VBO-MI: A Differentiable Bayesian Optimization Framework via Variational Mutual Information Estimation

Anonymous authors
Paper under double-blind review

Abstract

Many real-world tasks require optimizing expensive black-box functions accessible only through noisy evaluations, a setting commonly addressed with Bayesian optimization (BO). While Gaussian processes (GPs) and Bayesian neural networks (BNNs) have driven significant progress, GP-based methods scale cubically with observations, and BNN-based approaches remain burdened by expensive posterior sampling such as Hamiltonian Monte Carlo, and inner-loop acquisition optimization in high-dimensional setting. In this work, we propose **VBO-MI** (Variational Bayesian Optimization with Mutual Information), a fully gradient-based BO framework that requires no assumptions on a Gaussian prior or a variational posterior family over f. While our framework uses gradient-based updates to train the internal action and information-estimation networks, it treats the objective function f as a strict black box, requiring only zeroth-order evaluations (input-output pairs). To enable end-to-end gradient flow, we employ an actor-critic architecture consisting of an action-net to navigate the input space and a variational critic to estimate information gain. This formulation effectively eliminates the traditional inner-loop acquisition optimization bottleneck, achieving up to a $10^2\times$ reduction in FLOPs compared to BNN-BO baselines. Additionally, we design a lightweight surrogate that approximates the objective from observed data using a replay buffer, which reduces the total required real function queries to one batch per iteration. We provide theoretical analysis for our framework, and establish the consistency of VBO-MI estimates of the optimum point. We evaluate our method on a diverse suite of benchmarks, including high-dimensional synthetic functions (Ackley, Levy, Griewank) and complex real-world tasks such as Rover Trajectory planning optimization, the Lunar Lander, and Pest Control problems. Our experiments demonstrate that VBO-MI provides the same or superior optimization performance over the various baselines.

1 Introduction

Bayesian optimization (BO) has matured into a de-facto tool for expensive global black-box optimization with applications in hyperparameter tuning, materials design, and experimental sciences (Frazier, 2018; Garnett, 2023). In Bayesian Optimization, the algorithm treats f as a black-box function and aims to optimize it sequentially under a finite budget N that is, with access to only N evaluations (input–output pairs) of the function (Cox & John, 1992). At each evaluation, the algorithm builds a probabilistic model of f, using the observed input–output pairs and uses this model to guide the selection of the next evaluation points, by maximizing an acquisition function that trades off exploration and exploitation with the goal of finding the global optimum as efficiently as possible.

Traditionally, Gaussian process (GP) assumptions for the surrogate coupled with various acquisition functions, such as expected improvement (Jones et al., 1998), entropy search (Hennig & Schuler, 2012) (Hennig et al., 2022), and Upper confidence bound (UCB) (Srinivas et al., 2010) have been effective in developing various efficient Bayesian optimization algorithms (Rudner et al., 2022). However, because standard GPs require inverting a covariance matrix (which scales cubically in the number of observations), they become computationally prohibitive for large datasets. Although approximate GPs such as sparse inducing-point models have been proposed (McIntire et al., 2016), and more recent techniques based on GP approximation scale to high-dimensional BO problems (Jimenez & Katzfuss, 2023), these methods still face trade-offs between the accuracy of finding optimal points, local fidelity, and uncertainty quantification.

Recently, several works have been focused on the replacing GPs with Bayesian neural networks (BNNs) in Bayesian optimization algorithms (Li et al., 2024; Springenberg et al., 2016). Various methods have been used in the literature for computing posteriors in the BNNs, these include variational inference with a Sparse Gaussian assumption (Wei et al., 2024), Gamma variational distribution (Cheng et al., 2025), Hamiltonian Monte Carlo (HMC) among others (Li et al., 2024). However, all of the methods rely on the specific form of distribution for posterior sampling, which limits their capability to search entire space, specially when the underlying function is complex and high dimensional. Another limitation is computational efficiency (Cheng et al., 2025) as training BNNs (Li et al., 2024) is prohibitive.
To address these limitations, this work take a different approach and develops a framework that models the f without assuming a specific parametric family. By leveraging a flexible variational representation, the proposed approach enables more expressive uncertainty modeling and improved exploration in complex, high-dimensional search spaces. Our method enables full gradient flow through the internal optimization process, resulting in faster training. Critically, while the framework utilizes gradient-based updates for its internal neural components, it maintains the standard Bayesian optimization requirement of working strictly with zeroth-order evaluations (input-output pairs) of the black-box function f. Specifically, we leverage recent advances in variational mutual information estimation (Song & Ermon, 2020; Belghazi et al., 2018) to design a novel acquisition function that is directly approximated from data samples. This acquisition function is jointly trained with a separate neural network, termed the action-net, which explores the input space and guides the optimization process. Our main contributions are as follows:

1. We propose a general Bayesian optimization framework that does not assume any specific variational form for the surrogate model, allowing greater flexibility in modeling complex objective functions.

2. We introduce a novel variational acquisition function (which is used as a loss function) derived from a variational mutual information formulation, which provides an explicit and principled trade-off between exploration and exploitation.

3. We provide theoretical analysis establishing the consistency and convergence properties of the proposed variational estimates and loss functions.

4. We benchmark VBO-MI across a diverse suite of baselines, including BNN based Bayesian optimization methods, and test it on high-dimensional synthetic functions and complex real-world control tasks such as Rover trajectory optimization, Pest control, and the Lunar lander problem. Our results show that VBO-MI provides competetive optimization performance over various set of baselines with reduced complexity overhead.

Notation. Random vectors are denoted by bold lowercase symbols (e.g., $\mathbf{x}_t \in \mathbb{R}^d$). The sequence of random vectors observed up to step t is written as $\mathbf{X}_t = [\mathbf{x}_1, \ldots, \mathbf{x}_t] \in \mathbb{R}^{t\times d}$. A realization (sample) of a random vector uses a superscript index (i). Thus $\mathbf{x}_t^{(i)} \in \mathbb{R}^d$, $\mathbf{X}_t^{(i)} = [\mathbf{x}_1^{(i)}, \ldots, \mathbf{x}_t^{(i)}] \in \mathbb{R}^{t\times d}$. Here, the index i distinguishes samples in a mini-batch or dataset. Moreover, scalar components of a realization are written as $\mathbf{x}_t^{(i)} = [\, x_{1,t}^{(i)}, \ldots, x_{d,t}^{(i)} \,]$. Calligraphic letters (e.g. $\mathcal{D} \subset \mathbb{R}^d$) denote domains or sets. Expectations are written as $\mathbb{E}_{P_X}[\cdot]$ under distribution P_X. The joint and product-of-marginals distributions are denoted by $P_{X,Y}$ and $P_X \times P_Y$, respectively. Probability density functions use lowercase notation, e.g. $p(\mathbf{x})$. Mutual information between random vectors $\mathbf{X}$ and random variable y is denoted $I(\mathbf{x}; y)$, and $D_{\mathrm{KL}}(P\|Q)$ denotes the Kullback–Leibler divergence between two probability measures P and Q. We use $\rightarrow$ to denote a Markov chain.

2 Related work

The literature on Bayesian Optimization (BO) has increasingly focused on overcoming the computational limitations of traditional Gaussian Processes for high dimensional and large datasets and application and the sampling complexity of Bayesian neural networks. We categorize recent advancements into three distinct research frontiers.

Scalable Bayesian optimization Gaussian Processes (GPs) remain the canonical choice for BO due to their well-calibrated uncertainty (Srinivas et al., 2010). However, their $\mathcal{O}(T^3)$ complexity has led to a surge in scalable alternatives. Recent advancements include Vecchia approximations (Jimenez & Katzfuss, 2023),

which achieve linear scaling by leveraging local conditioning sets, and Trust-Region Bayesian Optimization (TuRBO) (Eriksson et al., 2019), which decomposes the search space into local regions to handle high-dimensional tasks. More recently, Focalized Sparse GPs (Wei et al., 2024) have been proposed to focus inducing points on promising regions of the input space. While effective, these methods often rely on stationary kernel assumptions that struggle to capture the complex, non-linear landscape of neural network landscapes.

Variational and Sampling-Based BNNs. Bayesian Neural Networks (BNNs) offer superior scalability and representation power compared to GPs. Recent efforts integrated BNNs using Hamiltonian Monte Carlo (HMC) (Springenberg et al., 2016; Li et al., 2024) for surrogate modeling of BO. To reduce the prohibitive cost of sampling, Variational Inference (VI) has emerged as a dominant paradigm. This includes the use of Variational Last Layers (VBLL) (Harrison et al., 2024), which treats only the final layer stochastically, and Prior-Fitted Networks (PFNs) (Müller et al., 2023), which use Transformers to amortize the cost of Bayesian inference. Despite these gains, many VI-based BO methods still rely on specific parametric families (e.g., Gaussian or Gamma) for the posterior, which can lead to model misspecification in complex search spaces.

Information-Theoretic acquisition function desgin Information-theoretic acquisition functions, such as *Predictive Entropy Search* (PES) (Hernandez-Lobato et al., 2015) and *Max-Value Entropy Search* (MES) (Wang & Jegelka, 2017), are prized for their principled approach to non-myopic exploration. Recent works like *Joint Entropy Search* (JES) (Hvarfner et al., 2022) attempt to enhance robustness by modeling the joint density over optimal inputs and values; however, due to the inherent complexity of this joint density, these methods typically revert to Gaussian Process (GP) assumptions for simplification.
Furthermore, in a recent work, Cheng et al. (2025) proposes a variational entropy search framework, and shows its connection expected improvement acquistion functin. However, the introduced acquisition functions lacks interpretability regarding the relative contributions of exploration and exploitation due to its complexity, while being constrained by specific variational families such as the Gamma distribution (Cheng et al., 2025). In another recent work, Ishikura & Karasuyama (2025) propose a variational lower bound maximization for entropy search. However, their formulation remains rooted in Gaussian Process framework and relies on combining over- and under-truncation approximation to maximize the bound. As it is mentioned in Ishikura & Karasuyama (2025) finding a full gradient-based approach remains a open future work.
In contrast, our work reformulates the search for optimal queries as a *max-min* optimization problem, which we solve through a novel *actor-critic* inspired framework. Unlike previous work, our approach makes no Gaussian Process assumptions on f, allowing for greater expressiveness. Essentially our framework, jointly addresses the problem of uncertainty modeling and data-driven acquisition function variational mutual information estimation. We utilize two neural networks trained in an alternating, gradient-based fashion: an the action-net to identify optimal inputs, and a *neural information gain estimator* (the helper) which provides the feedback to compute the acquisition function in the form of loss of the action-net. By optimizing the Donsker-Varadhan variational bound through direct gradient flow, our framework eliminates the need for both the rigid kernel dependencies of GPs and the expensive posterior sampling of traditional BNNs.

3 Problem Setting and Background

3.1 Problem Setting

We consider the task of maximizing an unknown, scalar-valued objective function $f : \mathcal{D} \to \mathbb{R}$ defined over a compact domain $\mathcal{D} \subseteq \mathbb{R}^d$. The function is a black box: its analytical form and gradients are unavailable, and each evaluation is expensive or noisy. Formally, the optimization problem is expressed as

$$\mathbf{x}^* = \arg\max_{\mathbf{x} \in \mathcal{D}} f(\mathbf{x}), \tag{1}$$

where $\mathbf{x}^*$ denotes the global maximizer of f.
At iteration t, the optimizer selects the input $\mathbf{x}_t$ and receives a noisy observation

$$y_t = f(\mathbf{x}_t) + \varepsilon_t, \tag{2}$$

where ε_t denotes independent, zero-mean observation noise. The history up to step t is denoted by $\mathbf{X}_t = [\mathbf{x}_1, \ldots, \mathbf{x}_t]$. Given a fixed evaluation budget, the goal here is to design a sequential policy that efficiently identifies $\mathbf{x}^*$ using T steps. At each of the T main iterations, f is evaluated at a batch of B points all

drawn from the same query distribution, $\mathbf{x}_t$ at step t; these B samples serve as Monte Carlo estimates of expectations (as discussed in section 4) at that step. Hence, we denote by $N = T \cdot B$ the total number of real function evaluations in T steps, the history $\mathbf{X}_t = [\mathbf{x}_1, ..., \mathbf{x}_t]$ records one query distribution per iteration, and each $\mathbf{x}_t$ representing the query at step t. We discuss further about this query distribution in the Section 4 of the paper.[1]. Querying the function in batches is practical in many real world tasks (Jain et al., 2022), and has been pursued in many recent works in high-dimensional Bayesian Optimization (Yun et al., 2025; Li et al., 2024).

3.2 Bayesian Optimization.

Bayesian Optimization (BO) traditionally addresses the problem by maintaining a probabilistic surrogate model over f and using an *acquisition function* $\alpha(\mathbf{x} \mid \mathbf{X}_t, \mathbf{Y}_t)$ to determine the next input point in T steps. Specifically, at each step, next input is chosen as :

$$\mathbf{x}_t = \arg\max_{\mathbf{x}} \alpha(\mathbf{x} \mid \mathbf{X}_{t-1}, \mathbf{Y}_{t-1}), \qquad t = 1, ..., T \tag{3}$$

At each step, BO updates the probabilistic model and chooses the next input based on previous observations up to step T. Various acquisition functions have been proposed in the literature, including Expected improvement (Jones et al., 1998), UCB (Jia et al., 2021), Entropy search (Hennig et al., 2022), and (Cheng et al., 2025), among others.

