BayMOTH: Bayesian optiMizatiOn with meTa-lookahead – a simple approacH

Anonymous authors
Paper under double-blind review

Abstract

Bayesian optimization (BO) has for sequential optimization of expensive black-box functions demonstrated practicality and effectiveness in many real-world settings. Meta-Bayesian optimization (meta-BO) focuses on improving the sample efficiency of BO by making use of information from related tasks. Although meta-BO is sample-efficient when task structure transfers, poor alignment between meta-training and test tasks can cause suboptimal queries to be suggested during online optimization. To this end, we propose a simple meta-BO algorithm that utilizes related task information when determined useful, falling back to lookahead BO otherwise, within a unified framework. We demonstrate competitiveness of our method with existing approaches on function optimization tasks, while retaining strong performance in low task-relatedness regimes where test tasks share limited structure with the meta-training set.

1 Introduction

For realistic applications that require optimization of an unknown function f, high sample evaluation costs often constrains the sampling budget. Examples of such applications arise in hyperparameter tuning of machine learning models (Snoek et al., 2012; Zoph & Le, 2016; Feurer & Hutter, 2019), in materials design (Frazier & Wang, 2015), and in physics applications such as in the domain of laser-driven nuclear fusion experiments (ICF) (Ejaz et al., 2023; Gundecha et al., 2024). For such tasks, Bayesian optimization (BO) (Frazier, 2018) has proven its utility. However, on the quest of more performant methods i.e. higher sample efficiency, approaches such as meta-BO and lookahead Bayesian optimization (lookahead BO) have emerged. Both improve sample efficiency albeit both utilize different approaches. In order to facilitate discussion, we borrow from the nomenclature of reinforcement learning and refer to the sampling strategy as the sampling policy denoted by π.

In meta-BO various strategies are used to transfer information (Bai et al., 2023; Ma et al., 2025) drawn from related tasks into components of the optimization methods' pipeline. This enables a more efficient policy π to be realized, especially compared with baselines such as random sampling and BO (Volpp et al., 2019). In contrast to meta-BO, lookahead BO does not rely on knowledge from related tasks and seeks to improve upon BO by recognizing that conventional BO operates under a one-step assumption without consideration of future samples allowed within the sample budget. Polices focused on maximizing the unknown function f in the immediate step are commonly referred to as "myopic" policies and recent work (Wu & Frazier, 2019; Cheon et al., 2024) demonstrates that during online optimization, lookahead policies reach near-optimal regions in fewer samples across a variety of benchmarks, indicating a superior sampling policy.

Combining meta-BO and lookahead BO appears to be a natural extension. However, it has been relatively unexplored, motivating a straightforward framework combining the two approaches. The presented method, which is referred to as BayMOTH, makes use of alternative virtual environments for inferring the presence of shared structure during online optimization between the test task (i.e., unknown function) and source tasks (training functions) provided to it offline (Levine et al., 2020) during meta-training. The shared structure information is used as a task descriptor similar in spirit to usage in multi-task learning (Zhao et al., 2019) to

gate into the pertinent alternative virtual environment which provides meta-data for informing the lookahead based policy.

We demonstrate the competitiveness of BayMOTH on hyperparameter optimization tasks and scientific experiments (ICF experiments) optimization tasks and compare it to incumbent meta-BO methods, multi-fidelity BO, lookahead BO, and conventional BO. The experimental tests are conducted under the context of relevant meta-data being available for meta-BO. Details are provided in Section 4.[1]

In many practical applications, however, it is often unclear if prior meta-information is useful. BayMOTH is, by construction, more robust to useless prior information, unlike meta-BO. We demonstrate this through targeted experiments, in which to the meta-BO methods the training tasks that are provided are not strongly related to the unknown function f. This experimental setup emulates real-world scenarios where the assumed relationship between the meta-information and the unknown function may not hold. In autonomous applications where the value of prior information is unknown, it is vital that the sampling policy must be able to self-assess the value of its priors. In existing BO works, this mismatch has been remarkably understudied, and to the best of our knowledge has only been considered previously by the robust multi-fidelity BO (rMFBO) of Mikkola et al. (2023) which accepts a MFBO proposal over a single-fidelity BO proposal, if the MFBO surrogate passes certain conditions.

Moreover, as becomes evident in Section 3, BayMOTH is a relatively simple algorithm when compared with popular meta-BO approaches, which employ reinforcement learning (RL) to incorporate meta-information into their respective sampling policies. It is well known that training RL based policies can be challenging due to the deadly triad of RL (Sutton et al., 1998). To counteract such effects, state-of-the-art (SOTA) meta-BO methods often include auxiliary tasks for stable policy training such as in the work of Maraval et al. (2023), which complicates the training process. While simplicity is not sufficient in isolation as a merit-able attribute, the complexity in algorithms and training procedure can preclude usage from practitioners outside of ML/RL expertise. For such practitioners sample efficient black-box optimization can be of great interest due to high cost of evaluations for their respective domains such as for scientific experiments. Therefore, the competitive performance and simplicity of BayMOTH, we deem useful to practitioners.

Contributions. Our main contributions are summarized as follows:

- We highlight suboptimal performance in meta-BO under source task mismatch.
- We propose a simple meta-Bayesian optimization algorithm (BayMOTH), that remains robust under source task mismatch and leverages related task structure without relying on RL pretraining.
- BayMOTH unifies lookahead and meta-BO perspectives in a simple algorithm.
- We show empirically that BayMOTH performs competitively against state-of-the-art meta-BO methods and strong established baselines, and we situate its performance relative these approaches.

2 Background

2.1 Problem formulation

We wish to find the optimum of an unknown expensive to evaluate black-box function $f : \mathcal{D} \to \mathbb{R}$ on a known domain $\mathcal{D} \subseteq \mathbb{R}^d$, within a sample budget T. A policy π chooses points to sample $x \in \mathcal{D}$ from $f(x)$ in a sequential manner resulting in a evaluation history $\mathcal{H}_t = \{x_i, f(x_i)\}_{i=1}^t$ after t sample steps. A sampling policy π is considered superior if the simple regret $r(t) := \max_{x\in\mathcal{D}} f(x) - \max_{1\leq i\leq t} f(x_i)$ is minimized in fewer sample steps.

2.2 Bayesian Optimization

In Bayesian optimization (BO) (Shahriari et al., 2015; Frazier & Wang, 2015) a probabilistic surrogate model, commonly a *Gaussian Process* ($\mathcal{GP}$) (Williams & Rasmussen, 2006) is used in conjunction with an

[1]All datasets and code are provided as supplementary material.

acquisition function (AF) for choosing sample locations that potentially maximize $f(x)$. Given an evaluation history $\mathcal{H}_t = \{x_i, f(x_i)\}_{i=1}^t$, the $\mathcal{GP}$ posterior is characterized by the mean $\mu_t(x) = \mathbb{E}[f(x)|\mathcal{H}_t]$ and variance $\sigma_t = \mathbb{V}[f(x)|\mathcal{H}_t]^{\frac{1}{2}}$, which have closed analytic forms (Williams & Rasmussen, 2006). Utilizing the $\mathcal{GP}$ posterior, an acquisition function provides a score value for $x \in \mathcal{D}$ taking also into account exploitation-exploration tradeoffs. A widely used AF is *Expected Improvement* (EI) (Jones et al., 1998) which at sample step t is given as:

$$\mathrm{EI}_{t+1}(x) = \mathbb{E}[\max(f(x) - \max_{1 \leq i \leq t} f(x_i), 0)]$$

The BO policy, $\pi_{\mathrm{BO}} \rightarrow x_{t+1}$ is $\mathrm{argmax}_{x \in \mathcal{D}} \mathrm{EI}_{t+1}(x)$.

2.3 Meta-Bayesian Optimization

In contrast to BO, where only the evaluation history informs the policy, meta-BO makes use of information available from related functions which is typically available in the form of sample-evaluation pairs as shown:

$$\mathcal{D}_\tau = \{(x_1, f_\tau(x_1)), \ldots, (x_N, f_\tau(x_N))\},$$

from any arbitrary policy forming task-specific datasets $\mathcal{D}_1, \mathcal{D}_2, \ldots \mathcal{D}_\tau$ that collectively constitute the meta-dataset ($\mathcal{D}_{\text{meta-train}}$). Approaches on how the meta-dataset is utilized differs among different approaches. Prior work has focused on the surrogate model component of BO, such as through ranking-weighted Gaussian process ensembles (Feurer et al., 2018) and multi-task Gaussian processes, or on the acquisition function component of BO, for example via transfer acquisition functions (Wistuba et al., 2018), whereas the work of (Maraval et al., 2023) focuses on both components in an end-to-end framework. A comprehensive survey of meta-BO can be found in Bai et al. (2023) and Ma et al. (2025).

Here, we expand on some foundational meta-BO methods. Generally, these methods learn a general sampling rule (policy) that at test-time require less sample steps for regret reduction compared with non-meta-BO. This is achieved by incorporating information from the meta-dataset obtained from related functions. Examples of related functions may include synthetic families of a well defined function such as sine waves with varying phase and amplitude over a shared domain (Finn et al., 2017), or in more abstract settings can be in the context of hyperparameter optimization (HPO) where model accuracies as a function of different hyperparameter configurations and search spaces are available for multiple classification problems (Arango et al., 2021).

The approach of Volpp et al. (2019) (known as MetaBO) makes use of a $\mathcal{GP}$ surrogate model similar to the traditional BO setup, but makes use of a *neural acquisition function* (NAF) in lieu of a classical AF. The meta-dataset is used to train the NAF with the Proximal Policy Optimization (PPO) RL algorithm where NAF takes as input $[\mu_t(x), \sigma_t(x), x]$ for the state-action representation at each sample step to output a score-value on a discrete set of points on the optimization domain. During PPO training the current policy rolls-out $\mathcal{H}_t = \{x_i, f(x_i)\}_{i=1}^t$ from f available in $\mathcal{D}_{\text{meta-train}}$, and a respective $\mathcal{GP}_t$ provides $\mu_t(x)$ and $\sigma_t(x)$ to the NAF. In the training loop the input information for NAF along with the auxiliary information (i.e. reward) needed for PPO is recorded for an episode, batches of which are used for the NAF policy update. Conceptually, after meta-training, NAF has learned how to use $[\mu_t(x), \sigma_t(x), x]$ for providing a high-score for promising sample locations over a variety of functions. As mentioned by Volpp et al. (2019) the inclusion of x as an input enables it to identify shared structure between functions used during offline meta-training and functions it will encounter in usage.

Similar to MetaBO (Volpp et al., 2019), FSAF (Hsieh et al., 2021) also uses $[\mu_t(x), \sigma_t(x)]$ from a $\mathcal{GP}$ as part of an embedding for the state-action representation. For the acquisition function a Bayesian approach is used to learn N deep-Q-networks (DQNs) in a MAML bi-level framework (Finn et al., 2017; Yoon et al., 2018), such that from the collection of $f \in \mathcal{D}_{\text{meta-train}}$ a generalized policy is learned that is well suited for one-shot adaption to the unknown function f, by making gradient updates to the DQN using $\mathcal{H}_t = \{x_i, f(x_i)\}_{i=1}^t$.

2.4 Multi-fidelity Bayesian optimization (MFBO)

MFBO extends standard BO by allowing (during online sampling) a query be made to the expensive test task or cheaper auxiliary information sources, by utilizing a joint surrogate over all fidelities in an augmented

space and an acquisition function that trades off informativeness vs. cost across fidelities. Its benefit relies on the auxiliary fidelities being sufficiently informative about the test task. However, when this assumption fails, MFBO can waste budget on misleading sources and even under perform single-fidelity BO. The robust MFBO (rMFBO) of Mikkola et al. (2023) addresses this issue through a wrapper for GP-based MFBO methods. The rMFBO approach accepts a multi-fidelity proposal when reliability conditions are met and otherwise falls back to a conservative single-fidelity alternative. We draw attention that BayMOTH is related to this perspective at a high-level, as both are motivated by the principle that source information should be exploited only when it is determined useful and the sampling policy should remain robust under source-test task mismatch. However, both approaches are fundamentally different. Furthermore, in the MFBO paradigm, auxiliary source information is revealed only online through sequential queries to that fidelity, with the objective being more focused on reducing costly evaluations on the target task. In contrast, in BayMOTH and meta-BO more broadly, source task information is available a priori through meta-training, therefore, source task information is already available before online optimization begins.

2.5 Lookahead Bayesian Optimization

Lookahead BO polices aim to improve upon BO by taking into consideration the allowance of future evaluations within the sample budget, with the assumption that an exploration sample taken at a current sample step t can reveal promising locations for finding the optimum location in future samples.

Remark 2.1. The meta-BO methods mentioned in Section 2.3 also utilize t/T as part of their state-action representation which can influence future sample points. However, this approach remains myopic, as, unlike lookahead methods, it does not explicitly simulate possible future sample outcomes.

The practical two-step lookahead acquisition function proposed by (Wu & Frazier, 2019), known as 2-OPT, demonstrates that even limited-horizon (two-step) lookahead can yield meaningful performance gains over standard one-step BO. The 2-OPT acquisition function itself consists of two EI acquisition functions each pertaining to a stage in the two-stage process, The first EI component (EI_0) considers $\mathcal{GP}_0$ from $\mathcal{H}_t = \{x_i, f(x_i)\}_{i=1}^{t}$, whereas EI_1 considers $\mathcal{GP}_1$ from $\mathcal{H}_{t_{\text{sim}}} = \mathcal{H}_t \cup \{x_{t+1}, f(x_{t+1}) \sim \mathcal{GP}_0\}$. Thus, the 2-OPT acquisition function takes the form

$$\text{2-OPT}(x_{t+1}) \;\approx\; \mathrm{EI}_0(x_{t+1}) \;+\; \mathbb{E}\left[\max_{x_{t+2}} \mathrm{EI}_1(x_{t+2})\right].$$

Where, the expectation is taken under $\mathcal{GP}_0$. The BayMOTH policy is based on the 2-OPT acquisition function, its details are presented in section 3.

Multi-step lookahead aims to maximize long-horizon optimization performance. However, due to scalability issues, emulating multi-step BO has been considered instead and is shown to be performant in the work by Cheon et al. (2024) by making use of an Attention-Deep-Sets encoder for projecting state information onto a latent space suitable for a RL framework.

