2024.

69 Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Numerical gaussian processes for time-dependent and nonlinear partial differential equations. *SIAM Journal on Scientific Computing*, 40(1):A172–A198, 2018.

70 Herilalaina Rakotoarison, Steven Adriaensen, Neeratyoy Mallik, Samir Garibov, Eddie Bergman, and Frank Hutter. In-context freeze-thaw bayesian optimization for hyperparameter optimization. In *International Conference on Machine Learning*, pages 41982–42008. PMLR, 2024.

71 Olivier Roustant, Esperan Padonou, Yves Deville, Aloïs Clément, Guillaume Perrin, Jean Giorla, and Henry Wynn. Group kernels for gaussian process metamodels with categorical inputs. *SIAM/ASA Journal on Uncertainty Quantification*, 8(2):775–806, 2020.

72 Kellin N Rumsey, Graham C Gibson, Devin Francom, and Reid Morris. All emulators are wrong, many are useful, and some are more useful than others: A reproducible comparison of computer model surrogates. *arXiv preprint arXiv:2512.09060*, 2025.

73 Annie Sauer, Andrew Cooper, and Robert B Gramacy. Non-stationary gaussian process surrogates. *arXiv preprint arXiv:2305.19242*, 2023.

74 Paul Saves, Youssef Diouane, Nathalie Bartoli, Thierry Lefebvre, and Joseph Morlier. A mixed-categorical correlation kernel for gaussian process. *Neurocomputing*, 550:126472, 2023.

75 Paul Saves, Rémi Lafage, Nathalie Bartoli, Youssef Diouane, Jasper H. Bussemaker, Thierry Lefebvre, John. T. Hwang, Joseph Morlier, and Joaquim R. R. A. Martins. SMT 2.0: A surrogate modeling toolbox with a focus on hierarchical and mixed variables Gaussian processes. *Advances in Engineering Sofware*, 188:1–15, 2024.

76 Amar Shah, AC UK, Zoubin Ghahramani, and CAM AC. Pareto frontier learning with expensive correlated objectives. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 1919–1927, 2016.

77 Mrinank Sharma, Sebastian Farquhar, Eric Nalisnick, and Tom Rainforth. Do bayesian neural networks need to be fully stochastic? In *International Conference on Artificial Intelligence and Statistics*, pages 7694–7722. PMLR, 2023.

78 Jasper Snoek, Oren Rippel, Kevin Swersky, Ryan Kiros, Nadathur Satish, Narayanan Sundaram, Mostofa Patwary, Mr Prabhat, and Ryan Adams. Scalable bayesian optimization using deep neural networks. In *International conference on machine learning*, pages 2171–2180. PMLR, 2015.

79 Jasper Snoek, Kevin Swersky, Richard S Zemel, and Ryan P Adams. Input warping for bayesian optimization of non-stationary functions. In *International Conference on Machine Learning*, pages 1674–1682, 2014.

80 Babacar Sow, Rodolphe Le Riche, Julien Pelamatti, Merlin Keller, and Sanaa Zannane. Optimization and metamodeling of functions defined over clouds of points. In *workshop SAMOURAI*, Paris – FR, December 2024. réseau thématique Uncertainty Quantification (RT UQ). HAL document hal-04830176, `https://hal.science/hal-04830176`.

81 Jost Tobias Springenberg, Aaron Klein, Stefan Falkner, and Frank Hutter. Bayesian optimization with robust bayesian neural networks. *Advances in neural information processing systems*, 29, 2016.

82 Michalis K Titsias and Miguel Lázaro-Gredilla. Variational heteroscedastic gaussian process regression. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 841–848, 2011.

83 Selvakumar Ulaganathan, Ivo Couckuyt, Tom Dhaene, Joris Degroote, and Eric Laermans. High dimensional kriging metamodelling utilising gradient information. *Applied Mathematical Modelling*, 40(9-10):5256–5270, 2016.

84 Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

85 Ke Wang, Geoff Pleiss, Jacob Gardner, Stephen Tyree, Kilian Q Weinberger, and Andrew Gordon Wilson. Exact gaussian processes on a million data points. *Advances in neural information processing systems*, 32, 2019.

86 Florian Wenzel, Kevin Roth, Bastiaan S. Veeling, Jakub Świątkowski, Linh Tran, Stephan Mandt, Jasper Snoek, Tim Salimans, Rodolphe Jenatton, and Sebastian Nowozin. How good is the bayes posterior in deep neural networks really? In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

87 Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

88 Andrew Wilson and Ryan Adams. Gaussian process kernels for pattern discovery and extrapolation. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 1067–1075, 2013.

89 Anqi Wu, Mikio C Aoi, and Jonathan W Pillow. Exploiting gradients and hessians in bayesian optimization and bayesian quadrature. *arXiv preprint arXiv:1704.00060*, 2017.

90 Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. In *The Twelfth International Conference on Learning Representations*, 2023.

91 Rosen Ting-Ying Yu, Cyril Picard, and Faez Ahmed. Git-bo: High-dimensional bayesian optimization with tabular foundation models. *arXiv preprint arXiv:2505.20685*, 2025.

92 Ning Zhang and Daniel W Apley. Brownian integrated covariance functions for gaussian process modeling: Sigmoidal versus localized basis functions. *Journal of the American Statistical Association*, 111(515):1182–1195, 2016.

93 Yichi Zhang, Siyu Tao, Wei Chen, and Daniel W Apley. A latent variable approach to gaussian process modeling with qualitative and quantitative factors. *Technometrics*, 62(3):291–302, 2020.

94 Qiang Zhou, Peter ZG Qian, and Shiyu Zhou. A simple approach to emulation for computer models with qualitative and quantitative factors. *Technometrics*, 53(3), 2011.