3.2.1 Exploration-Exploitation trade-off

Since our goal is to maximize $f(\mathbf{x})$ with as few queries as possible, BO seeks to simultaneously approximate f globally while identifying its maximum. This corresponds to an exploration-exploitation strategy, where the posterior mean of the surrogate function drives the exploitation, and the posterior variance or uncertainty of the surrogate guides the exploration. As discussed, this is done by optimizing an acquisition function, which guides the sampling of the best next query. Consider a set of sampled input points $\mathbf{X}_T = [\mathbf{x}_1, ..., \mathbf{x}_T]$, and $\mathbf{Y}_T = [y_1, ..., y_T]$ their corresponding noisy observations, where $y_t = f(\mathbf{x}_t) + \epsilon_t$, and $\epsilon_t \sim \mathcal{N}(0, \sigma_t^2)$, and $\mathbf{f}_{\mathbf{X}_T} = [f(\mathbf{x_1}), ..., f(\mathbf{x}_T)]$ are unknown function values. We aim to choose a set of points that provides the maximal information to approximate f, and approximate it with the fewest possible queries. To do that, the concept of information gain (Srinivas et al., 2010) has been used in the literature, which quantifies how much information is gained about f through observations $\mathbf{Y}_T$ as follows:

$$I(\mathbf{Y}_T; f) = H(\mathbf{Y}_T) - H(\mathbf{Y}_T \mid f), \tag{4}$$

$H(\cdot)$ and $H(\cdot \mid \cdot)$ denote the entropy and conditional entropy. Considering the GP assumption, it can be seen that we have:

$$F(\mathbf{X}_T) = I(\mathbf{Y}_T; \mathbf{f}_{\mathbf{X}_T}) = \frac{1}{2} \log(\mid \mathbf{I} + \sigma^{-2} \mathbf{K}_{\mathbf{X}_T} \mid), \tag{5}$$

where $\mathbf{K}_{\mathbf{X}_T} = [k(\mathbf{x}, \mathbf{x}')]_{\mathbf{x}, \mathbf{x}' \in \mathbf{X}_T}$, where $k(\mathbf{x}, \mathbf{x}')$ represent covariance function between two input points $\mathbf{x}$, and $\mathbf{x}'$.[2] Here the goal is to select points that maximize information gain. Although finding these points among different $\mathbf{X} \subset \mathcal{D}$, $\mid \mathbf{X} \mid \leq T$ is NP hard, it can be done by sequentially maximizing F as follows (Husain et al., 2023; Srinivas et al., 2010):
Imagine that we have observed up to $t-1$ input points (set $\mathbf{X}_{t-1}$). The optimal selection strategy for choosing the next input is such that

$$\mathbf{x}_t = \arg\max_{\mathbf{x} \in \mathcal{D}} F(\mathbf{X}_{t-1} \cup \{\mathbf{x}\}), \tag{6}$$

In case of Gaussian assumption on f this reduces to (Srinivas et al., 2010)

$$\mathbf{x}_t = \arg\max_{\mathbf{x} \in \mathcal{D}} \sigma_{t-1}(\mathbf{x}), \tag{7}$$

where σ_{t-1} is the posterior standard deviation at point $\mathbf{x}$ after observing previous points. In this case posterior variance has a closed-form (See Appendix B). In next section we discuss how we will solve for exploration

[1] As discussed in Sec.4, we also perform W warm-up iterations which query a batch of B points.
[2] See Appendix B.

term to a general case other than Gaussian. Although using the strategy of using only equation 7 is a good way to explore the function globally and approximate it, it is not particularly well suited to find the maximum of the function f. In this case, the objective is to (Srinivas et al., 2010) quickly identify locations $\mathbf{x}$ where $f(\mathbf{x})$ attains high values, allowing the algorithm to focus sampling efforts there and effectively exploit the current knowledge. The idea is to select points that maximize posterior mean at sampled point. In case of Gaussian assumption (Srinivas et al., 2010) we have:

$$\mathbf{x}_t = \arg\max_{\mathbf{x}\in\mathcal{D}} \mu_{t-1}(\mathbf{x}), \tag{8}$$

to maximize the posterior GP mean, $\mu_t(\mathbf{x})$, of the function f so far, which has closed-form expression (Appendix B). To take into account both exploring and exploiting strategies in equation 7, and equation 8 GP-UCB rule has been applied as (Srinivas et al., 2010; Husain et al., 2023; Kirschner et al., 2020; Tay et al., 2022):

$$\mathbf{x}_t = \arg\max_{\mathbf{x}\in\mathcal{D}} \sqrt{\beta}\,\sigma_{t-1}(\mathbf{x}) + \mu_{t-1}(\mathbf{x}), \tag{9}$$

where β is a constant. This sampling rule sequentially selects points $\mathbf{x}$ such that the selected input points maximize an upper quantile of the marginal of the posterior on f up to time $t-1$. In other words, with high probability: $f(\mathbf{x}) \leq \sqrt{\beta}\sigma_{t-1}(\mathbf{x}) + \mu_{t-1}(\mathbf{x})$ which accounts for choosing the name of GP-UCB bound.
Despite their success, GP-based BO methods scale poorly with the number of observations and rely on restrictive kernel assumptions, while BNN-based approaches often incur significant computational overhead due to posterior sampling and inner-loop acquisition optimization. These limitations motivate the need for BO frameworks that are both scalable and fully differentiable. In the next section, we introduce VBO-MI, a gradient-based Bayesian optimization framework that addresses these challenges via variational mutual information estimation.

4 Proposed Bayesian Optimization Framework with Variational MI Estimation

To develop our framework, Our first observation is that the policy described in equation 9 implicitly depends on the MI between the unknown function f and the observed rewards $\mathbf{y}_t$, conditioned on the selected actions $\mathbf{x}_t$, as established in equation 7 and equation 9. In the following sections, we leverage the results of Belghazi et al. (2018) to derive a variational bound on the exploration term in equation 9. We then introduce a data-driven approach for estimating the exploitation term. Building on these two components, we design a loss function that enables the joint training of two neural networks aimed at solving our Bayesian Optimization problem: one network responsible for identifying promising actions, and a second auxiliary (helper) network that estimates information gain and provides guidance for training both models that are trained in alternative fashion.

4.1 Variational Mutual Information

To incorporate Mutual Information (MI) into our optimization strategy without relying on restrictive assumptions (e.g., Gaussianity), we employ a variational lower bound based on the Donsker–Varadhan (DV) representation (Belghazi et al., 2018; Donsker & Varadhan, 1975) of the Kullback–Leibler (KL) divergence. Let P and Q be two probability measures over the same measurable space. The DV representation expresses the KL divergence as

$$D_{\mathrm{KL}}(P \,\|\, Q) = \sup_{D} \mathbb{E}_P[D] - \log \mathbb{E}_Q[e^D], \tag{10}$$

where the supremum is taken over all measurable functions D such that the expectations are finite (Belghazi et al., 2018; Donsker & Varadhan, 1975). Now, consider random vector $\mathbf{x}$ and random variable y with joint distribution $P_{\mathbf{x},y}$, and let $Q = P_{\mathbf{x}} \times P_y$ denote the product of marginals. Then, the MI between $\mathbf{x}$ and y can be written as

$$I(\mathbf{x}; y) = D_{\mathrm{KL}}\big(P_{\mathbf{x},y} \,\|\, P_{\mathbf{x}\times y}\big) = \tag{11}$$

$$\sup_{D} \mathbb{E}_{P_{\mathbf{x},y}}[D(\mathbf{x}, y)] - \log \mathbb{E}_{P_{\mathbf{x}}\times P_y}[e^{D(\mathbf{x},y)}]. \tag{12}$$

In practice, we parameterize $D(\mathbf{x}, y)$ using a Neural Network $D_\theta(\mathbf{x}, y)$ with parameters θ, and estimate the expectations via sampling. This yields the following variational lower bound:

$$I(\mathbf{x}; y) \geq \max_{\theta \in \Theta} \mathbb{E}_{P_{\mathbf{x},y}}[D_\theta(\mathbf{x}, y)] - \log \mathbb{E}_{P_{\mathbf{x}} \times P_y}[e^{D_\theta(\mathbf{x},y)}]. \tag{13}$$

The goal here would be to maximize this variational lower (by finding the optimal weights of a neural network through gradient descent) to approximate the mutual information.
Given samples of distributions $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{n=1}^N$, the Neural Network D_θ is trained by minimizing the following loss function:

$$L_{D_\theta} = -\frac{1}{B} \sum_{i=1}^{B} D_\theta(\mathbf{x}^{(i)}, y^{(i)}) + \log \left(\frac{1}{B^2} \sum_{i=1}^{B} \sum_{j=1}^{B} e^{D_\theta(\mathbf{x}^{(i)}, y^{(j)})} \right), \tag{14}$$

where B denotes the minibatch size. This formulation provides a scalable, sample-based estimator of MI, and integrate into our Bayesian Optimization framework to guide exploration.

4.2 Exploration and Mutual Information Gain Estimation

Using results of previous section, our goal here is to find a variational bound for exploring the function f. Based on the defined setting, at time step t we have access to $\mathbf{X}_t = [\mathbf{x}_1, ..., \mathbf{x}_t]$, of current chosen actions. We also, assume that these actions are chosen according to $P_{\mathbf{X}_t}$. Moreover, corresponding observations are $\mathbf{Y}_t = [y_1, .., y_t]$, where $y_t = f(\mathbf{x}_t) + \epsilon_t$, and ϵ_t is zero mean, independent, identically distributed (i.i.d.) noise. Our goal in this section is to find a lower bound on the information gain quantity which corresponds to the exploration term in equation 9, which is:

$$I_G = I(f(\mathbf{X}_t); \mathbf{Y}_t), \tag{15}$$

where we have[3] $\mathbf{Y}_t = f(\mathbf{X}_t) + \mathcal{E}_t$, $\mathcal{E}_t = [\epsilon_i]_{i=1}^t$, and $f(\mathbf{X}_t) = [f(\mathbf{x}_i)]_{i=1}^t$.[4] To find a simple bound on this quantity using the bound of Belghazi et al. (2018), and equation 12, we first note that we have the following Markov chain Cover & Thomas (2012):

$$\mathbf{X}_t \longleftrightarrow f(\mathbf{X}_t) \longleftrightarrow \mathbf{Y}_t \longleftrightarrow y_t. \tag{16}$$

Based on the data-processing inequality (Cover & Thomas, 2012) we have

$$I(\mathbf{X}_t; y_t) \leq I_G = I(f(\mathbf{X}_t), \mathbf{Y}_t). \tag{17}$$

By expanding the mutual information via chain rule we have:

$$I(\mathbf{X}_t; y_t) = I(\mathbf{x}_t; y_t) + I(\mathbf{x}_{t-1}; y_t \mid \mathbf{x}_t) + I(\mathbf{x}_{t-2}; y_t \mid \mathbf{x}_t, \mathbf{x}_{t-1}) + \cdots + I(\mathbf{x}_1; y_t \mid \mathbf{x_t}, \ldots \mathbf{x}_2), \tag{18}$$

except for the first term, all the conditional terms are zero due to presence of condition on $\mathbf{x}_t$, and conditional mutual information[5] definition (Cover & Thomas, 2012). On the other hand, by using the variational representation in equation 10:

$$I_\theta(\mathbf{x}_t; y_t) = \sup_{\theta \in \Theta} \mathbb{E}_{P_{\mathbf{x}_t, y_t}}[D_\theta(\mathbf{x}_t, y_t)] - \log(\mathbb{E}_{P_{\mathbf{x}_t} \times P_{y_t}}[e^{D_\theta(\mathbf{x}_t, y_t)}]). \tag{19}$$

By noting that $I_\theta(\mathbf{x}_t, y_t) \leq I(\mathbf{x}_t, y_t)$, and combining equation 17 and equation 19 we have the following bound on the exploration term, or IG bound in equation 9:

$$\mathbb{E}_{P_{\mathbf{x}_t, y_t}}[D_\theta(\mathbf{x}_t, y_t)] - \log(\mathbb{E}_{P_{\mathbf{x}_t} \times P_{y_t}}[e^{D_\theta(\mathbf{x}_t, y_t)}]) \leq I_G. \tag{20}$$

[3]Depending on its argument, $f(\mathbf{x}_t)$ represents a scalar value, or a vector, $f(\mathbf{X}_t)$.

[4]Note that here we don't have a Gaussian assumption (for f or noise), and therefore we cannot use the closed form for Gaussian distribution entropy)

[5]Given $\mathbf{x}_t$, the observation $y_t = f(\mathbf{x}_t) + \varepsilon_t$ is determined directly through the fixed objective f and on the i.i.d. noise ε_t. Assuming sequential selection of $\mathbf{x}_t$, neither of the previous $\mathbf{x}_{t-k}$ $k \geq 1$ affect observation if $\mathbf{x}_t$ is known. Therefore $\mathbf{x}_{t-k} \perp y_t \mid \mathbf{x}_t$ for all $k \geq 1$.

We choose y_t, the current observation, rather than the full history $\mathbf{Y}_t$: by the Markov chain in equation 16, y_t captures the *marginal* information gained at step t given the already-observed history, avoiding double-counting of information already incorporated in previous iterations. This choice allows the exploration signal to remain focused on the incremental uncertainty reduction provided by the newest query, rather than considering redundant historical observations which would also increase the complexity of evaluating the mutual information bound.

4.3 A Data-Driven Exploitation term

To take into account the exploitation capability of our algorithm, we also estimate exploitation term using a data-driven approach. The goal is use these results in our acquisition function (loss of a neural network) to choose the best actions to approximate and maximize f. In other word, our loss function should simultaneously consider the balance of exploration and exploitation in searching the input space. In the previous subsection, we have seen how the information gain quantity can be leveraged to consider the exploration of the input space. To account for the exploitation capability of our algorithm, we need to consider finding the maximum of f by utilizing our current knowledge. A straightforward approach to find the maximum is to consider the expectation of the function f conditioned on previous points (posterior of f), e.g:

$$\mathbb{E}[f(\mathbf{x}_t) \mid y_{t-1}, .., y_1, \mathbf{x}_{t-1}, .., \mathbf{x}_1]. \tag{21}$$

Furthermore, we note that by assuming that f and noise are independent, and noise has a zero mean, we have:

$$\begin{aligned} &\mathbb{E}[f(\mathbf{x}_t) \mid y_{t-1}, .., y_1, \mathbf{x}_{t-1}, .., \mathbf{x}_1] = \\ &\mathbb{E}[f(\mathbf{x_t}) + \epsilon_t \mid y_{t-1}, .., y_1, \mathbf{x}_{t-1}, .., \mathbf{x}_1] = && (22) \\ &\mathbb{E}[y_t \mid y_{t-1}, .., y_1, \mathbf{x}_{t-1}, .., \mathbf{x}_1]. && (23) \end{aligned}$$

In practice, computing $\mathbb{E}[y_t \mid y_{t-1}, \ldots, y_1, \mathbf{x}_{t-1}, \ldots, \mathbf{x}_1]$ analytically requires a parametric surrogate such as a Gaussian Process, which reintroduces the assumptions we seek to avoid. Instead, we approximate this conditional expectation directly from data using a neural network surrogate $\hat{f}_\psi : \mathcal{D} \to \mathbb{R}$, trained by minimizing the mean squared error on all past observations in replay buffer $\mathcal{H} = \{(\mathbf{x}_{t'}^{(i)}, y_{t'}^{(i)})\}$, $t' \leq t$:

$$\mathcal{L}_{\text{MSE}}(\psi) = \frac{1}{B} \sum_{i=1}^{B} \big(\hat{f}_\psi(\mathbf{x}^{(i)}) - y^{(i)}\big)^2, \quad \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{B} \overset{\text{i.i.d.}}{\sim} \text{Uniform}(\mathcal{H}). \tag{24}$$

The trained surrogate $\hat{f}_\psi(\mathbf{x})$ then serves as a point estimate of $\mathbb{E}[y_t \mid \mathcal{H}]$, which we call it $\hat{y}_t$ at any proposed action $\mathbf{x}$, and its gradient $\nabla_{\mathbf{x}} \hat{f}_\psi$ provides a differentiable exploitation signal for updating E_ϕ without any parametric assumption on f and without querying the black-box function.
In the next part, we define a loss function and state our main framework, and algorithm for the proposed variational Bayesian Optimization.

4.4 Proposed variational BO with MI and Algorithm

In this section, we develop our proposed variational acquisition function and algorithm to find the best possible inputs of f at each step. More specifically, we aim to solve the Bayesian Optimization problem described in equation 1 in T steps with batch queries of function f. To do so, we find the best possible actions for f using a neural network (called action-net, E_ϕ), based on maximizing an acquisition function (minimizing as a loss function as described later), which we call g, through the following optimization problem:

$$\phi^* = \arg\max_{\phi} \min_{\theta} \; g(\phi, \theta), \qquad \mathbf{x}_t = E_{\phi^*}(\mathbf{s}_t), \tag{25}$$

where ϕ, θ represent parameters of two neural networks, E_ϕ and D_θ, which we call the action-net and Helper network respectively. These networks are trained in an alternating manner (similar to an actor-critic architecture) using gradient descent. Here $g(\phi, \theta)$ denotes the variational acquisition objective combining exploitation and exploration, defined explicitly in *equation* 31 after the two components are developed in Sections 4.4.1 and 4.4.2. To the best of our knowledge, this is the first work in this area that applies variational formulations to design a fully gradient-based Bayesian Optimization framework using deep learning, in contrast to earlier works that either rely on Hamiltonian Monte Carlo sampling with BNNs (Ishikura & Karasuyama, 2025) or on specific variational distribution families (Cheng et al., 2025).

