

---

# Informed Initialization for Bayesian Optimization and Active Learning

---

Anonymous Author(s)

Affiliation

Address

email

## Abstract

Bayesian Optimization is a widely used method for optimizing expensive black-box functions, relying on probabilistic surrogate models such as Gaussian Processes. The quality of the surrogate model is crucial for good optimization performance, especially in the few-shot setting where only a small number of batches of points can be evaluated. In this setting, the initialization plays a critical role in shaping the surrogate’s predictive quality and guiding subsequent optimization. Despite this, practitioners typically rely on (quasi-)random designs to cover the input space. However, such approaches neglect two key factors: (a) random designs may not be space-filling, and (b) efficient hyperparameter learning during initialization is essential for high-quality prediction, which may conflict with space-filling designs. To address these limitations, we propose Hyperparameter-Informed Predictive Exploration (HIPE), a novel acquisition strategy that balances space-filling exploration with hyperparameter learning using information-theoretic principles. We derive a closed-form expression for HIPE in the Gaussian Process setting and demonstrate its effectiveness through extensive experiments in active learning and few-shot BO. Our results show that HIPE outperforms standard initialization strategies in terms of predictive accuracy, hyperparameter identification, and optimization performance, particularly in large-batch, few-shot settings relevant to many real-world Bayesian Optimization applications.

## 1 Introduction

Bayesian Optimization (BO) (Frazier, 2018; Garnett, 2023; Jones et al., 1998; Mockus et al., 1978) is a principled framework for sample-efficient global optimization of black-box functions with applications across diverse fields such as biological discovery (Griffiths and Hernández-Lobato, 2020; Stanton et al., 2022), materials science (Ament et al., 2023b; Attia et al., 2020; Frazier and Wang, 2016), online A/B testing (Agarwal et al., 2018; Feng et al., 2025; Letham et al., 2019), and machine learning hyperparameter optimization (HPO) (Feurer et al., 2015; Snoek et al., 2014). BO combines a probabilistic surrogate model—commonly a Gaussian Process (GP)—with an acquisition function to select where to evaluate the unknown objective function. In many applications, the runtime of a single black-box function evaluation may restrict the experimenter to a small number of *batches*—the number of sequential rounds of experiments—but many real-world experimental setups permit conducting multiple experiment simultaneously (e.g., on a parallel compute cluster, in a randomized controlled trial, or batch-testing multiple specimens in a lab).

The success of Bayesian Optimization in practice is highly sensitive to the quality of the surrogate model (Eriksson and Jankowiak, 2021; Hvarfner et al., 2023). This is a challenge particularly during the early stages of optimization when few observations are available. In these early stages, a small set of inputs is typically selected at random, or via space-filling or quasi-random sampling

strategies such as Latin Hypercube Sampling (LHS) or scrambled Sobol’ sequences (Bossek et al., 2020; Owen, 2023). While such strategies aim to achieve broad coverage of the input space for initialization of the surrogate model, they are not necessarily the ideal choice to satisfy this criterion. Moreover, they neglect a second crucial aspect of modeling: the need to accurately infer the model’s hyperparameters (Zhang et al., 2019), such as the kernel lengthscales of a GP. With accurate hyperparameter estimates, variation in unimportant dimensions will have less influence on the selection of points in subsequent iterations of BO, leading to more sample-efficient optimization (Eriksson and Jankowiak, 2021; Hvarfner et al., 2023; Müller et al., 2023). Conversely, poor hyperparameter estimation may cause subsequent BO iterations to fail to make meaningful progress (Berkenkamp et al., 2019), which may be caused by misidentifying signal for noise, exploring irrelevant dimensions, or returning poor terminal recommendations (Hvarfner et al., 2022).

In this work, we provide a principled approach to addressing this initialization challenge. Our main contributions are as follows:

1. We propose Hyperparameter-Informed Predictive Exploration (HIPE), an acquisition function for initialization that jointly optimizes for both predictive uncertainty and hyperparameter inference.
2. We derive a closed-form expression for this objective in the case of Gaussian Process models, and implement a practical Monte Carlo approximation.
3. We conduct extensive experiments in active learning and Bayesian Optimization on synthetic and real-world BO tasks, demonstrating that HIPE outperforms competing methods in terms of both model accuracy metrics and BO performance in few-shot, large-batch settings.

## 2 Background

### 2.1 Gaussian Processes

Gaussian Processes (GPs) are a widely used surrogate model in BO due to their flexibility, closed-form and well-calibrated predictive distributions. GPs define a distribution over functions,  $\hat{f} \sim \mathcal{GP}(m(\cdot), k(\cdot, \cdot))$ , specified by a mean function  $m(\cdot)$  and a covariance (kernel) function  $k(\cdot, \cdot)$ . For a given location  $\mathbf{x}$ , the function value  $\hat{f}(\mathbf{x})$  is normally distributed, with closed-form expressions for the predictive mean  $\mu(\mathbf{x})$  and variance  $\sigma^2(\mathbf{x})$ . In practice, the mean function is often kept constant, leaving the covariance function to capture the structural properties of the objective.

To model differences in variable importance in GPs with stationary kernels, each input dimension is commonly scaled by a lengthscale hyperparameter  $\ell_i$ , a practice known as Automatic Relevance Determination (ARD) (Williams and Rasmussen, 1995). Additional, optional hyperparameters include a learnable noise variance  $\sigma_\varepsilon^2$  and signal variance  $\sigma_f^2$ . The full set of hyperparameters  $\boldsymbol{\theta} = \{\ell, \sigma_\varepsilon^2, \sigma_f^2\}$  may be learned either by maximizing the marginal likelihood  $p(\mathcal{D} \mid \boldsymbol{\theta})$  (Maximum Likelihood Estimation, MLE) or by incorporating hyperpriors  $p(\boldsymbol{\theta})$  to perform Maximum A Posteriori (MAP) estimation. Alternatively, a fully Bayesian treatment (Lalchand and Rasmussen, 2020; Osborne, 2010) integrates over  $\boldsymbol{\theta}$  to approximate the full Bayesian posterior distribution using Markov Chain Monte Carlo (MCMC) methods, thereby explicitly accounting for hyperparameter uncertainty. For additional background on GPs, see (Rasmussen and Williams, 2005).

### 2.2 Bayesian Optimization

Bayesian Optimization (BO) (Frazier, 2018) is a sample-efficient framework for finding the maximizer  $\mathbf{x}_* = \arg \max_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x})$  of a black-box function  $f : \mathcal{X} \rightarrow \mathbb{R}$  over a  $D$ -dimensional input space  $\mathcal{X} = [0, 1]^D$ . The function  $f$  is assumed to be expensive to evaluate and observable only through noisy point-wise measurements,  $y(\mathbf{x}) = f(\mathbf{x}) + \varepsilon$ , where  $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2)$ .

At the core of BO is an *acquisition function*, which uses a surrogate model to quantify the (expected) utility of candidate points  $\mathbf{x}$ . Acquisition functions balance exploration and exploitation typically through greedy heuristics. Popular examples include Expected Improvement (EI) (Bull, 2011; Jones et al., 1998) and its numerically stable variant, LogEI (Ament et al., 2023a), as well as the Upper Confidence Bound (UCB) (Srinivas et al., 2012, 2010). In batch BO, multiple points are selected in parallel to accelerate data collection. This is often achieved by computing and jointly

optimizing a (quasi-)MC estimate of the utility  $u$  associated with acquisition function over the full batch  $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_q\}$  of size  $q$  (Balandat et al., 2020; Wilson et al., 2017; Wilson et al., 2020).

### 2.3 Bayesian Active Learning

Bayesian Active Learning (BAL) and Bayesian Experimental Design (BED) (Chaloner and Verdinelli, 1995) aims to improve predictive models by strategically selecting data points that are most informative, either with regard to the model or to future predictions. A central quantity is the Expected Information Gain (EIG):

$$\text{EIG}(\xi; y(\mathbf{x})) = H[\xi] - \mathbb{E}_{\xi} [H[\xi|y(\mathbf{x})]], \quad (1)$$

where  $H$  is the Shannon (differential) entropy and  $\xi$  is a parameter of interest. Importantly, EIG is symmetric, and can be equivalently formulated as an entropy reduction over  $y(\mathbf{x})$ , instead of  $\xi$ .

**Bayesian Active Learning by Disagreement** Bayesian Active Learning by Disagreement (BALD) (Houlsby et al., 2011; Kirsch et al., 2019) selects query points that maximize the mutual information between model predictions and hyperparameters  $\theta$ :

$$\text{BALD}(\mathbf{x}) = \text{EIG}(y(\mathbf{x}); \theta) = H[y(\mathbf{x})|\mathcal{D}] - \mathbb{E}_{\theta} [H[y(\mathbf{x})|\mathcal{D}, \theta]]. \quad (2)$$

BALD identifies locations where models within an ensemble exhibit the greatest disagreement in predictive uncertainty. In the GP setting, this often leads to axis-aligned queries when there is high uncertainty in the lengthscales, and to repeated queries when observation noise is highly uncertain. Notably, BALD is model-agnostic and can be applied to a wide range of surrogate models and hyperparameters, including subspace models (Garnett et al., 2014) and additive decompositions (Gardner et al., 2017; Hvarfner et al., 2023).

**Negative Integrated Posterior Variance and Expected Predictive Information Gain** The Negative Integrated Posterior Variance (NIPV) (Chen and Zhou, 2014; Seo et al., 2000) criterion selects queries that minimize the expected posterior variance over a test distribution  $p_*(\mathbf{x})$ :

$$\text{NIPV}(\mathbf{x}; p_*) = -\mathbb{E}_{\mathbf{x}_* \sim p_*} [\sigma^2(\mathbf{x}_*) | \mathbf{x}, \mathcal{D}]. \quad (3)$$

Similarly, Expected Predictive Information Gain (EPIG) (Bickford Smith et al., 2023) selects queries that minimize the expected predictive entropy over  $p_*(\mathbf{x})$ :

$$\text{EPIG}(\mathbf{x}; p_*) = -\mathbb{E}_{\mathbf{x}_* \sim p_*} [H[y(\mathbf{x}) | \mathbf{x}_*, \mathcal{D}]]. \quad (4)$$

Both objectives promote the selection of data that reduces uncertainty over the test distribution, but without considering the effect on hyperparameter learning. The test distribution encodes how much emphasis is put on different parts of the domain. It can be specified by subject matter experts; in the case of no prior knowledge it is typically the uniform distribution. Throughout the remainder of the paper, we will exclusively consider NIPV with a uniform  $p_*$ .