5.5 GenAI and Bayesian Optimization

Raul Astudillo (California Institute of Technology – Pasadena, US)
Gautam Dasarathy (Arizona State University – Tempe, US)
Peter Frazier (Cornell University – Ithaca, US)
Daolang Huang (Aalto University, FI)
Bryan Kian Hsiang Low (National University of Singapore, SG)
Giulia Pedrielli (Arizona State University – Tempe, US)
Matthias Poloczek (Amazon.com – San Francisco, US)
Elena Raponi (Leiden University, NL)

5.5.1 Overview

Working Group 5 explored emerging intersections between generative AI (GenAI) and Bayesian optimization (BO). Our discussions converged on four overarching themes:

1. **LLM-based synthetic task generation** to improve meta-BO, mitigate prior misspecification, and broaden the distribution of training tasks (including connections to probabilistic numerics and PFNs).
2. **Bayesian optimization across the LLM training and inference pipeline**, including data mixture optimization, model merging, and inference-time orchestration.
3. **BO guided by generative models** (e.g., diffusion models for protein design and safety verifiers), using learned priors to constrain search to plausible regions while enabling effective exploration.
4. **Rich text-based human feedback for BO**, where LLMs allow algorithms to interpret semantically rich user feedback, greatly increasing information per human interaction.

Together, these directions point toward a long-term vision of a *task-agnostic neural optimization engine*: a foundation model for optimization that can be deployed across scientific and engineering domains, guided by symbolic structure, generative priors, and human intent.

5.5.2 Research Ideas

5.5.2.1 LLM-Based Synthetic Task Generation (for Meta-BO and PFNs)

Contributors

- First-draft writers: Giulia Pedrielli, Daolang Huang
- Other interested parties: Elena Raponi, Raul Astudillo

Motivation. Meta–Bayesian optimization (meta-BO) has shown impressive gains in sample efficiency by learning reusable optimization policies across tasks. However, its success hinges on the quality and breadth of the pre-training task distribution. Most prior work (e.g., NAP, PFN v1, ACE) synthesizes tasks from Gaussian process (GP) priors or structural causal models (SCMs). While convenient, these generators induce two key practical failure modes:

1. **Prior misspecification.** Many real optimization problems differ substantially from GP/SCM assumptions, causing meta-policies to overfit the synthetic pre-training distribution and degrade at deployment.

2. **Real-data scarcity.** In domains such as protein or material design, collecting diverse, well-labeled optimization traces is expensive, making it difficult to correct misspecification with sufficient supervision.

This motivates a principled way to systematically generate better-matched pre-training data. Our idea is to leverage LLMs to generate task priors and simulators, broadening the support of synthetic tasks beyond GP/SCM and targeting high-information training signals aligned with downstream objectives.

Problem Formulation. We aim to define a generative–symbolic pipeline that produces a distribution over optimization tasks

$$T = (\mathcal{X}, f, c, \theta_T),$$

where tasks differ in domain, structure, constraints, and parameters but expose a common query interface. A meta-BO policy

$$\pi_\theta : \mathcal{H}_n \to \mathcal{X}$$

maps a history of queries and outcomes to the next query point. The training objective is to minimize distributional regret:

$$\min_\theta \ \mathbb{E}_{T\sim\mathcal{T}} \left[R(\pi_\theta; T) \right].$$

5.5.2.2 Main Challenges

Symbolic task generation. We identified three macro areas as especially relevant:

1. **Compositional and operator-based modeling.** Symbolic and graph-based equation modeling, code generation, and scientific machine learning (SciML) offer formalisms for composing simulators from reusable operators. Probabilistic programs and category-theoretic approaches provide rigorous semantics for composing open systems as "boxes" with typed ports. These can underpin an operator algebra for task generation. Key challenges include:
   - designing operators with sufficient expressivity while avoiding computational intractability,
   - handling heterogeneous aspects and domains within a common representation.
2. **Equation discovery.** Methods such as SINDy and its variants (with control, boundary-value problems, etc.) discover sparse governing equations from trajectories by selecting terms from a dictionary of candidate operators. These equations can be turned into simulation code directly. Symbolic regression methods (e.g., AI-Feynman, PySR) discover closed-form expressions, with recent work incorporating units and dimensional constraints to maintain physical validity. Such approaches yield human-readable right-hand sides that can be integrated into simulation frameworks like Modelica.
3. **Domain-Specific Language (DSL) synthesis.** DSLs allow LLMs to operate over *operators* (parameterizable objects) rather than raw files, making search and verification easier. Systems such as DreamCoder, which learn and refine a DSL of primitives while solving tasks, exemplify this paradigm. This aligns well with the idea of "teaching the model an operator algebra, then synthesizing programs that compose them."

Meta-BO. Assuming that a symbolic/LLM pipeline can provide a stream of verified simulators and task specifications, the next challenge is to learn meta-BO policies that systematically exploit this rich task distribution:

- **Robust training paradigms.** Pure behavior cloning from expert BO traces is stable but capped by the expert's biases. Pure RL on regret or feasibility objectives is well aligned but unstable and expensive. A key challenge is to combine them: pre-train on high-quality trajectories and then fine-tune with regret-, feasibility-, or level-set–aligned objectives.
- **Architectures for heterogeneous histories.** The policy must handle variable input dimensions, constraints, multi-objective settings, and tasks where success includes not only finding the global optimum but also locating feasible points, reaching target level-sets, or covering a Pareto front.
- **Alternative utilities beyond the optimal value.** Practitioners often care about feasibility rate, time-to-feasibility, level-set coverage, or cost-aware regret. The challenge is to define unified, learnable utilities and surrogates that make meta-training stable while reflecting these practical objectives.
- **Robustness to generator imperfections.** Even with verification, the task generator may be biased. The meta-BO model should not exploit artifacts of the generator but instead learn algorithms that transfer to real-world problems.

Pipeline. At a high level, the system consists of three main stages:

1. **Symbolic proposal via LLMs.** LLMs propose task specifications, simulators, and operator compositions within a DSL.
2. **Verification and compilation.** Candidate simulators are checked for consistency (e.g., units, dimensions), tested via automatic unit tests, and compiled into executable form.
3. **Meta-BO training.** A meta-BO policy π_θ is trained on tasks sampled from the verified generator using a combination of imitation learning and utility-aligned optimization to produce decision rules that generalize across tasks.
4. **(Optional) Co-learning.** An additional direction is to close the loop between the task generator and the meta-BO policy. Instead of treating the LLM-based generator as fixed, we can use the performance of the learned policy as feedback to refine the task distribution: identify failure modes, adjust prompts, modify operator choices or parameter ranges, and encourage more diverse and informative tasks.

Outlook: Toward a Task-Agnostic Neural Optimization Engine. Ultimately, the vision underlying this line of work is a task-agnostic neural optimization engine: a foundation model for optimization that can be deployed across domains with minimal adaptation. Given a standardized description of a new task, the model should rapidly propose effective queries without rebuilding a bespoke BO pipeline.