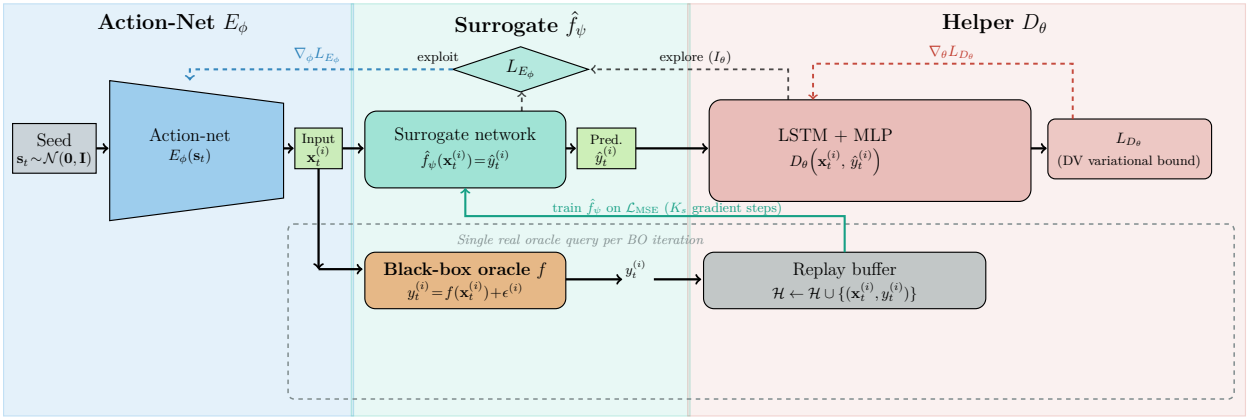


Figure 1: VBO-MI framework. An *action-net* E_ϕ maps random seeds $\mathbf{s}_t$ to candidate inputs $\mathbf{x}_t^{(i)}$. A *surrogate* $\hat{f}_\psi$ predicts rewards $\hat{y}_t^{(i)}$, enabling oracle-free updates of the helper D_θ (K_a steps) and action-net (K_b steps) via the acquisition loss L_{E_ϕ}, which balances exploitation (surrogate mean) and exploration (variational MI estimate I_θ). One real oracle query per iteration yields $y_t^{(i)}$, which replenishes the replay buffer $\mathcal{H}$ used to retrain $\hat{f}_\psi$ via MSE (K_s steps). Dashed arrows indicate gradient flows.

4.4.1 Phase I: Surrogate Training and Helper Network Update

At each BO iteration t, the algorithm maintains a replay buffer $\mathcal{H} = \{(\mathbf{x}_{t'}^{(i)}, y_{t'}^{(i)})\}$ of all past input and output observations collected from f. Before updating the helper network D_θ, we first use $\mathcal{H}$ to train a lightweight surrogate network $\hat{f}_\psi : \mathcal{D} \to \mathbb{R}$, a feed-forward neural network that approximates the conditional expectation $\mathbb{E}[y_t \mid \mathbf{x}, \mathcal{H}]$. The optimal point estimator is:

$$\psi^* = \arg\min_{\psi} \; \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{H}}\big[\big(\hat{f}_\psi(\mathbf{x}) - y\big)^2\big], \tag{26}$$

since the minimizer of the expected squared error is the conditional mean $\hat{f}_{\psi^*}(\mathbf{x}) = \mathbb{E}[y \mid \mathbf{x}, \mathcal{H}]$. Because the joint distribution under $\mathcal{H}$ is unknown, we approximate the objective in equation 26 by its sample average over a mini-batch $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^B \overset{\text{i.i.d.}}{\sim} \text{Uniform}(\mathcal{H})$:

$$\mathcal{L}_{\text{MSE}}(\psi) = \frac{1}{B}\sum_{i=1}^{B}\big(\hat{f}_\psi(\mathbf{x}^{(i)}) - y^{(i)}\big)^2, \quad \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^B \overset{\text{i.i.d.}}{\sim} \text{Uniform}(\mathcal{H}). \tag{27}$$

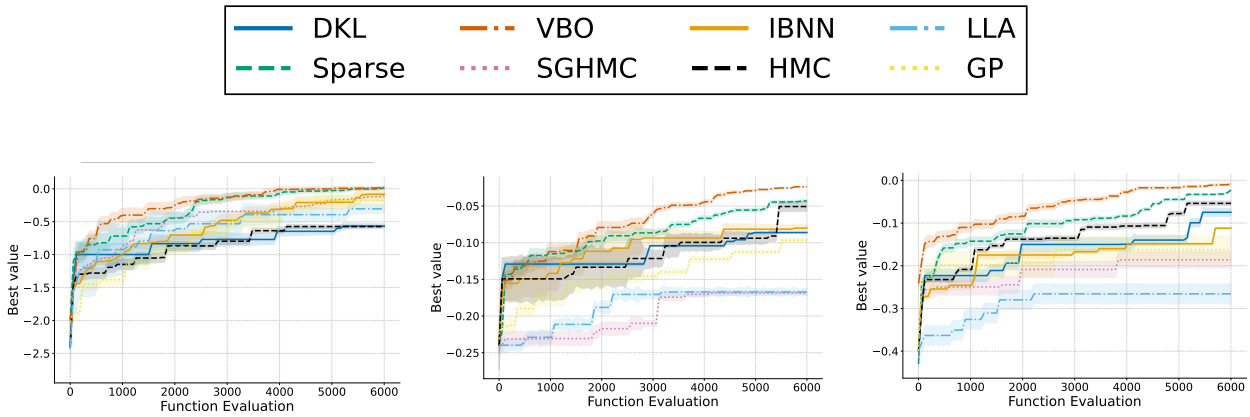
Because $\hat{f}_\psi$ is trained exclusively on observed input–output pairs and its gradient $\nabla_{\mathbf{x}}\hat{f}_\psi$ is computed through the surrogate network rather than through f, the black-box condition on f is never violated.
Once the surrogate is updated, its predictions serve as a query-free substitute for real function evaluations inside the helper update. Specifically, the action-net E_ϕ proposes a batch of actions $\{\mathbf{x}_t^{(i)}\}_{i=1}^B$, and the surrogate produces corresponding predictions:

$$\hat{y}_t^{(i)} = \hat{f}_\psi(\mathbf{x}_t^{(i)}), \quad i = 1, \ldots, B. \tag{28}$$

These surrogate predictions $\hat{y}_t$ replace real observations in the helper's loss, allowing D_θ to be updated without additional queries to f. The helper network D_θ is then trained to estimate the information gain between the proposed actions and these predicted observations via the Donsker-Varadhan variational bound equation 12:

$$L_{D_\theta}(\mathbf{x}_t, \hat{y}_t) = \; -I_\theta(\mathbf{x}_t, \hat{y}_t) = \; -\mathbb{E}_{P_{\mathbf{x}_t, \hat{y}_t}}[D_\theta(\mathbf{x}_t, \hat{y}_t)] + \log \mathbb{E}_{P_{\mathbf{x}_t} \times P_{\hat{y}_t}}[e^{D_\theta(\mathbf{x}_t, \hat{y}_t)}], \tag{29}$$

where the weights ϕ of the action-net are frozen during this update, and $\hat{f}_\psi$ is kept fixed. This phase runs for K_a gradient steps and produces an updated estimate $I_\theta(\mathbf{x}_t, \hat{y}_t)$ of the information gain, which is then passed to Phase II.



(a) Ackley ($d = 10$) (b) Levy ($d = 20$) (c) Griewank ($d = 20$)

Figure 2: Performance comparison on synthetic benchmark tasks (a–c). Global optimum is 0 for all functions (dashed line). All curves show mean ± one standard deviation over 5 trials.

4.4.2 Phase II: Action-Net Update via Proposed Loss Function

With the helper network D_θ updated and frozen, the action-net E_ϕ is trained to propose inputs that jointly maximise exploitation of the surrogate's predicted reward and exploration measured by the MI estimate from Phase I. The action-net maps a random seed $\mathbf{s}_t \sim \mathcal{N}(0, I)$ to a candidate action:

$$E_\phi(\mathbf{s}_t) = \mathbf{x}_t, \tag{30}$$

and its loss combines the two objectives. The *exploitation* term is the surrogate's mean predicted reward at the proposed actions, providing a differentiable signal that pushes E_ϕ toward high-reward regions without querying f. The *exploration* term is the negative variational MI estimate $-I_\theta(\mathbf{x}_t, \hat{y}_t)$ from Phase I, which encourages E_ϕ to propose inputs that are informative relative to the surrogate's current predictions. Together, by combining the exploitation term in equation 23 with the exploration bound in equation 20, the loss (or acquisition function) takes the form:

$$L_{E_\phi}(\mathbf{x}_t, \hat{y}_t) = -g(\phi, \theta) = -\hat{y}_t - \sqrt{\beta}\, \mathbb{E}_{P_{\mathbf{x}_t, \hat{y}_t}}[D_\theta(\mathbf{x}_t, \hat{y}_t)] + \sqrt{\beta} \log \mathbb{E}_{P_{\mathbf{x}_t} \times P_{\hat{y}_t}}[e^{D_\theta(\mathbf{x}_t, \hat{y}_t)}] \tag{31}$$

where β controls the trade-off between exploration and exploitation, and L_{E_ϕ} is a *loss to be minimized* by E_ϕ, equivalent to maximizing $\hat{f}_\psi(\mathbf{x}_t) + \sqrt{\beta}\, I_\theta(\mathbf{x}_t, \hat{y}_t)$.
Here the weights of θ and ψ are frozen throughout. Since both losses contain only expectation terms, they are implemented as sample averages over the batch. Specifically we approximate it as:

$$L_{E_\phi} = -\frac{1}{B}\sum_{i=1}^{B} \hat{y}_t^{(i)} - \sqrt{\beta}\left[\frac{1}{B}\sum_{i=1}^{B} D_\theta(\mathbf{x}_t^{(i)}, \hat{y}_t^{(i)}) - \log \frac{1}{B^2}\sum_{i=1}^{B}\sum_{j=1}^{B} e^{D_\theta(\mathbf{x}_t^{(i)}, \hat{y}_t^{(j)})}\right] \tag{32}$$

where $\hat{y}_t^{(i)} = \hat{f}_\psi(\mathbf{x}_t^{(i)})$. This update runs for K_b gradient steps with D_θ and $\hat{f}_\psi$ fixed.
Our framework has three neural network components. E_ϕ searches the input space $\mathcal{D}$ and selects the next action $\mathbf{x}_t$; the helper D_θ estimates the information gain and provides the exploration signal for E_ϕ; and the surrogate $\hat{f}_\psi$ approximates the conditional expectation of f from past observations and supplies the exploitation gradient to E_ϕ. All three networks are trained in an alternating fashion toward convergence (see Fig. 1). In order to give the algorithm smooth, and efficient performance we have included warm-up step where only helper network, and surrogate are updated. Our detailed algorithm is summarized in Algorithm 1.

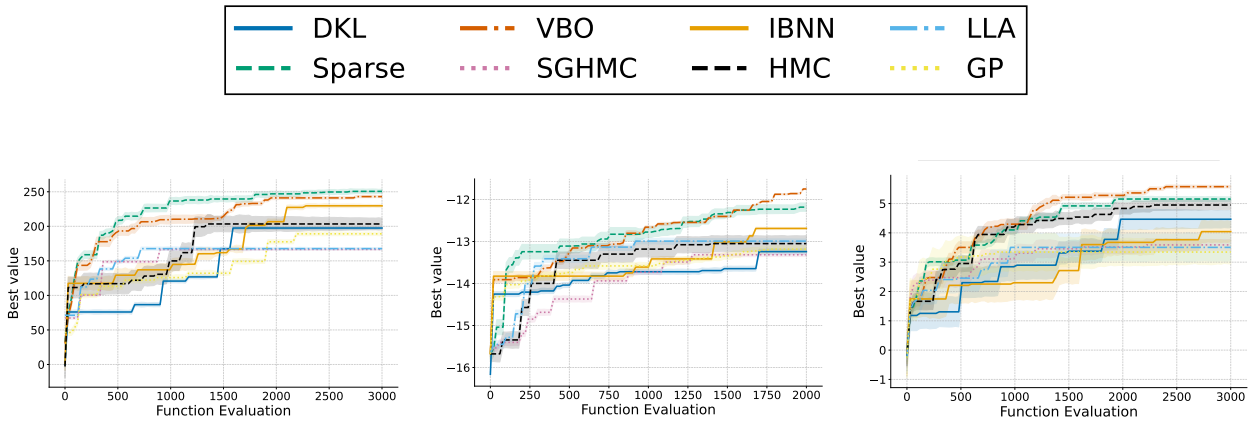
Remark 4.1. It is important to distinguish between the optimization of our neural networks and the optimization of f. VBO-MI uses gradients to update network parameters ϕ and θ with respect to the variational loss. However, these gradients are computed through the neural architecture; the objective function f itself remains a black box, and no gradients of f are required or used.

Algorithm 1 VBO-MI

Input: Initialized weights of networks E_ϕ (action-network), D_θ (MI network-Helper), and $\hat{f}_\psi$ (surrogate network); initial random seed $\mathbf{s}_0 \sim \mathcal{N}(0, I)$, number of warm-up steps W, number of BO iterations T, iteration counts K_a, K_b, and K_s, batch size B.
Output: Optimized weights of E_ϕ, D_θ, and $\hat{f}_\psi$; estimated optimum $\mathbf{x}^*$.

Warm-up Phase:
Initialize replay buffer $\mathcal{H} \leftarrow \emptyset$.
for $k = 1, \ldots, W$ **do**
  Using the seeds $\mathbf{s}_t^{(i)}$ compute (a batch of) actions $\mathbf{x}_t^{(i)} = E_\phi(\mathbf{s}_t^{(i)}),\ i = 1, \ldots, B$.
  Evaluate observations $y^{(i)} = f(\mathbf{x}^{(i)}) + \epsilon^{(i)}$.
  Store $\mathcal{H} \leftarrow \mathcal{H} \cup \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^B$.
  Compute $-$MI using $\{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^B$ and helper network.
  Update D_θ via gradient descent on L_{D_θ}, keeping E_ϕ fixed.
end for
Pre-train surrogate: for $k = 1, \ldots, 200$, sample a mini-batch $\{(\mathbf{x}^{(i)}, y^{(i)})\} \subset \mathcal{H}$ and update $\hat{f}_\psi$ by gradient descent on $\mathcal{L}_{\text{MSE}} = \frac{1}{B}\sum_i (\hat{f}_\psi(\mathbf{x}^{(i)}) - y^{(i)})^2$.
Main VBO-MI Phase:
for $t = 1, \ldots, T$ **do**
  Compute (a batch of) actions $\mathbf{x}_t^{(i)} = E_\phi(\mathbf{s}_t^{(i)}),\ i = 1, \ldots, B$.
  Update D_θ **(Helper Network):**
  for $k = 1, \ldots, K_a$ **do**
    Compute surrogate predictions $\hat{y}_t^{(i)} = \hat{f}_\psi(\mathbf{x}_t^{(i)}),\ i = 1, \ldots, B$.
    Update D_θ by gradient descent on L_{D_θ} with E_ϕ fixed, using $\{\mathbf{x}_t^{(i)}, \hat{y}_t^{(i)}\}_{i=1}^B$.
  end for
  for $k = 1, \ldots, K_s$ **do**
    Sample a mini-batch $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^B \subset \mathcal{H}$.
    Update $\hat{f}_\psi$ by gradient descent on $\mathcal{L}_{\text{MSE}} = \frac{1}{B}\sum_i (\hat{f}_\psi(\mathbf{x}^{(i)}) - y^{(i)})^2$.
  end for
  Update E_ϕ **(Action-Net):**
  for $k = 1, \ldots, K_b$ **do**
    Compute (a batch of) actions $\mathbf{x}_t^{(i)} = E_\phi(\mathbf{s}_t^{(i)}),\ i = 1, \ldots, B$.
    Compute surrogate predictions $\hat{y}^{(i)} = \hat{f}_\psi(\mathbf{x}^{(i)})$.
    Compute MI estimate $I_\theta(\mathbf{x}^{(i)}, \hat{y}^{(i)})$ via helper D_θ (weights frozen).
    Update E_ϕ by SGD on L_{E_ϕ} in equation 32 with D_θ and $\hat{f}_\psi$ fixed.
  end for
  Compute actions $E_\phi(\mathbf{s}_t) = \mathbf{x}_t^{(i)},\ i = 1, \ldots, B$.
  Evaluate $y_t^{(i)} = f(\mathbf{x}_t^{(i)}) + \epsilon^{(i)}$.
  Store $\mathcal{H} \leftarrow \mathcal{H} \cup \{(\mathbf{x}_t^{(i)}, y_t^{(i)})\}_{i=1}^B$.
  Save estimated optimum $\mathbf{x}_t^* = \arg\max_{(\mathbf{x},y)\in\mathcal{H}}\ y$.
end for
Return: final network weights $(\phi^*, \theta^*, \psi^*)$ and estimated optimum $\mathbf{x}_t^*$.