3 BayMOTH

In this section we introduce BayMOTH, a meta-BO policy that extends the 2-OPT acquisition function by incorporating meta-information through a well-defined selection rule and fallback to 2-OPT when such information is deemed uninformative. A key feature of BayMOTH is that it is capable of autonomously deciding when the meta-information is useful. We first describe the inclusion procedure of meta-information based on shared structure identification with the related functions, and then present the BayMOTH policy and simplifications made to the policy for practical usage, finally we present an informal theoretical analysis for BayMOTH providing insight for regret performance. The full algorithm for BayMOTH is presented in Section A.1

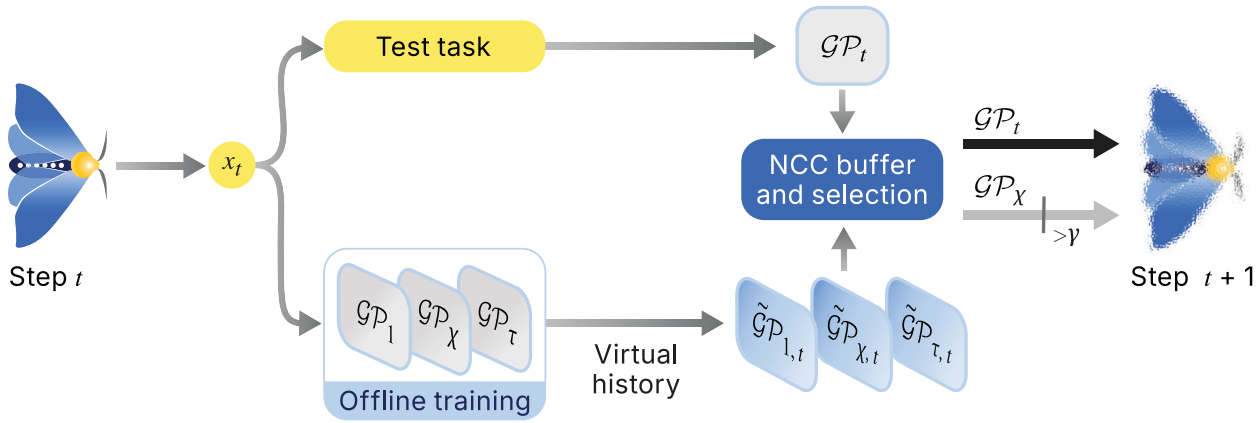


Figure 1: *An overview of BayMOTH.* During the online BO step t, the sample proposal is evaluated on the unknown test task and across the source task virtual environments. The alignment between each induced virtual environment surrogate $\tilde{\mathcal{GP}}_{\chi,t}$, and the test task surrogate, $\mathcal{GP}_t$, is quantified using NCC. If the maximum NCC score across virtual environments exceeds the γ threshold, BayMOTH uses the corresponding $\mathcal{GP}_\chi$ to guide proposal generation at step $t+1$.

3.1 Shared structure formulism

A common peril for meta-learning based approaches is the memorization problem (Yin et al., 2019). The memorization problem is a form of task overfitting where the meta-learner overfits to the finite set of provided training tasks[2] by implicitly encoding task-specific solutions, rather than learning transferable attributes that enable good performance on unseen tasks. This form of overfitting is subtle as the meta-learning based model can still be performant if the unseen task is similar to the training tasks. This concept is formalized in Yin et al. (2019). Meta–BO methods are also susceptible to memorization, as strong meta-learned surrogates or acquisition policies can achieve low meta-training regret by encoding task-specific structure present in the training set rather than learning robust exploration/exploitation mechanisms that generalize to unseen tasks. As a result, when task diversity is limited, performance gains may diminish when evaluated on unseen objectives or complex tasks. A mitigating strategy is to increase the number of training tasks (Volpp et al., 2019), though, its applicability is limited in scientific or HPO settings where datasets are expensive to generate and the available meta-data is inherently fixed. Other strategies include restricting model capacity and Yin et al. (2019) propose a meta-regularizer based on information theory.

For robustness against memorization and explicit usage of only relevant meta-information, BayMOTH makes use of alternative virtual environments. A alternative virtual environment for each dataset in $\mathcal{D}_{\text{meta-train}}$ is constructed by fitting a $\mathcal{GP}$ to $\mathcal{D}_\chi = \{(x_i, f_\chi(x_i))\}_{i=1}^n$, where $\mathcal{D}_\chi \in \mathcal{D}_{\text{meta-train}}$. Consequently, given τ related tasks available for meta-training, i.e., $\{\mathcal{D}_1, \ldots, \mathcal{D}_\chi, \ldots, \mathcal{D}_\tau\}$, this procedure induces a collection of τ task-specific alternative virtual environments containing $\{\mathcal{GP}_1, \ldots, \mathcal{GP}_\chi, \ldots, \mathcal{GP}_\tau\}$.

At optimization step $t+1$ on an unknown task, the observed sample history $\mathcal{H}_t = \{(x_i, f(x_i)\}_{i=1}^t$ is reused to query in each virtual environment yielding virtual sample histories $\tilde{\mathcal{H}}_t^\chi = \{(x_i, \tilde{f}_\chi(x_i)\}_{i=1}^t$, where $\tilde{f}_\chi(x_i) = \mu_\chi(x_i)$ from $\mathcal{GP}_\chi(x_i)$. Each virtual history $\tilde{\mathcal{H}}_t^\chi$ is then used to fit a $\tilde{\mathcal{GP}}_{\chi,t}$ using the same kernel and hyperparameter configuration as the surrogate $\mathcal{GP}_t$ fitted to the unknown task. The resulting surrogates are compared to $\mathcal{GP}_t$ using normalized cross-correlation (NCC) between their predictive mean functions, defined as:

$$\text{NCC}(\mu_t, \tilde{\mu}_{\chi,t}) = \frac{\sum_{j=1}^M (\mu_t(x_j) - \bar{\mu}_t)(\tilde{\mu}_{\chi,t}(x_j) - \bar{\tilde{\mu}}_{\chi,t})}{\|\mu_t - \bar{\mu}_t\|_2 \, \|\tilde{\mu}_{\chi,t} - \bar{\tilde{\mu}}_{\chi,t}\|_2},$$

where $\mu_t(\cdot)$ and $\tilde{\mu}_{\chi,t}(\cdot)$ denote the predictive means of $\mathcal{GP}_t$ and $\tilde{\mathcal{GP}}_{\chi,t}$, respectively, evaluated on a shared reference set $\{x_j\}_{j=1}^M$, and $\bar{\mu}_t$ and $\bar{\tilde{\mu}}_{\chi,t}$ denote their empirical means. The meta-training task achieving the

[2]In the meta-learning literature, the terms *related tasks* (Finn et al., 2017) and *related functions* are often used interchangeably.

highest NCC score is selected as exhibiting the most shared structure with the unknown task. This score is used as a task descriptor to gate into the pertinent alternative virtual environment through the selection rule, whose corresponding $\mathcal{GP}$ now denoted as $\mathcal{GP}_{\chi^*}$ is used in Eq 1. If each alternative virtual environment's NCC score is below a threshold, BayMOTH fallbacks to 2-OPT. Routing and source task usage behavior is elaborated on in the ablation provided in Appendix A.4.

Normalized cross-correlation is well suited for this comparison, as it is invariant to affine scaling of function values and emphasizes structural alignment, making it robust to differences in magnitude and noise levels even in high-dimensional input spaces. Our use of NCC is tied directly to its role in BayMOTH: it serves as a routing criterion for deciding whether a source task exhibits sufficiently similar structure to the test task for meta-informed lookahead to be beneficial. Alternative routing criteria, such as KL-divergence-based measures or cost-adjusted information gain (Mikkola et al., 2023), are also conceivable. However, for our algorithmic objective, NCC computed on predictive means isolates the quantity most relevant to our intent, namely structural alignment is informative for meta-lookahead to provide operational benefit. Furthermore, to the best of our knowledge, existing literature does not provide a formal definition of task relatedness in the context of BO, as is also noted by Mikkola et al. (2023). Here we do not attempt to introduce one. Instead, we adopt NCC as a pragmatic relatedness measure that is well matched to the intended function it serves in BayMOTH.

While this procedure requires fitting τ auxiliary Gaussian processes at each optimization step, the computational overhead remains modest relative to the cost of unknown function evaluations in many real-world usage scenarios. The computational cost of BayMOTH is explored in Appendix A.7. Additionally, the resulting simplicity and interpretability of the task-selection mechanism provides a favorable trade-off in practice.

3.2 BayMOTH policy

Given data $\mathcal{H}_{1:t}$ from the unknown function f, where,

$$\mathcal{H}_{1:t} = \Big\{ (\mathbf{x}_1, f_1), (\mathbf{x}_2, f_2), \ldots, (\mathbf{x}_t, f_t) \Big\},$$

the next sample location is provided from BayMOTH according to its policy π_{BayMOTH} which is given by:

$$\arg\max_{x_{t+1}} \Big[\Lambda_0(x_{t+1}) + \mathbb{E}_{\mathcal{GP}_{\chi^*}} \Big[(1-\alpha) \max_{x_{t+2}} \Lambda_1(x_{t+2}) + \alpha\, \Omega \Big] \Big] \tag{1}$$

Here, Λ_0 is the EI acquisition function considering the $\mathcal{GP}$ posterior from data $\mathcal{H}_{1:t}$, and Λ_1 is EI considering the $\mathcal{GP}$ posterior from data $\mathcal{H}_{1:t_{\text{sim}}}$. Where , $\mathcal{H}_{1:t_{\text{sim}}}$ contains the simulated sample evaluation at x_{t+1},

$$\mathcal{H}_{1:t_{\text{sim}}} = \mathcal{H}_{1:t} \cup \Big\{ x_{t+1},\ f(x_{t+1}) \sim \mathcal{GP}_{\chi^*} \Big\}. \tag{2}$$

The Ω term is a greedy improvement component defined as $\Omega = \max\big(0,\ f(x_{t+1}) - f^{(t)}_{\max}\big)$ where $f^{(t)}_{\max} = \max_{1 \le i \le t} f_i$. The term in the expectation is interpreted as a convex combination of a meta-information derived future expected improvement term with an immediate greedy improvement term. By interpolating these terms as such, we are balancing horizon planning with direct exploitation, with both terms being informed by meta-information. The behavior of BayMOTH with this balancing is elaborated through an ablation study on α in Appendix A.6.

3.3 BayMOTH simplification

The second term in Eq.1 is an expectation under $\mathcal{GP}_{\chi^*}$, considering the randomness of $\mathcal{GP}_{\chi^*}$ as $\mathcal{P}(\omega)$ we define this term as,

$$\mathbb{J}(\omega) := \mathbb{E}_{\omega \sim \mathcal{P}} \Big[(1-\alpha) \max_{x_{t+2}} \Lambda_1(x_{t+2}; \omega) + \alpha \Omega(\omega) \Big]$$

and,

$$\mathbb{Z}(\omega) := (1-\alpha)\max_{x_{t+2}} \Lambda_1(x_{t+2};\omega) + \alpha\Omega(\omega)$$

then,

$$\mathbb{J}(\omega) = \int \mathbb{Z}(\omega)\mathcal{P}(\omega)\mathrm{d}\omega$$

$$= \int \Big((1-\alpha)\max_{x_{t+2}} \Lambda_1(x_{t+2};\omega) + \alpha\Omega(\omega)\Big)\mathcal{P}(\omega)\mathrm{d}\omega$$

as this is intractable analytically, we use a Monte Carlo estimator

$$\hat{J}_M \;=\; \frac{1}{M}\sum_{m=1}^{M} \mathbb{Z}(\omega^{(m)}), \qquad \omega^{(m)} \sim \mathrm{GP}_{\chi^*} \text{ i.i.d.} \tag{3}$$

This estimator is unbiased and converges to $\mathbb{J}$ as $M \to \infty$ by the law of large numbers, with variance $\mathbb{V}(\hat{J}_M) = \sigma^2/M$ where σ^2 is the variance of $\mathbb{Z}$. These properties are detailed in Appendix A.2.

Although the variance of the estimator decays at rate ($O(1/M)$), in practice, relatively small number of samples is sufficient for estimator accuracy (Bishop & Bishop, 2023). As GP_{χ^*} posterior depends on the samples drawn from $\mathcal{D}_\chi$, the variance on the random variable $\mathbb{Z}(\omega)$ depends directly on how well $\mathcal{D}_\chi$ covers the underlying function over the domain of interest. When the domain is densely sampled, the posterior variance of GP_{χ^*} is typically small throughout the domain, which in turn implies that σ^2 is also small. Since the estimator has variance σ^2/M, a modest number of samples is a sufficient approximation for small σ^2.

In instances where the domain is sparsely sampled for $\mathcal{D}_\chi$, BayMOTH is still performant as a relative ranking of candidate points is the quantity of interest over accurately estimating $\mathbb{Z}$. Moreover, the stochasticity in $\mathbb{Z}$ can even encourage exploration. In all experiments we set $M = 5$, which we found to provide a good trade-off between computational cost and performance. A small ablation in Appendix A.5 shows that larger M yields only marginal performance gains.

3.4 Theoretical Analysis

Here, we present an informal theoretical analysis. BayMOTH is built on a gated two-branch architecture. When the inferred task-relatedness is low, it reverts to the 2-OPT acquisition; when the inferred task-relatedness is sufficiently high (γ dependent), it uses a meta-informed lookahead branch that modifies the horizon planning term through the selected alternative virtual environment and the greedy improvement component. Accordingly, rather than attempting to derive a formal regret bound, we analyze it relative to an idealized oracle version of itself. This oracle comparator uses the correct branch, correct virtual environment (when applicable), together with the exact expectation in the meta branch, and in particular reduces to 2-OPT on rounds where fallback is the appropriate choice.

To formalize this, let

$$f^\star := \max_{x\in D} f(x), \qquad r_T^{\mathrm{BM}} := f^\star - \max_{1\le t\le T} f(x_t^{\mathrm{BM}})$$

denote the simple regret of BayMOTH after T evaluations.