## 3 Related Work

Initialization has received surprisingly little attention in the context of BO. Bossek et al. (2020) conducts a study on the effect of various random initial designs on BO performance. Alternatively, minimax or maximin criteria (Johnson et al., 1990) may be used to accomplish evenly distributed designs. Maybe closest to our work is (Zhang et al., 2019), which proposes LHS-Beta, an initial design criterion which alters samples drawn by LHS to achieve pairwise distances between points which matches a Beta distribution. LHS-Beta pursues diverse pairwise distances in the data, in order to best learn the lengthscale of a GP with an isotropic kernel. Müller and Zimmerman (1999); Zimmerman (2006) address the problem of learning parameters of Kriging estimators using a empirical estimates of optimal experimental design criteria, limiting candidates to a fixed grid of points.

Entropy-maximizing (Guestrin et al., 2005; MacKay, 1992, 1995) or variance-minimizing (Park et al., 2024) designs have been explored in active learning for optimal sensor placement (Krause et al., 2006, 2008) and other applications involving GPs, such as geostatistics (Sauer et al., 2023) and contour finding (Cole et al., 2023). Moreover, parameter-related EIG criteria are a bedrock

of the broader topic of BED (Bickford Smith et al., 2023; Chaloner and Verdinelli, 1995; Kirsch et al., 2021; Rainforth et al., 2024), which focuses on selecting data that is most informative about model parameters or future predictions. These prediction or model-oriented criteria have yet to see widespread use in BO, particularly for initialization. However, information-theoretic acquisition functions (Hennig and Schuler, 2012; Hernández-Lobato et al., 2014; Hvarfner et al., 2022; Moss et al., 2021; Neiswanger et al., 2021, 2024; Tu et al., 2022; Wang and Jegelka, 2017) address the optimization problem from an information theoretic perspective, albeit not with a primary focus on initialization or model predictive performance.

The problem of actively learning model hyperparameters during BO (post initialization) has previously been investigated by Hvarfner et al. (2023), who propose a combined BO-BAL framework to actively learn the hyperparameters of the GP along with the optimum, and demonstrate that better-calibrated surrogates significantly enhance BO performance. Houlsby et al. (2011) proposes BALD, an active learning acquisition functions for hyperparameters in preference learning in GPs. Riis et al. (2022) proposes a Query-By-Committee-oriented acquisition function for BAL in GPs. Lastly, Berkenkamp et al. (2019); Ziomek et al. (2024) address BO performance under hyperparameter uncertainty from a theoretical perspective, proving regret bounds when hyperparameters of the objective are unknown.

## 4 Method

We consider the case of large-batch, few-shot BO, where the batch size  $q$  is large ( $q \geq 10$ ), and only a small number of sequential batches  $B$  can be evaluated (often only one for initialization and one for BO, so  $B = 2$ ). This setting is common if evaluations are lengthy but can be effectively parallelized, which is the case for instance in online A/B tests or lab experiments for biology or material design.

In practice, the objective(s) often exhibit complex structure such as moderate or high dimensionality  $D$  ( $D > 5$ ) and significant uncertainty in the lengthscales  $\ell$ , noise and signal variance  $\sigma_\varepsilon^2$  and  $\sigma_f^2$ , respectively, and the importance of each input dimension. Standard space-filling designs fail to reliably uncover these hyperparameter dependencies (Zhang et al., 2019), often leading to poor surrogate model calibration and degraded acquisition decisions in the subsequent BO phase. At the same time, purely hyperparameter-focused designs such as BALD tend to cluster evaluations along specific dimensions, resulting in poor coverage of the input space and ineffective exploration. This tension is exacerbated by the limited number of batches: there is little or no opportunity to correct for poor initialization in later iterations. This highlights the need for initial designs that simultaneously reduce uncertainty in hyperparameters and provide sufficient coverage for downstream optimization.

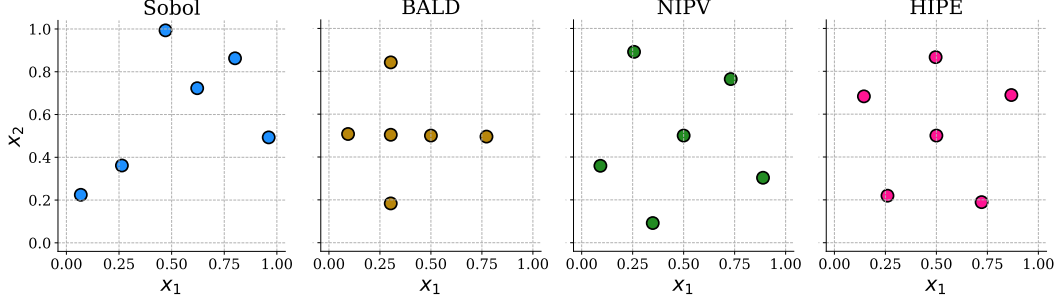
This situation is illustrated in Fig. 1, which visualizes different initialization strategies applied to a GP model under uncertainty in its lengthscale hyperparameters. Each strategy exhibits distinct behaviors that directly impact the model’s ability to make accurate predictions and effectively guide subsequent optimization. Scrambled Sobol sequences, while widely used for space-filling designs, often fail to achieve the desired coverage in practice. BALD, an active learning strategy, exclusively focuses on reducing uncertainty in hyperparameters, leading to axis-aligned, clustered queries to identify the lengthscales. NIPV explicitly minimizes predictive uncertainty but ignores hyperparameter informativeness (Zhang et al., 2019). In contrast, our proposed method, HIPE, balances space-filling exploration and hyperparameter-awareness, leading to initial designs that effectively reduce both predictive and hyperparameter uncertainty.

### 4.1 Hyperparameter-Informed Predictive Exploration

We approach the problem of initialization through the lens of optimization, by simultaneously maximizing the coverage of a region of interest and the information acquired about model-level uncertainty. A natural way to achieve this is to optimize a criterion that combines the EPIG and BALD objectives:

$$\text{HIPE}_\beta(\mathbf{X}) := \underbrace{-\mathbb{E}_{\boldsymbol{\theta}, y(\mathbf{X})} [\mathbb{E}_{\mathbf{x}_*} [\mathbf{H}[y(\mathbf{x}_*) \mid \boldsymbol{\theta}, y(\mathbf{X})]]]}_{\text{EPIG objective}} + \beta \underbrace{(\mathbf{H}[y(\mathbf{X})] - \mathbb{E}_{\boldsymbol{\theta}} [\mathbf{H}[y(\mathbf{X}) \mid \boldsymbol{\theta}]])}_{\text{BALD objective}} \quad (5)$$

where  $\beta > 0$  is a scalar weighting parameter. For large  $\beta$ , this objective favors hyperparameter learning, while for small  $\beta$  it favors space-filling designs.



**Figure 1:** Visual comparison of initialization strategies BO for a GP in two dimensions with uncertainty in the lengthscales  $\ell$ . While Sobol emphasizes designs that are space-filling, it may not accomplish this to the desired degree. BALD focuses on reducing lengthscale uncertainty by choosing axis-aligned queries. With a uniform test distribution, NIPV spreads out points to minimize average predictive variance over the input space. Lastly, HIPE balances space-filling and hyperparameter-awareness, choosing spread-out points while preserving axis-alignment between queries.

It turns out that for the choice of  $\beta = 1$ , the maximizer of Eq. (7) is exactly the maximizer of the joint information gain over test function values and model hyperparameters (for proof see Appendix B):

**Proposition 1** (Equivalence of  $\text{HIPE}_{\beta=1}$  to Joint Information Gain). *The  $\text{HIPE}_{\beta}$  acquisition function with  $\beta = 1$  is equivalent to maximizing the expected joint information gain over test function values  $y(\mathbf{x}_*)$  and model hyperparameters  $\theta$  acquired by a candidate batch  $\mathbf{X}$ . Formally,*

$$\arg \max_{\mathbf{X} \in \mathbb{R}^{q \times D}} \text{HIPE}_{\beta=1}(\mathbf{X}; p_*) = \arg \max_{\mathbf{X} \in \mathbb{R}^{q \times D}} \mathbb{E}_{\mathbf{x} \sim p_*} [\text{EIG}(y(\mathbf{x}_*), \theta; \mathbf{X})]. \quad (6)$$

While Proposition 1 provides an intuitive connection between HIPE, EPIG, and BALD, the constituent quantities often have vastly different scales. Therefore the choice of  $\beta = 1$  will generally not result in optimal performance since the optimization will inadvertently focus on the larger of the two terms.

A key observation is that not all hyperparameter information gain amounts to information gained on the test set – this depends on multiple aspects, including downstream test distribution and hyperparameterization. To what extent the reduction in hyperparameter entropy *manifests in a reduction in test set entropy* can be quantified through the mutual information between the hyperparameters  $\theta$  and the test points  $y(\mathbf{x}_*)$ ,  $\mathbf{x}_* \sim p_*(\mathbf{x})$ :

$$\text{EIG}(y(\mathbf{x}_*); \theta) = \mathbb{E}_{\mathbf{x}_*} [\mathbb{H}[y(\mathbf{x}_*)] - \mathbb{E}_{\theta} [\mathbb{H}[y(\mathbf{x}_*) | \theta]]]. \quad (7)$$

Intuitively, Eq. (7) quantifies how well the knowledge of the hyperparameters, in expectation, informs us about the values of  $y(\mathbf{x}_*)$ . Importantly,  $\text{EIG}(y(\mathbf{x}_*); \theta)$  does not depend on the candidate set  $\mathbf{X}$  and can thus be pre-computed.