Now we resort to the loss functions L_{E_ϕ} and L_{D_θ}, and the outputs of E_ϕ and D_θ, to propose our algorithm for the BO defined in Section 3. In the proposed method the action-net E_ϕ, searches the input space $\mathcal{D}$ and selects the next query $\mathbf{x}_t$; the helper D_θ, which estimates information gain and provides the exploration signal for E_ϕ; and the surrogate $\hat{f}_\psi$, which approximates the conditional expectation of f from accumulated observations and supplies the exploitation gradient. The core optimization follows an alternating actor-critic scheme between E_ϕ and D_θ: at each iteration, D_θ is updated first to sharpen the MI estimate, then E_ϕ is updated using the frozen D_θ. The surrogate $\hat{f}_\psi$ is updated separately via MSE minimization on the replay buffer $\mathcal{H}$, serving as an oracle-free substitute for f throughout both updates. The complete procedure is summarized in Algorithm 1 and illustrated in Fig. 1.



(a) Lunar Lander ($d = 12$) (b) Pest Control ($d = 25$) (c) Rover Trajectory ($d = 60$)

Figure 3: Performance comparison on real-world benchmark tasks (a–c). All curves show mean ± one standard deviation over 5 trials.

Now we state our theorem regarding the consistency of the estimates in our VBO-MI framework for finding optimal point of function f, and it's input optimizer.

Theorem 4.2 (Consistency of VBO-MI Optimization)**.** *Let $f : \mathcal{D} \to \mathbb{R}$ be a continuous black-box function, and let $\widehat{g}(\phi, \theta)$ be the sample-average approximation of the variational loss $g(\phi, \theta)$ as defined in equation 25 and equation 32. Assume the following conditions hold:*

(i) ***(Finite DV expectations)*** *For all $\theta \in \Theta$, the expectations in the Donsker–Varadhan bound are finite under the surrogate observation $\hat{y}_t = \hat{f}_\psi(\mathbf{x}_t)$, i.e.,*

$$\mathbb{E}_{P_{\mathbf{x}_t, \hat{y}_t}}\left[D_\theta(\mathbf{x}_t, \hat{y}_t)\right] < \infty \quad \textit{and} \quad \log \mathbb{E}_{P_{\mathbf{x}_t} \times P_{\hat{y}_t}}\left[e^{D_\theta(\mathbf{x}_t, \hat{y}_t)}\right] < \infty. \tag{33}$$

(ii) ***(Universal approximation)*** *The networks E_ϕ, D_θ, and $\hat{f}_\psi$ belong to universal approximation classes feed-forward networks for E_ϕ and $\hat{f}_\psi$, and LSTMs for D_θ (Hornik et al., 1989; Aguiar et al., 2023) with sufficient capacity.*

(iii) ***(Surrogate consistency)*** *The replay buffer $\mathcal{H} = \{(\mathbf{x}_{t'}^{(i)}, y_{t'}^{(i)})\}$ constitutes a stationary ergodic sequence whose marginal distribution over inputs has full support on $\mathcal{D}$, and ψ^* achieves the global minimum of the sample-average MSE loss. The surrogate is trained by minimizing this loss over mini-batches drawn uniformly from $\mathcal{H}$.*

(iv) ***(Compact parameter spaces)*** *The parameter spaces Φ, Θ, and Ψ are compact.*

Then:

(a) The sample average approximation $\widehat{g}(\phi, \theta)$ converges uniformly to $g(\phi, \theta)$ $\mathbb{P}$-almost surely as the batch size $B \to \infty$.

(b) ***(Surrogate convergence)*** *Under condition (iii), the surrogate network $\hat{f}_{\psi^*}$ trained on $\mathcal{H}$ converges uniformly to f on $\mathcal{D}$, $\mathbb{P}$-almost surely, as $|\mathcal{H}| \to \infty$. Consequently, the exploitation term $\frac{1}{B}\sum_i \hat{f}_{\psi^*}(\mathbf{x}^{(i)})$ in L_{E_ϕ} converges to the true conditional expectation $\mathbb{E}[f(\mathbf{x}) \mid \mathcal{H}]$.*

(c) ***(Action-net convergence)*** *Given conclusion (b), the joint optimizer*

$$\phi^* = \arg\max_{\phi} \min_{\theta} \widehat{g}(\phi, \theta)$$

converges $\mathbb{P}$-almost surely to the optimal action-network parameters with respect to $\widehat{g}(\phi, \theta)$, and the corresponding output $\mathbf{x}^* = E_{\phi^*}(\mathbf{s})$ converges to the maximizer of the variational acquisition objective

$g(\phi, \theta)$. Under the surrogate consistency of conclusion (b), this yields an ε-optimal solution to the original problem:

$$f(\mathbf{x}^*) \geq \sup_{\mathbf{x} \in \mathcal{D}} f(\mathbf{x}) - \varepsilon, \quad \textit{for any } \varepsilon > 0.$$

Proof sketch. The proof proceeds in following steps.
(a) Uniform convergence of $\widehat{g}$: The exploitation term $\mathbb{E}[y_t \mid \mathcal{H}]$ is estimated by its sample average using the consistency result for neural network MSE estimators (Chen & White, 1999). For the two DV terms, we cannot apply the standard LLN because the selected actions $\mathbf{x}_t$ are not i.i.d. Instead, we apply the Generalised AEP (Algoet & Cover, 1988) to the joint expectation $\mathbb{E}_{P_{\mathbf{x},y}}[D_\theta(\mathbf{x}, y)]$ and Birkhoff's Ergodic Theorem (Breiman, 1957) to the log-marginal term $\log \mathbb{E}_{P_{\mathbf{x}} \times P_y}[e^{D_\theta(\mathbf{x},y)}]$, both under the ergodicity assumption on the input sequence.
(b) Surrogate convergence: Uniform convergence of $\hat{f}_{\psi^*}$ to f follows from the consistency result for neural network MSE estimators (Chen & White, 1999) applied to mini-batches drawn from the stationary ergodic replay buffer $\mathcal{H}$.
(c) Action-net convergence: Given (a) and (b), the optimal policy map $\pi^*(\mathbf{s}) = \arg\max_{\mathbf{x}} g(\phi, \theta)$ is continuous on compact $\mathcal{D}$; the Universal Approximation Theorem then guarantees that $E_{\hat{\phi}}$ approximates π^* uniformly, and uniform continuity of f on $\mathcal{D}$ gives the global optimum result.
In proof, it is also shown for the helper D_θ: the RNN UAT (Aguiar et al., 2023) establishes $|I(\mathbf{x}; y) - I_\theta(\mathbf{x}, y)| \leq \eta, \eta \geq 0$ via the triangular inequality and exponential bounds.
Full details are in Appendix A. □

5 Numerical Evaluation

In this section, we present numerical evaluations of our proposed method, Variational Bayesian Optimization with Mutual Information (VBO-MI), and compare its performance with several baselines (Li et al., 2024; Wei et al., 2024). These baselines include the following methods for Bayesian neural networks: Laplace Linear Approximation (LLA) (Immer et al., 2021; Li et al., 2024), Hamiltonian Monte Carlo (HMC) (Izmailov et al., 2021; Li et al., 2024), Stochastic Gradient Hamiltonian Monte Carlo (SGHMC) (Chen et al., 2014; Li et al., 2024), Deep Kernel Learning (DKL) (Wilson et al., 2016; Li et al., 2024), Infinite-width Bayesian Neural Networks (IBNN) (Lee et al., 2018; Li et al., 2024), and Gaussian Processes (GPs) (Snoek et al., 2012; Li et al., 2024). In addition, we include the Focal Sparse Gaussian Process method (Wei et al., 2024) as a recent baseline. We present experiments on two settings: first, three synthetic benchmark functions; then, three real-world tasks of increasing dimensionality.

Implementation. For VBO-MI, we use a batch size of $B = 30$ and $B = 60$ for Real world task and synthetic functions respectively, a learning rate of 2×10^{-3} with Adam optimizer for all three networks, $K_a = 5$ helper update steps, $K_b = 10$ action-net update steps, $K_s = 10$ surrogate update steps per BO iteration, and $W = 30$ warm-up iterations. The total number of evaluations considers both evaluations needed during the warm-up, and main phase of algorithm (see 1). Baselines use the hyperparameter configurations reported in Li et al. (2024). All results are averaged over 5 random seeds; shaded regions denote ± 1 standard deviation. Full architecture details and per-task configurations are provided in Appendix C.

5.1 Synthetic Functions

We evaluate our algorithm on three widely used benchmark functions: Ackley, Levy, and Griewank (Li et al., 2024; Eriksson et al., 2019). All three functions are highly non-convex, contain many local optima, and have their global minimum at zero—making them a diverse and challenging suite for evaluating BO methods. Because these functions are conventionally used as minimization problems, we optimize $-f$ instead of f; maximizing $-f$ is equivalent to minimizing f, and the global maximum of $-f$ is therefore 0 in all cases.
In Fig. 2a, we compare all methods on the 10-dimensional Ackley function. Ackley is particularly challenging due to its near-flat outer region and sharp global basin, which makes exploration critical. We observe that VBO-MI achieves competitive performance, consistently reaching higher final average rewards than the BNN baselines. The Focal Sparse GP baseline performs strongly in the early iterations but VBO-MI closes the gap and surpasses it as the number of iterations increases.
Fig. 2b presents results on the 20-dimensional Levy function. Levy is a scalable multimodal function whose difficulty grows with dimension, making the 20-dimensional setting a meaningful test of high-dimensional

exploration. In this setting, VBO-MI demonstrates faster initial convergence and achieves a higher final reward compared to all baselines, including GP and IBNN.
Finally, Fig. 2c evaluates all methods on the 20-dimensional Griewank function. Unlike Ackley and Levy, Griewank is non-separable due to its product term, meaning that dimensions interact—a property that makes surrogate-based methods with stationary kernels less effective. VBO-MI achieves the highest final reward in this setting, which we attribute to its non-parametric variational exploration signal, which does not assume any specific covariance structure over the input space.

5.2 Real-World Tasks

We further evaluate VBO-MI on three real-world benchmark tasks of increasing dimensionality, following the experimental protocol of prior work (Li et al., 2024; Deng et al., 2022). These tasks test whether the method scales to higher-dimensional, non-stationary objectives where standard GP surrogates are known to struggle (Li et al., 2024). For all methods, we report the best value so far across different evaluations and report results averaged over multiple trials with standard deviation. Further experimental details are provided in the Appendix.

Lunar Lander ($d = 12$). In Fig. 3a, we evaluate all methods on the OpenAI Gym LunarLander-v2 control task (Deng et al., 2022). The goal is to optimise the 12 parameters of a controller such that the average episode reward over 50 randomly generated environments is maximised. VBO-MI achieves the highest final reward among all methods, with a clear margin over the BNN baselines. The non-stationarity of the reward landscape as a function of controller parameters makes this task well-suited to our method, which does not rely on stationary kernel assumptions.

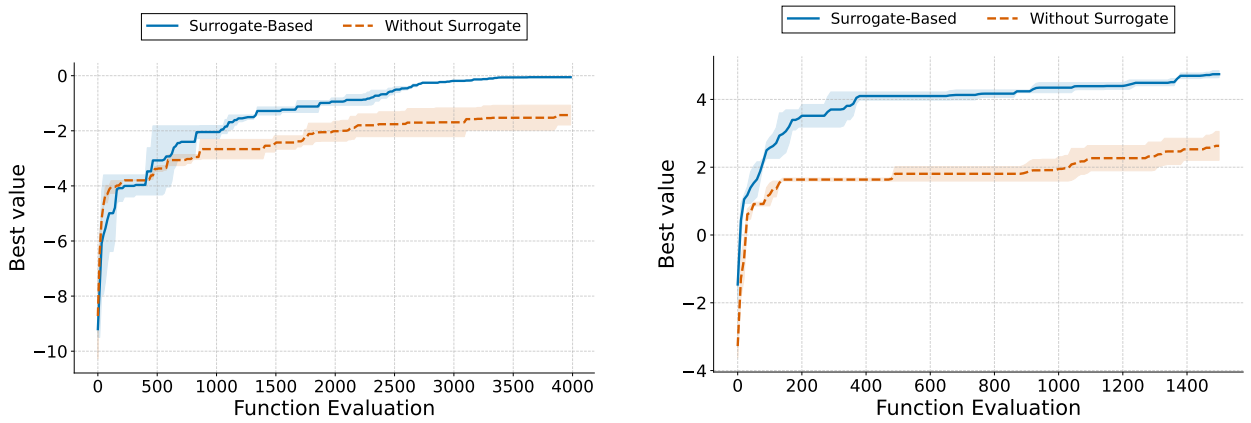
Pest Control ($d = 25$, categorical). Fig. 3b shows results on the Pest Control problem (Eriksson et al., 2019), a combinatorial optimization task with 25 categorical input variables each taking 5 values. We apply one-hot encoding following Hernandez-Lobato et al. (2015), and use their wrapper transformation $T(\cdot)$ to project the continuous output of E_ϕ back to a categorical input at each oracle query. This is the highest-cardinality benchmark in our evaluation ($5^{25} \approx 3 \times 10^{17}$ configurations), making exhaustive search infeasible. VBO-MI achieves superior performance compared to all baselines, and we observe that the advantage over the Focal Sparse GP grows as more iterations are completed, consistent with the observation that our method benefits from accumulated replay data.

Rover Trajectory ($d = 60$). In Fig. 3c, we evaluate on the Rover trajectory planning benchmark (Deng et al., 2022), which optimises the positions of 30 waypoints in a 2D environment (60 total dimensions). This is the highest-dimensional task in our evaluation and directly tests scalability. We observe that VBO-MI maintains competitive performance relative to the strongest baselines at this dimensionality. The Focal Sparse GP and IBNN also perform well at this scale, consistent with findings in the high-dimensional BO literature (Li et al., 2024), but VBO-MI achieves comparable or superior final reward values while requiring significantly fewer real function evaluations per iteration due to its surrogate-assisted inner loops.