At round t, let $\mathcal{X}_t$ denote the candidate set considered by the optimization routine. We write the fallback 2-OPT acquisition as

$$A_t^{\mathrm{2OPT}}(x) := \Lambda_{0,t}(x) + \mathbb{E}_{GP_t}\left[\max_{x'\in\mathcal{X}_t} \Lambda_{1,t}(x';x)\right],$$

and let $A_t^{\mathrm{meta},\dagger}(x)$ denote the *oracle* meta-informed acquisition, defined as the exact meta-lookahead branch evaluated using the correct source task and exact expectation. Let $g_t^\dagger \in \{0,1\}$ denote the oracle branch selection decision, where $g_t^\dagger = 0$ selects the fallback (2-OPT) and $g_t^\dagger = 1$ selects the oracle meta branch. The resulting oracle-gated comparator is

$$A_t^\dagger(x) = (1-g_t^\dagger)A_t^{\mathrm{2OPT}}(x) + g_t^\dagger A_t^{\mathrm{meta},\dagger}(x).$$

Let

$$x_t^\dagger \in \arg\max_{x\in\mathcal{X}_t} A_t^\dagger(x), \qquad r_T^\dagger := f^\star - \max_{1\le t\le T} f(x_t^\dagger)$$

denote the corresponding oracle-gated simple regret.

BayMOTH itself uses a NCC based branch selection decision $g_t \in \{0,1\}$. When $g_t = 0$, it optimizes the same fallback 2-OPT branch. When $g_t = 1$, it optimizes a Monte Carlo approximation of the meta branch based on the selected alternative virtual environment. We denote the acquisition actually optimized by BayMOTH as $\hat{A}_t^{\mathrm{BM}}(x)$.

We make the following assumptions.

Assumption 1 (bounded objective). There exists $B > 0$ such that

$$0 \le f(x) \le B, \qquad \forall x \in D.$$

Assumption 2 (meta-vs-fallback branch disagreement probability). Let

$$\mathcal{E}_t^{\mathrm{route}} := \{g_t \ne g_t^\dagger\}$$

denote the event that BayMOTH disagrees with the oracle branch selection rule at round t. Assume

$$\Pr(\mathcal{E}_t^{\mathrm{route}}) \le p_t.$$

Assumption 3 (within-meta branch virtual environment selection error). For each round t, let $\chi_t^\dagger$ denote the oracle virtual environment index, i.e., the source task that the oracle meta branch would use when meta-routing is appropriate. Let $\hat{\chi}_t$ denote the virtual environment index selected by BayMOTH's NCC rule on rounds where $g_t = 1$.

Let

$$A_t^{\mathrm{meta}}(x;\chi)$$

denote the *common exact meta-acquisition template* evaluated using alternative virtual environment GP_χ and with the expectation computed exactly (that is, without Monte Carlo approximation). Then, on rounds where both BayMOTH and the oracle use the meta branch,

$$A_t^{\mathrm{meta},\dagger}(x) = A_t^{\mathrm{meta}}(x;\chi_t^\dagger), \qquad A_t^{\mathrm{meta},\star}(x) = A_t^{\mathrm{meta}}(x;\hat{\chi}_t),$$

and we assume

$$\sup_{x\in\mathcal{X}_t} \left| A_t^{\mathrm{meta}}(x;\hat{\chi}_t) - A_t^{\mathrm{meta}}(x;\chi_t^\dagger) \right| \le \eta_t.$$

If $(g_t, g_t^\dagger) \ne (1,1)$, this assumption is not invoked: the case $g_t \ne g_t^\dagger$ is handled by Assumption 2, and when $g_t = g_t^\dagger = 0$, both methods use the same fallback 2-OPT branch.

Assumption 4 (Monte Carlo approximation error). Let $\hat{J}_{t,M}(x)$ denote the Monte Carlo estimator used by BayMOTH in the meta branch with M samples. Assume

$$\mathrm{Var}(\hat{J}_{t,M}(x)) = \frac{\sigma_t^2(x)}{M},$$

and define

$$\sigma_t := \sup_{x\in\mathcal{X}_t} \sigma_t(x).$$

Assumption 5 (acquisition-to-value stability). There exists $L_t > 0$ such that for any two acquisitions A_t and $\tilde{A}_t$ on $\mathcal{X}_t$,

$$\sup_{x \in \mathcal{X}_t} |A_t(x) - \tilde{A}_t(x)| \leq \varepsilon \quad \Longrightarrow \quad f(x_{\tilde{A}_t}) - f(x_{A_t}) \leq L_t \varepsilon,$$

where $x_{A_t} \in \arg\max_{x \in \mathcal{X}_t} A_t(x)$ and similarly for $x_{\tilde{A}_t}$.

Proposition 1 (Informal simple-regret decomposition for BayMOTH)**.** *For any sequence $\{\delta_t\}_{t=1}^T$ with $\delta_t \in (0,1)$, with probability at least*

$$1 - \sum_{t=1}^{T} |\mathcal{X}_t| \delta_t - \sum_{t=1}^{T} p_t,$$

the simple regret of BayMOTH satisfies

$$r_T^{\mathrm{BM}} \leq r_T^{\dagger} + \max_{1 \leq t \leq T} g_t^{\dagger} L_t \left(\eta_t + \frac{\sigma_t}{\sqrt{M \delta_t}} \right).$$

Moreover,

$$\mathbb{E}[r_T^{\mathrm{BM}}] \leq \mathbb{E}[r_T^{\dagger}] + \max_{1 \leq t \leq T} g_t^{\dagger} L_t \left(\eta_t + \frac{\sigma_t}{\sqrt{M \delta_t}} \right) + B \left(\sum_{t=1}^{T} |\mathcal{X}_t| \delta_t + \sum_{t=1}^{T} p_t \right).$$

Proof sketch. The proof sketch is given in A.3 □

This proposition should be interpreted as a crude regret decomposition rather than as a sharp problem-dependent theorem. It shows that BayMOTH tracks an oracle-gated version of its own two-branch architecture. In particular, BayMOTH differs from it only through three BayMOTH-specific sources of error: meta-vs-fallback branch disagreement, mismatch between the selected alternative virtual environment and the oracle selection on meta-routed rounds, and Monte Carlo approximation error in the expectation.

4 Experiments

To position BayMOTH relative to competitive existing approaches, we compare against a broad set of strong baselines. These include established and state-of-the-art meta-BO methods, namely MetaBO (Volpp et al., 2019), NAP (Maraval et al., 2023), and HyperBO(Wang et al., 2024), as well as the robust multi-fidelity BO (rMF-MES) of Mikkola et al. (2023). We also include standard myopic and non-myopic BO baselines, GPBO-EI, GPBO-LogEI (Ament et al., 2023) and 2-OPT (Wu & Frazier, 2019), and additionally include a random search (RS) baseline. We evaluate across a variety of optimization tasks under a meta-learning context.

For scientific experiment optimization, we focus on synthetic inertial confinement fusion (ICF) tasks. Beyond assessing optimization performance, these experiments allow us to systematically study how different methods degrade as the distribution shift between source and test tasks increases. Robustness to source-test task mismatch is further investigated in Appendix B by making use of the benchmark functions for global optimization (Momin & Yang, 2013), specifically the Hartmann-6D and the Rosenbrock-6D function. With the experimental setup being similar to the setup found in Mikkola et al. (2023). For hyperparameter optimization tasks we use the HBO-B benchmark (Arango et al., 2021) and the HPOBench dataset (Eggensperger et al., 2021).

Our results are reported in terms of simple regret as a function of BO step, shown in the corresponding figures is the <> line and $\pm\sigma$ shaded region from optimization trajectories obtained over randomly initialized starting locations on the test task.

4.1 Synthetic ICF Experiments

Inertial confinement fusion (ICF) is a laser-driven scheme for nuclear fusion and offers a path to a clean and abundant energy source (Atzeni & Meyer-ter Vehn, 2004). A key hindrance towards the realization of optimal

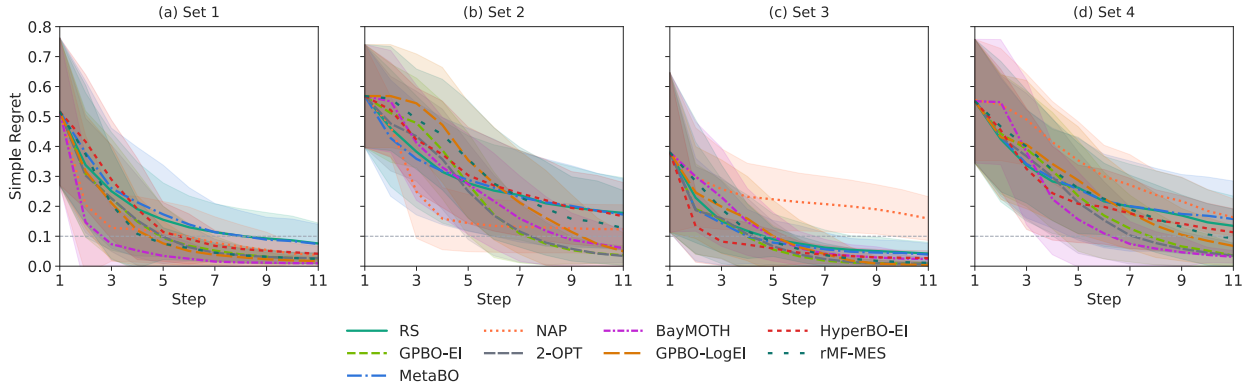


Figure 2: Optimization trajectories obtained over randomly initialized (100) starting locations across relatedness settings (Sets 1–4) for synthetic ICF experiments. Sample budget is restricted to 10, reflecting the realistic sample budget for ICF.

experimental designs is the limited predictive capability via physics modeling tools, therefore, approaches that use direct experimental feedback for optimization are increasingly becoming popular (Ejaz et al., 2024). However, as ICF experiments are extremely costly and allocation of facility time is competitive, we first demonstrate the practicality of BayMOTH for this purpose through synthetic ICF optimization experiments, similar to the studies of Gundecha et al. (2024) and Gutierrez et al. (2024). In addition, we study the behavior of BO approaches under progressively increasing distribution shift between the test task and the source tasks used in meta-training. Specifically, we consider four evaluation regimes spanning low, moderate, medium and high degrees of relatedness to the source tasks. This experimental design is intended to emulate realistic scenarios in which the assumed physics-based models are only weakly related, or potentially unrelated, to the experimental response. For the ICF relatedness experiments, we set γ=0.7 as in this setting source task usage remains active in roughly 90% of steps on the high-relatedness split. We then keep γ fixed across all other relatedness settings to avoid performance gains from set-specific retuning. Source task usage dependence on γ is provided in the ablation study in Appendix A.4. Details on task generation, source-test tasks sets' relatedness determination and visualizations of the sets are provided in Appendix B.2.

High Relatedness (Set 1): In the high-relatedness setting, BayMOTH achieves the strongest performance among all methods, both in terms of the initial rate of regret reduction and the final regret attained. This suggests that BayMOTH is able to effectively exploit highly related source task information when such information is available. NAP exhibits with a discernible advantage the second-fastest initial reduction in regret through approximately the first three optimization steps, indicating that its learned acquisition policy can make use of source task structure effectively early in the trajectory.

Medium/Moderate Relatedness (Set 2/3): In Set 2, NAP achieves the fastest initial rate of regret reduction up to approximately step 6. However, it attains less favorable final regret compared with BayMOTH. More broadly, the standard meta-BO methods, namely HyperBO, rMF-MES, MetaBO, and NAP, fail to reduce regret below 0.1 within the sample budget. In contrast, BayMOTH is able to reach the within 10% of the optimum region (< 0.1 simple regret). This behavior is consistent with the intended design of BayMOTH, where it explicitly gates the use of source task information and falls back to a robust lookahead policy when transfer is unreliable. We attribute the difficulty of this set to the lack of overlap between the optimum of the test task and the optima of the available source tasks, together with the smaller optimum region compared with the other ICF sets. As a result, methods that rely more directly on source task structure are more likely to propose samples that are informative under the meta-training distribution but suboptimal for the test task. In Set 3, HyperBO achieves the fastest initial reduction in regret. However, most methods are able to reduce regret below 0.1 well within the available sample budget, indicating that this setting is comparatively easier once a few samples have been collected. The main exception is NAP which achieves poor final regret relative to the other methods.

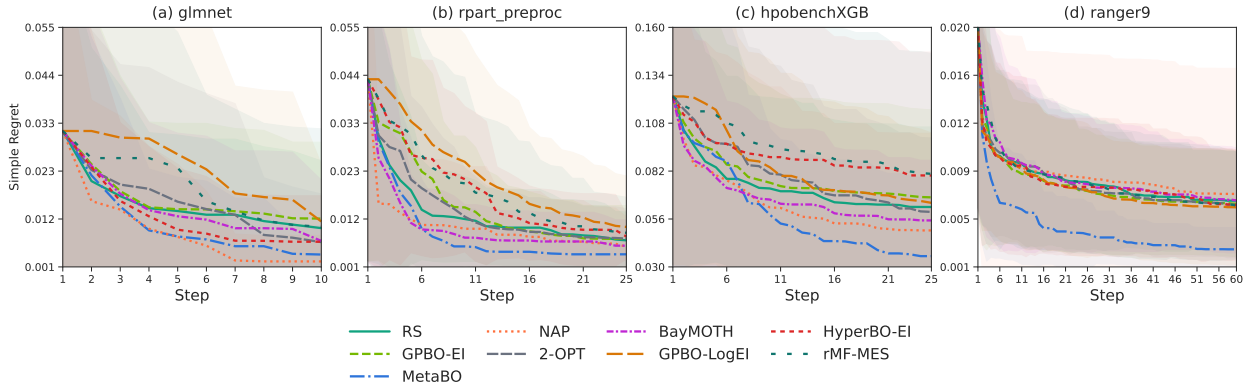


Figure 3: Simple regret trajectories on HPO search spaces. For each search space, results are aggregated across three randomly drawn test tasks, each evaluated from randomly initialized (10) starting locations.

Low Relatedness (Set 4): In the low-relatedness setting, the standard meta-BO methods HyperBO, MetaBO, and NAP all exhibit suboptimal performance, with MetaBO and NAP degrading most severely. NAP also demonstrates the slowest rate of regret reduction. This suboptimal behavior of meta-BO methods is consistent with source task mismatch. When the meta-training tasks provide misleading information about the test task, meta-learned policies can continue to exploit priors that are no longer useful. In contrast, BayMOTH remains substantially more robust because it is designed to disregard source-task information when the inferred relatedness is low and revert to the 2-OPT fallback policy. This allows BayMOTH to continue reducing regret even when source task information is weakly informative or misleading. We note that rMF-MES also remains effective in this setting as it is designed to guard against unreliable auxiliary information sources, however, it is less performant than BayMOTH.