Setting  $\beta = \text{EIG}(y(\mathbf{x}_*); \theta)$  balances the two competing objectives in Eq. (5) according to their effect on downstream predictive uncertainty, without introducing any additional hyperparameters. We refer to the resulting acquisition function as Hyperparameter-Informed Predictive Exploration (HIPE):

$$\text{HIPE}(\mathbf{X}) := -\mathbb{E}_{y(\mathbf{X})} [\mathbb{E}_{\mathbf{x}_*} [\mathbb{H}[y(\mathbf{x}_*) | \theta, y(\mathbf{X})]] + \text{EIG}(y(\mathbf{x}_*); \theta) \mathbb{E}_{\theta} [\mathbb{H}[y(\mathbf{X}) | \theta] - \mathbb{H}[y(\mathbf{X})]] \quad (8)$$

Notably, optimizing HIPE does not require having observed *any* data – for this problem to be well-posed only requires test distribution, model structure, and model parameter hyperpriors.

## 4.2 A Parallel Monte Carlo Implementation of HIPE

Following the Monte Carlo approach used in modern acquisition functions (Balandat et al., 2020; Wilson et al., 2020), we implement a parallel version of HIPE that enables joint optimization. For a candidate batch  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_q\}$ , we estimate the acquisition objective using  $M$  MC samples over the hyperparameters,  $T$  test locations of the EPIG objective, and  $N$  samples from the predictive posterior for the BALD objective:

$$\alpha_{\text{BALD}}(\mathbf{X}; \theta) = -\log p(\mathbb{E}_{\theta}[y(\mathbf{X}) | \theta]) + \mathbb{E}_{\theta} [\log p(y(\mathbf{X}) | \theta)] \quad (9a)$$

$$\approx -\frac{1}{N} \sum_{n=1}^N \left[ \log \left( \frac{1}{M} \sum_{m=1}^M p(Y^{(n)} | \theta^{(m)}) \right) + \frac{1}{M} \sum_{m=1}^M \log p(Y^{(n)} | \theta^{(m)}) \right], \quad (9b)$$

where  $\theta^{(m)} \sim p(\theta \mid \mathcal{D})$  are i.i.d samples from the belief over hyperparameters, and  $Y^{(n)} \sim y(\mathbf{X})$  are i.i.d,  $q$ -dimensional (joint) samples from the predictive posterior. Thus, the BALD objective amounts to repeated evaluation the  $q$ -dimensional multivariate normal posterior  $y(\mathbf{X})$  for each candidate  $\mathbf{X}$ , and estimating the posterior entropy from it. Secondly, we estimate the EPIG objective analogously to Balandat et al. (2020) as

$$\alpha_{\text{EPIG}}(\mathbf{X}; p_*, \theta) \approx \frac{1}{M} \frac{1}{T} \sum_{m=1}^M \sum_{t=1}^T \left[ C - \mathbb{H}[y(\mathbf{x}_*^{(t)}) \mid \mathbf{X}, \theta^{(m)}, \mathcal{D}] \right], \quad (10)$$

where  $C = \mathbb{H}[y(\mathbf{x}_*^{(t)}) \mid \mathcal{D}]$  is constant w.r.t.  $\mathbf{X}$  and does not need to be computed. Since the entropy term in Eq. (10) is Gaussian, we can evaluate it in closed form after conditioning each of the  $M$  GPs on the set  $\mathbf{X}$ . Notably, the posterior entropy is independent of the observed value at these locations.

With these two MC estimators, the subsequent optimization can be carried out jointly for the entire batch in a  $qD$ -dimensional space. Using Sample Average Approximation (Balandat et al., 2020), the HIPE objective is deterministic and differentiable via auto-differentiation. Notably, HIPE does not face the same optimization difficulties as other BO acquisition functions such as EI (Ament et al., 2023a; Gramacy et al., 2022; Swersky, 2017; Wilson et al., 2020) where optima may be sharp, in hard-to-locate regions, and the implementation may suffer from vanishing gradients. On the contrary, space-filling designs constitute a good initial condition for the numerical optimization.

One downside of this formulation is that the nested MC estimator imposes substantial computational runtime, making HIPE less suited for high-throughput applications (Daulton et al., 2022; Eriksson et al., 2019; Maus et al., 2022). However, in those applications the quality of the initialization batch is generally much less crucial, so this is not a limitation in practice.

## 5 Results

We evaluate HIPE across two main types of tasks: Active Learning (AL) and Bayesian Optimization (BO). We consider various synthetic and real-world problems and a number of different baselines. In both settings, we maintain large batch sizes and few batches.

**Setup** For AL, we simply run a number of batches with HIPE and the other baselines with the goal of achieving the best model fit. For BO, we consider the “two-shot” setting in which we first have to select an initialization batch and then can perform a single iteration of (batch) BO using qLogNEI (Ament et al., 2023a) as the acquisition function. We benchmark against the conventional initializations Sobol and Random Search, as well as BALD, NIPV and LHS-Beta (Zhang et al., 2019). On all tasks, we utilize a fully Bayesian GP (Eriksson and Jankowiak, 2021; Snoek et al., 2012) using MCMC with NUTS (Hoffman and Gelman, 2014) in Pyro (Bingham et al., 2018). We implement HIPE and all baselines in BoTorch (Balandat et al., 2020). For all experiments, the hyperparameter set  $\theta$  consists of lengthscales  $\ell$  for each dimension with a  $D$ -scaled prior (Hvarfner et al., 2024), a constant mean  $c$ , and an inferred noise variance  $\sigma_\epsilon^2$  unless otherwise specified. All baselines, including random algorithms, are given the center of the search space as part of their initial design, as both HIPE and NIPV select the center of the search space by design under a uniform  $p_*$ . Complete details on the experimental setup and all benchmarks can be found in Appendix A.1.

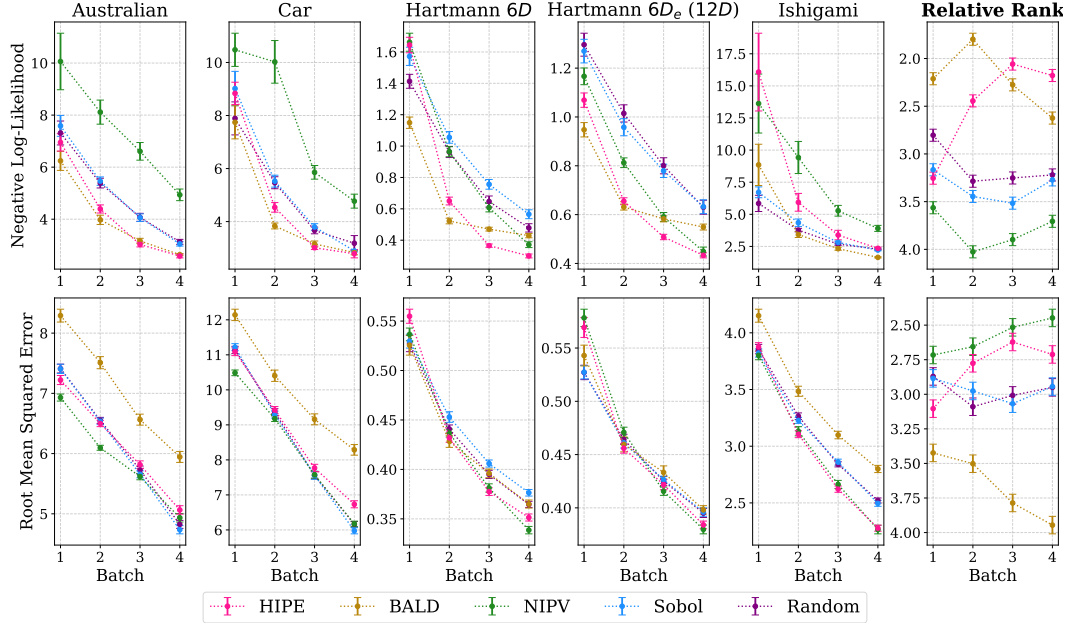
**Evaluation Criteria** We measure model fit quality with Root Mean Square Error (RMSE) of the mean prediction and Negative Log-Likelihood (NLL) against a large number of (ground truth) test point sampled uniformly from the domain. In the “two-shot” optimization setting, we are interested in how the initialization affects the quality of the GP surrogate after both the first (initialization) and second (BO) batch. We compute relative rankings, the performance of each algorithm compared to its competitors, for each seed of each function, and average across the task type. As such, the relative rankings aggregate inter-algorithm performance across rows for Figs 2-4.

We also study how these model quality improvements translate to better optimization performance. To this end, we consider the *out-of-sample inference performance* (Hernández-Lobato et al., 2014; Hvarfner et al., 2022), that is, the performance of the point  $\mathbf{x}' = \arg \max \mu(\mathbf{x} \mid \mathcal{D})$  selected as the maximizer of the posterior mean of the surrogate model fit on the data available in each batch. We choose this metric since using observed points directly performs very poorly in noisy settings, and only considering in-sample points is rather limiting in the few-shot setting.

## 5.1 Batch Active Learning on GP Surrogates and Synthetic Functions

We first evaluate the ability of HIPE to learn accurate surrogate models through batch active learning on noisy synthetic test functions and surrogate LCBench (Zimmer et al., 2021) tasks. The LCBench tasks are derived from complete neural network training runs on various OpenML (Vanschoren et al., 2014) datasets, with 7D GP surrogate models fitted as described in Appendix A.1. Additionally, we evaluate performance on the Hartmann 6D function and a high-dimensional Hartmann 6D (12D) variant, where dummy input dimensions are added following standard practice in high-dimensional BO (Eriksson and Jankowiak, 2021). These dummy dimensions introduce an additional challenge, as effectively identifying and ignoring irrelevant features is critical for accurate predictions. All evaluations are subject to substantial observation noise, detailed in Appendix A.1.