Realizing this vision raises several additional challenges, including:

- defining universal task representations that capture symbolic, statistical, and geometric structure,
- aligning training task distributions with real-world variability,
- transferring operator-based semantics across scientific domains,
- providing safety, interpretability, and robustness guarantees for generated tasks and learned policies.

5.5.2.3 Bayesian Optimization for the LLM Training and Inference Pipeline

Contributors

- First-draft writers: Bryan Low, Gautam Dasarathy
- Other interested parties: Peter Frazier

Motivation. Bayesian optimization (BO) and bandit methods offer a unified lens on several high-impact but expensive decision problems in the large language model (LLM) pipeline:

1. selecting data mixtures for training and adaptation,
2. merging or interpolating pretrained models,
3. directing agentic workflows or routing decisions at inference time.

Each problem involves a costly, noisy black-box objective, a structured decision space, and opportunities for exploiting multi-fidelity or derivative-like information.

Optimizing Data Mixtures. Data composition determines how training samples from different domains are combined under a fixed training protocol. BO constructs a surrogate model of validation performance based on a small number of observed mixtures and selects new candidates via acquisition functions such as expected improvement or knowledge gradient. Multi-fidelity BO integrates cheaper proxies – smaller models, truncated runs, or reduced datasets – with occasional full evaluations.

Recent work suggests that performance as a function of mixture proportions often obeys predictable *data mixing laws* that transfer across scales, enabling nested optimization where low-fidelity surrogates guide higher-fidelity decisions. Derivative-like signals (e.g., sensitivity of validation loss to marginal increases in data buckets) could further accelerate convergence. While full pretraining remains too costly to explore, adaptation and post-training reweighting provide practical arenas for BO-driven mixture optimization.

As an example, [4] introduces DUET, a global-to-local algorithm that combines influence functions for data selection with Bayesian optimization to iteratively refine data mixtures based on feedback from an unseen task. It provides both theoretical convergence guarantees and empirical improvements over competing methods.

Optimizing Model Mixtures. Model mixing or merging seeks to combine the capabilities of separately trained LLMs – whether checkpoints, domain specialists, or fine-tuned variants – into a single unified model. When models share architecture and tokenization, parameters can often be interpolated or aligned directly; when they differ, merging typically proceeds through ensembling or distillation rather than literal weight fusion.

Given a set of models and merge parameters (e.g., interpolation coefficients), the performance of the merged model defines another complex black-box optimization landscape. BO can efficiently explore low-dimensional merge spaces using multi-fidelity evaluations such as limited benchmarks or reduced-context inference. Incorporating similarity features between models, such as representation overlap or calibration distance, into the surrogate may improve sample efficiency. Future directions include BO-guided tuning of structured merges (mixtures-of-experts, layerwise blends) and developing reliable low-cost proxies for alignment or robustness metrics.

Optimizing Agentic Workflows and Routing Policies. As language models are increasingly embedded in multi-step or agentic workflows, a key challenge is deciding which model, prompt, or tool to invoke at each stage. These routing or orchestration policies must balance accuracy, latency, and cost, often under uncertainty about user intent or task difficulty. Evaluating a routing decision typically requires running full interactions or multi-turn rollouts, which are expensive and noisy.

Bayesian optimization can guide the offline design of such policies by exploring high-level configuration choices – such as thresholds, model assignments, or prompt templates – using cheaper approximations like partial rollouts, simulated environments, or reduced-context prompts. Once deployed, online bandit algorithms can adapt these decisions based on real feedback, allocating traffic toward more promising routes while maintaining safety constraints. Together, offline BO for global structure and online bandits for continual adaptation offer a practical path toward efficient and robust agentic control.

As an example of a related perspective, [4] also introduces A-BAD-BO, a BO-based algorithm for optimizing complex AI systems ("AutoAI") by leveraging local loss signals from system components.

Cross-Cutting Themes. The three settings we discussed – data mixing, model merging, and routing – share a common structure. Each involves a search over a large and expensive design space where direct experimentation is costly. In every case, there exist cheaper or approximate sources of information that can guide this search, such as small-scale models, truncated training runs, or limited-context evaluations. These can be organized into a hierarchy of fidelities that BO is naturally equipped to exploit.

Additional signals can also be incorporated into this process. Examples include estimated sensitivities of validation loss to different data sources, measures of similarity between models before merging, and statistics collected from partial rollouts during inference. These signals provide side information that can shape the surrogate model used by BO, improving both efficiency and robustness.

Although these problems are usually treated independently, they are in fact coupled. The data mixture chosen during training affects the diversity and quality of models available for merging. The merged model influences how routing or orchestration policies should behave at inference time. In the opposite direction, observations from deployment can identify where the model's behavior is lacking, which in turn informs how future training data or compute should be allocated.

Viewing the entire training and inference pipeline as a single, hierarchical optimization process makes these dependencies explicit. Information collected at one stage can serve as prior knowledge at another, reducing redundant experimentation. Surrogate models built for data selection may inform later stages such as model merging or routing. This integrated perspective does not yet have a practical implementation, but it points toward a coherent optimization framework in which choices about data, model parameters, and inference strategies are treated as parts of one evolving system.

Outlook. BO's strength lies in sample-efficient search under extreme cost asymmetries. Near-term gains are likely in post-training data reweighting and model merging, where iteration is feasible. Longer-term opportunities include multi-fidelity theory for scaling transfer, gradient-informed surrogates, and unified optimization frameworks that bridge training, adaptation, and inference.

5.5.2.4 BO with Generative Priors for Protein Design, Safety Verifiers, and Beyond

Contributors

- First-draft writers: Raul Astudillo, Elena Raponi
- Other interested parties: Peter Frazier, Gautam Dasarathy

Context. Recent work on steering generative models with experimental data for protein fitness optimization includes, for example, [13] and [14]. These and related papers illustrate how diffusion and other generative models can be used as learned priors in scientific design problems.

Problem Setting. Generative models capture rich, high-dimensional structure from large datasets. In domains such as protein design, they learn distributions over biologically valid sequences, implicitly encoding physical, evolutionary, and functional constraints.