General observations from all Sets: Overall, the synthetic ICF experiments highlight the central motivation of BayMOTH. When source tasks are highly related to the test task, BayMOTH is able to exploit this information effectively and achieve strong performance. When source-task relatedness decreases or becomes unreliable, BayMOTH remains robust and performant by avoiding over reliance on misleading meta-information. The results also show that performance in meta-BO is strongly method-dependent. HyperBO, MetaBO, and NAP respond differently across the ICF regimes, and NAP in particular exhibits the most degradation in mismatch settings. This is suggestive that learned policy and inductive bias of NAP are not as robust to mismatch for the synthetic ICF optimization problem class.

4.2 Hyper Parameter Optimization Experiments

To further assess the performance of BayMOTH in general Bayesian optimization settings, we experiment on hyperparameter optimization tasks from the HPO-B benchmark (Arango et al., 2021) and the HPOBench dataset (Eggensperger et al., 2021). These datasets contain accuracies of classification models as a function of model hyperparameters. The HPO-B datasets contains different search spaces comprising of model class, hyperparameter settings, and search domain which are evaluated on different supervised training datasets and splits. This benchmark provides predefined datasets to be used for meta-training, meta-validation and meta-testing. From this benchmark dataset we select the search spaces glmnet(ID:5860, dim=2), rpart.preproc(ID:4796, dim=3) and ranger9(ID:5965, dim=10). For these search spaces all meta-training datasets are used for training and evaluation is aggregated across three randomly drawn tasks from the meta-test datasets. We refer the reader to Arango et al. (2021) for more details on this benchmark. Pertaining to HPOBench (Eggensperger et al., 2021) we use the dataset utilized by Maraval et al. (2023), which is a hyperparameter optimization task (6D) for XGBoost. The same datasets as Maraval et al. (2023) are used for meta-training and testing is again aggregated across three randomly drawn tasks from the meta-test datasets as is specified in Maraval et al. (2023). For glmnet, rpart.preproc, XGBoost, and ranger, the number of meta-training datasets is 27, 36, 20, and 60, respectively. Consequently, here it is also tested how BayMOTH performs when a large number of virtual environments are induced for BayMOTH's task selection mechanism.

For each benchmark, the routing threshold γ is set to the average normalized cross-correlation (NCC) between the meta-validation and meta-training datasets; the corresponding values are provided in Appendix B.3.

Across the HPO tasks, BayMOTH exhibits consistently strong performance, both in terms of the initial rate of regret reduction and the final regret achieved. Overall, BayMOTH ranks among the top three methods across these tasks. On glmnet, HyperBO achieves a faster initial reduction in regret; however, BayMOTH matches the final regret attained by HyperBO, which is the third-best performing method on this task. On rpart.preproc and HPOBench XGBoost, BayMOTH substantially outperforms HyperBO while maintaining a top-three ranking, indicating stronger performance across general HPO settings. Overall, MetaBO attains the strongest performance across the HPO tasks, followed by NAP and BayMOTH. The ranger9 task is a notable exception in which MetaBO is highly dominant, while the remaining methods perform comparably.

5 Discussion

In this work, we introduce BayMOTH, a simple yet effective framework that unifies meta–Bayesian optimization and two-step lookahead BO. To the best of our knowledge, no prior work combines meta-BO with lookahead planning while explicitly accounting for the relatedness between source and target tasks and its impact on the robustness of the sampling policy. BayMOTH extends the 2-OPT acquisition function to operate with meta-information and also includes an immediate greedy improvement term. Both the horizon planning component and the immediate improvement component influence the sample suggestion strategy, as is analyzed in Appendix A.6. Despite the simplicity of BayMOTH, our experiments demonstrate that it is competitive with state-of-the-art meta-BO approaches. Moreover, by selectively incorporating meta-information, BayMOTH exhibits a robust sampling policy even when training tasks have limited similarity to the test task, a behavior that is observed across our experiments.

Our findings support a broader claim that performance in meta-BO and robustness to source–test mismatch is not automatic and is inherently method-dependent. Different methods encode different inductive biases and therefore respond differently across BO problem classes and source–test relatedness regimes. This is evident in the synthetic ICF experiments, where NAP achieves rapid early regret reduction in some regimes but exhibits degraded final regret across other sets. However, across all the HPO tasks, NAP exhibits strong performance. Nevertheless, across our experiments, BayMOTH demonstrates consistently strong performance across diverse BO settings, including synthetic ICF optimization, global optimization benchmarks (in Appendix B), and HPO tasks. Moreover, within individual problem classes, such as the ICF and global optimization benchmark settings, BayMOTH remains robust across source–test task relatedness mismatch.

We acknowledge that BayMOTH's interpretability and ease of implementation come at the cost of increased computational overhead, a systematic breakdown of the computational cost and comparison to offline trained meta-BO baselines is explored in Appendix A.7. More efficient optimization strategies (Wu & Frazier, 2019) for the 2-OPT acquisition function could therefore be explored for the horizon-planning term in Equation 1 for future versions of the method. Nevertheless, in practical HPO settings, where per-evaluation costs are often orders of magnitude larger this overhead is negligible. Moreover, our experiments suggest that BayMOTH often attains near-optimal configurations 2–3 evaluations earlier, making this trade-off favorable in practice. BayMOTH is particularly appealing in low to moderately high-dimensional settings where source task relatedness is uncertain and complicated training procedures are undesirable.

In future works, we plan to investigate learned task-similarity identification mechanisms and more sophisticated routing strategies spanning multiple environments. This can potentially further improve sample efficiency by allowing for exploitation of both global and local task features across multiple virtual environments. These directions may benefit from ideas drawn from mixture-of-experts (MoE) models (Mu & Lin, 2025). Additionally, we aim to further explore BayMOTH's core principle, leveraging meta-information when it is informative and reverting to a robust default strategy otherwise, through alternative approaches such as Q-transformers (Chebotar et al., 2023), which are well suited for multi-task learning and could naturally integrate within a MoE framework.

References

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.

Kenneth Scott Anderson. *Adiabat shaping in direct-drive inertial confinement fusion implosions*. University of Rochester, 2006.

Sebastian Pineda Arango, Hadi S Jomaa, Martin Wistuba, and Josif Grabocka. Hpo-b: A large-scale reproducible benchmark for black-box hpo based on openml. *arXiv preprint arXiv:2106.06257*, 2021.

Stefano Atzeni and Jürgen Meyer-ter Vehn. *The Physics of Inertial Fusion: BeamPlasma Interaction, Hydrodynamics, Hot Dense Matter*. Number 125. Oxford University Press, 2004.

Tianyi Bai, Yang Li, Yu Shen, Xinyi Zhang, Wentao Zhang, and Bin Cui. Transfer learning for bayesian optimization: A survey. *arXiv preprint arXiv:2302.05927*, 2023.

Maximilian Balandat, Brian Karrer, Daniel R. Jiang, Samuel Daulton, Benjamin Letham, Andrew Gordon Wilson, and Eytan Bakshy. BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization. In *Advances in Neural Information Processing Systems 33*, 2020. URL `http://arxiv.org/abs/1910.06403`.

R Betti, AR Christopherson, BK Spears, R Nora, A Bose, J Howard, KM Woo, MJ Edwards, and J Sanz. Alpha heating and burning plasmas in inertial confinement fusion. *Physical review letters*, 114(25):255003, 2015.

Ricardo Betti, PY Chang, BK Spears, KS Anderson, J Edwards, M Fatenejad, JD Lindl, RL McCrory, R Nora, and D Shvarts. Thermonuclear ignition in inertial confinement fusion and comparison with magnetic confinement. *Physics of Plasmas*, 17(5), 2010.

Christopher M Bishop and Hugh Bishop. *Deep learning: Foundations and concepts*. Springer Nature, 2023.

Yevgen Chebotar, Quan Vuong, Karol Hausman, Fei Xia, Yao Lu, Alex Irpan, Aviral Kumar, Tianhe Yu, Alexander Herzog, Karl Pertsch, et al. Q-transformer: Scalable offline reinforcement learning via autoregressive q-functions. In *Conference on Robot Learning*, pp. 3909–3928. PMLR, 2023.

Mujin Cheon, Jay H Lee, Dong-Yeun Koh, and Calvin Tsay. Earl-bo: Reinforcement learning for multi-step lookahead, high-dimensional bayesian optimization. *arXiv preprint arXiv:2411.00171*, 2024.

AR Christopherson, R Betti, J Howard, KM Woo, A Bose, EM Campbell, and V Gopalaswamy. Theory of alpha heating in inertial fusion: Alpha-heating metrics and the onset of the burning-plasma regime. *Physics of Plasmas*, 25(7), 2018.

AR Christopherson, R Betti, S Miller, V Gopalaswamy, OM Mannion, and D Cao. Theory of ignition and burn propagation in inertial fusion implosions. *Physics of Plasmas*, 27(5), 2020.

RS Craxton, KS Anderson, TR Boehly, VN Goncharov, DR Harding, JP Knauer, RL McCrory, PW McKenty, DD Meyerhofer, JF Myatt, et al. Direct-drive inertial confinement fusion: A review. *Physics of Plasmas*, 22(11), 2015.

J Delettrez, R Epstein, MC Richardson, PA Jaanimagi, and BL Henke. Effect of laser illumination nonuniformity on the analysis of time-resolved x-ray measurements in uv spherical transport experiments. *Physical Review A*, 36(8):3926, 1987.

Katharina Eggensperger, Philipp Müller, Neeratyoy Mallik, Matthias Feurer, René Sass, Aaron Klein, Noor Awad, Marius Lindauer, and Frank Hutter. Hpobench: A collection of reproducible multi-fidelity benchmark problems for hpo. *arXiv preprint arXiv:2109.06716*, 2021.

Rahman Ejaz, Varchas Gopalaswamy, Aarne Lees, Duc Cao, Soumyendu Sarkar, and Christopher Kanan. Direct-drive implosion performance optimization using gaussian process modeling and reinforcement learning. In *APS Division of Plasma Physics Meeting Abstracts*, volume 2023, pp. BO07–007, 2023.

Rahman Ejaz, Varchas Gopalaswamy, Ricardo Luna, Vineet Gundecha, Aarne Lees, Riccardo Betti, Sahand Ghorbanpour, Soumyendu Sarkar, and Christopher Kanan. Optimizing the performance of direct-drive implosion experiments using meta-bayesian optimization. *Bulletin of the American Physical Society*, 2024.

Matthias Feurer and Frank Hutter. Hyperparameter optimization. In *Automated machine learning: Methods, systems, challenges*, pp. 3–33. Springer International Publishing Cham, 2019.

Matthias Feurer, Benjamin Letham, and Eytan Bakshy. Scalable meta-learning for bayesian optimization. *stat*, 1050(6):7, 2018.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pp. 1126–1135. PMLR, 2017.

Peter I Frazier. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

Peter I Frazier and Jialei Wang. Bayesian optimization for materials design. In *Information science for materials discovery and design*, pp. 45–75. Springer, 2015.

Varchas Gopalaswamy, Riccardo Betti, JP Knauer, N Luciani, D Patel, KM Woo, A Bose, IV Igumenshchev, EM Campbell, KS Anderson, et al. Tripled yield in direct-drive laser fusion through statistical modelling. *Nature*, 565(7741):581–586, 2019.

Vineet Gundecha, Ricardo Luna Gutierrez, Sahand Ghorbanpour, Rahman Ejaz, Varchas Gopalaswamy, Riccardo Betti, Avisek Naug, Paolo Faraboschi, and Soumyendu Sarkar. Meta-learned bayesian optimization for energy yield in inertial confinement fusion. In *NeurIPS CCAI Workshop*, 2024.

Ricardo Luna Gutierrez, Sahand Ghorbanpour, Vineet Gundecha, Rahman Ejaz, Varchas Gopalaswamy, Riccardo Betti, Avisek Naug, Desik Rengarajan, Ashwin Ramesh Babu, Paolo Faraboschi, et al. Explainable meta bayesian optimization with human feedback for scientific applications like fusion energy. In *NeurIPS 2024 Workshop on Tackling Climate Change with Machine Learning*, 2024.

Bing-Jing Hsieh, Ping-Chun Hsieh, and Xi Liu. Reinforced few-shot acquisition function learning for bayesian optimization. *Advances in Neural Information Processing Systems*, 34:7718–7731, 2021.

Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.

A Lees, R Betti, JP Knauer, V Gopalaswamy, D Patel, KM Woo, KS Anderson, EM Campbell, D Cao, J Carroll-Nellenback, et al. Experimentally inferred fusion yield dependencies of omega inertial confinement fusion implosions. *Physical review letters*, 127(10):105001, 2021.

A Lees, R Betti, V Gopalaswamy, JP Knauer, LA Ceurvorst, D Patel, R Ejaz, KS Anderson, KA Bauer, MJ Bonino, et al. Approaching hydro-equivalent ignition in laser direct-drive via target design optimization using novel statistical modeling. *Physics of Plasmas*, 32(7), 2025.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Zeyuan Ma, Hongshu Guo, Yue-Jiao Gong, Jun Zhang, and Kay Chen Tan. Toward automated algorithm design: A survey and practical guide to meta-black-box-optimization. *IEEE Transactions on Evolutionary Computation*, 2025.

Alexandre Maraval, Matthieu Zimmer, Antoine Grosnit, and Haitham Bou Ammar. End-to-end meta-bayesian optimisation with transformer neural processes. *Advances in Neural Information Processing Systems*, 36: 11246–11260, 2023.

Petrus Mikkola, Julien Martinelli, Louis Filstroff, and Samuel Kaski. Multi-fidelity bayesian optimization with unreliable information sources. In *International Conference on Artificial Intelligence and Statistics*, pp. 7425–7454. PMLR, 2023.

Jamil Momin and Xin-She Yang. A literature survey of benchmark functions for global optimization problems. *Journal of Mathematical Modelling and Numerical Optimisation*, 4(2):150–194, 2013.

Siyuan Mu and Sen Lin. A comprehensive survey of mixture-of-experts: Algorithms, theory, and applications. *arXiv preprint arXiv:2503.07137*, 2025.

PB Radha, TJB Collins, JA Delettrez, Y Elbaz, R Epstein, V Yu Glebov, VN Goncharov, RL Keck, JP Knauer, JA Marozas, et al. Multidimensional analysis of direct-drive, plastic-shell implosions on omega. *Physics of Plasmas*, 12(5), 2005.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012.

Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

Michael Volpp, Lukas P Fröhlich, Kirsten Fischer, Andreas Doerr, Stefan Falkner, Frank Hutter, and Christian Daniel. Meta-learning acquisition functions for transfer learning in bayesian optimization. *arXiv preprint arXiv:1904.02642*, 2019.