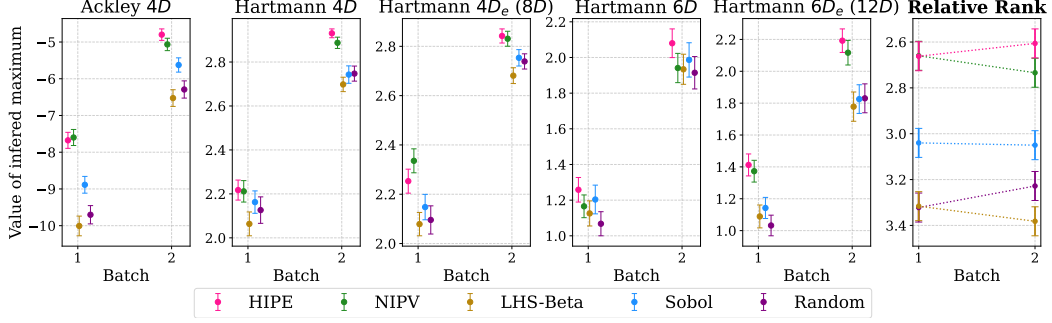
We run each algorithm for 4 batches of size  $q = 16$  and measure the NLL and RMSE after each batch, displaying mean and one standard error on all tasks. In Fig. 2 we see that, across all tasks, HIPE is the only method that consistently ranks in the top two for both RMSE and NLL, demonstrating that the models it produces are both accurate and well-calibrated. On NLL, HIPE performs comparably to BALD, which targets hyperparameter learning and thus excels at model calibration. Similarly, HIPE is competitive with NIPV on RMSE, a metric for which NIPV is particularly well-suited (Gramacy and Lee, 2009).



**Figure 2:** Model accuracy results in the batch active learning setting. We report RMSE across various synthetic and LCBench surrogate tasks over 4 batches of  $q = 16$  evaluations across 100 seeds per benchmark. HIPE consistently ranks in the top two in relative rankings on both metrics, achieving a strong balance between hyperparameter learning and predictive accuracy. BALD performs competitively on marginal log-likelihood (MLL) but underperforms on RMSE due to limited space-filling behavior, while NIPV excels at reducing RMSE but struggles with model calibration. On aggregate, random initialization methods lag significantly behind across all benchmarks.

## 5.2 Noisy Synthetic Test Functions

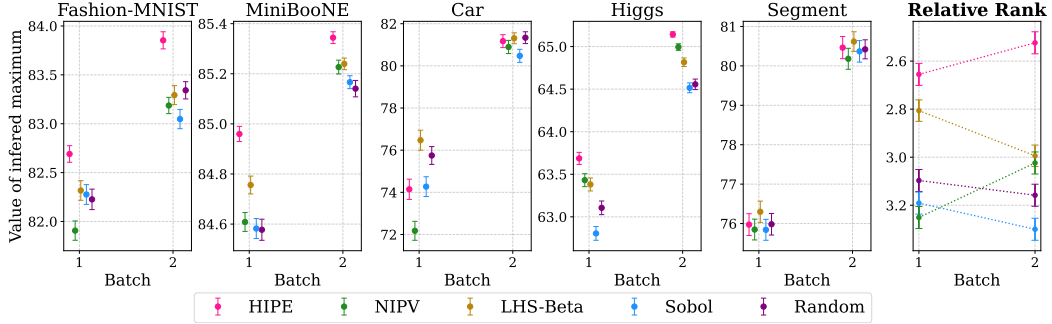
Next, we evaluate HIPE on synthetic benchmark functions in the two-shot Bayesian optimization setting, using  $B = 2$  batches and a batch size of  $q = 24$ . We consider three standard test functions—Ackley (4D), Hartmann (4D), and Hartmann (6D)—as well as two higher-dimensional variants of the Hartmann function. Observation noise is added to all tasks ( $\sigma_\varepsilon = 2$  for Ackley and  $\sigma_\varepsilon = 0.5$  for Hartmann), further increasing task difficulty. In Fig. 3 shows that across all benchmarks, HIPE consistently achieves the best or second-best performance, followed by NIPV. Random and space-filling initialization methods (LHS-Beta, Random, and Sobol) perform noticeably worse across all settings.



**Figure 3:** Out-of-sample inference optimization performance on noisy synthetic benchmark functions under the two-shot setting ( $B = 2$ ,  $q = 24$ ) across 100 seeds per benchmark. HIPE outperforms or matches the best-performing method across all benchmarks, including on the high-dimensional Hartmann variants with added dummy variables. NIPV performs well on most tasks, but its performance degrades on tasks where hyperparameter identification is critical. Random initialization strategies perform the worst throughout.

### 278 5.3 LCBench HPO Tasks

279 We evaluate five additional tasks from LCBench in the two-shot optimization setting: Fashion-MNIST,  
 280 MiniBooNE, Car, Higgs, and Segment, using fixed, minimal observation noise. Fig. 4 displays that  
 281 HIPE substantially outperforms the competition on Fashion-MNIST, Mini-BooNE and Higgs, and  
 282 performs competitively on the remaining Segment and Car. Overall, HIPE consistently delivers the  
 283 highest relative rank, outperforming competing algorithms by a substantial margin.



**Figure 4:** Two-shot optimization results on five hyperparameter optimization tasks from LCBench across 200 seeds for each task. HIPE achieves the highest final performance on four of the five tasks and remains competitive on the remaining one (Segment). Relative rankings across the five problems show that HIPE consistently outperforms other methods. This confirms the practical relevance of our approach for real-world HPO scenarios, where both accurate predictive modeling and effective exploration are crucial.

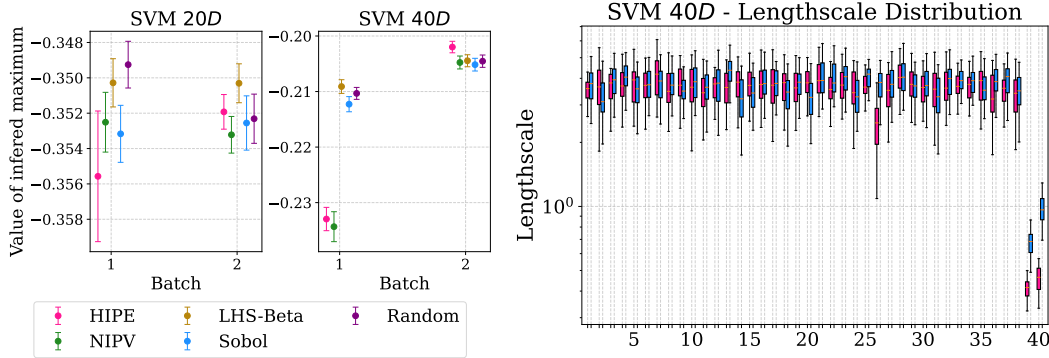
### 284 5.4 High-Dimensional SVM Tasks

285 Finally, we evaluate HIPE and baseline methods on challenging high-dimensional SVM hyperparam-  
 286 eter optimization tasks with  $D = 20$  and  $D = 40$  input dimensions, considered in similar variants by  
 287 Ament et al. (2023a); Eriksson and Jankowiak (2021); Hvarfner et al. (2024); Papenmeier et al. (2023).  
 288 For both tasks, only the last two dimensions—corresponding to the SVM’s global regularization  
 289 parameters—significantly influence the objective, while the remaining dimensions, corresponding to  
 290 feature-specific lengthscales, are of lesser importance. Effectively identifying and prioritizing these  
 291 relevant dimensions is critical for successful optimization. We again consider the two-shot setting,  
 292 using a larger batch size of  $q = 32$  due to the higher dimensionality of the problems.

293 The left panel of Fig. 5 reports the out-of sample inference performance after each batch. On the  
 294 20D task, HIPE achieves competitive, mid-range performance relative to the evaluated methods.  
 295 On the more challenging 40D task, HIPE obtains the highest performance, albeit by a narrow  
 296 margin over the next-best alternatives. The limited budget relative to the dimensionality presents a  
 297 significant challenge, as the ability to accurately learn the model hyperparameters diminishes with



298 increasing dimensionality. Despite this, we observe in the right panel of Fig. 5 that HIPE identifies  
 299 the important hyperparameters remarkably well after initialization on the 40D task—assigning the  
 300 last two dimensions lengthscales that are, on average, nearly half an order of magnitude smaller  
 301 than those inferred under a Sobol initialization. Finally, we note that the best solutions to the SVM  
 302 problem were almost exclusively located near the boundaries of the search space—particularly in the  
 303 last two dimensions—which neither NIPV nor HIPE naturally explore during initialization.



**Figure 5: Results on 20D and 40D SVM hyperparameter optimization tasks. Left:** Objective value of inferred maximum after each batch. On 20D, HIPE achieves average performance, climbing to a mid-tier result in the second batch after focusing on hyperparameter learning in the first batch. HIPE obtains a large standard error in the first batch, as two repetitions poorly infer the maximizer and suggests ill-performing points as a result. On 40D, it outperforms all baselines, demonstrating strong robustness in higher dimensions and the ability to recover after a less successful first batch. Notably, the performance of Random decreases between batches on the 20D task, demonstrating the difficulty of accurately inferring the optimum. **Right:** Log-mean estimated lengthscales hyperparameters *after initialization* on the 40D task. HIPE identifies the last two dimensions—corresponding to the SVM’s global regularization parameters – much more effectively than Sobol, assigning substantially smaller lengthscales to the relevant inputs.

## 304 6 Discussion

305 **Contributions** We introduced HIPE, a principled, hyperparameter-free information-theoretic  
 306 method for initializing Bayesian Optimization and Bayesian Active Learning algorithms. its HIPE  
 307 yields initial designs that balance coverage of (relevant) areas of the domain with the ability to  
 308 effectively learn model hyperparameters. HIPE is especially useful in the few-shot, large-batch  
 309 setting, where it achieves superior surrogate model quality compared to various initialization baselines,  
 310 as demonstrated by our experiments.