Bayesian optimization (BO), by contrast, is a principled framework for optimizing expensive black-box objectives, balancing exploration and exploitation through probabilistic modeling. Yet, BO lacks a natural mechanism to incorporate the powerful prior knowledge encoded by generative models. This gap motivates a central question: *How can learned generative priors guide Bayesian optimization to search more efficiently while remaining faithful to data-driven structure?*

Let $\mathbb{X}$ denote the input space and $f : \mathbb{X} \to \mathbb{R}$ the objective function to be optimized. We assume access to a generative model that defines a prior distribution $\pi_0(x)$, highlighting regions of $\mathbb{X}$ corresponding to plausible or desirable configurations. Our goal is to identify evaluation points that maximize f while effectively leveraging the structural information encoded in $\pi_0(x)$.

Relation to Prior Work. Given data $\mathcal{D}_n = \{(x_i, y_i)\}_{i=1}^n$, let $\alpha_n : \mathbb{X} \to \mathbb{R}$ denote a user-defined acquisition function. Prior work has explored various approaches to integrate the inductive bias encoded by π_0 with the predictive guidance provided by α_n across different settings.

Hvarfner et al. introduced π-BO [6], scaling the acquisition function by a user-specified prior $\pi(x)$ to encode expert beliefs:

$$\tilde{\alpha}_n(x) = \pi(x)\, \alpha_n(x).$$

Yang et al. [13] explored a related idea in the context of priors encoded by generative models such as diffusion models, yielding a sampling policy

$$x_{n+1} \sim \pi_{n+1} \propto \pi_0(x) \exp\left(\frac{\alpha_n(x)}{\beta}\right), \tag{19}$$

where β balances exploration and adherence to the generative prior.

Opportunities for Exploration

Coupled vs. decoupled formulations. Most existing approaches combine the generative prior π_0 and acquisition function α_n post hoc. A key question is whether these components should instead be co-learned or co-adapted – for example, through conditional parameterization.

Prior alignment and bias control. Generative priors capture *plausibility*, not necessarily *optimality*. How can we align or temper π_0 to balance structural validity with performance? Adaptive temperature schedules, mixture priors, or meta-learned weighting schemes may help prevent over-constraining the search while retaining useful bias.

Generative acquisition functions. Rather than using a fixed acquisition function, could a generative model learn to *represent and sample from* the acquisition landscape itself? This raises the possibility of *generative acquisition functions* – models that internalize both uncertainty and prior structure, potentially bridging BO and reinforcement learning.

Theoretical understanding. What are the convergence, regret, or consistency guarantees for sampling-based policies such as Eq. (19)? Can they be formalized as entropy-regularized optimization, energy-based inference, or constrained search in distribution space? Clarifying these connections could ground new hybrid methods in a principled theoretical framework.

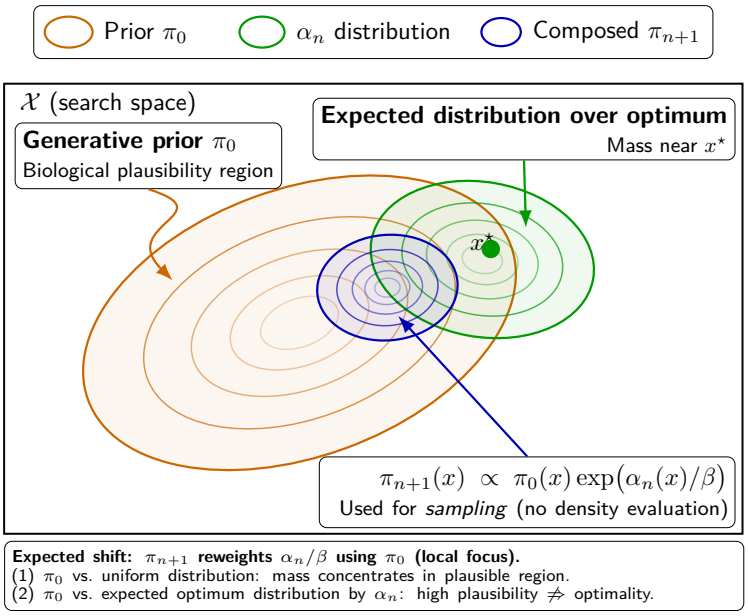


Figure 9 Conceptual overview: Bayesian optimization guided by a generative prior π_0 learned from data. The generative model constrains exploration to biologically plausible regions of the search space, while the acquisition function $\alpha_n(x)$ drives optimization toward higher predicted performance.

5.5.2.5 Rich Text-Based Human Feedback

Contributors

- First-draft writers: Matthias Poloczek, Peter Frazier
- Other interested parties: Raul Astudillo, Elena Raponi, Matthias Feurer, Antonio, Gautam Dasarathy, Daolang Huang

Introduction

There is a long history of work on BO and human interaction. This includes:

- **BO for preference learning**, e.g., see [5, 10, 2, 8, 9, 11]. Here, our objective function in a BO problem is a human decision-maker's utility function, which the human cannot specify as a mathematical expression. Instead, we interrogate this utility via preference queries.
- **BO with human-in-the-loop feasibility assessment**, e.g., [1, 12], where a human expert reviews inputs proposed by a BO algorithm and may label them infeasible before they are evaluated.
- **Elicitation of informative priors**, where human expertise is translated into prior distributions through structured questioning (e.g., quantiles, ranges, or plausible values).

Traditionally, human–algorithm interactions in BO have a simple form where the human provides only a few bits of feedback per interaction. For example, in BO for preference learning, the most common interaction is a pairwise comparison: the user is presented with two items and asked which one they prefer. In feasibility assessment, the human often provides a single bit of information (feasible or infeasible). In prior elicitation, humans answer structured questions that are then converted into prior distributions.

Large language models (LLMs) present an opportunity to interpret *richer text-based feedback from the human* and incorporate this into our algorithms. This promises many more bits per interaction, potentially enabling much more sample-efficient optimization.

Novel Challenges

Effective use of rich text-based human feedback raises novel challenges. Many of these can be categorized into questions about inference and questions about experiment and interaction design.

Inference.

- How do we interpret text-based human feedback?
- How do we leverage LLMs in this interpretation while maintaining probabilistic coherence?
- How do we combine information from LLMs with traditional probabilistic models (e.g., Gaussian processes or other surrogates)?
- How do we deal with variability, bias, and lack of reliability in LLM outputs?

Interaction and experiment design.