5.3 Ablation

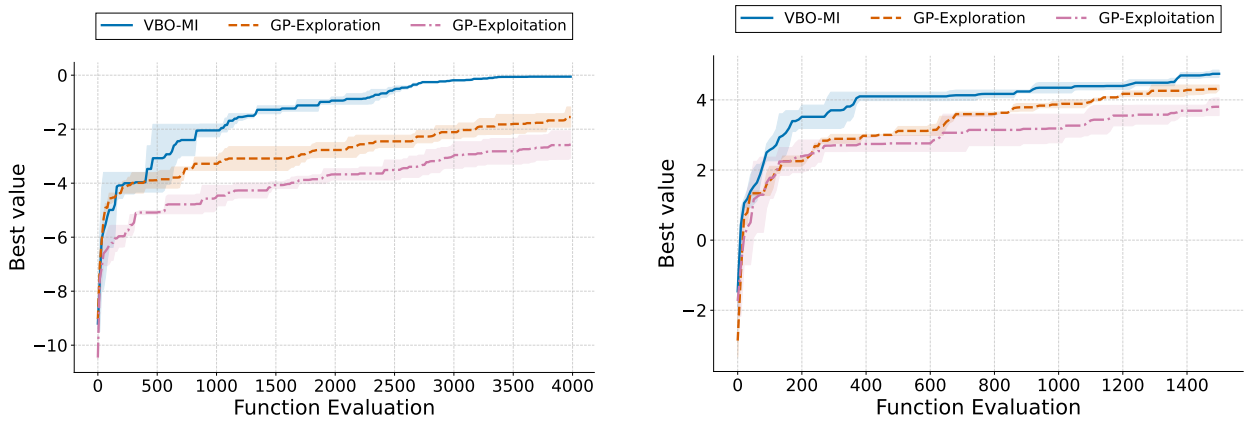
We present ablation studies to isolate the contribution of the surrogate network $\hat{f}_\psi$ and each term of the variational acquisition function L_{E_ϕ}. All ablations are evaluated on Ackley ($d = 10$) and Rover trajectory planning ($d = 60$), and all other hyperparameters are held fixed.

Surrogate network vs Direct queries To validate the role of the surrogate network $\hat{f}_\psi$ in VBO-MI, we compare the complete implementation of the surrogate-assisted implementation against an ablated baseline in which the surrogate is removed and the helper and action-net updates return to querying the true objective f directly inside the loops K_a and K_b. All other components network architectures, learning rates, batch size, and number of iterations are held fixed, so the surrogate is the sole variable.
Figs. 4a and 4b report best-so-far reward averaged over multiple random seeds (shaded regions: ± 1 standard deviation) on Ackley ($d = 10$) and Rover trajectory planning ($d = 60$), respectively.
On the lower-dimensional Ackley task (Fig. 4a), both variants converge toward the global optimum, and the surrogate-assisted method achieves a higher final reward. The improvement is already visible in the early iterations, suggesting that the surrogate provides a useful exploitation gradient from the outset once it has been pre-trained on the warm-up observations.



(a) Ackley ($d = 10$) (b) Rover trajectory ($d = 60$)

Figure 4: Ablation I: surrogate network contribution. Surrogate-assisted VBO-MI vs. the original variant that queries f directly inside the K_a and K_b loops. Mean $\pm$ 1 std over three trials.



(a) Ackley ($d = 10$) (b) Rover trajectory ($d = 60$)

Figure 5: Ablation II: acquisition function term contribution. Full VBO-MI vs. two hybrid variants: one replaces the exploration term with the GP posterior variance (σ^2_{GP}), and one replaces the exploitation term with the GP posterior mean (μ_{GP}). Mean $\pm$ 1 std over three trials.

The advantage of the surrogate becomes substantially more pronounced on the higher-dimensional Rover task (Fig. 4b). Here, the original variant without surrogate converges more slowly and plateaus at a noticeably lower reward value, whereas the surrogate-assisted method continues to improve throughout the budget. This behaviour is consistent with the role of the surrogate as a query-efficient approximation to $\mathbb{E}[y_t \mid \mathcal{H}]$ as in higher-dimensional spaces the action-net requires more gradient steps to identify promising regions, and the surrogate makes those steps possible without incurring additional real function evaluations.
Our ablation provides us with the conclusion that the surrogate provides a meaningful benefit as dimensionality increases, justifying its inclusion as a core component of VBO-MI rather than an optional efficiency add-on. From an oracle-efficiency perspective, the surrogate-assisted variant reduces real f queries per iteration from $(K_a + K_b + 1) \times B$ to B which is a $16\times$ reduction with our default hyperparameters while achieving superior optimization performance across both tasks when compared on the same budget.

Acquisition function terms. We assess the individual contributions of the exploration and exploitation terms in L_{E_ϕ} equation 32 by constructing two hybrid variants. In the first, the exploration term $\sqrt{\beta}\, I_\theta(\mathbf{x}, \hat{y})$ is replaced by the posterior variance of a GP surrogate, $\sqrt{\beta}\, \sigma^2_{\text{GP}}(\mathbf{x})$, while the exploitation term $\hat{f}_\psi(\mathbf{x})$ is kept intact. In the second, the exploitation term is replaced by the GP posterior mean $\mu_{\text{GP}}(\mathbf{x})$, while the exploration term I_θ is kept intact. Results are shown in Fig. 5.

On both tasks (Figs. 5a–5b), the full VBO-MI outperforms both hybrid variants, confirming that both terms contribute to the performance of the algorithm.
Replacing the exploitation term with the GP posterior mean produces a larger degradation than replacing the exploration term with the GP variance, particularly on the higher-dimensional Rover task.
This indicates that the exploitation gradient provided by $\hat{f}_\psi$ plays a more critical role in steering the action-net toward high-reward regions than the exploration signal alone. Nonetheless, removing the variational MI exploration term also degrades performance, demonstrating that the information-theoretic exploration in VBO-MI is not redundant and cannot be replaced by standard GP-based uncertainty without loss.
Taken together, both terms are necessary for the full performance of VBO-MI, and neither can be substituted by its GP-based counterpart without incurring a measurable cost.

Additional ablations. Appendix E reports three further sensitivity studies. The acquisition term ablation of Fig. 11 is extended to four additional real-world tasks (PDE optimization, Interferometer, Lunar Lander, Pest Control), confirming that the finding — exploitation removal hurts more than exploration removal — holds consistently across task types and dimensionalities. Fig. 9 shows that VBO-MI's performance improves with batch size up to a threshold, beyond which gains diminish, a pattern shared by all baselines. Fig. 10 shows that VBO-MI is less sensitive to the exploration parameter β than GP-based baselines, maintaining competitive final reward across $\beta \in [10, 200]$.

6 Conclusion

We presented VBO-MI, a fully gradient-based Bayesian optimization framework that eliminates GP priors and specific variational family assumptions. By combining a variational MI estimator for exploration with a data-driven surrogate for exploitation, all three networks in VBO-MI are trained via backpropagation while treating f as a strict black box. The surrogate reduces oracle queries to one batch per iteration (a $16\times$ reduction), and the amortised action-net replaces the inner-loop acquisition optimizer (a $10^2\times$ FLOP reduction). We established consistency guarantees and demonstrated competitive or superior performance on diverse benchmarks spanning $d = 10$ to $d = 60$. Future work includes extending VBO-MI to multi-objective settings and investigating tighter MI estimators for sparse reward landscapes.

Impact Statement

This work proposes a general-purpose framework for Bayesian optimization based on variational mutual information estimation, and a neural actor-critic structure. As a methodological contribution, the proposed approach is intended to improve optimization performance across a wide range of tasks and does not target any specific application domain. Bayesian optimization techniques are commonly used in areas such as engineering design, scientific experimentation, and machine learning model tuning. Such methods may be applied in diverse fields including biology, chemistry, or finance, and as a result there are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Miguel Aguiar, Amritam Das, and Karl H. Johansson. Universal approximation of flows of control systems by recurrent neural networks. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pp. 2320–2327, 2023. doi: 10.1109/CDC49753.2023.10383457.

Paul H. Algoet and Thomas M. Cover. A Sandwich Proof of the Shannon-McMillan-Breiman Theorem. *The Annals of Probability*, 16(2):899 – 909, 1988. doi: 10.1214/aop/1176991794.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 531–540. PMLR, 10–15 Jul 2018.

Leo Breiman. The Individual Ergodic Theorem of Information Theory. *The Annals of Mathematical Statistics*, 28(3):809 – 811, 1957. doi: 10.1214/aoms/1177706899.

Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient hamiltonian monte carlo. In Eric P. Xing and Tony Jebara (eds.), *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 1683–1691, Bejing, China, 22–24 Jun 2014. PMLR.

Xiaohong Chen and H. White. Improved rates and asymptotic normality for nonparametric neural network estimators. *IEEE Transactions on Information Theory*, 45(2):682–691, 1999. doi: 10.1109/18.749011.

Nuojin Cheng, Leonard Papenmeier, Stephen Becker, and Luigi Nardi. A unified framework for entropy search and expected improvement in bayesian optimization. In *Forty-second International Conference on Machine Learning*, 2025.

T.M. Cover and J.A. Thomas. *Elements of Information Theory*. Wiley, 2012. ISBN 9781118585771.

D.D. Cox and S. John. A statistical method for global optimization. In *[Proceedings] 1992 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 1241–1246 vol.2, 1992. doi: 10.1109/ICSMC.1992.271617.

Zhijie Deng, Feng Zhou, and Jun Zhu. Accelerated linearized laplace approximation for bayesian deep learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 2695–2708. Curran Associates, Inc., 2022.

M. D. Donsker and S. R. S. Varadhan. Asymptotic evaluation of certain markov process expectations for large time, i. *Communications on Pure and Applied Mathematics*, 28(1):1–47, 1975. doi: https://doi.org/10.1002/cpa.3160280102.

David Eriksson, Michael Pearce, Jacob Gardner, Ryan D Turner, and Matthias Poloczek. Scalable global optimization via local bayesian optimization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Peter I. Frazier. A Tutorial on Bayesian Optimization. *arXiv e-prints*, art. arXiv:1807.02811, July 2018. doi: 10.48550/arXiv.1807.02811.

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

Eduardo C. Garrido-Merchán and Daniel Hernández-Lobato. Dealing with categorical and integer-valued variables in bayesian optimization with gaussian processes. *Neurocomputing*, 380:20–35, 2020. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2019.11.004.

James Harrison et al. Variational last layer Bayesian neural networks. In *International Conference on Learning Representations (ICLR)*, 2024.

Philipp Hennig and Christian J. Schuler. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 13(57):1809–1837, 2012.

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

Jose Miguel Hernandez-Lobato, Michael Gelbart, Matthew Hoffman, Ryan Adams, and Zoubin Ghahramani. Predictive entropy search for bayesian optimization with unknown constraints. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1699–1707, Lille, France, 07–09 Jul 2015. PMLR.

Kurt Hornik, Maxwell B. Stinchcombe, and Halbert L. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.

Hisham Husain, Vu Nguyen, and Anton van den Hengel. Distributionally robust bayesian optimization with φ-divergences. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 20133–20145. Curran Associates, Inc., 2023.

Carl Hvarfner, Frank Hutter, and Luigi Nardi. Joint entropy search for maximally-informed bayesian optimization. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 11494–11506. Curran Associates, Inc., 2022.

Alexander Immer, Maciej Korzepa, and Matthias Bauer. Improving predictions of bayesian neural nets via local linearization. In Arindam Banerjee and Kenji Fukumizu (eds.), *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 703–711. PMLR, 13–15 Apr 2021.

Masanori Ishikura and Masayuki Karasuyama. Pareto-frontier entropy search with variational lower bound maximization. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 26490–26522. PMLR, 13–19 Jul 2025. URL `https://proceedings.mlr.press/v267/ishikura25a.html`.

Pavel Izmailov, Sharad Vikram, Matthew D Hoffman, and Andrew Gordon Gordon Wilson. What are bayesian neural network posteriors really like? In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 4629–4640. PMLR, 18–24 Jul 2021.

Moksh Jain, Emmanuel Bengio, Alex Hernandez-Garcia, Jarrid Rector-Brooks, Bonaventure F. P. Dossou, Chanakya Ajit Ekbote, Jie Fu, Tianyu Zhang, Michael Kilgour, Dinghuai Zhang, Lena Simine, Payel Das, and Yoshua Bengio. Biological sequence design with GFlowNets. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 9786–9801. PMLR, 17–23 Jul 2022. URL `https://proceedings.mlr.press/v162/jain22a.html`.

Huiwen Jia, Cong Shi, and Siqian Shen. Multi-armed bandit with sub-exponential rewards. *Operations Research Letters*, 49(5):728–733, 2021. ISSN 0167-6377. doi: https://doi.org/10.1016/j.orl.2021.08.004.

Felix Jimenez and Matthias Katzfuss. Scalable bayesian optimization using vecchia approximations of gaussian processes. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent (eds.), *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 1492–1512. PMLR, 25–27 Apr 2023. URL `https://proceedings.mlr.press/v206/jimenez23a.html`.

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

Johannes Kirschner, Ilija Bogunovic, Stefanie Jegelka, and Andreas Krause. Distributionally robust Bayesian optimization. In Silvia Chiappa and Roberto Calandra (eds.), *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 2174–2184. PMLR, 26–28 Aug 2020.

Jaehoon Lee, Jascha Sohl-dickstein, Jeffrey Pennington, Roman Novak, Sam Schoenholz, and Yasaman Bahri. Deep neural networks as gaussian processes. In *International Conference on Learning Representations*, 2018.

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

Mitchell McIntire, Daniel Ratner, and Stefano Ermon. Sparse gaussian processes for bayesian optimization. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, UAI'16, pp. 517–526, Arlington, Virginia, USA, 2016. AUAI Press. ISBN 9780996643115.

Samuel Müller, Noah Hollmann, Sebastian P Arango, Josif Grabocka, and Frank Hutter. PFNs: Prior-fitted networks for efficient Bayesian optimization. In *International Conference on Learning Representations (ICLR)*, 2023.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 11 2005. ISBN 9780262256834. doi: 10.7551/mitpress/3206.001.0001.

Tim G. J. Rudner, Freddie Bickford Smith, Qixuan Feng, Yee Whye Teh, and Yarin Gal. Continual learning via sequential function-space variational inference. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 18871–18887. PMLR, 17–23 Jul 2022.

Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical bayesian optimization of machine learning algorithms. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'12, pp. 2951–2959, Red Hook, NY, USA, 2012. Curran Associates Inc.

Jiaming Song and Stefano Ermon. Understanding the limitations of variational mutual information estimators. In *International Conference on Learning Representations*, 2020.

Jost Tobias Springenberg, Aaron Klein, Stefan Falkner, and Frank Hutter. Bayesian optimization with robust bayesian neural networks. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on Machine Learning*, pp. 1015–1022. Omnipress, 2010.

Sebastian Shenghong Tay, Chuan Sheng Foo, Urano Daisuke, Richalynn Leong, and Bryan Kian Hsiang Low. Efficient distributionally robust Bayesian optimization with worst-case sensitivity. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 21180–21204. PMLR, 17–23 Jul 2022.

Dor Tsur, Ziv Aharoni, Ziv Goldfeld, and Haim H. Permuter. Neural estimation and optimization of directed information over continuous spaces. *IEEE Transactions on Information Theory*, 69:4777–4798, 2022.

Zi Wang and Stefanie Jegelka. Max-value entropy search for efficient Bayesian optimization. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3627–3635. PMLR, 06–11 Aug 2017. URL `https://proceedings.mlr.press/v70/wang17e.html`.

Yunyue Wei, Vincent Zhuang, Saraswati Soedarmadji, and Yanan Sui. Scalable bayesian optimization via focalized sparse gaussian processes. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 120443–120467. Curran Associates, Inc., 2024.

Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P. Xing. Deep kernel learning. In Arthur Gretton and Christian C. Robert (eds.), *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pp. 370–378, Cadiz, Spain, 09–11 May 2016. PMLR.

Taeyoung Yun, Kiyoung Om, Jaewoo Lee, Sujin Yun, and Jinkyoo Park. Posterior inference with diffusion models for high-dimensional black-box optimization. In *Frontiers in Probabilistic Inference: Learning meets Sampling*, 2025. URL `https://openreview.net/forum?id=7VzwNqQu9x`.