311 **Limitations** HIPE can become computationally expensive, especially for large batch sizes  $q$  in  
 312 higher dimensions  $D$ , though in many applications this cost is still insignificant compared to the  
 313 time and resources required to evaluate the underlying black-box function. Finally, the current paper  
 314 focuses on GP surrogates, and while our main insights and the general approach apply also to other  
 315 surrogate types, our implementation does not translate directly.

316 **Future work** Here we studied the “cold start” problem of initializing Bayesian Optimization and  
 317 Bayesian Active Learning from scratch. In practice, we may have access to data from related (but not  
 318 necessarily identical) problems. This motivates an extension of HIPE to the transfer-learning setting,  
 319 e.g., by means of using a multi-task GP surrogate. Additionally, incorporating prior knowledge of  
 320 domain experts in a principled fashion is of high practical relevance, and can readily be utilized in  
 321 the form of a non-uniform  $p_*$ . Finally, we are also interested in studying the multi-objective setting  
 322 in which different surrogates of potentially different form with different hyperparameters and priors  
 323 model different objectives but share observations at the same input locations.

## 324 References

325 Deepak Agarwal, Kinjal Basu, Souvik Ghosh, Ying Xuan, Yang Yang, and Liang Zhang. Online  
 326 parameter selection for web-based ranking problems. In *Proceedings of the 24th ACM SIGKDD*

327 *International Conference on Knowledge Discovery & Data Mining*, pages 23–32, 2018.

328 Sebastian Ament, Samuel Daulton, David Eriksson, Maximilian Balandat, and Eytan Bakshy. Unex-  
 329 pected improvements to expected improvement for bayesian optimization. In A. Oh, T. Naumann,  
 330 A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information*  
 331 *Processing Systems*, volume 36, pages 20577–20612. Curran Associates, Inc., 2023a.

332 Sebastian Ament, Andrew Witte, Nishant Garg, and Julius Kusuma. Sustainable concrete via bayesian  
 333 optimization. *NeurIPS 2023 Workshop on Adaptive Experimentation in the Real World*, 2023b.

334 Peter M Attia, Aditya Grover, Norman Jin, Kristen A Severson, Todor M Markov, Yang-Hung Liao,  
 335 Michael H Chen, Bryan Cheong, Nicholas Perkins, Zi Yang, et al. Closed-loop optimization of  
 336 fast-charging protocols for batteries with machine learning. *Nature*, 578(7795):397–402, 2020.

337 Maximilian Balandat, Brian Karrer, Daniel Jiang, Samuel Daulton, Ben Letham, Andrew G Wilson,  
 338 and Eytan Bakshy. Botorch: A framework for efficient monte-carlo bayesian optimization. In  
 339 *Advances in Neural Information Processing Systems*, volume 33, pages 21524–21538, 2020.

340 Felix Berkenkamp, Angela P. Schoellig, and Andreas Krause. No-regret bayesian optimization with  
 341 unknown hyperparameters. *Journal of Machine Learning Research*, 20(50):1–24, 2019.

342 Freddie Bickford Smith, Andreas Kirsch, Sebastian Farquhar, Yarin Gal, Adam Foster, and Tom  
 343 Rainforth. Prediction-oriented bayesian active learning. In *Proceedings of The 26th International*  
 344 *Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine*  
 345 *Learning Research*, pages 7331–7348, 2023.

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

349 Jakob Bossek, Carola Doerr, and Pascal Kerschke. Initial design strategies and their effects on  
 350 sequential model-based optimization: an exploratory case study based on bbob. In *Proceedings of*  
 351 *the 2020 Genetic and Evolutionary Computation Conference, GECCO ’20*, page 778–786, 2020.

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

354 Kathryn Chaloner and Isabella Verdinelli. Bayesian experimental design: A review. *Statistical*  
 355 *Science*, 10(3):273–304, 1995.

356 Xi Chen and Qiang Zhou. Sequential experimental designs for stochastic kriging. In *Proceedings of*  
 357 *the 2014 Winter Simulation Conference, WSC ’14*, page 3821–3832, 2014.

358 D. Austin Cole, Robert B. Gramacy, James E. Warner, Geoffrey F. Bomarito, Patrick E. Leser, and  
 359 William P. Leser and. Entropy-based adaptive design for contour finding and estimating reliability.  
 360 *Journal of Quality Technology*, 55(1):43–60, 2023.

361 Samuel Daulton, David Eriksson, Maximilian Balandat, and Eytan Bakshy. Multi-objective bayesian  
 362 optimization over high-dimensional search spaces. In *Proceedings of the Thirty-Eighth Conference*  
 363 *on Uncertainty in Artificial Intelligence*, Proceedings of Machine Learning Research. PMLR, 2022.

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

366 David Eriksson, Michael Pearce, Jacob Gardner, Ryan D Turner, and Matthias Poloczek. Scalable  
 367 global optimization via local bayesian optimization. In *Advances in Neural Information Processing*  
 368 *Systems*, volume 32, 2019.

369 Qing Feng, Samuel Daulton, Benjamin Letham, Maximilian Balandat, and Eytan Bakshy. Experi-  
 370 menting, fast and slow: Bayesian optimization of long-term outcomes with online experiments. In  
 371 *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1*,  
 372 KDD ’25, pages 2235–2246, 2025.

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

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

377 Peter I Frazier and Jiale Wang. Bayesian optimization for materials design. In *Information science*  
 378 *for materials discovery and design*, pages 45–75. Springer, 2016.

379 Jacob Gardner, Chuan Guo, Kilian Weinberger, Roman Garnett, and Roger Grosse. Discovering and  
380 Exploiting Additive Structure for Bayesian Optimization. In *Proceedings of the 20th International  
381 Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning  
382 Research*, pages 1311–1319. PMLR, 20–22 Apr 2017.

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

386 Roman Garnett. *Bayesian Optimization*. Cambridge University Press, 2023.

387 Roman Garnett, Michael A. Osborne, and Philipp Hennig. Active learning of linear embeddings  
388 for gaussian processes. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial  
389 Intelligence*, UAI’14, page 230–239, 2014.

390 Robert B Gramacy and Herbert KH Lee. Adaptive design and analysis of supercomputer experiments.  
391 *Technometrics*, 51(2):130–145, 2009.

392 Robert B. Gramacy, Annie Sauer, and Nathan Wycoff. Triangulation candidates for bayesian opti-  
393 mization. In *Proceedings of the 36th International Conference on Neural Information Processing  
394 Systems*, NIPS ’22, 2022.

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

397 Carlos Guestrin, Andreas Krause, and Ajit Paul Singh. Near-optimal sensor placements in gaussian  
398 processes. In *Proceedings of the 22nd international conference on Machine learning*, pages  
399 265–272, 2005.

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

402 J. M. Hernández-Lobato, M. W. Hoffman, and Z. Ghahramani. Predictive entropy search for efficient  
403 global optimization of black-box functions. In *Advances in Neural Information Processing Systems*,  
404 2014.

405 Matthew D. Hoffman and Andrew Gelman. The no-u-turn sampler: Adaptively setting path lengths  
406 in hamiltonian monte carlo. *Journal of Machine Learning Research*, 15(47):1593–1623, 2014.

407 Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian Active Learning for  
408 Classification and Preference Learning. *arXiv e-prints*, page arXiv:1112.5745, December 2011.

409 Carl Hvarfner, Frank Hutter, and Luigi Nardi. Joint entropy eearch for maximally-informed bayesian  
410 optimization. In *Proceedings of the 36th International Conference on Neural Information Process-  
411 ing Systems*, 2022.

412 Carl Hvarfner, Erik Hellsten, Frank Hutter, and Luigi Nardi. Self-correcting bayesian optimization  
413 through bayesian active learning. In *Thirty-seventh Conference on Neural Information Processing  
414 Systems*, 2023.

415 Carl Hvarfner, Erik O. Hellsten, and Luigi Nardi. Vanilla bayesian optimization performs great  
416 in high dimensions. In *Proceedings of the 41st International Conference on Machine Learning*,  
417 ICML’24, 2024.

418 M.E. Johnson, L.M. Moore, and D. Ylvisaker. Minimax and maximin distance designs. *Journal of  
419 Statistical Planning and Inference*, 26(2):131–148, 1990. ISSN 0378-3758.

420 D. Jones, M. Schonlau, and W. Welch. Efficient global optimization of expensive black-box functions.  
421 *Journal of Global Optimization*, 13:455–492, 12 1998.

422 Andreas Kirsch, Joost Van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch  
423 acquisition for deep bayesian active learning. *Advances in neural information processing systems*,  
424 32, 2019.

425 Andreas Kirsch, Tom Rainforth, and Yarin Gal. Test distribution-aware active learning: A principled  
426 approach against distribution shift and outliers, 2021.

427 Andreas Krause, Carlos Guestrin, Anupam Gupta, and Jon Kleinberg. Near-optimal sensor place-  
428 ments: Maximizing information while minimizing communication cost. In *Proceedings of the 5th  
429 international conference on Information processing in sensor networks*, pages 2–10, 2006.

- Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9(8):235–284, 2008.
- Vidhi Lalchand and Carl Edward Rasmussen. Approximate inference for fully bayesian gaussian process regression. In *Symposium on Advances in Approximate Bayesian Inference*, pages 1–12. PMLR, 2020.
- Benjamin Letham, Brian Karrer, Guilherme Ottoni, and Eytan Bakshy. Constrained bayesian optimization with noisy experiments. *Bayesian Analysis*, 14(2):495–519, 2019.
- David JC MacKay. Information-based objective functions for active data selection. *Neural computation*, 4(4):590–604, 1992.
- David J.C MacKay. Bayesian neural networks and density networks. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 354(1):73–80, 1995. Proceedings of the Third Workshop on Neutron Scattering Data Analysis.
- Natalie Maus, Haydn Jones, Juston Moore, Matt J Kusner, John Bradshaw, and Jacob Gardner. Local latent space bayesian optimization over structured inputs. In *Advances in Neural Information Processing Systems*, volume 35, pages 34505–34518, 2022.
- J. Mockus, V. Tiesis, and A. Zilinskas. The application of Bayesian methods for seeking the extremum. *Towards Global Optimization*, 2(117-129):2, 1978.
- Henry B. Moss, David S. Leslie, Javier Gonzalez, and Paul Rayson. Gibbon: General-purpose information-based bayesian optimisation. *Journal of Machine Learning Research*, 22, 2021.
- Samuel Müller, Matthias Feurer, Noah Hollmann, and Frank Hutter. PFNs4BO: In-context learning for Bayesian optimization. In *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.
- Werner G Müller and Dale L Zimmerman. Optimal designs for variogram estimation. *Environmetrics: The official journal of the International Environmetrics Society*, 10(1):23–37, 1999.
- Willie Neiswanger, Ke Alexander Wang, and Stefano Ermon. Bayesian algorithm execution: Estimating computable properties of black-box functions using mutual information. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8005–8015, 2021.
- Willie Neiswanger, Lantao Yu, Shengjia Zhao, Chenlin Meng, and Stefano Ermon. Generalizing bayesian optimization with decision-theoretic entropies. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, 2024.
- Michael A Osborne. *Bayesian Gaussian processes for sequential prediction, optimisation and quadrature*. PhD thesis, Oxford University, UK, 2010.
- Art B. Owen. *Practical Quasi-Monte Carlo Integration*. <https://artowen.su.domains/mc/practicalqmc.pdf>, 2023.
- Leonard Papenmeier, Luigi Nardi, and Matthias Poloczek. Bounce: a Reliable Bayesian Optimization Algorithm for Combinatorial and Mixed Spaces. In *Advances in Neural Information Processing Systems*, 2023.
- Chiwoo Park, Robert Waelder, Bonggwon Kang, Benji Maruyama, Soondo Hong, and Robert Gramacy. Active learning of piecewise gaussian process surrogates, 2024. URL <https://arxiv.org/abs/2301.08789>.
- Tom Rainforth, Adam Foster, Desi R. Ivanova, and Freddie Bickford Smith. Modern Bayesian Experimental Design. *Statistical Science*, 39(1):100 – 114, 2024.
- Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.
- Christoffer Riis, Francisco Antunes, Frederik Hüttel, Carlos Lima Azevedo, and Francisco Pereira. Bayesian active learning with fully bayesian gaussian processes. In *Advances in Neural Information Processing Systems*, volume 35, pages 12141–12153, 2022.
- Annie Sauer, Robert B. Gramacy, and David Higdon and. Active learning for deep gaussian process surrogates. *Technometrics*, 65(1):4–18, 2023.

483 Sambu Seo, M. Wallat, T. Graepel, and K. Obermayer. Gaussian process regression: active data  
484 selection and test point rejection. In *Proceedings of the IEEE-INNS-ENNS International Joint Con-*  
485 *ference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives*  
486 *for the New Millennium*, volume 3, 2000.

487 J. Snoek, H. Larochelle, and R. P. Adams. Practical bayesian optimization of machine learning  
488 algorithms. In *Advances in Neural Information Processing Systems*, 2012.

489 J. Snoek, K. Swersky, R. Zemel, and R. Adams. Input warping for bayesian optimization of non-  
490 stationary functions. In *Proceedings of the 31st International Conference on Machine Learning*,  
491 volume 32 of *Proceedings of Machine Learning Research*, 2014.

492 N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger. Information-theoretic regret bounds for  
493 gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58  
494 (5), 2012.

495 Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process opti-  
496 mization in the bandit setting: no regret and experimental design. In *Proceedings of the 27th*  
497 *International Conference on International Conference on Machine Learning*, ICML’10, 2010.

498 Samuel Stanton, Wesley Maddox, Nate Gruver, Phillip Maffettone, Emily Delaney, Peyton Greenside,  
499 and Andrew Gordon Wilson. Accelerating Bayesian optimization for biological sequence design  
500 with denoising autoencoders. In *Proceedings of the 39th International Conference on Machine*  
501 *Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 20459–20478. PMLR,  
502 2022.

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

505 Ben Tu, Axel Gandy, Nikolas Kantas, and Behrang Shafei. Joint entropy search for multi-objective  
506 bayesian optimization. In *Advances in Neural Information Processing Systems*, 2022.

507 Joaquin Vanschoren, Jan N Van Rijn, Bernd Bischl, and Luis Torgo. Openml: networked science in  
508 machine learning. *ACM SIGKDD Explorations Newsletter*, 15(2):49–60, 2014.

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

511 Christopher Williams and Carl Rasmussen. Gaussian processes for regression. In *Advances in Neural*  
512 *Information Processing Systems*, volume 8, 1995.

513 James T. Wilson, Riccardo Moriconi, Frank Hutter, and Marc Peter Deisenroth. The reparameteriza-  
514 tion trick for acquisition functions. *arXiv e-prints*, 2017.

515 James T. Wilson, Viacheslav Borovitskiy, Alexander Terenin, Peter Mostowsky, and Marc Peter  
516 Deisenroth. Efficiently sampling functions from gaussian process posteriors. In *International*  
517 *Conference on Machine Learning*, 2020.

518 Boya Zhang, D. Austin Cole, and Robert B. Gramacy. Distance-distributed design for gaussian  
519 process surrogates, 2019.

520 Lucas Zimmer, Marius Lindauer, and Frank Hutter. Auto-pytorch tabular: Multi-fidelity metalearning  
521 for efficient and robust autodl. *IEEE Transactions on Pattern Analysis and Machine Intelligence*,  
522 43(9):3079 – 3090, 2021.

523 Dale L Zimmerman. Optimal network design for spatial prediction, covariance parameter estimation,  
524 and empirical prediction. *Environmetrics: The official journal of the International Environmetrics*  
525 *Society*, 17(6):635–652, 2006.

526 Juliusz Ziomek, Masaki Adachi, and Michael A. Osborne. Bayesian optimisation with unknown hy-  
527 perparameters: Regret bounds logarithmically closer to optimal. In *Advances in Neural Information*  
528 *Processing Systems*, volume 37, pages 86346–86374, 2024.

## NeurIPS Paper Checklist

### 1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: All claims made in the abstract and in the introduction are detailed and backed up with theoretical and empirical results. The content in the paper does not go beyond the broad scope of what is claimed in abstract and introduction.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

### 2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: Limitations such as computational scalability are discussed in Section 6 and are acknowledged throughout the paper.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

### 3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The only theoretical contribution in this paper is Proposition 1, for which assumptions are discussed and a proof is provided in Appendix B.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

#### 4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Implementation details on our main contribution are discussed in the paper, the experimental setup is detailed in the Appendix, and our results can easily be reproduced using the accompanying code submission.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

## 5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Our code submission allows to easily reproduce the results in the paper. Furthermore, our main contribution, HIPE, will be made available in the popular open source library BoTorch (<https://github.com/pytorch/botorch>) upon publication.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

## 6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Details on the experimental setup are provided in Appendix A.1 as well as part of the code submission.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

## 7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: All experimental results provide confidence intervals and experiments have been performed with sufficiently many replications to highlight any claimed differences between our contributions and baselines.

Guidelines:

- The answer NA means that the paper does not include experiments.



- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

## 8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: The compute resources used for the experiments in the paper are summarized in Appendix A.1.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

## 9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: The research done for this paper neither involved any human subjects nor did it generate or use any data sets that could raise potential privacy concerns. Moreover, since it focuses on foundational methodological contributions to Bayesian Optimization and Bayesian Active Learning, it is highly unlikely to generate any adverse societal impact and potential harmful consequences.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

## 10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: This paper focuses on foundational methodological contributions to Bayesian Optimization and Bayesian Active Learning. As such, the contributions are highly unlikely to generate any adverse societal impact and potential harmful consequences.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

## 11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: This work does not involve the release of any data or models that would pose any risk for misuse.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

## 12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Work on this paper did not require the use of any models. Any code or data sets that were used are properly credited and their respective licenses are mentioned and properly respected.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, [paperswithcode.com/datasets](https://paperswithcode.com/datasets) has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

**13. New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [\[Yes\]](#)

Justification: The code submission includes documentation both in the form of a readme file as well as docstrings throughout the code.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

**14. Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [\[NA\]](#)

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

**15. Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

844 Answer: [NA]  
845 Justification: The paper does not involve crowdsourcing nor research with human subjects.  
846 Guidelines:  
847 • The answer NA means that the paper does not involve crowdsourcing nor research with  
848 human subjects.  
849 • Depending on the country in which research is conducted, IRB approval (or equivalent)  
850 may be required for any human subjects research. If you obtained IRB approval, you  
851 should clearly state this in the paper.  
852 • We recognize that the procedures for this may vary significantly between institutions  
853 and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the  
854 guidelines for their institution.  
855 • For initial submissions, do not include any information that would break anonymity (if  
856 applicable), such as the institution conducting the review.

857 **16. Declaration of LLM usage**

858 Question: Does the paper describe the usage of LLMs if it is an important, original, or  
859 non-standard component of the core methods in this research? Note that if the LLM is used  
860 only for writing, editing, or formatting purposes and does not impact the core methodology,  
861 scientific rigorousness, or originality of the research, declaration is not required.

862 Answer: [NA]

863 Justification: The core method development in this research does not involve LLMs in any  
864 way.

865 Guidelines:  
866 • The answer NA means that the core method development in this research does not  
867 involve LLMs as any important, original, or non-standard components.  
868 • Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>)  
869 for what should or should not be described.

## 870 A Experimental Setup

871 We describe the full experimental setup used in the paper: the design of the Bayesian optimization and  
872 active learning loops, the benchmarks used, the compute as well as the licenses of all software and  
873 datasets. Our code, including all the benchmarks and plotting to reproduce our results, is available at  
874 <https://github.com/hipeneurips/HIPE>.