- **Form of interaction.** What forms of interaction are most effective? For example, in preferential BO, we could ask:
  - "Tell me what you are looking for in a solution."
  - "Which solution attributes are most important?"
  - "Compare solutions A and B. Tell me which one you prefer and why."
  - "Please improve solution A by making a small change and tell me why it is better."

  Effectiveness encompasses useful bits per interaction, ease of interpretation, and cognitive load for the human.
- **Question selection.** What is the best way to choose questions to ask the human? Should we outsource this to an LLM, rely on traditional acquisition functions, or combine both?
- **Resource allocation.** In light of this new way of getting feedback, how do we balance effort spent on querying the expensive black box with effort spent on interacting with the human?
- **Supporting human introspection.** Humans use introspection to explore their own preferences as they consider options in multi-objective problems. An ideal algorithm would clearly identify key tradeoffs and present them in a way that helps the human focus their introspective efforts.
- **Evaluation.** How do we evaluate such systems? Possible directions include:
  - using an LLM as a judge in simulated user studies,
  - online studies with real user traffic,
  - offline evaluation on comprehensive user datasets where multiple forms of feedback are available.

Related Work

- Max: `https://arxiv.org/abs/2510.17671`
- Peter: `https://arxiv.org/abs/2510.25799`
- Papers identified by Matthias: [7, 3]

[7] considers engineering tasks where the feasible space does not have an explicit mathematical form and proposes eliciting human prior knowledge in natural language to learn the feasible set for BO.

[3] proposes a human–AI collaborated experimental workflow, "Bayesian Optimized Active Recommender System" (BOARS), where the user can dynamically change the target objective on the fly.

More Specific Discussion of Preferential BO

Formal problem description. Formally, we have a feasible set $\mathbb{X}$, a computationally expensive black box that computes a collection of metrics $y = f(x)$ for any $x \in \mathbb{X}$, and an unknown user utility function $u(y)$. We seek to optimize $u(y)$ subject to the expense of evaluations and the constraint that u can only be interrogated through human feedback, potentially in rich textual form.

Important problem characteristics.

- **Dimensionality of y.** Is y low-dimensional (e.g., a small set of KPIs) or high-dimensional (e.g., images, videos, or long-form text)? The nature of y profoundly affects the form and cost of human feedback.
- **Need for physical instantiation.** Do we need to physically instantiate x to observe y and thus obtain feedback? For example, in materials design, x might be a composition, and y could involve physical properties that require lab experiments.

Connections to Other Areas. Connections to work on LLM alignment and assurance are natural: in both settings, one seeks to interpret human feedback and use it to guide system behavior. However, in our setting we typically do not modify the LLM itself; rather, we use the LLM as a tool to interpret feedback and support BO.

BO for Prompt Optimization

- Names from shared notes: Elena, Matthias P., Matthias Feurer, Antonio

This topic was identified as an important direction but was not developed during the seminar. We anticipate that many of the ideas above – LLM-based task generation, generative priors, and rich text-based human feedback – will play a role in BO for prompt and prompt-pipeline optimization. We leave a detailed treatment of this topic for future work.

References

1 Arun Kumar A V, Santu Rana, Alistair Shilton, and Svetha Venkatesh. Human–AI collaborative Bayesian optimisation. In *Advances in Neural Information Processing Systems*, pages 16233–16245, 2022.

2 Raul Astudillo and Peter Frazier. Multi-attribute Bayesian optimization with interactive preference learning. In *International Conference on Artificial Intelligence and Statistics*, pages 4496–4507, 2020.

3 Arpan Biswas, Yongtao Liu, Nicole Creange, Yu-Chen Liu, Stephen Jesse, Jan-Chi Yang, Sergei V Kalinin, Maxim A Ziatdinov, and Rama K Vasudevan. A dynamic Bayesian optimized active recommender system for curiosity-driven partially human-in-the-loop automated experiments. *npj Computational Materials*, 10(1):29, 2024.

4 Zhiliang Chen, Gregory Kang Ruey Lau, Chuan-Sheng Foo, and Bryan Kian Hsiang Low. Duet: Optimizing training data mixtures via feedback from unseen evaluation tasks, 2025.

5 Wei Chu and Zoubin Ghahramani. Preference learning with Gaussian processes. In *International Conference on Machine Learning*, pages 137–144, 2005.

6 Carl Hvarfner, Danny Stoll, Artur Souza, Marius Lindauer, Frank Hutter, and Luigi Nardi. πBO: Augmenting acquisition functions with user beliefs for Bayesian optimization, 2022.

7 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.

8 Zhiyuan Jerry Lin, Raul Astudillo, Peter Frazier, and Eytan Bakshy. Preference exploration for efficient Bayesian optimization with multiple outcomes. In *International Conference on Artificial Intelligence and Statistics*, pages 4235–4258, 2022.

9 Ryota Ozaki, Kazuki Ishikawa, Youhei Kanzaki, Shion Takeno, Ichiro Takeuchi, and Masayuki Karasuyama. Multi-objective Bayesian optimization with active preference learning. In *AAAI Conference on Artificial Intelligence*, volume 38, pages 14490–14498, 2024.

10 Juan Ungredda and Juergen Branke. When to elicit preferences in multi-objective Bayesian optimization. In *Genetic and Evolutionary Computation Conference Companion Proceedings*, pages 1997–2003, 2023.

11 Hanyang Wang, Juergen Branke, and Matthias Poloczek. Bayesian optimization with preference exploration by monotonic neural network ensemble. In *Advances in Neural Information Processing Systems*, 2025.

12 Wenjie Xu, Masaki Adachi, Colin N Jones, and Michael A Osborne. Principled Bayesian optimization in collaboration with human experts. *Advances in Neural Information Processing Systems*, 37:104091–104137, 2024.

13 Jason Yang, Wenda Chu, Daniel Khalil, Raul Astudillo, Bruce James Wittmann, Frances H. Arnold, and Yisong Yue. Steering generative models with experimental data for protein fitness optimization. In *Advances in Neural Information Processing Systems*, 2025.

14 Taeyoung Yun, Kiyoung Om, Jaewoo Lee, Sujin Yun, and Jinkyoo Park. Posterior inference with diffusion models for high-dimensional black-box optimization. In *Forty-second International Conference on Machine Learning*, 2025.