A Consistency Analysis of Losses

In this section, our objective is to provide theoretical guaranties for our approach based on neural networks (NN) to estimate and approximate losses L_{E_ϕ} and L_{D_θ}. Specifically, we seek to prove the consistency of estimates of these losses. Consistency, in this context, refers to the property of an estimator that ensures that

as the number of data points increases indefinitely, the sequence of estimates converges in probability to the true parameter. Our goal is twofold:
First, to demonstrate that the expected values can be replaced with averages using tools from the generalized Birkhoff Theorem (Breiman, 1957) and generalized AEP theorem (Algoet & Cover, 1988), this is not trivial since the selected actions are not independent; we cannot directly use the law of large numbers in this case (LLN). In our proofs, we follow a similar approach to other neural information estimation methods specifically, Belghazi et al. (2018), and Tsur et al. (2022).
Second, to show that NNs can effectively approximate the underlying functions to compute these estimations using the universal approximation theorem (Hornik et al., 1989). This allows us to establish the consistency of our results. We focus on proving consistency for L_{E_ϕ}, noting that the same approach can be extended to L_{D_θ}. We do not go through all the details of proof for L_{D_θ}.
1. Estimation from samples: In the first step, we aim to demonstrate the consistency of estimating the actual loss. Specifically, we aim to show our loss:

$$\mathbb{E}[y_t \mid y_{t-1}, .., y_1, \mathbf{x}_{t-1}, .., \mathbf{x}_1] + \sqrt{\beta}\big[- \mathbb{E}_{P_{\mathbf{x}_t, \mathbf{Y}_t}}[D_\theta(\mathbf{x}_t, y_t)] + \log(\mathbb{E}_{P_{\mathbf{x}_t} \times P_{y_t}}[e^{D_\theta(\mathbf{x}_t, y_t)}])\big], \tag{34}$$

can be estimated using samples:

$$L_{E_\phi} = \frac{1}{B}\sum_{i=1}^{B} y_t^{(i)} - \sqrt{\beta}\big(\frac{1}{B}\sum_{i=1}^{B} D_\theta(\mathbf{x}^{(i)}, y^{(i)}) + \log \frac{1}{B^2}\sum_{i=1}^{B}\sum_{j=1}^{B} e^{D_\theta(\mathbf{x}^{(i)}, y^{(j)})}\big), \tag{35}$$

where $y_t^{(i)} = f(\mathbf{x}_t^{(i)}) + \epsilon_t^{(i)}$[6]
In our analysis, we separately consider each term in the above formulation. To establish consistency. To that end, we rely on the following two key theorems to prove our results:

Theorem A.1. *Generalized AEP(Generalized Asymptotic Equipartition Property for Markov Approximation). (Algoet & Cover, 1988) Let $\mathcal{H}$ be a standard Borel space, and consider the infinite product space $\mathcal{H}_0^\infty = \prod_{t=0}^{\infty} \mathcal{H}$, equipped with its canonical Borel σ-algebra $(\Omega, \mathcal{F})$. Let $\mathbb{P}$ represent the probability law of a stationary, ergodic stochastic process $\{X_t\}$ defined over $(\Omega, \mathcal{F})$, and let $\mathbb{M}$ denote a finite-order Markov measure that has a stationary transition mechanism. Suppose that for each finite n, the n-dimensional marginal of $\mathbb{P}$ is absolutely continuous with respect to that of $\mathbb{M}$, with associated joint density $p(x_0, \ldots, x_{n-1})$. Then, the generalized asymptotic equipartition property asserts that:*

$$\frac{1}{n}\log p(X_0, \ldots, X_{n-1}) \xrightarrow{n\to\infty} \lim_{k\to\infty} \mathbb{E}\left[\log p(X_k \mid X_{k-1}, \ldots, X_0)\right]. \tag{36}$$

Generalized AEP is particularly useful when we do not have the i.i.d. assumption, and when we are dealing with a stationary ergodic process. By applying the generalized AEP for the first two terms in L_{E_ϕ}, we conclude:

$$\mathbb{E}[y_t \mid y_{t-1}, .., y_1, \mathbf{x}_{t-1}, .., \mathbf{x}_1] \to \frac{1}{B}\sum_{i=1}^{B} \hat{y}_t^{(i)}, \tag{37}$$

$$\mathbb{E}_{P_{\mathbf{x}_t, y_t}}[D_\theta(\mathbf{x}_t, y_t)] \to \frac{1}{B}\sum_{i=1}^{B} D_\theta(\mathbf{x}^{(i)}, y^{(i)}). \tag{38}$$

Where the first term follows from equation 37 follows from consistency result for MSE estimation of neural networks (Chen & White, 1999), and second term directly from Generalized AEP. For sample estimation of the other term in L_{E_ϕ}, we need another theorem, as noted in other works in the literature (such as Tsur et al. (2022)). As we are dealing with a function of expectation, we invoke the Generalized Birkhoff Theorem (Breiman, 1957) to estimate

$$\log(\mathbb{E}_{P_{\mathbf{x}_t} \times P_{y_t}}[e^{D_\theta(\mathbf{x}_t, y_t)}]). \tag{39}$$

[6] $s_t^{(i)}$ refers to element i of random seed vector at time step t in Alg.1

Theorem A.2. ***Birkhoff's Ergodic Theorem. (Breiman, 1957)***
Let $(X, \mathcal{F}, \mu)$ be a probability space, and suppose $T : X \to X$ is a measure-preserving transformation, meaning that for all $A \in \mathcal{F}$,

$$\mu(T^{-1}(A)) = \mu(A).$$

Assume $f : X \to \mathbb{R}$ is integrable with respect to μ (i.e., $f \in L^1(\mu)$).
Then the following statements hold:

1. ***Almost Sure Convergence:*** *For μ-almost every $\mathbf{x} \in X$, the time average of f along the orbit of $\mathbf{x}$ under T converges:*

$$\lim_{n\to\infty} \frac{1}{n} \sum_{k=0}^{n-1} f(T^k(\mathbf{x})) = f^*(\mathbf{x}),$$

where f^* is a T-invariant function, meaning $f^*(T(\mathbf{x})) = f^*(\mathbf{x})$.

2. ***In the Ergodic Case:*** *If T is ergodic (i.e., the only T-invariant sets have measure 0 or 1), then $f^*(\mathbf{x})$ is constant μ-almost everywhere and equal to the integral of f:*

$$f^*(\mathbf{x}) = \int f \, d\mu \quad \textit{for } \mu\textit{-almost every } \mathbf{x}.$$

By the assumption of Ergodicity of input data sequences, the T transform in this case translates into the condition $\mathbb{P}(A) = \mathbb{P}(T^{-1}(A))$ for any $A \in \mathcal{F}$, where in this case we use a time-shift. By applying this to the first term, we conclude:

$$\log(\mathbb{E}_{P_{\mathbf{x}_t} \times P_{y_t}}[e^{D_\theta(\mathbf{x}_t, y_t)}]) \underset{\mathbb{P}-as}{\overset{n\to\infty}{\to}} \log \frac{1}{B^2} \sum_{i=1}^{B} \sum_{j=1}^{B} e^{D_\theta(\mathbf{x}^{(i)}, y^{(j)})}. \tag{40}$$

By putting together all terms, we conclude:

$$\begin{aligned} &\frac{1}{B} \sum_{i=1}^{B} \hat{y}_t^{(i)} + \frac{1}{B} \sum_{i=1}^{B} D_\theta(\mathbf{x}^{(i)}, y^{(i)}) - \log \frac{1}{B^2} \sum_{i=1}^{B} \sum_{j=1}^{B} e^{D_\theta(\mathbf{x}^{(i)}, y^{(j)})} \overset{n\to\infty}{\to} \\ &\mathbb{E}[y_t \mid y_{t-1}, .., y_1, \mathbf{x}_{t-1}, .., \mathbf{x}_1] + \mathbb{E}_{P_{\mathbf{x},y}}[D_\theta(\mathbf{x}, y)] - \log \mathbb{E}_{P_{\mathbf{x}\times y}}[e^{D_\theta(\mathbf{x}, y)}]. \end{aligned} \tag{41}$$

This completes the estimation step from samples.
2. Approximation Step: Up to now, we have provided the proof for the consistent estimation of our loss. In this stage, we aim to show the possibility to approximate functional space with the space of Feed-forward network (FNN), $E_\phi(\mathbf{s}) = \mathbf{x}_t, f(\mathbf{x}_t)$, and RNN networks, D_θ. we apply universal approximation theorem for FNN [Theorem.2.2 (Hornik et al., 1989)], and RNNs (Aguiar et al., 2023).
Our proof in this section is a generalization of Belghazi et al. (2018); Tsur et al. (2022), as our loss has an additional term, and both feed-forward and RNN neural networks.
First, consider the first network, E_ϕ. The action-net E_ϕ does not approximate f directly — since f is a strict black box, it is never accessible pointwise to E_ϕ. Instead, E_ϕ approximates the *optimal policy map* $\pi^* : \mathcal{S} \to \mathcal{D}$, defined as

$$\pi^*(\mathbf{s}) = \arg\max_{\mathbf{x}\in\mathcal{D}} \; g_\theta(\mathbf{x}), \tag{42}$$

where g_θ is the acquisition objective with the helper network parameters θ fixed, and $\mathcal{S}$ is the seed space from which $\mathbf{s}_t$ is drawn. Since $\mathcal{D}$ is compact and f is continuous by assumption, g_θ is continuous and π^* is well-defined. As $\pi^* : \mathcal{S} \to \mathcal{D}$ is a continuous map between compact subsets of Euclidean space, the Universal Approximation Theorem (Hornik et al., 1989) guarantees that for any $\epsilon \geq 0$, there exists a feedforward network $E_{\hat{\phi}}$ with sufficient capacity such that:

$$\sup_{\mathbf{s}\in\mathcal{S}} \|E_{\hat{\phi}}(\mathbf{s}) - \pi^*(\mathbf{s})\| \leq \epsilon. \tag{43}$$

Furthermore, since f is uniformly continuous on the compact domain $\mathcal{D}$, there exists a modulus of continuity L_f such that:

$$\left|f(E_{\hat{\phi}}(\mathbf{s})) - f(\pi^*(\mathbf{s}))\right| \leq L_f \cdot \epsilon. \tag{44}$$

Since $\pi^*(\mathbf{s})$ achieves $\max_{\mathbf{x}\in\mathcal{D}} f(\mathbf{x})$ by definition, equation 44 establishes that the action-net output approaches the global maximum of f as $\epsilon \to 0$, completing the approximation argument for E_ϕ.
Now our main focus is on proving that it is possible for our "helper" network, which consists of RNNs, to approximate the functional space to find information gain between selected actions and observations. Specifically, we want to show there is an RNN with parameters θ such that:

$$\mid I(\mathbf{x};\mathbf{y}) - I_\theta(\mathbf{x},\mathbf{y}) \mid \leq \eta, \tag{45}$$

where $I_\theta(\mathbf{x},\mathbf{y}) = \max_{\theta\in\Theta}(\mathbb{E}_{P_{x,y}}[T_\theta(\mathbf{x},\mathbf{y})] - \log(\mathbb{E}_{P_{\mathbf{x}\times\mathbf{y}}}[e^{T_\theta(\mathbf{x},y)}]))$.
Based on DV representation (Donsker & Varadhan, 1975)
Let: $T^*(\mathbf{x},\mathbf{y}) = \log\frac{p(\mathbf{x},\mathbf{y})}{p(\mathbf{x})p(\mathbf{y})}$, where be the discriminator function for estimating information gain. Based on Aguiar et al. (2023) there exist a RNN, for any $\epsilon \geq 0$ such that:

$$\sup_{\{\mathbf{x}\},\{y\}} \mid T^*(\mathbf{x},\mathbf{y}) - \hat{T}(\mathbf{x},\mathbf{y}) \mid \leq \epsilon. \tag{46}$$

Since equation 46 implies $|T^*(\mathbf{x},\mathbf{y}) - \hat{T}(\mathbf{x},\mathbf{Y})| \leq \epsilon \quad \forall(\mathbf{X},\mathbf{y})$, By applying expectation with respect to $p(\mathbf{x},\mathbf{y})$ we conclude:

$$\left|\mathbb{E}_{p(\mathbf{x},\mathbf{y})}[T^*(\mathbf{x},\mathbf{y}) - \hat{T}(\mathbf{x},\mathbf{y})]\right| \leq \epsilon. \tag{47}$$

Thus, the first term DV representation is satisfied.
Now we turn to the second term in DV representation:
[7]. Now define:

$$A = \mathbb{E}_{p(\mathbf{x})p(\mathbf{y})}[e^{T^*(\mathbf{x},\mathbf{y})}], \quad B = \mathbb{E}_{p(\mathbf{x})p(\mathbf{y})}[e^{\hat{T}(\mathbf{x},\mathbf{y})}]. \tag{48}$$

Now it is easy to see that by using the triangular inequality, and Taylor expansion, we can find the following lower bound

$$|\log A - \log B| \leq \frac{|A-B|}{\min(A,B)}. \tag{49}$$

Now to find the bound on $\mid \log A - \log B \mid$ we bound each term, $A - B$, and $\min(A,B)$:
If the T^* is bounded by M, e.g, $T^* \leq M$ we have:

$$|T^*(\mathbf{x},\mathbf{y}) - \hat{T}(\mathbf{x},\mathbf{y})| \leq \epsilon \implies \left|e^{T^*(\mathbf{x},\mathbf{y})} - e^{\hat{T}(\mathbf{x},\mathbf{y})}\right| \leq e^M \cdot \epsilon. \tag{50}$$

Now we turn to bound $\min(A-B)$.
If we assume that T^* is bounded from below, e.g, $T^* \geq L$, then we conclude:

$$A \geq e^L, \quad B \geq C.e^L, \tag{51}$$

where $C = e^{-|\epsilon|}$. So, as exp is monotonic, we have: $\min(A,B) \geq C.e^L$.
Now, substitute the bounds for $|A-B|$ and $\min(A,B)$ in equation 49 to conclude:

$$|\log A - \log B| \leq \frac{e^M \cdot \epsilon}{e^{L-\epsilon}} = e^{M-L+\epsilon} \cdot \epsilon. \tag{52}$$

Since e^{M-L} is bounded, as $\epsilon \to 0$:

$$\mid \log A - \log B \mid \leq \mathcal{O}(\epsilon) \to 0. \tag{53}$$

By combining equation 47, and equation 53 we conclude equation 45. Now by combining equation 43, and equation 45 we conclude that estimation and approximation of L_{E_ϕ} is consistent. As L_{D_θ} has similar terms to L_{E_ϕ}, the proof is similar and we omit the details here.