### 875 A.1 Bayesian Optimization Loop

876 All experiments were conducted using a standardized pipeline based on BoTorch (Balandat et al.,  
877 2020) and GPyTorch (Gardner et al., 2018). We use fully Bayesian Gaussian Process (GP) models,  
878 with hyperparameter inference performed via the No-U-Turn Sampler (NUTS) (Hoffman and Gelman,  
879 2014) as implemented in Pyro (Bingham et al., 2018). Unless otherwise specified, we draw 192  
880 burn-in samples followed by 288 hyperparameter samples, retaining every 24<sup>th</sup> sample for evaluation.

881 Our GP prior is adapted from Hvarfner et al. (2024) to better suit a fully Bayesian setting. Specifically,  
882 we set  $\mu_0 = -0.75$  and  $\sigma = 0.75$ , resulting in  $\ell_d \sim \mathcal{LN}(0.75 + \log(D)/2, 0.75)$ . The noise  
883 standard deviation is modeled as  $\sigma_\varepsilon \sim \mathcal{LN}(-5.5, 0.75)$ , and the constant mean parameter follows  
884  $c \sim \mathcal{N}(0, 0.25)$ . These modifications ensure that the means of the priors for  $\ell_d$  and  $\sigma_\varepsilon^2$  approximately  
885 match the modes of the corresponding parameters in the prior proposed by Hvarfner et al. (2024),  
886 producing similar hyperparameter values in practice.

887 Acquisition functions are optimized jointly over the batch using multi-start L-BFGS-B optimization  
888 with 4 random restarts and 384 initial samples drawn from a scrambled Sobol sequence. For the  
889 optimization of LogEI, we additionally sample 384 points from a Gaussian distribution centered at  
890 the current incumbent to improve local search performance.

891 For our proposed HIPE method and relevant baselines (BALD and NIPV), Monte Carlo estimators  
892 use  $M = 12$  hyperparameter samples,  $T = 1024$  test points drawn uniformly from the search space,  
893 and  $N = 128$  predictive posterior samples.

### 894 A.2 Benchmarks

895 We use three types of benchmarks: synthetic optimization test functions, surrogate-based hyper-  
896 parameter optimization tasks from LCBench (Zimmer et al., 2021), and high-dimensional SVM  
897 hyperparameter optimization problems. Synthetic functions are standard benchmarks for evaluating  
898 active learning and Bayesian optimization under controlled noise and dimensionality. LCBench tasks  
899 are GP surrogate models trained on 2000 evaluations of multi-layer perceptrons (MLPs) on real-world  
900 datasets, trained with a Matern 3/2 kernel. We make all surrogates as part of our code. The SVM  
901 benchmarks mimic the setup in Ament et al. (2023a), where feature selection is applied to the SVM  
902 problem originally proposed in Eriksson and Jankowiak (2021) to obtain a lower-dimensional HPO  
903 task than the original, 388D problem. In synthetic functions, additive Gaussian noise is introduced  
904 directly to the function evaluations. The LCBench benchmarks rely on pre-trained surrogate models,  
905 where the posterior mean is evaluated, and in cases where  $\sigma_\varepsilon$  is non-zero, additional noise is added to  
906 the evaluation of the posterior mean.

### 907 A.3 Compute Resources

908 All experiments were conducted using an NVIDIA A40 GPU cluster. The compute usage to run all  
909 experiments in the main paper amounts to approximately 1000 GPU hours, and an additional 500  
910 GPU hours to produce all the results provided in Appendix C.

### 911 A.4 Licenses

912 The following software packages, libraries and datasets were used in our experiments and for  
913 presenting the results in the paper:

- 914 • **GPyTorch, BoTorch, Hydra:** MIT License
- 915 • **PyTorch, NumPy, SciPy, Pandas:** BSD Licenses
- 916 • **Matplotlib, Seaborn:** PSF/BSD Licenses

**Table 1:** Noise levels for all Active Learning (AL) and Bayesian Optimization (BO) benchmarks.

Category	Benchmark	Task Type	Dimensionality	$\sigma_\epsilon$
Synthetic	Ackley (4D)	BO	4	2.0
	Hartmann (6D)	BO / AL	6	0.5
	Hartmann (6D)	BO / AL	12	0.5
	Hartmann (4D)	BO	4	0.5
	Hartmann (4D)	BO	8	0.5
LCBench	Car	AL	7	2.5
	Australian	AL	7	2.5
	Fashion-MNIST	BO	7	0.0
	MiniBooNE	BO	7	0.0
	Car	BO	7	0.0
	Higgs	BO	7	0.0
	Segment	BO	7	0.0
SVM	Feature-reduced SVM	BO	20	0.0
	Feature-reduced SVM	BO	40	0.0

• **LCBench:** Apache License

## B Derivation of HIPE as Joint Information Gain

Recall Proposition 1 from section 4.1.

**Proposition 1** (Equivalence of  $\text{HIPE}_{\beta=1}$  to Joint Information Gain). *The  $\text{HIPE}_\beta$  acquisition function with  $\beta = 1$  is equivalent to maximizing the expected joint information gain over test function values  $y(\mathbf{x}_*)$  and model hyperparameters  $\boldsymbol{\theta}$  acquired by a candidate batch  $\mathbf{X}$ . Formally,*

$$\arg \max_{\mathbf{X} \in \mathbb{R}^{q \times D}} \text{HIPE}_{\beta=1}(\mathbf{X}; p_*) = \arg \max_{\mathbf{X} \in \mathbb{R}^{q \times D}} \mathbb{E}_{\mathbf{x} \sim p_*} [\text{EIG}(y(\mathbf{x}_*), \boldsymbol{\theta}; \mathbf{X})]. \quad (6)$$

*Proof of Proposition 1.* For  $\beta = 1$ , we have

$$\text{HIPE}_1(\mathbf{X}) = -\mathbb{E}_{\boldsymbol{\theta}, y(\mathbf{X})} [\mathbb{E}_{\mathbf{x}_*} [\text{H}[y(\mathbf{x}_*) | \boldsymbol{\theta}, y(\mathbf{X})]]] + (\text{H}[y(\mathbf{X})] - \mathbb{E}_{\boldsymbol{\theta}} [\text{H}[y(\mathbf{X}) | \boldsymbol{\theta}]])$$

Therefore, with  $\mathbf{X}^* := \arg \max_{\mathbf{X} \in \mathbb{R}^{q \times D}} \text{HIPE}_1(\mathbf{X})$ ,

$$\mathbf{X}^* = \arg \max_{\mathbf{X} \in \mathbb{R}^{q \times D}} -\mathbb{E}_{\boldsymbol{\theta}, y(\mathbf{X})} [\mathbb{E}_{\mathbf{x}_*} [\text{H}[y(\mathbf{x}_*) | \boldsymbol{\theta}, y(\mathbf{X})]]] + (\text{H}[y(\mathbf{X})] - \mathbb{E}_{\boldsymbol{\theta}} [\text{H}[y(\mathbf{X}) | \boldsymbol{\theta}]]) \quad (11a)$$

$$= \arg \max_{\mathbf{X} \in \mathbb{R}^{q \times D}} -\mathbb{E}_{\boldsymbol{\theta}, y(\mathbf{X})} [\mathbb{E}_{\mathbf{x}_*} [\text{H}[y(\mathbf{x}_*) | \boldsymbol{\theta}, y(\mathbf{X})]]] - \mathbb{E}_{\boldsymbol{\theta}} [\text{H}[\boldsymbol{\theta} | y(\mathbf{X})]] \quad (11b)$$

$$= \arg \max_{\mathbf{X} \in \mathbb{R}^{q \times D}} \mathbb{E}_{\mathbf{x}_*} [\text{H}[y(\mathbf{x}_*), \boldsymbol{\theta}] - \mathbb{E}_{y(\mathbf{X})} [\text{H}[y(\mathbf{x}_*), \boldsymbol{\theta} | y(\mathbf{X})]]] \quad (\text{by def}) \quad (11c)$$

$$= \arg \max_{\mathbf{X} \in \mathbb{R}^{q \times D}} \mathbb{E}_{\mathbf{x}_*} [\text{EIG}(y(\mathbf{x}_*), \boldsymbol{\theta}; \mathbf{X})] \quad (11d)$$

where the equalities follow from the Bayes' rule of conditional entropy and the fact that the  $\arg \max$  is independent of quantities which do not involve  $\mathbf{X}$ .  $\square$

## C Additional Experiments

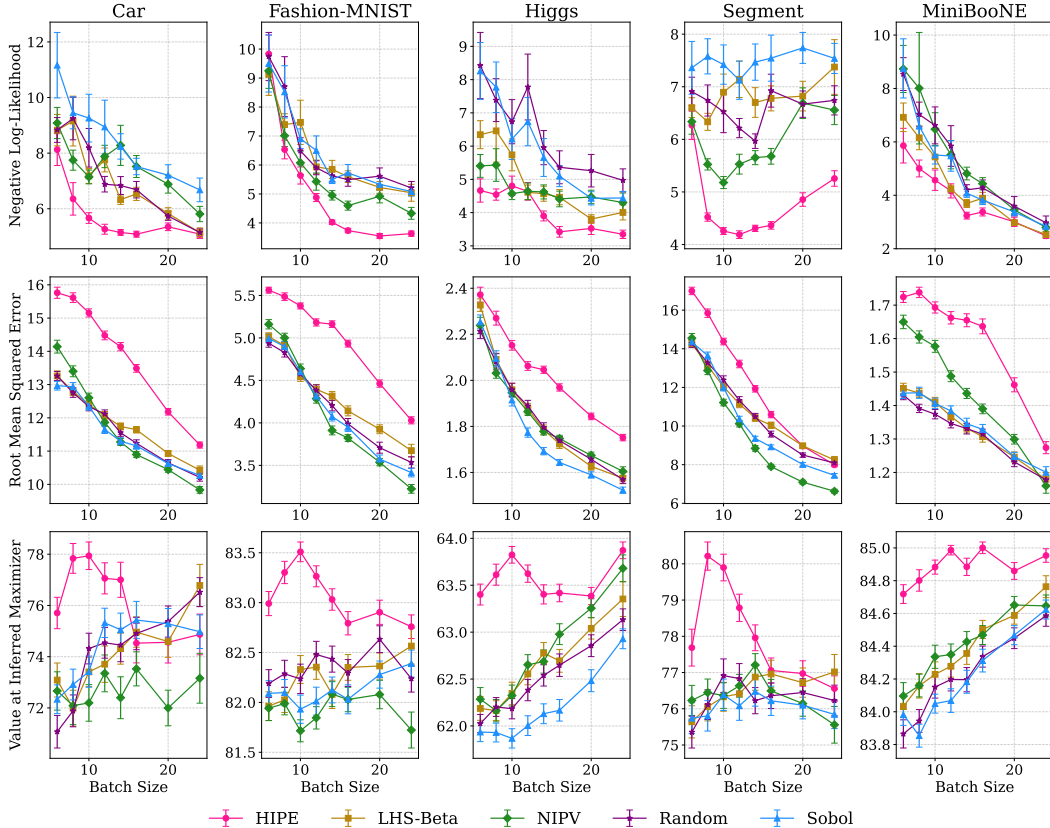
We present supplementary experiments to further analyze the behavior of the evaluated methods under varying experimental conditions. These results provide additional insights into the robustness of the methods with respect to batch size during initialization and active learning, as well as their performance trade-offs across different evaluation metrics.