6 Additional Content

6.1 Benchmarking Bayesian Optimization in Practice: Lessons from a Dagstuhl Seminar Exercise

Giulia Pedrielli (Arizona State University – Tempe, US)

This report summarizes a hands-on benchmarking exercise conducted during the Dagstuhl Seminar 25451 (November 2025), organized with the explicit goal of fostering discussion around practical aspects of Bayesian Optimization (BO). Rather than comparing abstract algorithmic formulations, the exercise was designed to surface the "tricks of the trade": implementation details and design choices that materially affect performance but are often under-specified or abstracted away in the literature.

The intent of this report is not to establish winners or propose a standardized benchmark, but to document observations that emerged from systematically varying algorithmic choices under controlled experimental conditions, and to articulate reflections relevant to both researchers and practitioners.

6.1.1 Scope of the Exercise and Experimental Setup

The exercise focused on single-objective, black-box optimization under finite evaluation budgets. Algorithms were evaluated on a small suite of well-known continuous test functions of varying dimensionality and landscape characteristics, including Ackley (100D), Rosenbrock (6D), and Michalewicz (10D). Budgets were deliberately limited to emphasize sample efficiency rather than asymptotic convergence.

6.1.1.1 Algorithm Characterization

A central design principle of the exercise was to characterize BO algorithms not as monolithic entities, but as compositions of interacting design choices, including:

- Surrogate model class (e.g., Gaussian process variants)
- Kernel choice and hyperparameter handling
- Acquisition function family
- Acquisition optimization strategy
- Initialization strategy (number and type of initial design points)
- Budget allocation policy (initial design vs adaptive phase)
- Noise modeling assumptions
- Restart or trust-region mechanisms

This decomposition allowed participants to reason about why performance differed, rather than merely whether it differed.

6.1.2 Performance Metrics and Visualization

Algorithmic choices were observed under multiple complementary views:

1. Sample efficiency curves, showing best-so-far objective value as a function of evaluations.
2. Rank-based comparisons, aggregating relative performance across functions.
3. Budget sensitivity plots, highlighting the impact of how evaluation budgets are allocated between initialization and adaptive optimization.

The use of rank-based summaries was motivated by the observation that raw objective values are often not comparable across problems or scales, while ranks more clearly expose robustness and consistency.

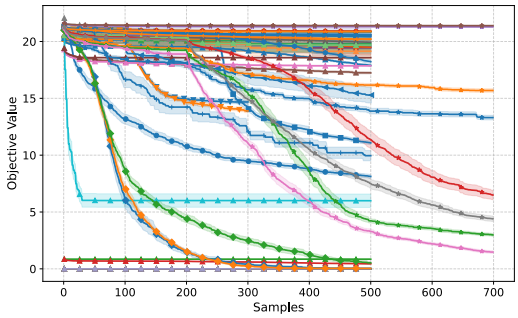
6.1.3 Empirical Results

6.1.3.1 Sample Efficiency

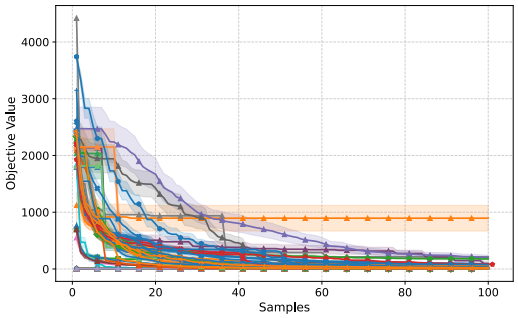
Across all test functions, we observed substantial variation in early-stage performance, even among algorithms that share nominally similar formulations. Figures illustrating convergence and sampled values (e.g., Ackley 100D and Rosenbrock 6D) show that initialization strategy and early hyperparameter behavior often dominate performance in the first 20–40% of the budget.

6.1.3.2 Rank-Based Comparisons

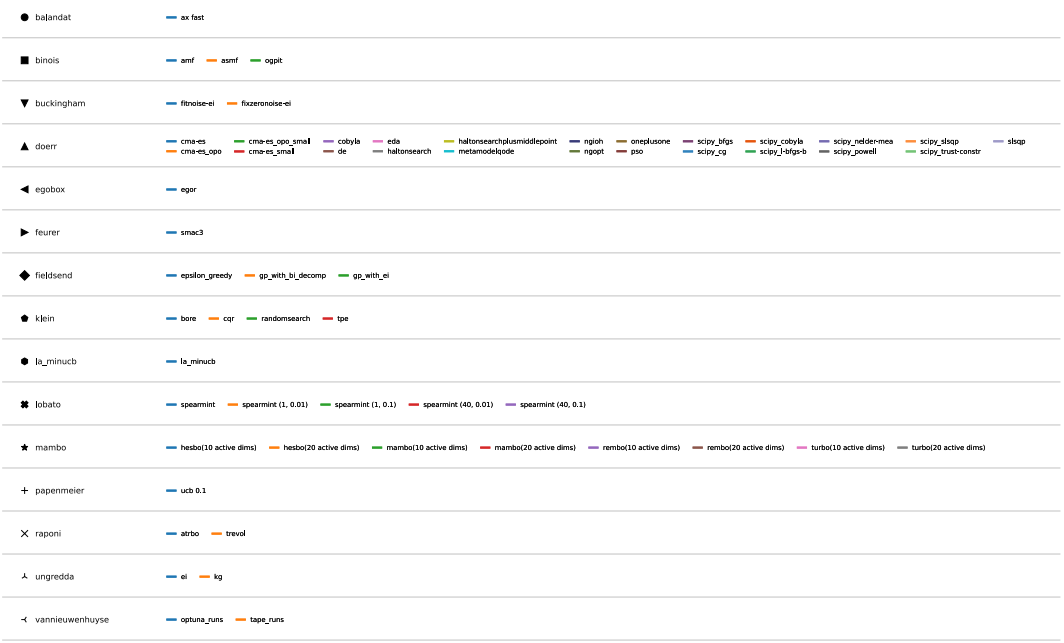
The overall average rank plot reveals that algorithm rankings are often compressed and unstable: small differences in setup can shift relative positions, and no single configuration consistently dominates across all problems. Aggregated rank statistics indicate that several algorithmic configurations perform comparably on average, with performance differences



(a) Ackley (100D)



(b) Rosenbrock (6D)



(c) Plots legend.

Figure 10 Sample efficiency of representative Bayesian optimization configurations. Plots report the best objective value observed so far as a function of the evaluation budget. Differences in early-stage behavior persist throughout the optimization horizon, highlighting the impact of initialization and surrogate configuration choices.