[7]We apply the universal approximation in the last step as a tool to prove the accurate estimation of $\mathbf{x}_t$, as an output of E_ϕ, and as result $y_t = f(\mathbf{x}_t) + \epsilon_t$

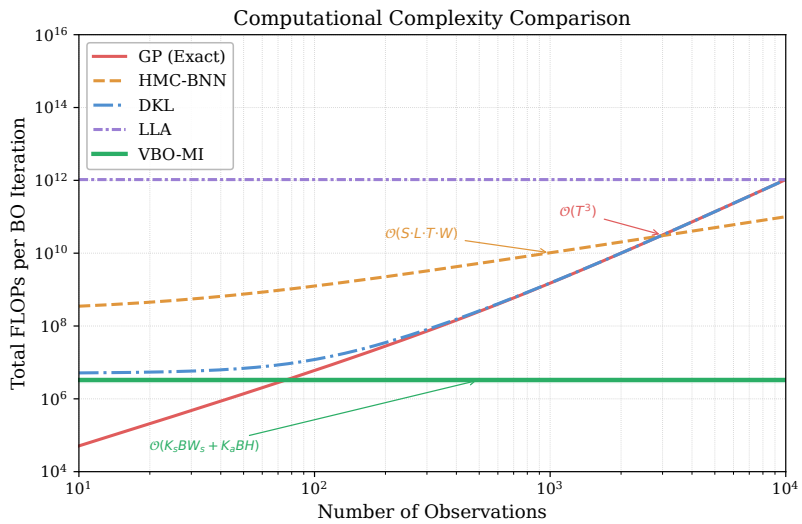


Figure 6: Theoretical computational complexity comparison measured in Floating-Point Operations (FLOPs) per Bayesian optimization iteration

B Gaussian BO

In this section we discuss some of the detailed results regarding the Gaussian Bayesian optimization discussed in Section 3. we summarize some of the main assumptions commonly used in the Gaussian literature (Srinivas et al., 2010) (Kirschner et al., 2020) (Tay et al., 2022) as below:

Assumption 1. Prior assumption on f: The function $f : \mathcal{D} \to \mathbb{R}$ is assumed to be a sample drawn from a Gaussian Process (GP), denoted as

$$f \sim \mathrm{GP}(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')), \tag{54}$$

where the mean function is $\mu(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$ and the covariance function is defined by

$$k(\mathbf{x}, \mathbf{x}') = \mathbb{E}\big[(f(\mathbf{x}) - \mathbb{E}[f(\mathbf{x})])(f(\mathbf{x}') - \mathbb{E}[f(\mathbf{x}')])\big]. \tag{55}$$

It is typically assumed that, without loss of generality, $\mu(\mathbf{x}) = 0$ and the variance of f prior is bounded, i.e., $k(\mathbf{x}, \mathbf{x}) \leq 1$ for all $\mathbf{x} \in \mathcal{D}$. In other words, for any finite collection of inputs $\{\mathbf{x}_t\}$, the corresponding (prior) function values $\{f(\mathbf{x}_t)\}$ follow a joint Gaussian distribution determined by the covariance function.

Assumption 2. Posterior assumption on f: After observing noisy samples, the posterior distribution of f conditioned on past inputs and observations remains a GP. Let $A_T = [\mathbf{x}_1, \ldots, \mathbf{x}_T]$ denote the set of previously sampled points, where each $\mathbf{x}_t \in \mathbb{R}^d$, and $\mathbf{Y}_T = [y_1, \ldots, y_T]^T$ be the vector of noisy observations, where $y_t = f(\mathbf{x}_t) + \epsilon_t,\ \epsilon_t \sim \mathcal{N}(0, \sigma^2)$ with $y_t \in \mathbb{R}$ scalar observations. It follows that the posterior distribution of f is given by (Rasmussen & Williams, 2005; Srinivas et al., 2010):

$$f \mid A_T, \mathbf{Y}_T \sim \mathrm{GP}(\mu_T(\mathbf{x}), k_T(\mathbf{x}, \mathbf{x}')), \tag{56}$$

where

$$\mu_T(\mathbf{x}) = \mathbf{k}_T(\mathbf{x})^T (\mathbf{K}_T + \sigma^2 \mathbf{I})^{-1} \mathbf{Y}_T, \tag{57}$$

$$k_T(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}, \mathbf{x}') - \mathbf{k}_T(\mathbf{x})^T (\mathbf{K}_T + \sigma^2 \mathbf{I})^{-1} \mathbf{k}_T(\mathbf{x}'). \tag{58}$$

Here, $\mathbf{k}_T(\mathbf{x}) = [k(\mathbf{x}_1, \mathbf{x}), \ldots, k(\mathbf{x}_T, \mathbf{x})]^T$ is the covariance vector between the new point $\mathbf{x}$ and the previously sampled points A_T, and $\mathbf{K}_T = [k(\mathbf{x}_i, \mathbf{x}_j)]_{i,j=1}^T$ is the kernel matrix. The posterior variance at $\mathbf{x}$ is given by $\sigma_T^2(\mathbf{x}) = k_T(\mathbf{x}, \mathbf{x})$.

C Experimental Setups

To ensure a fair comparison with previously reported results, we adopt the same benchmark problems and hyperparameter configurations as described in Li et al. (2024), applied to the experimental scenarios described below. Reported results are averaged over multiple random seeds to account for stochastic variability.

C.1 VBO-MI Implementation Details

Network architectures. The *action-net* E_ϕ consists of three fully connected layers of width 128 with `ReLU` activations and a `Tanh` output layer that bounds actions to $[-\texttt{bound}, \texttt{bound}]^d$. The *helper network* D_θ is a pairwise LSTM-MLP critic: two single-layer LSTMs (one for actions, one for observations, each with hidden size 64) whose final hidden states are concatenated and passed through a three-layer MLP with `ReLU` activations. The *surrogate network* $\hat{f}_\psi$ is a two-layer MLP with `ReLU` activations, mapping proposed actions to predicted reward values.

Optimisation hyperparameters. All three networks are optimised with Adam. The learning rate is 2×10^{-3} for E_ϕ and D_θ, and 1×10^{-3} for $\hat{f}_\psi$. The batch size is $B = 256$ across all experiments. The warm-up phase runs for $W = 30$ iterations, followed by a 200-step surrogate pre-training phase on the warm-up observations. In the main BO loop, the update counts are $K_a = 5$ (helper), $K_s = 10$ (surrogate), and $K_b = 10$ (action-net) per iteration.

BNN baselines. For all BNN baselines (HMC, SGHMC, LLA, DKL, IBNN), we use a three-layer MLP with `ReLU` activations and width 128 as the surrogate network, following the configuration in Li et al. (2024). For each baseline we perform a grid search over the method-specific hyperparameters and report the best performance.

C.2 Synthetic Benchmarks

We evaluate on three scalable multimodal functions. All three have their global minimum at zero and are framed as maximisation problems by optimising $-f$.
Ackley ($d = 10$, domain $[-32.768, 32.768]^{10}$) — A deceptive function with a near-flat outer region and a sharp global basin at the origin. Standard parameters: $a = 20$, $b = 0.2$, $c = 2\pi$.
Levy ($d = 20$, domain $[-10, 10]^{20}$) — A scalable multimodal function whose difficulty grows with dimension due to its sinusoidal structure; global minimum is $f(\mathbf{1}) = 0$.
Griewank ($d = 20$, domain $[-600, 600]^{20}$) — A non-separable function whose product term introduces interactions between dimensions, defined as $f(\mathbf{x}) = \sum_i x_i^2/4000 - \prod_i \cos(x_i/\sqrt{i}) + 1$, with global minimum $f(\mathbf{0}) = 0$.

C.3 Real-World Benchmarks

Lunar Lander ($d = 12$) — The objective is to learn an optimal control policy for the OpenAI Gym `LunarLander-v2` environment. The reward is computed over full episodes and the goal is to maximise the expected return averaged over 50 randomly generated environments. The controller is parameterised as a feed-forward neural network with two hidden layers of size 256 using `tanh` activations. A learning rate of 3×10^{-4} is used with the Adam optimiser.
Pest Control ($d = 25$, categorical) — This benchmark optimises a pest-management policy over multiple seasons, modelled as a combinatorial problem with 25 categorical variables each taking 5 values Eriksson et al. (2019). We apply one-hot vector encoding (Garrido-Merchán & Hernández-Lobato, 2020) to handle categorical inputs. The policy is parameterised by a two-layer neural network with 128 neurons per layer and `tanh` activations, optimized with Adam at learning rate 2×10^{-3}.
Rover Trajectory Planning ($d = 60$) — The goal is to find an optimal trajectory defined by 30 waypoints (2 coordinates each) in a 2D environment with obstacles (Eriksson et al., 2019). The reward penalises obstacle collisions and path deviation; higher is better with no known analytical optimum. This is the highest-dimensional real-world task in our evaluation and directly tests the scalability of each method.
For all experiments, Bayesian optimization is performed with an initial design of n_{init} points drawn via a Sobol sequence, followed by sequential acquisition steps until the total evaluation budget is reached. The evaluation budget and n_{init} match those of the baseline study of Li et al. (2024) for direct comparability.

D Computational Complexity and FLOP Analysis

A significant bottleneck in Bayesian optimization is the computational overhead associated with surrogate model updates, the inner-loop optimization of the acquisition function, and — most critically in expensive black-box settings — the number of real function evaluations required per iteration. We evaluate the complexity of our proposed VBO-MI framework along two axes: (i) the number of oracle calls to f per

BO iteration, which dominates cost when f is expensive; and (ii) the neural network FLOP count, which determines computational overhead when f is cheap. We compare against HMC-BNN, Deep Kernel Learning (DKL), and Linearized Laplace Approximation (LLA).

Oracle query complexity. In standard BNN-based BO, the surrogate is re-fitted and the acquisition function is re-optimised at every iteration, but the number of real queries to f per iteration remains B (one batch of candidate points). VBO-MI matches this budget: the surrogate $\hat{f}_\psi$, the helper D_θ, and the action-net E_ϕ are all updated using either the surrogate's own predictions or mini-batches from the replay buffer $\mathcal{H}$, so no additional oracle calls are incurred inside the K_a, K_s, or K_b loops. The *only* real query to f per iteration is the evaluation batch of B points collected at the end of each loop, which simultaneously calibrates the surrogate and tracks the best-so-far value. The per-iteration oracle cost is therefore $\mathcal{O}(B)$, matching the standard BO budget.

Neural network FLOP complexity. Table 1 summarises the per-iteration FLOP breakdown for each method.
Helper update (K_a steps). The helper D_θ receives surrogate predictions $\hat{y}_t = \hat{f}_\psi(\mathbf{x}_t)$ in place of real observations, at a combined cost of $\mathcal{O}(K_a \cdot B \cdot (W_s + H))$, where W_s is the surrogate parameter count and H is the LSTM hidden dimension.
Surrogate update (K_s steps). The surrogate is updated for K_s gradient steps on mini-batches of size B drawn from $\mathcal{H}$, at cost $\mathcal{O}(K_s \cdot B \cdot W_s)$. Since W_s is small (a three-layer MLP) and K_s is modest, this adds a low constant-factor overhead.
Action-net update (K_b steps). Each action-net step includes a frozen surrogate forward pass plus the action-net backward pass, at cost $\mathcal{O}(K_b \cdot (B \cdot W_s + W_e))$, where W_e is the action-net parameter count.
The key distinction from GP- and BNN-based methods is in the surrogate update complexity. GP-based methods require an $\mathcal{O}(T^3)$ Cholesky decomposition at each iteration; HMC-BNN requires $\mathcal{O}(S \cdot L \cdot T \cdot W)$ leapfrog steps over the full dataset; and DKL combines neural network training with a GP Cholesky. By contrast, VBO-MI's internal update costs are independent of T: the replay buffer $\mathcal{H}$ is capped at a fixed size, and all network updates process fixed-size mini-batches. The total per-iteration FLOP cost is therefore $\mathcal{O}(K_s \cdot B \cdot W_s + K_a \cdot B \cdot H + K_b \cdot B \cdot W_s)$, which is *constant in* T — a qualitatively different scaling regime from all GP- and BNN-based baselines. Furthermore, by eliminating the inner-loop acquisition optimizer (replaced by the amortised action-net), the acquisition phase costs only $\mathcal{O}(K_b \cdot W_e)$ rather than the $\mathcal{O}(N_{\text{starts}} \cdot N_{\text{steps}} \cdot S \cdot W)$ required by BNN-based methods, yielding a near $10^2\times$ reduction in acquisition FLOPs.
Fig. 6 plots the scaling behaviour and Table 1 summarises the asymptotic complexity. Table 2 lists the numerical values used for the theoretical estimates.

Table 1: Comparison of FLOP and oracle-call complexity per BO iteration. T: observations, W: BNN weights, W_s: surrogate weights, H: LSTM hidden dim, S: HMC samples, L: leapfrog steps, $K_{a,b,s}$: update steps. We assume $W > H$ and $S, L > K_{a,b}$.

Method	Surrogate Update	Acquisition Opt.	Real f/iter	Scaling
GP (Exact)	$\mathcal{O}(T^3)$	$N \cdot \mathcal{O}(T^2)$	B	Cubic
HMC-BNN	$\mathcal{O}(S \cdot L \cdot T \cdot W)$	$N \cdot \mathcal{O}(S \cdot W)$	B	High Linear
DKL	$\mathcal{O}(T \cdot W + T^3)$	$N \cdot \mathcal{O}(T^2 + W)$	B	Cubic
LLA	$\mathcal{O}(T \cdot W + W^3)$	$N \cdot \mathcal{O}(W^2)$	B	Linear
VBO-MI (ours)	$\mathcal{O}(K_s \cdot B \cdot W_s + K_a \cdot B \cdot H)$	$\mathcal{O}(K_b \cdot (B \cdot W_s + W_e))$	**B**	**Constant**

E Additional Numerical Evaluations

In this section, we provide extended numerical results across a broader set of benchmarks and hyperparameter conditions.

Additional real-world benchmarks. The main paper evaluates VBO-MI on three real-world tasks of increasing dimensionality: Lunar Lander ($d = 12$), Pest Control ($d = 25$), and Rover Trajectory ($d = 60$). Here, we supplement these results with two additional lower-dimensional real-world tasks that provide a complete picture of VBO-MI's performance across the full dimensionality range tested in the prior literature (Li

Table 2: Parameters used for the FLOP complexity analysis in Section D.

Parameter	Value	Description
W	10,000	BNN parameter count (weights + biases)
W_s	$\approx$500	Surrogate network parameter count (3-layer MLP)
W_e	$\approx$400	Action-net parameter count (3-layer MLP)
D	10	Input dimensionality of the objective function
H	64	Hidden state dimension of the Helper LSTM
S	50	Number of HMC posterior samples
L	20	Leapfrog steps per HMC sample
B	256	Batch size (oracle calls per eval block)
K_a	5	Gradient steps for the Helper network
K_b	10	Gradient steps for the Action-net
K_s	10	Gradient steps for the Surrogate network
N_{starts}	10	Restarts for acquisition function optimization
N_{steps}	50	Iterations per restart

et al., 2024): Brusselator PDE optimization ($d = 4$) and Interferometer position optimization ($d = 4$). The PDE task involves tuning the four coefficients of a Brusselator PDE to minimise the output variance of the simulated concentration field. The Interferometer task involves configuring an optical interferometer for maximum sensitivity by optimising the positions of two mirrors (Li et al., 2024). Both tasks serve as lower-dimensional reference points for the dimensionality sweep $d \in \{4, 12, 25, 60\}$ covered across the full paper.

Batch size sensitivity. Fig. 9 reports the effect of batch size B on final reward on PDE optimization, Interferometer, Lunar Lander, and Pest Control. VBO-MI generally improves with larger batch sizes up to a threshold, beyond which gains diminish — a pattern consistent across all baselines.

β sensitivity. Fig. 10 reports the sensitivity of each method to the exploration parameter β on the same four tasks. VBO-MI is notably less sensitive to the choice of β compared to GP-based baselines, maintaining competitive final reward values across a wide range of $\beta \in [10, 200]$.

Extended acquisition term ablation. Fig. 11 extends the acquisition term ablation of Section 5.3 to all four lower-dimensional real-world tasks (PDE, Interferometer, Lunar Lander, Pest Control). As in the main ablation on Ackley and Rover, replacing either term with its GP counterpart degrades performance, and replacing the exploitation term causes greater damage, particularly on higher-dimensional tasks. This confirms that the finding is not specific to the benchmarks shown in Section 5.3 and generalizes across task types.