Zi Wang, George E Dahl, Kevin Swersky, Chansoo Lee, Zachary Nado, Justin Gilmer, Jasper Snoek, and Zoubin Ghahramani. Pre-trained gaussian processes for bayesian optimization. *Journal of Machine Learning Research*, 25(212):1–83, 2024.

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

Martin Wistuba, Nicolas Schilling, and Lars Schmidt-Thieme. Scalable gaussian process-based transfer surrogates for hyperparameter optimization. *Machine Learning*, 107(1):43–78, 2018.

Jian Wu and Peter Frazier. Practical two-step lookahead bayesian optimization. *Advances in neural information processing systems*, 32, 2019.

Mingzhang Yin, George Tucker, Mingyuan Zhou, Sergey Levine, and Chelsea Finn. Meta-learning without memorization. *arXiv preprint arXiv:1912.03820*, 2019.

Jaesik Yoon, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. Bayesian model-agnostic meta-learning. *Advances in neural information processing systems*, 31, 2018.

Zhe Zhao, Lichan Hong, Li Wei, Jilin Chen, Aniruddh Nath, Shawn Andrews, Aditee Kumthekar, Maheswaran Sathiamoorthy, Xinyang Yi, and Ed Chi. Recommending what video to watch next: a multitask ranking system. In *Proceedings of the 13th ACM conference on recommender systems*, pp. 43–51, 2019.

Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.

A BayMOTH details and analysis

A.1 BayMOTH pseudo code

Algorithm 1 BayMOTH

Require: Meta-training datasets $\mathcal{D}_{\text{meta-train}} = \{\mathcal{D}_\chi\}_{\chi=1}^{\tau}$; greedy balancing parameter α; correlation threshold γ; kernel $k(\cdot,\cdot)$ and $\mathcal{GP}$ hyperparameters θ; unseen-task history $\mathcal{H}_0 = \emptyset$; domain $x \in D$; optimization routine over x; sample budget T.

alternative virtual environments: For each $\chi \in \{1,\dots,\tau\}$ fit $\mathcal{GP}_\chi$ using $\mathcal{D}_\chi$.
for $t = 1, 2, \dots, T$ **do**
  Fit target surrogate: Fit $\mathcal{GP}_t$ on $\mathcal{H}_t = \{(x_i, f(x_i)\}_{i=1}^{t}$ using (k, θ).
  Compute predictive mean $\mu_t(\cdot)$ from $\mathcal{GP}_t$.

  Construct virtual histories and select closest task:
  for $\chi = 1, 2, \dots, \tau$ **do**
    Form virtual history $\tilde{\mathcal{H}}_t^\chi = \{(x_i, \tilde{f}_\chi(x_i)\}_{i=1}^{t}$ with $\tilde{f}_\chi = \mu_\chi(x_i)$ from $\mathcal{GP}_\chi$.
    Fit auxiliary $\tilde{\mathcal{GP}}_{\chi,t}$ on $\tilde{\mathcal{H}}_t^\chi$ using the same (k, θ).
    Compute predictive mean $\tilde{\mu}_{\chi,t}(\cdot)$ from $\tilde{\mathcal{GP}}_{\chi,t}$.
    Compute similarity score

$$\mathcal{S}_{\chi,t} \leftarrow \mathrm{NCC}(\mu_t, \tilde{\mu}_{\chi,t})$$

  end for

  Sample selection:
  if $\max_{\chi\in\{1,\dots,\tau\}} \mathcal{S}_{\chi,t} > \gamma$ **then** ▷ correlation threshold γ
    $\chi^\star \leftarrow \arg\max_{\chi\in\{1,\dots,\tau\}} \mathcal{S}_{\chi,t}$.
    Compute x_{t+1} **using optimization routine:**
    $x_{t+1} \leftarrow \arg\max_{x_{t+1}} \Big[\Lambda_0(x_{t+1}) + \mathbb{E}_{\mathcal{GP}_{\chi^\star}}\Big[(1-\alpha)\max_{x_{t+2}} \Lambda_1(x_{t+2}) + \alpha\Omega\Big]\Big]$ ▷ π_{BayMOTH}
  else
    Compute x_{t+1} **using optimization routine:**
    $x_{t+1} \leftarrow \arg\max_{x_{t+1}} \Big[\Lambda_0(x_{t+1}) + \mathbb{E}_{\mathcal{GP}_t}\Big[\max_{x_{t+2}} \Lambda_1(x_{t+2})\Big]\Big]$ ▷ fallback
  end if
  Evaluate the true objective: $f_{t+1} \leftarrow f(x_{t+1})$.
  Update history: $\mathcal{H}_{t+1} \leftarrow \mathcal{H}_t \cup \{(x_{t+1}, f_{t+1})\}$.
end for

A.2 Proof of properties of the Monte Carlo Estimator

Consider the second term in Eq.1, also shown below

$$\mathbb{J}(\omega) = \mathbb{E}_{\omega\sim\mathcal{GP}_{\chi^*}}[\mathbb{Z}(\omega)], \qquad \mathbb{Z}(\omega) = (1-\alpha)\max_{x_{t+2}} \Lambda_1(x_{t+2};\omega) + \alpha\Omega(\omega)$$

to approximate $\mathbb{J}$ we draw M iid samples $\omega^{(1)}, \omega^{(2)}, \dots, \omega^{(M)} \sim \mathcal{GP}_{\chi^*}$ and compute the estimator

$$\hat{J}_M \;=\; \frac{1}{M}\sum_{m=1}^{M} \mathbb{Z}^{(m)}, \qquad \mathbb{Z}^{(m)} = \mathbb{Z}(\omega^{(m)})$$

The statistical guarantees for $\hat{J}_M$ are outlined below.

Estimator is unbiased:

$$
\begin{aligned}
\mathbb{E}[\hat{J}_M] &= \mathbb{E}\left[\frac{1}{M}\sum_{m=1}^{M}\mathbb{Z}^{(m)}\right] \\
&= \frac{1}{M}\sum_{m=1}^{M}\mathbb{E}[\mathbb{Z}^{(m)}] && \text{(linearity of expectation)} \\
&= \frac{1}{M}\sum_{m=1}^{M}\mathbb{E}[\mathbb{Z}] && (\mathbb{Z}^{(1)},\ldots,\mathbb{Z}^{(M)} \text{ identically distributed}) \\
&= \frac{1}{M}\,M\,\mathbb{E}[\mathbb{Z}] \\
&= \mathbb{E}[\mathbb{Z}]\,.
\end{aligned}
$$

Estimator variance:

$$
\begin{aligned}
\mathbb{V}[\hat{J}_M] &= \mathbb{E}[\hat{J}_M^2] - \mathbb{E}[\hat{J}_M]^2 \\
&= \mathbb{E}\Big[\Big(\frac{1}{M}\sum_{m=1}^{M}\mathbb{Z}^{(m)}\Big)^2\Big] - \mathbb{E}[\mathbb{Z}]^2 \\
&= \mathbb{E}\Big[\frac{1}{M^2}\sum_{m=1}^{M}(\mathbb{Z}^{(m)})^2 + \frac{1}{M^2}\sum_{m=1}^{M}\sum_{k=1,k\neq m}^{M}\mathbb{Z}^{(m)}\mathbb{Z}^{(k)}\Big] - \mathbb{E}[\mathbb{Z}]^2 \\
&= \frac{1}{M^2}\sum_{m=1}^{M}\mathbb{E}[(\mathbb{Z}^{(m)})^2] + \frac{1}{M^2}\sum_{m=1}^{M}\sum_{k=1,k\neq m}^{M}\mathbb{E}[\mathbb{Z}^{(m)}\mathbb{Z}^{(k)}] - \mathbb{E}[\mathbb{Z}]^2 \\
&= \frac{1}{M^2}\sum_{m=1}^{M}\mathbb{E}[(\mathbb{Z}^{(m)})^2] + \frac{1}{M^2}\sum_{m=1}^{M}\sum_{k=1,k\neq m}^{M}\mathbb{E}[\mathbb{Z}^{(m)}]\mathbb{E}[\mathbb{Z}^{(k)}] - \mathbb{E}[\mathbb{Z}]^2 \\
&= \frac{1}{M^2}\sum_{m=1}^{M}\mathbb{E}[\mathbb{Z}^2] + \frac{1}{M^2}\sum_{m=1}^{M}\sum_{k=1,k\neq m}^{M}\mathbb{E}[\mathbb{Z}]\mathbb{E}[\mathbb{Z}] - \mathbb{E}[\mathbb{Z}]^2 \\
&= \frac{1}{M^2}M * \mathbb{E}[\mathbb{Z}^2] + \frac{1}{M^2}M(M-1)\mathbb{E}[\mathbb{Z}]^2 - \mathbb{E}[\mathbb{Z}]^2 \\
&= \frac{1}{M}\mathbb{E}[\mathbb{Z}^2] - \frac{1}{M}\mathbb{E}[\mathbb{Z}]^2 \\
&= \frac{\sigma^2}{M}
\end{aligned}
$$

A.3 Proofs

Proof sketch of Proposition 1. Let $\mathcal{G}_T$ denote the event that, for every round $t \le T$, BayMOTH agrees with the oracle branch decision and the Monte Carlo estimator satisfies the required uniform accuracy bound over $\mathcal{X}_t$.

We first work on the event $\mathcal{G}_T$.

Fix a round t. Since $g_t = g_t^\dagger$ on $\mathcal{G}_T$, there are two cases.

If $g_t^\dagger = 0$, both BayMOTH and the oracle comparator optimize the same fallback acquisition A_t^{2OPT}, and hence there is no acquisition discrepancy at round t.

If $g_t^\dagger = 1$, both procedures use the meta branch, and the discrepancy can be decomposed as

$$\hat{A}_t^{\mathrm{BM}}(x) - A_t^\dagger(x) = \underbrace{\left(\hat{A}_t^{\mathrm{BM}}(x) - A_t^{\mathrm{meta},\star}(x)\right)}_{\text{Monte Carlo error}} + \underbrace{\left(A_t^{\mathrm{meta},\star}(x) - A_t^{\mathrm{meta},\dagger}(x)\right)}_{\text{virtual-environment mismatch}}.$$

Under Assumption 3,

$$\sup_{x\in\mathcal{X}_t} \left|A_t^{\mathrm{meta},\star}(x) - A_t^{\mathrm{meta},\dagger}(x)\right| \le \eta_t.$$

For the Monte Carlo term, Assumption 4 and Chebyshev's inequality imply that for each fixed $x \in \mathcal{X}_t$,

$$\Pr\left(|\hat{J}_{t,M}(x) - J_t^{\mathrm{meta},\star}(x)| \ge \frac{\sigma_t}{\sqrt{M\delta_t}}\right) \le \delta_t.$$

Equivalently, for each fixed $x \in \mathcal{X}_t$, with probability at least $1 - \delta_t$,

$$|\hat{J}_{t,M}(x) - J_t^{\mathrm{meta},\star}(x)| \le \frac{\sigma_t}{\sqrt{M\delta_t}}.$$

Now define the bad event

$$\mathcal{B}_t := \left\{\exists x \in \mathcal{X}_t : |\hat{J}_{t,M}(x) - J_t^{\mathrm{meta},\star}(x)| > \frac{\sigma_t}{\sqrt{M\delta_t}}\right\}.$$

By the union bound,

$$\Pr(\mathcal{B}_t) \le \sum_{x\in\mathcal{X}_t} \Pr\left(|\hat{J}_{t,M}(x) - J_t^{\mathrm{meta},\star}(x)| > \frac{\sigma_t}{\sqrt{M\delta_t}}\right) \le |\mathcal{X}_t|\,\delta_t.$$

Hence, with probability at least $1 - |\mathcal{X}_t|\delta_t$,

$$\forall x \in \mathcal{X}_t, \qquad |\hat{J}_{t,M}(x) - J_t^{\mathrm{meta},\star}(x)| \le \frac{\sigma_t}{\sqrt{M\delta_t}}.$$

Since the Monte Carlo approximation enters through the acquisition, this implies

$$\sup_{x\in\mathcal{X}_t} \left|\hat{A}_t^{\mathrm{BM}}(x) - A_t^{\mathrm{meta},\star}(x)\right| \le \frac{\sigma_t}{\sqrt{M\delta_t}}.$$

Therefore, on $\mathcal{G}_T$,

$$\sup_{x\in\mathcal{X}_t} \left|\hat{A}_t^{\mathrm{BM}}(x) - A_t^\dagger(x)\right| \le \sup_{x\in\mathcal{X}_t} \left|\hat{A}_t^{\mathrm{BM}}(x) - A_t^{\mathrm{meta},\star}(x)\right| + \sup_{x\in\mathcal{X}_t} \left|A_t^{\mathrm{meta},\star}(x) - A_t^{\mathrm{meta},\dagger}(x)\right| \le \eta_t + \frac{\sigma_t}{\sqrt{M\delta_t}}.$$

Multiplying by $g_t^\dagger$ makes the bound valid uniformly over both fallback and meta rounds:

$$\sup_{x\in\mathcal{X}_t} \left|\hat{A}_t^{\mathrm{BM}}(x) - A_t^\dagger(x)\right| \le g_t^\dagger \left(\eta_t + \frac{\sigma_t}{\sqrt{M\delta_t}}\right).$$

Assumption 5 then transfers this to an objective-value difference:

$$f(x_t^\dagger) - f(x_t^{\mathrm{BM}}) \le g_t^\dagger L_t \left(\eta_t + \frac{\sigma_t}{\sqrt{M\delta_t}} \right).$$

Since simple regret depends on the best value attained over all rounds, we obtain on $\mathcal{G}_T$

$$r_T^{\mathrm{BM}} \le r_T^\dagger + \max_{1 \le t \le T} g_t^\dagger L_t \left(\eta_t + \frac{\sigma_t}{\sqrt{M\delta_t}} \right).$$

Finally, by Assumption 2 and a union bound over rounds,

$$\Pr(\mathcal{G}_T^c) \le \sum_{t=1}^{T} |\mathcal{X}_t| \delta_t + \sum_{t=1}^{T} p_t,$$

which gives the stated high-probability bound.