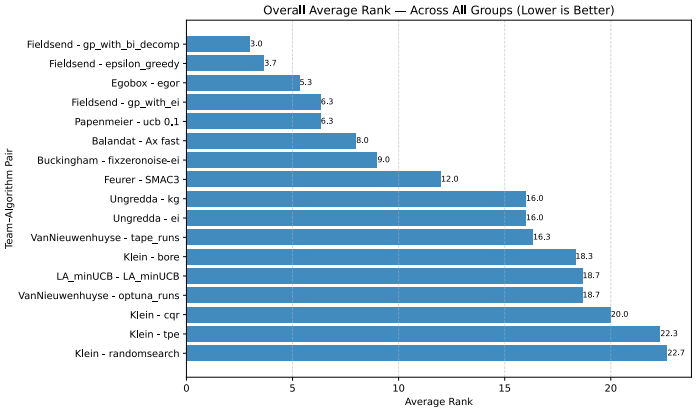


Figure 11 Average rank calculated for each approach across test functions.

frequently within statistical noise for the given budgets. This observation reinforces the notion that headline rankings without detailed configuration context can be misleading.

6.1.3.3 Budget Allocation Effects

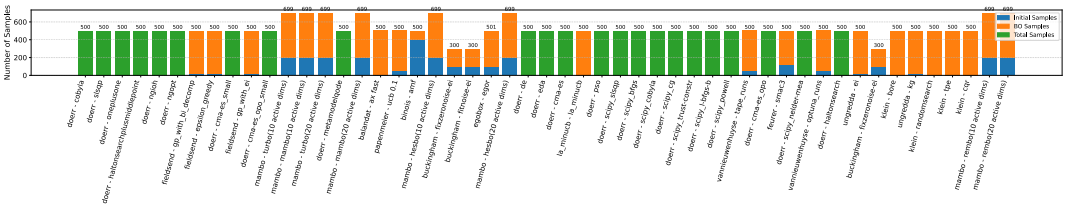
Explicit plots separating initial sampling from adaptive optimization show that how the budget is spent can matter as much as which acquisition function is used. Over-investment in initial design can delay exploitation, while overly aggressive early exploitation can lead to premature convergence. Sample allocation plots illustrate that moderate, problem-aware initialization strategies tend to offer better robustness across test functions.

6.1.4 Reflections on Algorithmic Choices

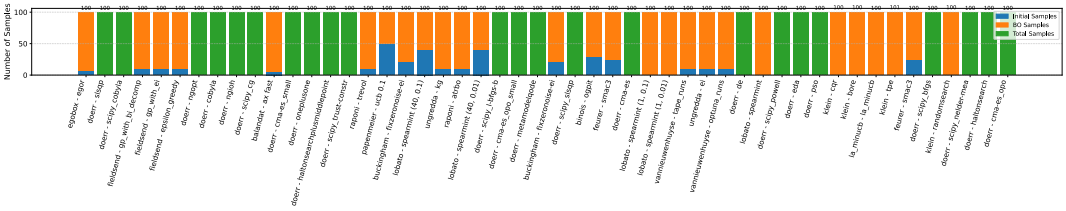
Several consistent themes emerged from the exercise and subsequent discussion:

- *Initialization Is Not a Detail*: The number, placement, and diversity of initial samples strongly influence surrogate fit and acquisition behavior. In high dimensions, naive space-filling designs can consume disproportionate budget with limited benefit.
- *Hyperparameter Handling Drives Early Behavior*: Surrogate hyperparameters, especially kernel length-scales, often dominate acquisition decisions early on. Aggressive re-estimation can destabilize optimization, while overly conservative strategies may delay adaptation.
- *Acquisition Optimization Quality Matters*: Differences in acquisition maximization (e.g., local vs global optimization, restart strategies) can outweigh differences between acquisition functions themselves, particularly under small budgets.
- *Budget Allocation Encodes Implicit Assumptions*: Choices about when to transition from exploration to exploitation reflect implicit beliefs about problem smoothness and noise. Making these assumptions explicit improves both interpretability and reproducibility.
- *More complex algorithm characterization may be needed*: it is a possibility that the lack of separation between algorithms partially lies in the choice of algorithm parameters. In other words, other working mechanisms of the methods may be required to understand the behaviors (e.g., use of a single/multiple surrogates, optimizer for the acquisition function, use of projections/embeddings).

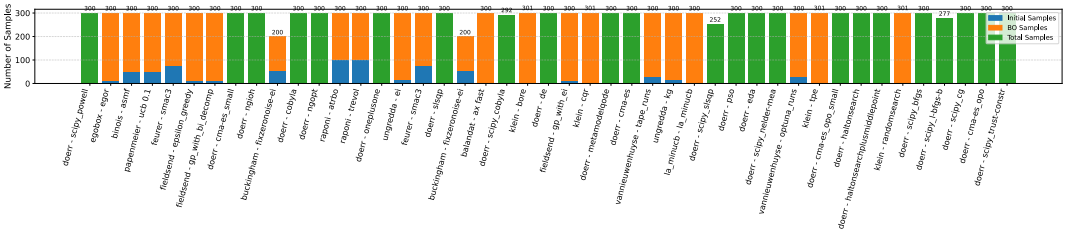
As an example of notable aspects that rose from the discussions during the seminar, the use of multiple surrogates dynamically and adaptively revealed to be potentially fruitful, on the



(a) Ackley (100D)



(b) Rosenbrock (6D)



(c) Michalewicz (10D)

Figure 12 Budget sensitivity.

other hand, strategies such as random projections revealed their lack of robustness in that they showed good performance but only due to the fact that the 100D function chosen was symmetric, and the same performance was not observed across benchmarks.

6.1.5 Concluding Remarks

This exercise highlights that practical BO performance is shaped less by isolated algorithmic components and more by the interaction of design choices under finite budgets. The results reinforce the need for benchmarking practices that foreground configuration metadata and encourage reflective analysis over leaderboard-style comparisons.

Rather than advocating for a particular "best" algorithm, the Dagstuhl exercise underscores the value of transparency, careful experimental design, and explicit acknowledgment of the practical decisions that ultimately govern performance.