Noise Effect Fig. 7 studies how observation noise affects the surrogate-assisted variant of VBO-MI relative to the original (no-surrogate) variant, evaluated on Ackley ($d = 10$) under three noise levels: $\sigma \in \{0.4, 0.8, 1.8\}$. Across all noise levels, the surrogate-assisted method (solid lines) achieves a higher final reward than the original variant (dashed lines). The performance gap between the two variants is consistent at low and moderate noise ($\sigma = 0.4$ and $\sigma = 0.8$), confirming that the surrogate remains a reliable approximation of f when noise is moderate. At high noise ($\sigma = 1.8$) both variants converge more slowly, as expected, but the surrogate-assisted method still maintains a clear advantage. This indicates that the surrogate $\hat{f}_\psi$ is robust to noisy observations: since it is trained by MSE minimisation over the replay buffer, it learns a smoothed estimate of f that is less sensitive to individual noisy evaluations than the direct oracle queries used by the original variant.

Warm-up steps Here, we investigate the impact of incorporating a surrogate model into VBO-MI and analyze the sensitivity of the algorithm to the number of warm-up steps. Specifically, we compare the performance of the proposed surrogate-based framework against a variant without surrogate modeling for different warm-up settings (20, 30, and 60 warm-up steps).

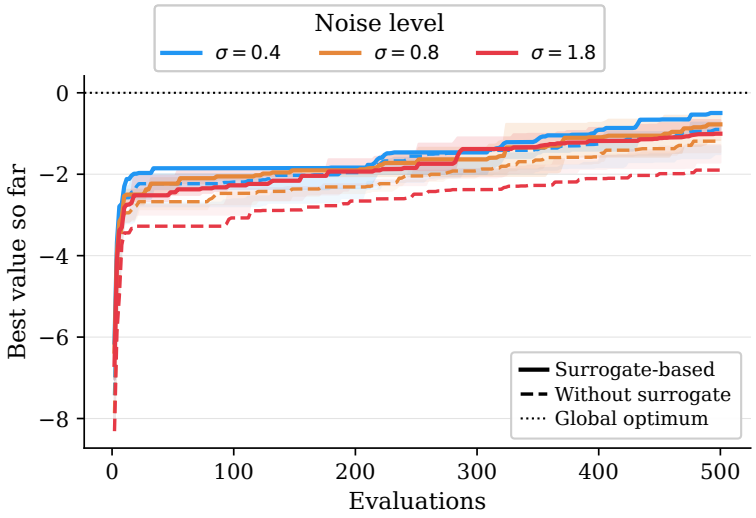


Figure 7: Effect of observation noise on VBO-MI. Solid lines: surrogate-assisted variant. Dashed lines: original variant (direct oracle queries in K_a, K_b loops).

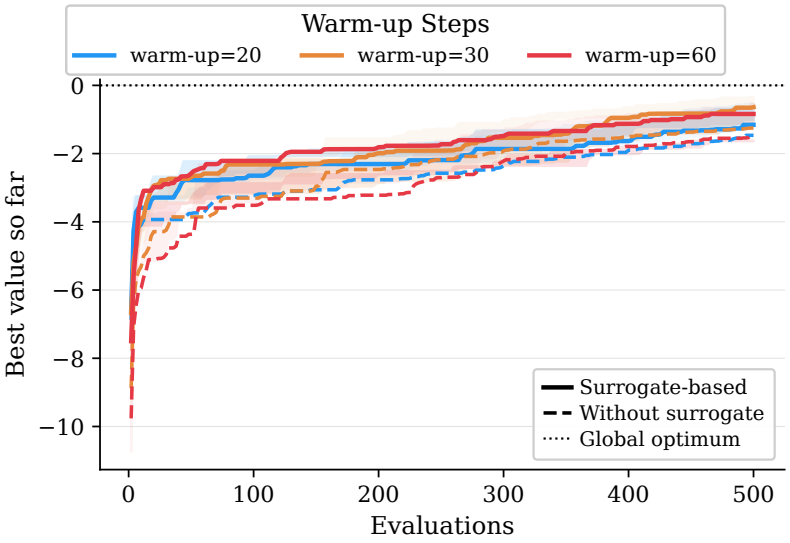
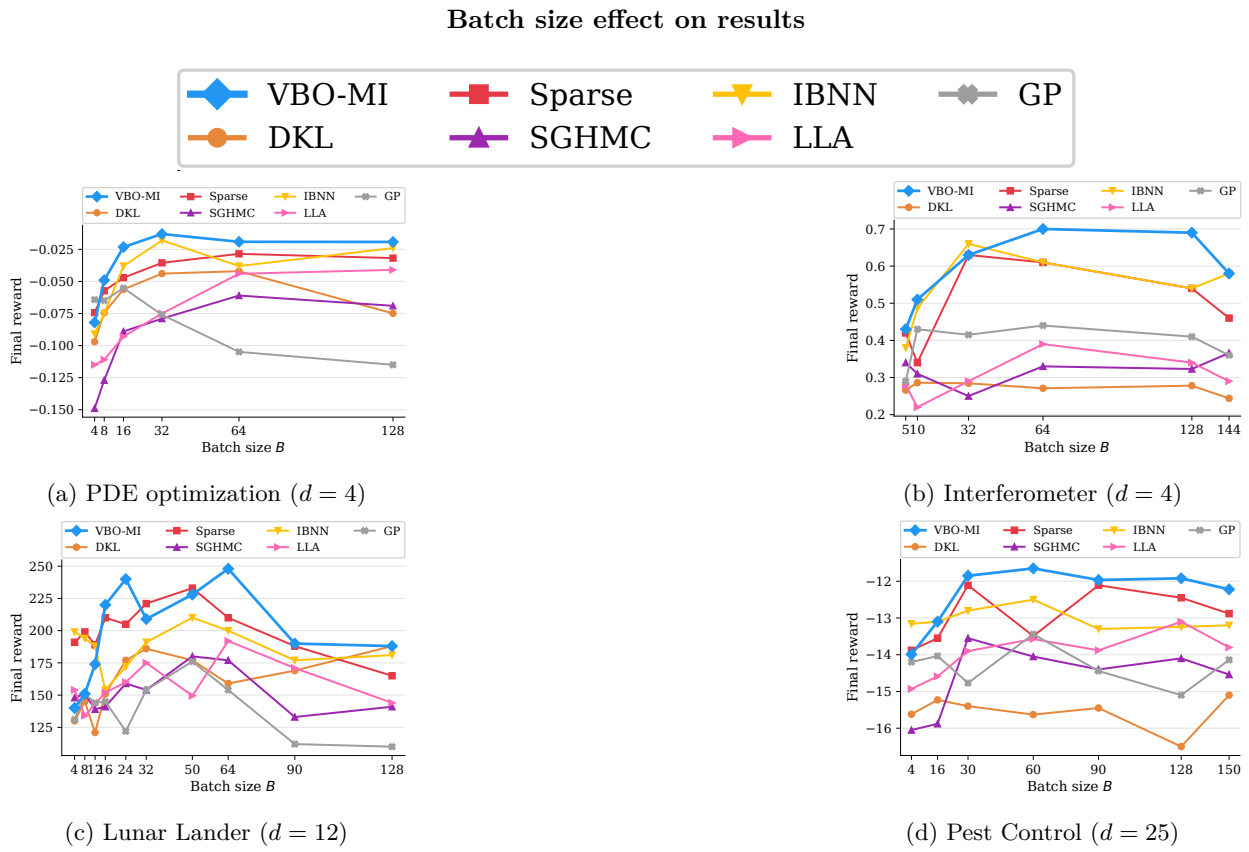


Figure 8: Comparison of VBO-MI with and without surrogate modeling for different numbers of warm-up steps. The surrogate-based version consistently outperforms the non-surrogate counterpart. Increasing the warm-up steps improves performance up to approximately 40 steps, while further increases provide marginal gains.

Fig. 8 illustrates the optimization performance over the course of function evaluations. Overall, the surrogate-based version consistently achieves superior optimization performance compared to the non-surrogate variant, demonstrating the effectiveness of the surrogate model in guiding the search process.
Moreover, increasing the number of warm-up steps from 20 to 30 improves the optimization performance, indicating that a moderate number of initial exploratory samples helps stabilize the Bayesian inference process and provides a better initialization for subsequent optimization. However, further increasing the warm-up period to 60 steps does not yield noticeable additional improvement, as the performance becomes similar to the 30-step setting. This observation suggests that approximately 30 warm-up steps provide a suitable trade-off between initial exploration and optimization efficiency for the considered setup.



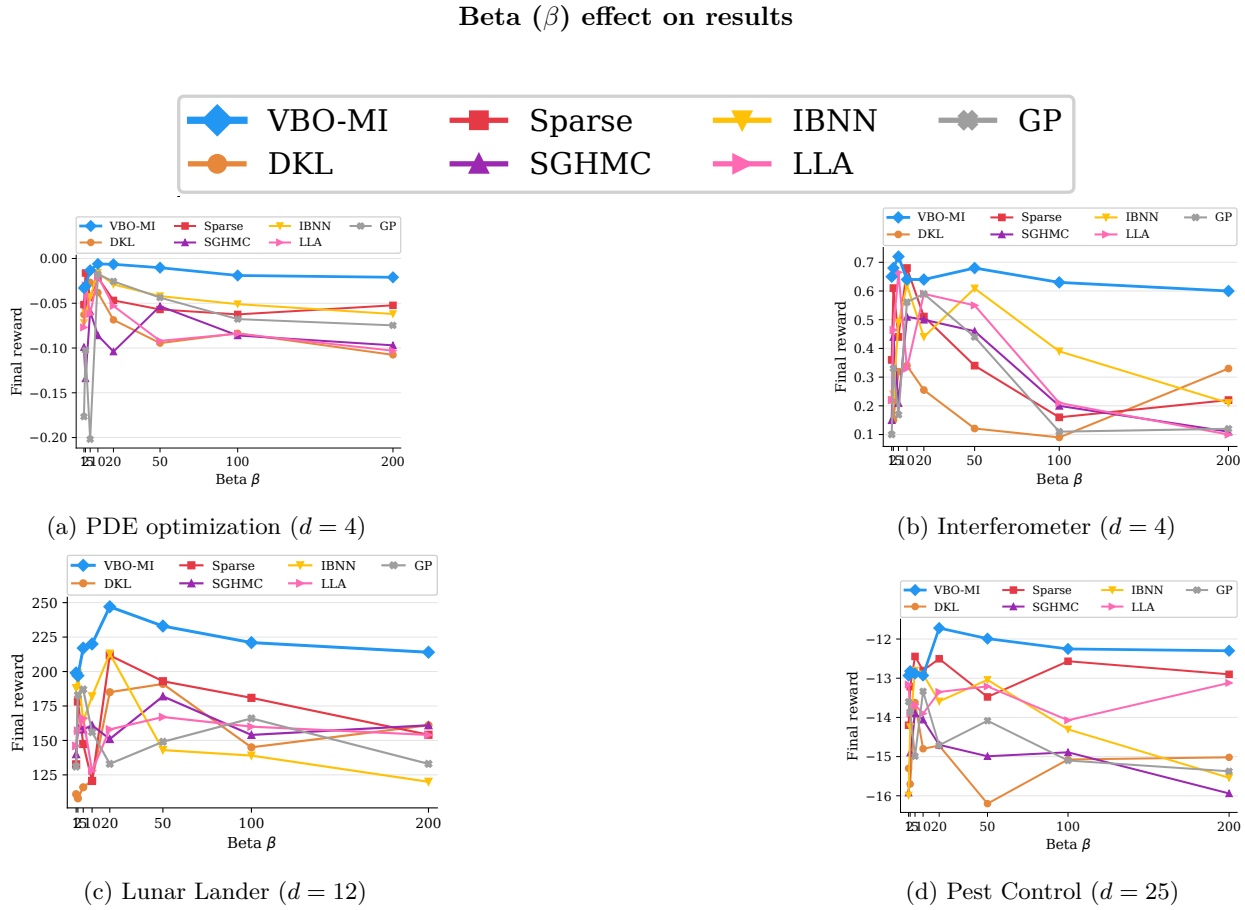
(a) PDE optimization ($d = 4$)

(b) Interferometer ($d = 4$)

(c) Lunar Lander ($d = 12$)

(d) Pest Control ($d = 25$)

Figure 9: Batch size sensitivity of VBO-MI and baselines on four real-world tasks. Final reward improves with batch size up to a threshold, after which gains diminish.



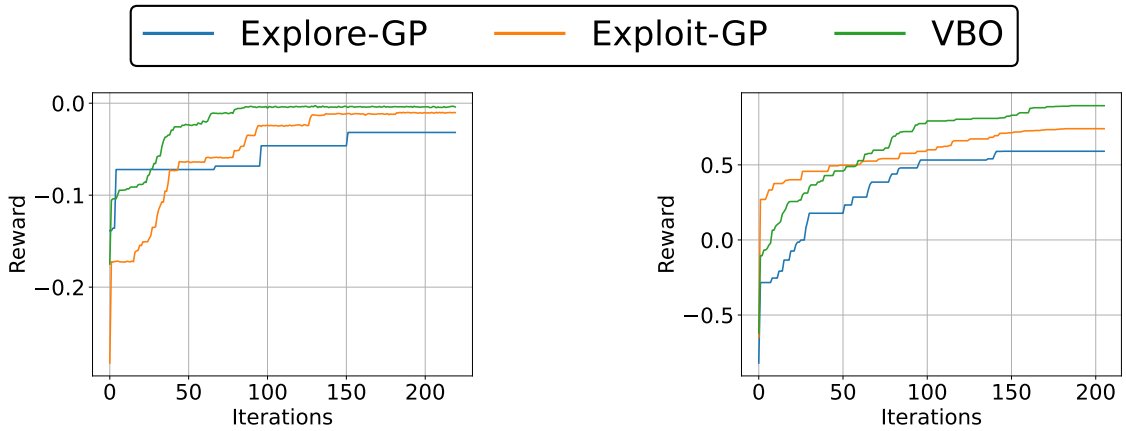
(a) PDE optimization ($d = 4$)

(b) Interferometer ($d = 4$)

(c) Lunar Lander ($d = 12$)

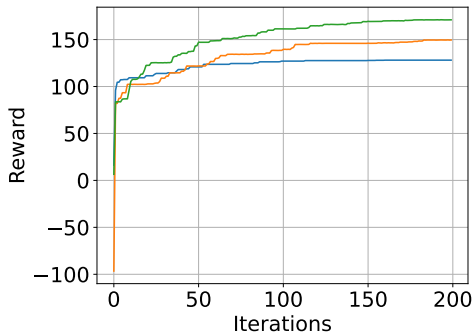
(d) Pest Control ($d = 25$)

Figure 10: Sensitivity to exploration parameter β across four real-world tasks. VBO-MI is less sensitive to β than GP-based baselines, maintaining high reward across $\beta \in [10, 200]$.

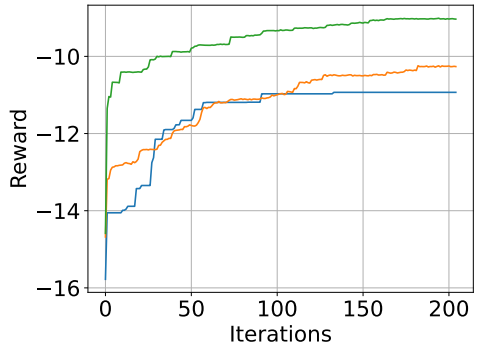


(a) PDE optimization ($d = 4$)

(b) Interferometer ($d = 4$)



(c) Lunar Lander ($d = 12$)



(d) Pest Control ($d = 25$)

Figure 11: Extended acquisition term ablation on four additional real-world tasks (cf. Section 5.3 for results on Ackley and Rover). *VBO-MI w/ GP exploration*: the exploration term $\sqrt{\beta}\, I_\theta$ is replaced by $\sqrt{\beta}\, \sigma^2_{\mathrm{GP}}$. *VBO-MI w/ GP exploitation*: the exploitation term $\hat{f}_\psi$ is replaced by μ_{GP}. Full VBO-MI outperforms both hybrids on all four tasks.