### 933 C.1 Impact of $q$ on Predictive Performance

934 We investigate how the initialization batch size  $q$  affects predictive quality and inference performance  
 935 on the LCBench tasks introduced in Section 5.3. We vary  $q$  from 6 to 24 in increments of 2 up to  
 936  $q = 16$ , then in steps of 4. For each batch size, we evaluate models trained on the initialization batch  
 937 only — before any active learning iterations — under a fixed-noise setting. This isolates the impact  
 938 of the initial design and avoids the confounding effects of subsequent data acquisition.

939 Fig. 6 summarizes the results across three key metrics: test-set NLL, test-set RMSE, and out-of-  
 940 sample inference performance. HIPE consistently achieves the lowest NLL across most benchmarks  
 941 and batch sizes, particularly in the  $q \in [12, 16]$  range, demonstrating strong calibration and hyper-  
 942 parameter learning. In contrast, RMSE results show that HIPE often underperforms relative to  
 943 simpler baselines like Sobol and Random, suggesting a trade-off between predictive calibration and  
 944 pointwise accuracy. Inference performance shows HIPE leading for smaller batch sizes ( $q < 16$ ),  
 945 although performance plateaus or declines at higher  $q$ , indicating that additional samples are not  
 946 necessarily helpful for inference when uncertainty in relevant parameters remains high. This can  
 947 be explained that the inferred  $\arg \max$  is further from observed data, for most methods, as batch  
 948 size increases. Naturally, this "aggressive" increases the risk of any algorithm to be incorrect in its  
 949 inference. Notably, as HIPE's initialization is generally more centered than competing methods', it  
 950 obtains higher in-sample values across most batch sizes.

951 Unlike the active learning setup, these tasks are noiseless and assume fixed, known noise levels during  
 952 training. This removes uncertainty in the noise model and creates a distinct evaluation setting focused  
 953 solely on input selection and hyperparameter inference.



**Figure 6:** Effect of initialization batch size  $q$  on predictive quality and inference across LCBench tasks. Each row shows performance on one of three metrics after the initialization batch: (top) NLL, (middle) RMSE, and (bottom) out-of-sample inference. HIPE consistently leads in NLL and small- $q$  inference, while other methods achieve lower RMSE, indicating a trade-off between model calibration and pointwise prediction accuracy.

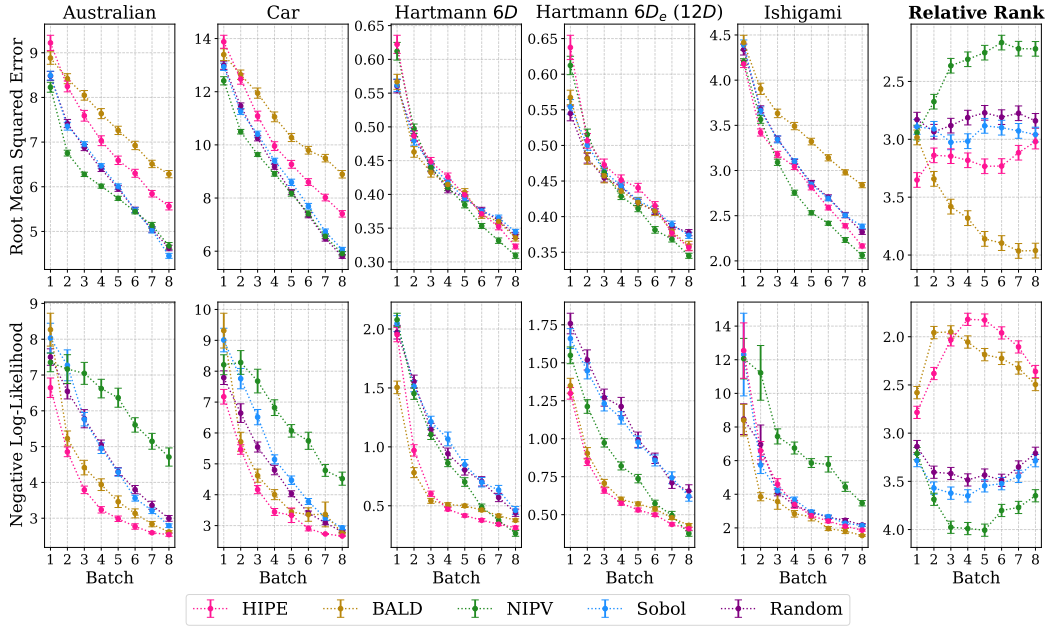
## 954 C.2 Active Learning with Different Batch Sizes

955

956 We analyze the effect of batch size in the active learning setting by comparing model performance  
 957 under small batches ( $q = 8$ ) and large batches ( $q = 24$ ). This evaluation assesses each method’s  
 958 robustness to changes in batch size and its capability to maintain predictive accuracy and effective  
 959 hyperparameter learning under varying evaluation budgets.

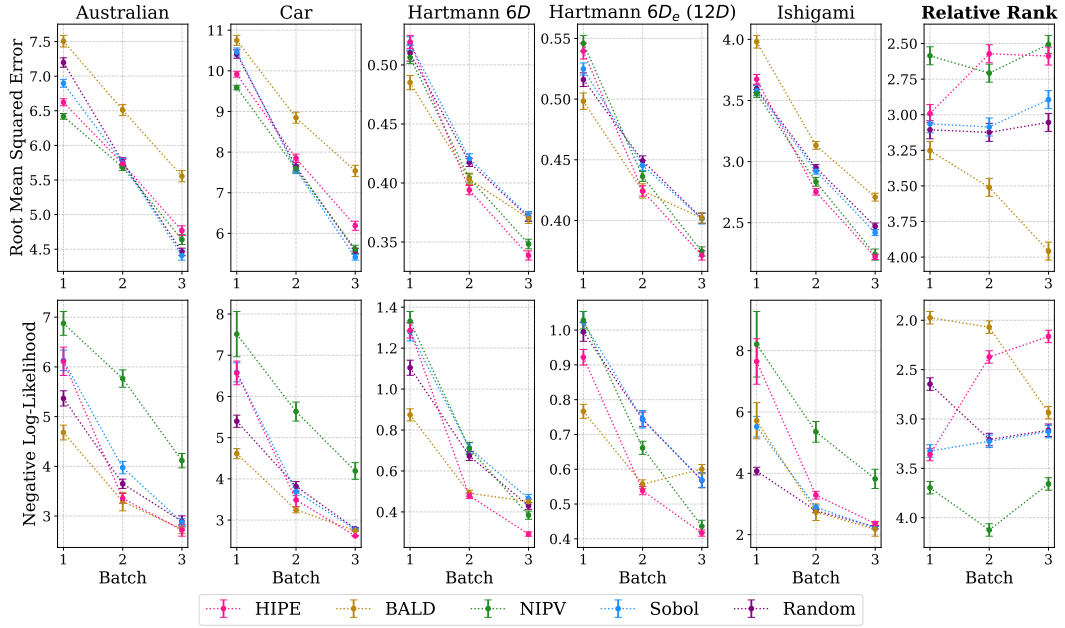
960 As shown in Fig. 7, for smaller batches of  $q = 8$ , HIPE achieves the best performance on the  
 961 NLL metric and significantly outperforms BALD—the second-best method on NLL—in terms of  
 962 MSE. In this setting, NIPV and BALD exhibit inconsistent behavior, alternating between the worst  
 963 performance on MLL and RMSE, respectively.

964 When the batch size increases to  $q = 24$ , as visualized in Fig. 7, HIPE continues to perform strongly,  
 965 consistently ranking among the top two methods across both evaluation metrics. Notably, its relative  
 966 performance improves with larger batch sizes, particularly in terms of RMSE, indicating that HIPE  
 967 scales more effectively with increased parallelism in query selection. This suggests that HIPE is  
 968 better suited for scenarios requiring efficient learning under larger batch evaluations.



**Figure 7:** Model accuracy results in the batch active learning setting with smaller batches ( $q = 8$ ). RMSE is reported across various synthetic and LCBench surrogate tasks over 8 batches, using 100 random seeds per benchmark. HIPE achieves the best performance on NLL and outperforms BALD, the second-best method on NLL, in terms of RMSE. NIPV and BALD alternate in exhibiting the worst performance on MLL and RMSE, respectively.





**Figure 8:** Model accuracy results in the batch active learning setting with larger batches ( $q = 24$ ). RMSE is reported across various synthetic and LCBench surrogate tasks over 3 batches, using 100 random seeds per benchmark. HIPE consistently ranks among the top two methods across both metrics, achieving a strong balance between hyperparameter learning and predictive accuracy. With larger batch sizes, HIPE exhibits improved relative performance, particularly on RMSE, demonstrating effective scalability under increased parallel evaluations.