For the expectation bound, decompose over $\mathcal{G}_T$ and $\mathcal{G}_T^c$. On $\mathcal{G}_T$, the above regret bound holds. On $\mathcal{G}_T^c$, Assumption 1 gives the crude bound $r_T^{\mathrm{BM}} \le B$. This yields

$$\mathbb{E}[r_T^{\mathrm{BM}}] \le \mathbb{E}[r_T^\dagger] + \max_{1 \le t \le T} g_t^\dagger L_t \left(\eta_t + \frac{\sigma_t}{\sqrt{M\delta_t}} \right) + B \left(\sum_{t=1}^{T} |\mathcal{X}_t| \delta_t + \sum_{t=1}^{T} p_t \right).$$

□

A.4 Ablation on γ

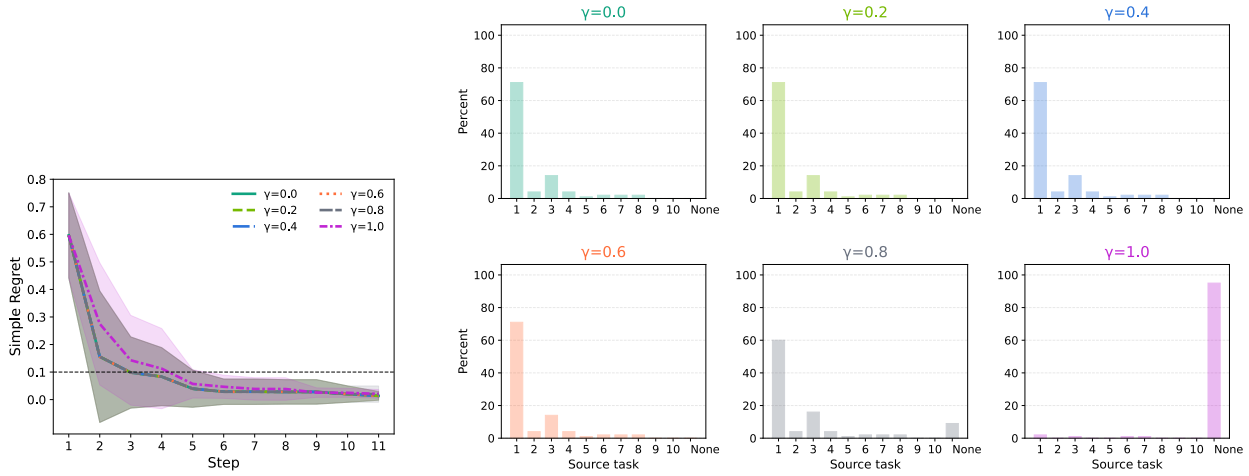


Figure 4: Set 1 (highest NCC in set ≈ 0.85) trajectory and source task usage across γ values.

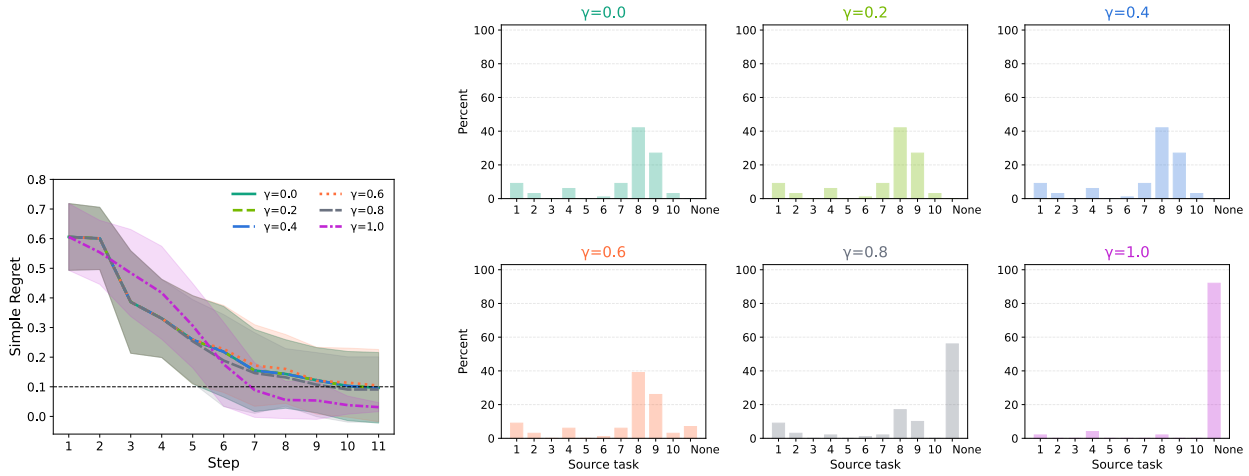


Figure 5: Set 2 (highest NCC in set ≈ 0.60) trajectory and source task usage across γ values.

This ablation studies the effect of the NCC threshold γ. This threshold controls the extent to which BayMOTH relies on source task transfer. We sweep γ and record the distribution of selected source tasks, measured as the percentage of total optimization steps across randomly initialized runs.

The results show that when γ is below the NCC value of the most correlated source task, the optimization trajectory remains largely unchanged. This is consistent with the source task usage distributions which are also nearly identical across these settings. Once γ exceeds the NCC of the most correlated source task, however, the method enters a weak-transfer regime: usage of the highest-NCC source task drops substantially, and BayMOTH increasingly selects no source task as $\gamma \to 1$. However, source-task usage does not vanish entirely even when $\gamma = 1$. This residual usage occurs in the early BO rounds, before enough target task samples have been acquired to estimate NCC reliably and rule out source task transfer. Thus, $\gamma = 1$ should be interpreted as strongly suppressing transfer once sufficient evidence is available, rather than disabling source task usage from the start.

On the high-relatedness set, increasing γ from 0.8 to 1.0 slows the initial reduction in regret, while leaving the final regret essentially unchanged. This behavior is consistent with the corresponding shift in source task usage, from predominantly selecting source task 1 and source task 3 to predominantly selecting no source

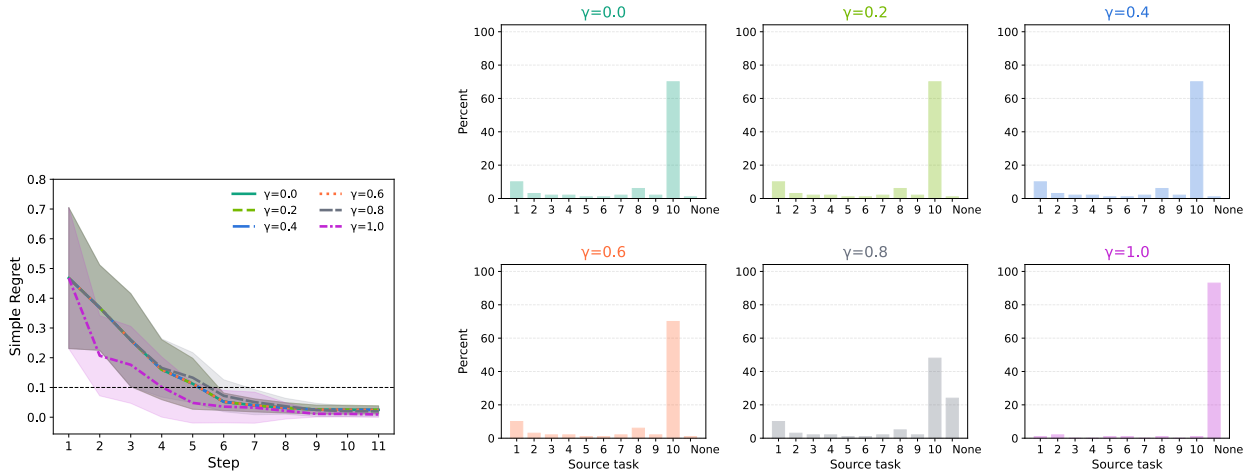


Figure 6: Set 3 (highest NCC in set ≈ 0.82) trajectory and source task usage across γ values.

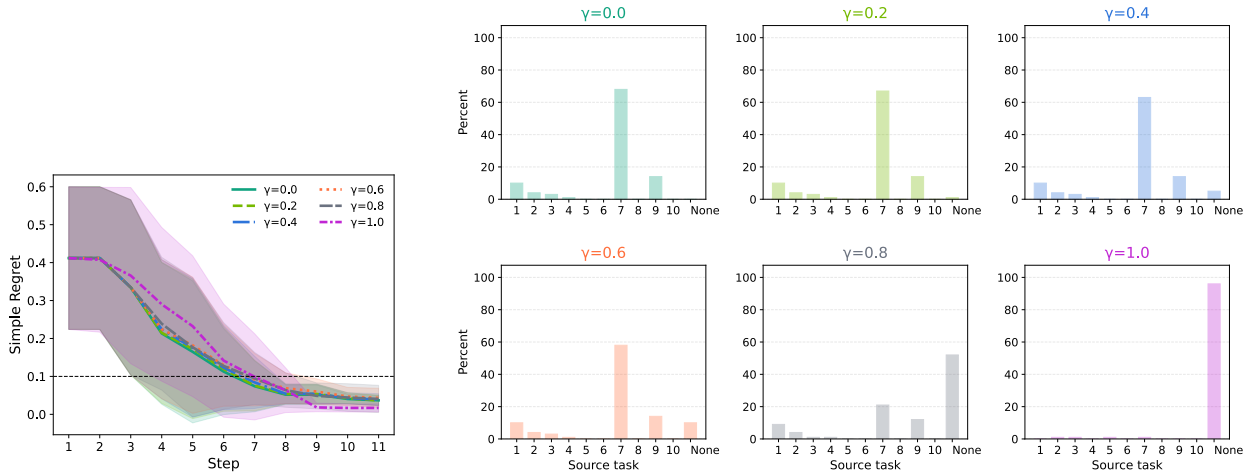


Figure 7: Set 4 (highest NCC in set ≈ 0.10) trajectory and source task usage across γ values.

task. On the low-relatedness set, the $\gamma = 1.0$ setting also yields a noticeably slower initial reduction in regret than the other threshold values, but achieves the best final performance, with regret approximately 2% lower than the other settings.

Overall, this ablation characterizes BayMOTH source task usage behavior across the spectrum from strong transfer to near always-fallback behavior.

A.5 Ablation on estimator number of samples M

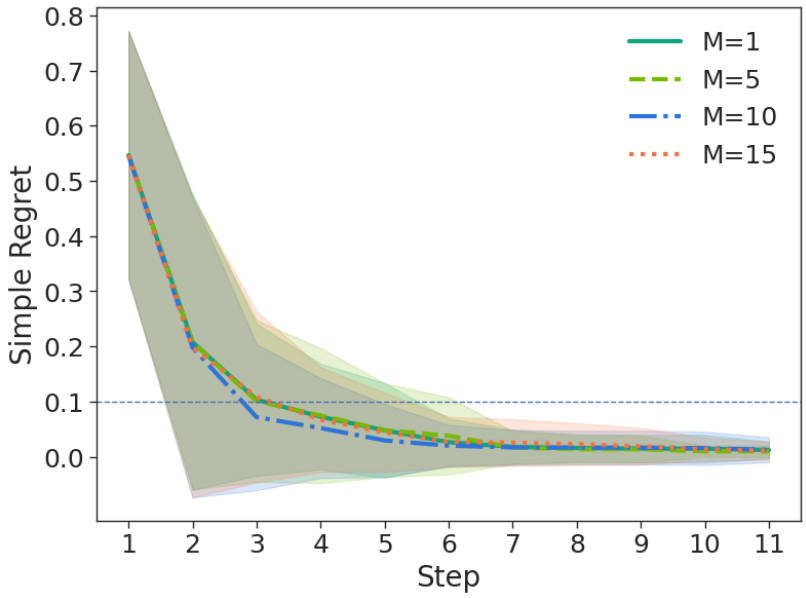


Figure 8: Scan on the number of samples for the Monte Carlo estimator, using set 1 of the synthetic ICF dataset. Shown is <> line and $\pm\sigma$ shaded region over 30 randomly initialized starting locations.

Here, we investigate the effect of the sampling number for the Monte Carlo estimator of the expectation term in Equation 1. We find that BayMOTH is only weakly sensitive to the choice of M on Set 1 of the synthetic ICF dataset. As is shown in Figure 8, increasing M beyond M=1 yields only marginal performance improvements, with M=10 performing the best, without a clear monotonic trend. This behavior is consistent with the chosen $\mathcal{GP}_{\chi^*}$ posterior being well-constrained in our experimental settings. Consequently, the estimator variance is small and a modest number of samples is sufficient. We anticipate that the dependence on M would become more pronounced in settings where the posterior variance of $\mathcal{GP}_{\chi^*}$ is larger, such as under severe data sparsity, higher-dimensional domains, or strongly misspecified kernels, where additional samples may be required to stabilize the estimator. We anticipate the M number to be treated as a hyperparameter which can be tuned using the training/validation tasks for a specific problem domain.

A.6 Ablation on α

Here, again using set 1 from the ICF datasets, we study the effect of the greedy balancing parameter α that controls the relative weighting between the immediate greedy improvement term and the meta-informed lookahead term in BayMOTH's policy. Several observations emerge. First, even for $\alpha = 0$ which corresponds to pure horizon planning under $\mathcal{GP}_{\chi^*}$, BayMOTH outperforms the 2-OPT baseline during the initial 5 sample steps (approx. 10% difference at sample step 3). In the domain of ICF experiments this is significant as sample evaluation costs are extremely high (500k USD - 1M USD per sample). This indicates that incorporating meta-information into Λ_1 provides a meaningful advantage in meta-BO settings even in the absence of the greedy improvement term. At the same time, intermediate values of α (specifically $\alpha \in \{0.25, 0.5\}$) yield the strongest overall performance and exhibit comparable behavior, suggesting that jointly leveraging both immediate and lookahead components is beneficial. While $\alpha = 1$, which places full emphasis on the immediate greedy improvement term, still improves upon $\alpha = 0$, it slightly under-performs relative to intermediate settings. This pattern indicates that neither component alone is sufficient to fully capture the benefits of the policy; rather, their combination provides a more effective balance between short-horizon planning and meta-informed greedy exploitation. Therefore, the greedy balancing parameter should be treated as hyperparameter by a user and based on the nature of the problem domain and goals should be tuned.