6.2 Bayesian Optimization Tricks Of The Trade

Jürgen Branke (University of Warwick, GB)
Giulia Pedrielli (Arizona State University – Tempe, US)
Matthias Poloczek (Amazon.com – San Francisco, US)

6.2.1 Introduction

At the Dagstuhl Seminar 25451 on Bayesian Optimization, we asked all participants to fill in a questionnaire and tell us their tricks of the trade. The questions we asked were:

1. What did you wish someone had told you before you started working in BO?
2. What's something you usually do but never see written in papers?
3. What's a mistake you made once and vowed never to repeat?
4. When reviewing others' work, what silen red flags tell you the're beginners?

In addition, we have asked chatGPT 5 to answer the same questions. The following summarizes the responses we received.

6.2.2 What I wish I had known ealier

- Always check and tune the model before doing optimisation.
- The details of implementation matter much more than expected (HP optimisation, AF optimisation).
- Having a solid foundation in GP bandits helps enormously.
- You don't have to be a strict Bayesian to do good BO.
- Try different kernels and trend functions; default choices rarely work best. If you go with one kernel, Matérn 5/2 seems a generally accepted choice.
- Nugget effect (modelling a small constant noise) improves stability in deterministic problems.
- Random axis-aligned subspace perturbations (RAASP) which samples around the current optimal point along which differ only in a random subset of dimensions or cylindrical Thompson sampling are good ways for candidate sampling especially in higher dimensions.
- Thompson sampling works when other AFs fail, probably because it is much easier to optimise than other AFs.
- Always start with smaller tests before scaling up.
- Establish clear baselines, such as random search, CMA-ES or vanilla BO.
- Save **EVERYTHING** you can, acqf values, commits, number of initial points, lengtscales, etc. Saves a lot of debugging time.
- Model mismatch and AF optimization can have a huge impact on results. Therefore one should always also test on a GP generated test problem on a discrete set where the AF can be optimized via complete enumeration.
- Setting the prior mean is application-dependent, often a constant, sometimes parametrized, for LLM even a scaling law.

6.2.3 Unwritten practices

- Scale all inputs to the unit hypercube
- Test on simple 2D problems first that allow visualization.
- Standardise objectives on the fly during multi-objective optimisation.

- Use multi-start when fitting the GP.
- Warm-start the model fit from the previous solution.
- Put lower bounds on kernel length scales to prevent numerical issues.
- Use logEI rather than EI.
- Smart points to optimize the acquisition function help a lot. Best posterior mean location, location of the best sampled location.
- Validate performance on a test set, not just on benchmarks used for deciding BO hyperparameters.

6.2.4 Mistakes to avoid

- Forgetting to update the reference point for expected hypervolume improvement calculation in multi-objective optimisation.
- Using unstable implementations of BFGS or other optimisers.
- Forgetting to check the GP fit.
- Failing to rescale data before fitting models.
- Starting benchmarking before checking that the code is correct.
- Not saving data, forcing repeated and inconsistent re-runs.
- Evaluating on too few problems or only on familiar ones.

6.2.5 Beginner red flags

- No error bars or uncertainty measures in reported results.
- Too many arbitrary "magic" constants in experiments.
- Only one example tested, with no justification.
- Misuse of terminology (e.g., calling BO "EGO").
- Wrong or missing citations for well-known work.
- Re-inventing already existing methods.
- Use squared exponential kernel without justification
- Irrelevant details swamping main explanations (move details to appendix).
- Experimental setups not clearly described.
- No ablation study on hyperparameters, or no explanation on how they were chosen.
- AF and HP optimization process not explained.
- Mixing up GP and acquisition-function concepts.
- Lack of replication or reproducibility checks.
- No scientific hypothesis that is clearly defined and rejected/not rejected at the end.
- Using an unsuitable method as baseline for the empirical evaluation.

6.2.6 Other good practices

- Report, not just validate, the performance on unseen test tasks.
- Accessible codebase in addition to high-level description of the code in the paper.

Participants

- Steven Adriaensen
Universität Freiburg, DE
- Mauricio A Álvarez
University of Manchester, GB
- Cedric Archambeau
Helsing – Berlin, DE
- Raul Astudillo
California Institute of Technology – Pasadena, US
- Maximilian Balandat
Meta – Menlo Park, US
- Nathalie Bartoli
ONERA – Toulouse, FR
- Thomas Bartz-Beielstein
TH Köln, DE
- Mickaël Binois
Inria Center at Université Côte d'Azur – Sophia Antipolis, FR
- Jürgen Branke
University of Warwick, GB
- Jack Buckingham
University of Warwick – Coventry, GB
- Antonio Candelieri
University of Milano-Bicocca, IT
- Ivo Couckuyt
Ghent University, BE
- Gautam Dasarathy
Arizona State University – Tempe, US
- Carola Doerr
Sorbonne University – Paris, FR
- Katharina Eggensperger
TU Dortmund, DE
- Matthias Feurer
TU Dortmund, DE
- Jonathan Fieldsend
University of Exeter, GB
- Peter Frazier
Cornell University – Ithaca, US
- Roman Garnett
Washington University – St. Louis, US
- Robert Gramacy
Virginia Polytechnic Institute – Blacksburg, US
- Daniel Hernández-Lobato
Autonomous University of Madrid, ES
- Daolang Huang
Aalto University, FI
- Frank Hutter
Prior Labs – Freiburg, DE & ELLIS Institute Tübingen, DE & Universität Freiburg, DE
- Aaron Klein
ScaDS.AI – Leipzig, DE
- Rodolphe Le Riche
CNRS – Aubière, FR
- Bryan Kian Hsiang Low
National University of Singapore, SG
- Neeratyoy Mallik
Universität Freiburg, DE
- Mike McCourt
Distributional – Toronto, CA
- Ruth Misener
Imperial College London, GB
- Szu Hui Ng
National University of Singapore, SG
- Leonard Papenmeier
Universität Münster, DE
- Giulia Pedrielli
Arizona State University – Tempe, US
- Matthias Poloczek
Amazon.com – San Francisco, US
- Jixiang Qing
Imperial College London, GB
- Elena Raponi
Leiden University, NL
- Sebastian Rojas Gonzalez
Ghent University, BE
- Alexander Terenin
Cornell University – Ithaca, US
- Juan Ungredda
ESTECO SpA – Trieste, IT
- Inneke Van Nieuwenhuyse
Hasselt University – Diepenbeek, BE