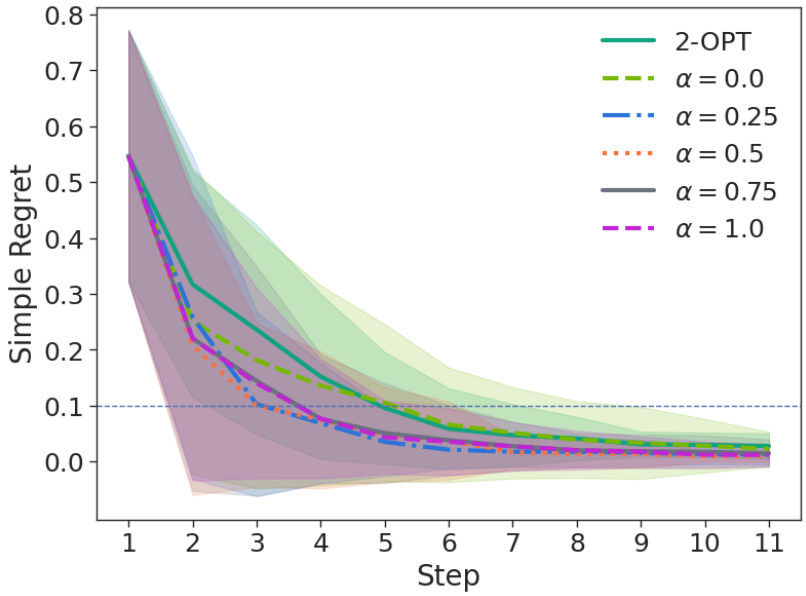


Figure 9: Ablation on the greedy balancing parameter α. Shown is <> line and $\pm\sigma$ shaded region over 30 randomly initialized starting locations.

A.7 Computational cost analysis

BayMOTH is not intended to reduce computational cost on a per-decision basis relative to offline-trained meta-BO methods. Instead, it makes a different computational tradeoff, rather than incurring the complexity of offline RL-based meta-training, it shifts computation to online planning within a simple and interpretable framework. Its intended advantage is therefore not lower proposal-time cost, but algorithmic simplicity, explicit decision logic, and the avoidance of a separate offline policy-training phase. In the settings considered in this paper, this tradeoff is favorable for the reasons discussed in the main text.

To make this tradeoff explicit, we compare the offline RL meta-training cost of MetaBO and NAP against BayMOTH's online cost on the `ranger9` test problem ($d = 10$), and additionally provide a profiling-based breakdown of BayMOTH's internal runtime components.

In our implementation, the source task GPs are fit once offline, while the comparison GPs used for virtual-environment selection are updated after each real observation. At proposal time, source task selection itself is lightweight and requires only NCC scoring across candidate source tasks. The dominant computational cost does not arise from this GP-based selection mechanism, but from the nested two-step acquisition optimization used to generate each proposal. Specifically, each proposal is obtained using multi-restart L-BFGS-B optimization (3 restarts, up to 100 iterations per restart). Each acquisition evaluation then performs a horizon-step simulation, which, over M Monte Carlo samples, requires future-GP fitting and future-EI evaluation. As a result, BayMOTH's runtime is driven primarily by the 2-OPT-style horizon-planning term rather than by the transfer-selection stage itself.

Overall, Table 2 shows that BayMOTH's computational overhead is dominated by explicit online lookahead planning, not by the GP-based source task selection mechanism. In particular, the overwhelming majority of proposal-time cost is spent inside acquisition evaluation and horizon-step simulation, while source task selection and orchestration contribute only a small fraction of runtime. This makes the computational tradeoff clear: BayMOTH avoids offline RL meta-training, but pays for this simplicity through a more expensive online planning step.

Computational resources. All experiments were conducted on a single workstation with an NVIDIA Quadro RTX 6000 GPU and an Intel(R) Xeon(R) W-2125 CPU @ 4.00GHz.

Table 1: High-level computational cost profile of BayMOTH and offline-trained baselines.

Method	Cost profile
BayMOTH	Offline source task GP meta-training (cheap $\approx$ 40s); primary computational cost is paid online during proposal generation.
MetaBO	Main cost is offline PPO training; approximately 4 hours for 2000 PPO iterations.
NAP	Main cost is offline PPO training; approximately 11 hours for 2000 PPO iterations.

Table 2: Profiling-based breakdown of BayMOTH proposal-time cost on `ranger9` ($d = 10$).

BayMOTH stage	Frequency	Share of proposal time	Avg per call	Meaning
source task GP training	once	N/A	40.00 s	One-time offline training.
Comparison GP update	once after observation	N/A	0.234 s	Online overhead outside the proposer.
Proposal optimization	once per suggestion	100.00%	44.84 s	Full online proposal cost.
BayMOTH acquisition evaluation	inside L-BFGS-B	98.66%	145.85 ms	Nearly all proposal-time computation is spent here.
Horizon-step simulation	once per acquisition evaluation	97.99%	144.85 ms	Main bottleneck.
Future EI (inside horizon)	M times per acquisition evaluation	72.17%	21.34 ms	Repeated future-EI computation dominates the inner loop.
Current EI (outside horizon)	once per acquisition evaluation	0.67%	0.99 ms	Negligible contribution.
Selection + NCC + optimizer orchestration	once per suggestion	1.34%	0.60 s	Upper bound on source selection and orchestration overhead.

B Additional experiments and details

B.1 Global optimization benchmark functions experiments

Here, we use an experimental setup similar to Mikkola et al. (2023) to further probe the robustness against source-test task mismatch. We make use of the Hartmann-6D and Rosenbrock-6D global optimization benchmark functions (Momin & Yang, 2013). These functions are provided below.

Hartmann-6D:

$$f_{\mathrm{H6}}(x,\ell) = \sum_{i=1}^{4} \alpha_i(\ell) \exp\left(-\sum_{j=1}^{6} A_{ij} \left(x_j - 10^{-4} P_{ij} \right)^2 \right), \qquad x \in [0,1]^6, \quad \ell \in \mathcal{L},$$

where

$$\mathcal{L} = \{0.0,\ 0.2,\ 0.4,\ 0.6,\ 0.8,\ 1.0\},$$

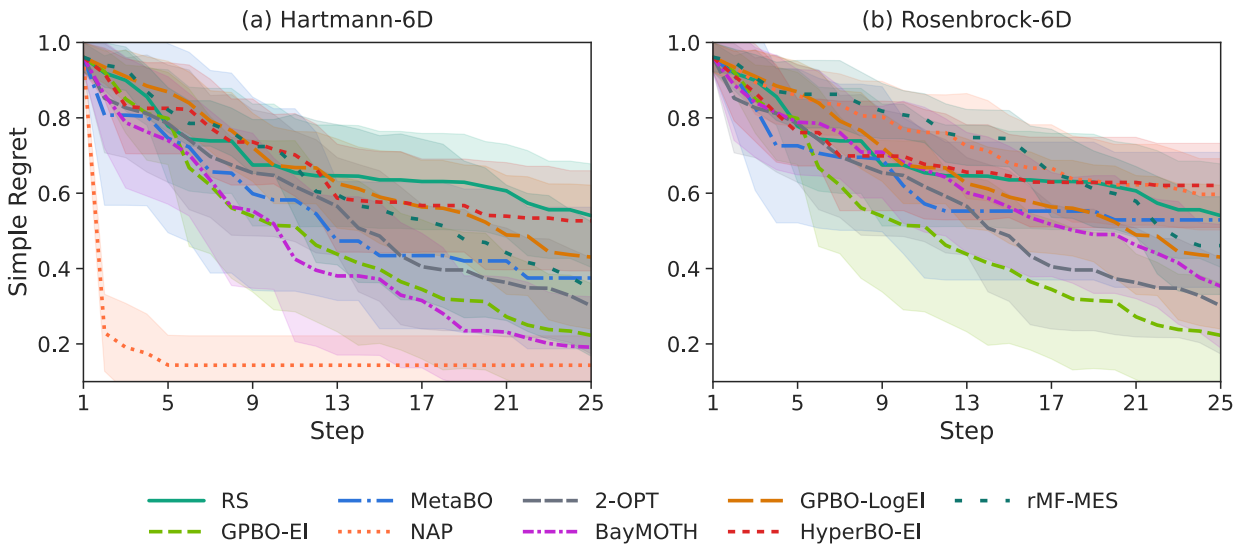


Figure 10: Experiments using global optimization benchmark functions (Momin & Yang, 2013), shown on left is the optimization trajectory when Hartmann-6D with $\ell = 0.2$ is used as the source task, shown on the right is the optimization trajectory when Rosenbrock-6D is used as the source task.

and the fidelity parameter modifies only the first Hartmann coefficient:

$$\alpha(\ell) = \begin{bmatrix} 1.0 - 0.1(1-\ell) \\ 1.2 \\ 3.0 \\ 3.2 \end{bmatrix}.$$

The remaining parameters are

$$A = \begin{bmatrix} 10 & 3 & 17 & 3.5 & 1.7 & 8 \\ 0.05 & 10 & 17 & 0.1 & 8 & 14 \\ 3 & 3.5 & 1.7 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 0.1 & 14 \end{bmatrix},$$

and

$$P = \begin{bmatrix} 1312 & 1696 & 5569 & 124 & 8283 & 5886 \\ 2329 & 4135 & 8307 & 3736 & 1004 & 9991 \\ 2348 & 1451 & 3522 & 2883 & 3047 & 6650 \\ 4047 & 8828 & 8732 & 5743 & 1091 & 381 \end{bmatrix}.$$

The highest-fidelity objective corresponds to $\ell = 1.0$, while lower values of ℓ induce controlled fidelity mismatch through the first coefficient $\alpha_1(\ell)$.

Rosenbrock-6D:

$$f_{\mathrm{R6}}(x) = -\sum_{j=1}^{5} \left[100 \left(x_{j+1} - x_j^2 \right)^2 + (1 - x_j)^2 \right], \qquad x \in [0,1]^6.$$

We consider the highest-fidelity Hartmann-6D function ($\ell = 1.0$) as the test task. To probe the effect of source task mismatch on meta-BO performance, we evaluate two source task configurations. In the first configuration, the source task is a lower-fidelity Hartmann-6D function with $\ell = 0.2$, which preserves the same underlying functional family as the test task while introducing controlled fidelity mismatch. In the

second configuration, the source task is the Rosenbrock-6D function. The optimization domain is kept fixed across both experimental setups. This latter setting provides a deliberately unrelated source task, allowing us to compare how the considered meta-BO methods behave when the available source information shifts from related but imperfect to unrelated and potentially misleading.

The results in Figure 10 and Table 3 show that BayMOTH effectively exploits related source task information while remaining comparatively robust when the source task becomes unrelated. When the lower-fidelity Hartmann-6D function is used as the source task, BayMOTH attains the second-best final regret overall, indicating that the meta-informed lookahead branch provides useful transfer when source and test tasks share structure. NAP demonstrates the strongest performance in this related-source setting, but degrades substantially when the source task is replaced by the unrelated Rosenbrock-6D function. In contrast, BayMOTH exhibits only a moderate increase in regret and remains the strongest-performing meta-BO method in the unrelated-source setting. Overall, these results suggest that BayMOTH provides reliable robustness–performance tradeoff among the meta-BO methods considered: it benefits from related source information while its gating mechanism limits the adverse effect of misleading source tasks.

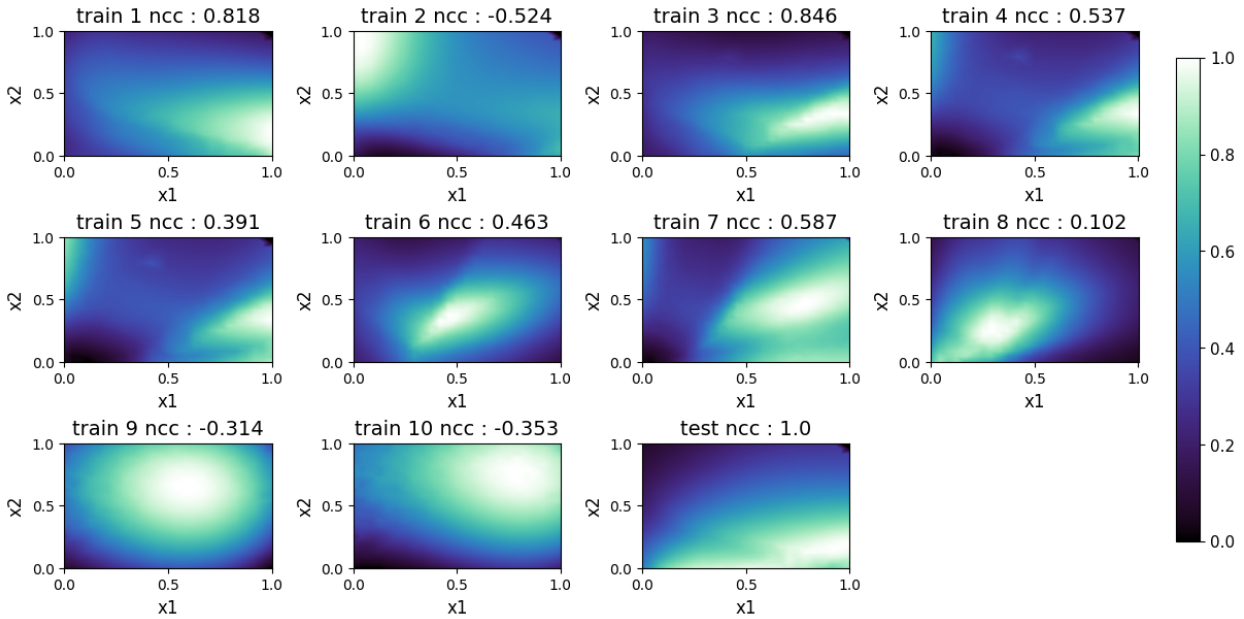
Table 3: Final regret comparison on the Hartmann-6D test task under related and unrelated source-task settings. The related-source setting uses Hartmann-6D with $\ell = 0.2$ as the source task, while the unrelated-source setting uses Rosenbrock-6D as the source task. Δ denotes the increase in regret from the related-source setting to the unrelated-source setting. Mean and worst-case regret are computed across the two source-task settings.

	Related source		Unrelated source		Summary		
Method	Regret	Rank	Regret	Rank	Δ Regret	Mean Regret	Worst Regret
BayMOTH	**0.19**	**2**	**0.35**	**3**	+0.16	**0.27**	**0.35**
NAP	0.14	1	0.60	8	+0.46	0.37	0.60
MetaBO	0.37	6	0.53	6	+0.16	0.45	0.53
HyperBO	0.53	8	0.62	9	+0.09	0.58	0.62
rMF-MES	0.34	5	0.46	5	+0.12	0.40	0.46
2-OPT	0.31	4	0.31	2	0.00	0.31	0.31
GPBO-EI	0.22	3	0.22	1	0.00	0.22	0.22
GPBO-LogEI	0.42	7	0.42	4	0.00	0.42	0.42
RS	0.54	9	0.54	7	0.00	0.54	0.54

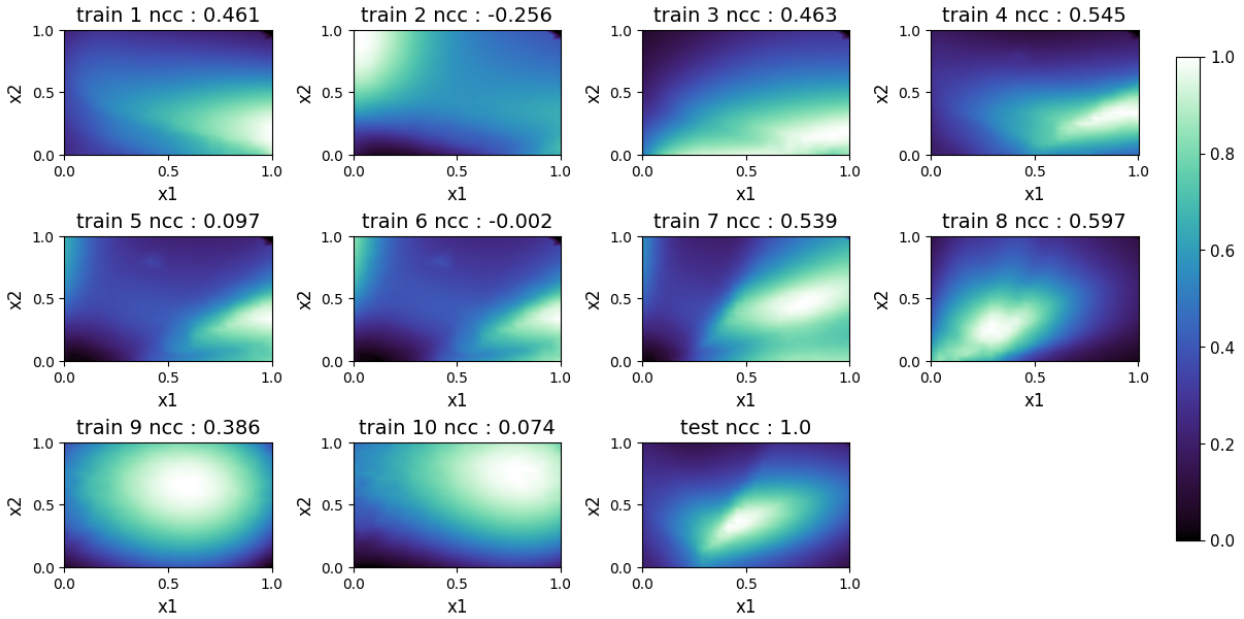
B.2 Synthetic ICF experiments

For the synthetic ICF experiments, the optimization is done with respect to parameters x1 and x2. These parameters correspond to features of laser pulse shape design, which along with a target specification make up the explicit inputs ($\approx$ 20D) for an ICF implosion. A number of performance metrics (Betti et al., 2010; 2015; Christopherson et al., 2018; 2020) have been devised to asses the quality of an implosion. The response of the performance metric with respect to $[\mathrm{x1}, \mathrm{x2}]$ is the focus of these optimization experiments. The parameters $[\mathrm{x1}, \mathrm{x2}]$ are consequential for performance through non-linear 1D and 3D physics effects, which are studied by using physics simulation codes such as LILAC (Delettrez et al., 1987), DRACO (Radha et al., 2005) and statistical models (Gopalaswamy et al., 2019; Lees et al., 2021; 2025). However, as simplifying assumptions and reduced order modeling methodologies are employed in the simulations and models, the true response of performance w.r.t x1 and x2 is potentially not captured, which contributes to the lack of a priori predictive ability. The model inaccuracies could mean the value of their prior information could be limited, motivating the use of BayMOTH.

For constructing source tasks, the physics models are varied in the physics simulator along with the coefficients of the 3D degradation model of (Lees et al., 2021) which is convolved with the simulation to produce a response function. This results in varied shock-timing (Craxton et al., 2015) from the front end of the pulse (x1 and x2 are part of the front end) resulting in a non-ideal variation of the entropy profile in the fusion fuel (Anderson, 2006) and consequently effecting the performance (Lees et al., 2025). Using this procedure, a total of 11 response functions were generated. We consider four distinct source-test task combinations for the experiments, these are shown in Figure 11. The four source-test task configurations are constructed to exhibit progressively increasing distributional shift between the test task and the source tasks, with the lowest shift setting corresponding to the configuration with the highest average positive NCC, and subsequent settings defined by decreasing average positive correlation. These experiments are designed to span scenarios ranging from cases in which the source tasks (i.e., physics-based models) accurately capture the experimental response to regimes in which these models are increasingly misaligned with, or fail to represent, the true response, and as such these experiments probe the robustness of the BO methods when the assumed physics-based models progressively diverge from the true experimental response. Additionally, these experiments assess sample efficiency in realistic settings where a candidate physics-based response is provided as a source task, but its validity with respect to the true experimental response is unknown a priori.

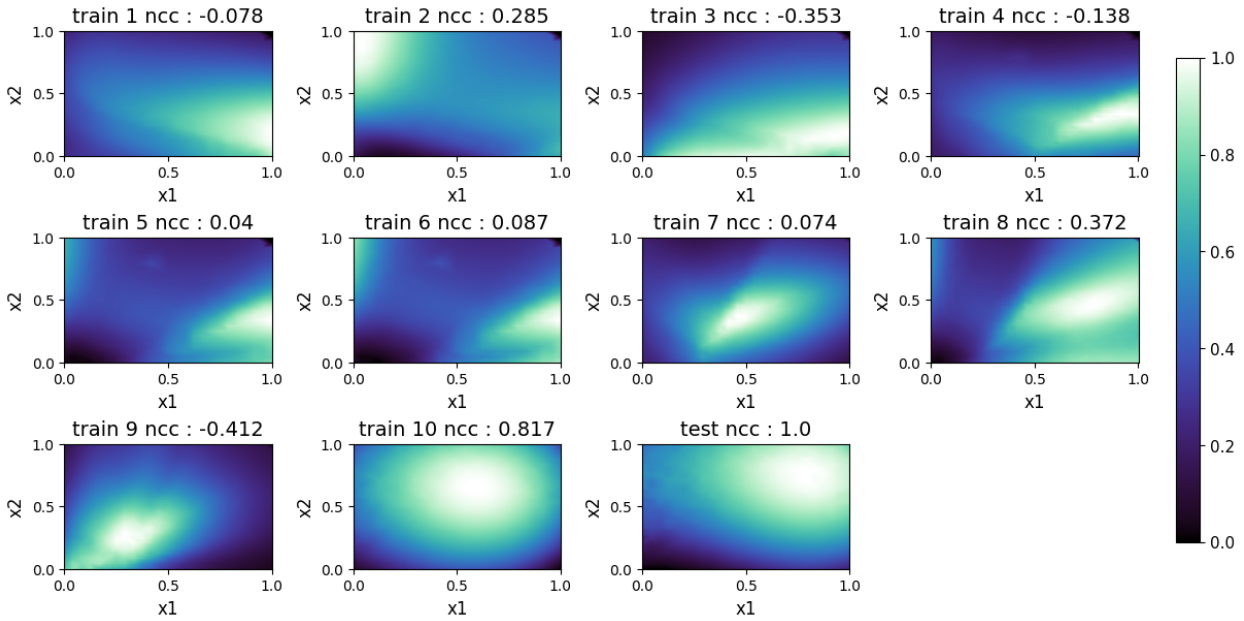


(a) Set 1: High relatedness

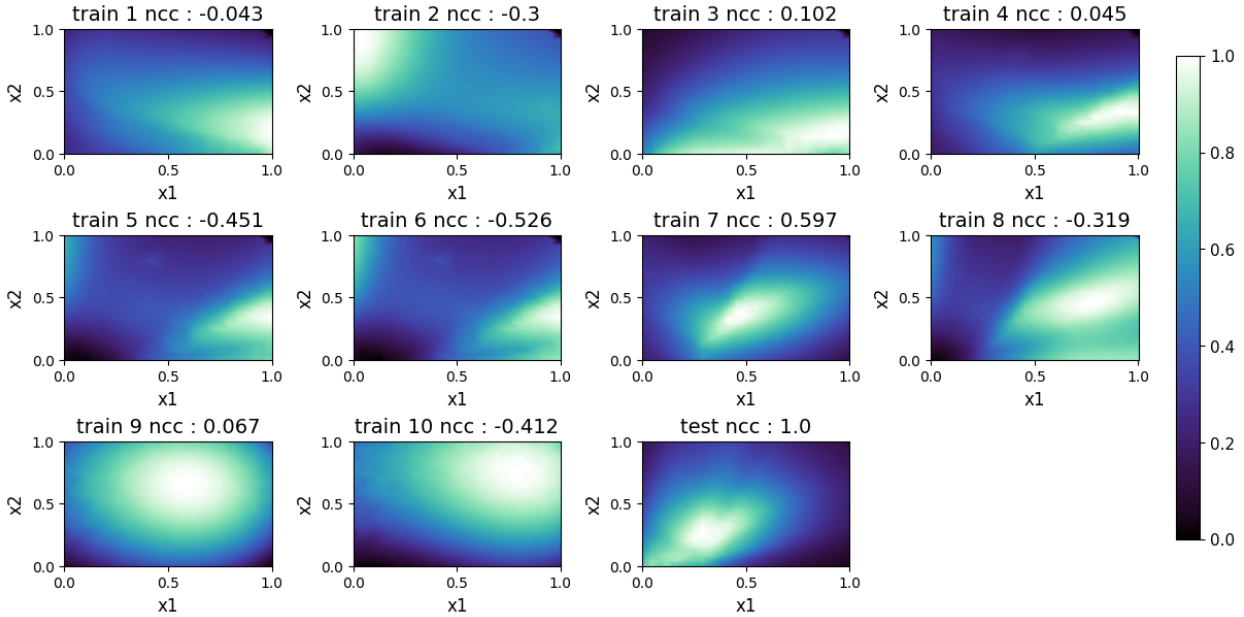


(b) Set 2: Medium relatedness

Figure 11: Source/test tasks across relatedness settings. (a) Set 1: High relatedness. (b) Set 2: Medium relatedness. (c) Set 3: Moderate relatedness. (d) Set 4: Low relatedness.



(c) Set 3: Moderate relatedness



(d) Set 4: Low relatedness

Figure 11: Source/test tasks across relatedness settings (continued).

B.3 Model hyperparameters and additional details

Here, we provide the associated hyperparameters used for the different baselines. Any additional details are present in the provided codebase. All hyperparameters were kept the same for all experiments (except otherwise noted). Minimal manual tuning was done for selection.

BayMOTH:

The hyperparameter configuration used for BayMOTH is noted below. Implementation for BayMOTH and code details on experimental tests are available in the provided codebase.

Table 4: BayMOTH configuration.

BayMOTH	
Λ_0 $\mathcal{GP}$ kernel	RBF
Λ_0 $\mathcal{GP}$ kernel length-scale	0.05*4
Λ_1 $\mathcal{GP}$ kernel	RBF
Λ_1 $\mathcal{GP}$ kernel length-scale	0.05
$\mathcal{GP}_\chi$ kernel	RBF
$\mathcal{GP}_\chi$ kernel length-scale	0.05
$\tilde{\mathcal{GP}}_\chi$ Kernel	RBF
$\tilde{\mathcal{GP}}_\chi$ Kernel length-scale	0.05*4
Estimator number of samples M	5
greedy balancing parameter α	0.5
Observation noise	10^{-6}
Input normalization	$[0,1]^d$

Table 5: Hyperparameter specification for the routing threshold γ used in each experimental setting.

Experimental setting	Threshold γ
ICF experiments	0.70
Global optimization benchmark functions	0.90
HPO: glmnet	0.62
HPO: rpart.preproc	0.52
HPO: HPOBenchXGB	0.13
HPO: ranger9	0.35

Robust MFBO (rMF-MES):
We note that for the rMF-MES (Mikkola et al., 2023) baseline, we follow the same test task evaluation accounting used for the meta-BO baselines. Although rMFBO is originally formulated with auxiliary information-source queries consuming online budget, in our setting source-task information is treated as available prior to online optimization. Therefore, auxiliary-source queries for rMF-MES are not considered to consume the test task BO budget.

The implementation from Mikkola et al. (2023) available at the following GitHub repository: `https://github.com/AaltoPML/rMFBO` was used. Any additional details on rMF-MES related to our experiments can be found in our submitted codebase.

Table 6: Hyperparameter specification for the rMF-MES baseline.

Hyperparameter	Value
c_1	0.10
c_2	0.10
Auxiliary IS cost	0.20

HyperBO
The HyperBO (Wang et al., 2024) codebase available at `https://github.com/google-research/hyperbo` was used for our experiments, we used the H-EKL variant of HyperBO with the EI (Jones et al., 1998) acquisition function, we refer to this baseline as HyperBO-EI. Our experimental implementation details are included in our provided codebase.

NAP and MetaBO:
The hyperparameters for MetaBO and NAP, were taken from their respective codebases, and the implementation from the NAP repository `https://github.com/huawei-noah/HEBO/tree/master/NAP` was used, all implementations and experimental tests scripts are included in our provided codebase.

2-OPT:

For the 2-OPT baseline, in our codebase, we include a companion implementation developed alongside BayMOTH, allowing comparison under a consistent implementation and evaluation framework. Furthermore, we note that an official implementation of 2-OPT was not available in the codebase referenced by Wu & Frazier (2019).

Table 7: 2-OPT configuration.

2-OPT	
Λ_0 $\mathcal{GP}$ kernel	RBF
Λ_0 $\mathcal{GP}$ kernel length-scale	0.05*4
Λ_1 $\mathcal{GP}$ kernel	RBF
Λ_1 $\mathcal{GP}$ kernel length-scale	0.05
Estimator number of samples M	5
Input normalization	$[0,1]^d$

GPBO-EI/GPBO-LogEI:

The provided codebase includes our implementation for GPBO-EI along with the experimental tests implementation. For the GPBO-LogEI baseline, BoTorch Balandat et al. (2020) was used, the experimental tests implementation is available in the provided codebase.

Table 8: GPBO-EI configuration.

GPBO-EI	
Λ_0 GP kernel	RBF
Λ_0 GP kernel length-scale	$0.05 * 4$
Input normalization	$[0,1]^d$