ENHANCED BAYESIAN OPTIMIZATION VIA PREFERENTIAL MODELING OF ABSTRACT PROPERTIES

Anonymous authors
Paper under double-blind review

A SUPPLEMENTARY MATERIAL

A.1 REPRODUCING KERNEL HILBERT SPACES

The kernel functions $k(\mathbf{x}, \mathbf{x}') : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ used in Gaussian Process (GP) uniquely define an associated Reproducing Kernel Hilbert Space (RKHS) (Aronszajn, 1950). Formally:

Definition A.1. Let $\mathcal{H}_k$ be a Hilbert space of real valued functions $f : \mathcal{X} \rightarrow \mathbb{R}$ on a non-empty set $\mathcal{X}$. A function $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is a reproducing kernel of $\mathcal{H}_k$, and $\mathcal{H}_k$ a **Reproducing Kernel Hilbert Space (RKHS)**, if

- $\forall \mathbf{x}, \mathbf{x}' \in \mathcal{X}$, $k(\mathbf{x}, \mathbf{x}') = \langle k(\cdot, \mathbf{x}), k(\cdot, \mathbf{x}') \rangle_{\mathcal{H}_k}$,
- k spans $\mathcal{H}_k$ i.e., $\forall \mathbf{x} \in \mathcal{X}$, $k(\cdot, \mathbf{x}) \in \mathcal{H}_k$,
- $\forall \mathbf{x} \in \mathcal{X}, \forall f \in \mathcal{H}_k$, $\langle f(\cdot), k(\cdot, \mathbf{x}) \rangle_{\mathcal{H}_k} = f(\mathbf{x})$ (the reproducing property).

There exists varieties of kernels that can be used in fitting a GP surrogate model. A kernel that depends only on the distance between two given points i.e., $k = k(\mathbf{x} - \mathbf{x}')$ is called as stationary kernels. Stationary kernels are also called as translation-invariant kernels. Some of the popular kernel functions are listed below.

A.1.1 MATÉRN KERNEL

Matérn kernel is a kernel that is commonly used in numerous machine learning applications. There are two variants of Matérn kernels that differ in their smoothness coefficient (ν) as shown below.

$$k_{\text{MAT}}(\mathbf{x}, \mathbf{x}')_{\nu=\frac{3}{2}} = \left(1 + \frac{\sqrt{3}}{l}\|\mathbf{x} - \mathbf{x}'\|\right) \exp\left(-\frac{\sqrt{3}}{l}\|\mathbf{x} - \mathbf{x}'\|\right) \tag{7}$$

$$k_{\text{MAT}}(\mathbf{x}, \mathbf{x}')_{\nu=\frac{5}{2}} = \left(1 + \frac{\sqrt{5}}{l}\|\mathbf{x} - \mathbf{x}'\| + \frac{\sqrt{5}}{3l^2}\|\mathbf{x} - \mathbf{x}'\|^2\right) \exp\left(-\frac{\sqrt{5}}{l}\|\mathbf{x} - \mathbf{x}'\|\right) \tag{8}$$

where l is the lengthscale hyperparameter of Matérn kernel.

A.1.2 SQUARED EXPONENTIAL KERNEL

Squared Exponential (SE) kernel or Radial Basis Function (RBF) or Gaussian kernel function is the popular stationary kernel function. The closed-form formulation of the SE kernel is represented as shown below.

$$k_{\text{SE}}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{1}{2l^2}\|\mathbf{x} - \mathbf{x}'\|^2\right) \tag{9}$$

where, σ_f^2 and l corresponds to the signal variance and lengthscale hyperparameter, respectively, collectively represented as $\Theta_k = \{l, \sigma_f^2\}$.

A.1.3 LINEAR KERNEL

Linear kernel is a commonly used non-stationary kernels defined as the inner product of the input data points. The mathematical formulation of a linear kernel is given by:

$$k_{\text{LIN}}(\mathbf{x}, \mathbf{x}') = \mathbf{x}'\mathbf{x}^{\intercal} + c \tag{10}$$

where c is the bias hyperparameter of linear kernels.

A.1.4 MULTI-KERNEL LEARNING

Multi-kernel is a non-stationary kernel function defined as a linear combination of stationary and non-stationary kernels. For instance, multi-kernels can be constructed as:

$$k_{\text{MKL}}(\mathbf{x}, \mathbf{x}') = w_1\ k_{\text{SE}}(\mathbf{x}, \mathbf{x}') + w_2\ k_{\text{MAT}}(\mathbf{x}, \mathbf{x}') + w_3\ k_{\text{LIN}}(\mathbf{x}, \mathbf{x}') \tag{11}$$

where $\mathbf{w} = [w_1\ w_2\ w_3]$ corresponds to the kernel weights, that are usually tuned by maximizing GP log-likelihood.

A.2 BAYESIAN OPTIMIZATION

The central idea of Bayesian Optimization (BO) strategy is to define a prior distribution over all the possible set of objective functions and then refine the model sequentially with the observed samples. BO is built on top of the Bayes theorem that incorporate prior belief about the black-box objective function under consideration. According to Bayes theorem, given a model $\mathcal{M}$ and data $\mathcal{D}$, the posterior probability of the model conditioned on data *i.e.,* $\mathcal{P}(\mathcal{M}|\mathcal{D})$ is directly proportional to the likelihood of data $\mathcal{D}$ conditioned on model $\mathcal{M}$ *i.e.,* $\mathcal{P}(\mathcal{D}|\mathcal{M})$, multiplied by the prior probability of model $\mathcal{P}(\mathcal{M})$,

$$\mathcal{P}(\mathcal{M}|\mathcal{D})\ \propto\ \mathcal{P}(\mathcal{D}|\mathcal{M})\,\mathcal{P}(\mathcal{M}) \tag{12}$$

The observation model of BO is collected as $\mathcal{D}_{1:t} = \{\mathbf{x}_{1:t}, \mathbf{y}_{1:t}\}$, where $y_t = f(\mathbf{x}_t) + \eta_t$ is a noisy observation of the black-box objective function f evaluated at input location $\mathbf{x}_t$ corrupted with a white Gaussian noise $\eta_t\ \sim \mathcal{GP}(0, \sigma_\eta^2)$. In BO, we compute the posterior distribution $\mathcal{P}(f\,|\mathcal{D})$ by combining the prior $\mathcal{P}(f)$ with the likelihood $\mathcal{P}(\mathcal{D}|\,f)$ represented as,

$$\mathcal{P}(f\,|\mathcal{D})\ \propto\ \mathcal{P}(\mathcal{D}|\,f)\,\mathcal{P}(f) \tag{13}$$

The posterior distribution $\mathcal{P}(f\,|\mathcal{D})$ computed captures our updated belief about the black-box objective function. BO can be perceived as a two step sequential strategy. First step focuses on defining the prior distribution that capture our prior beliefs. Usually GPs are used in placing prior distributions. Second step focuses on determining the best candidate that can be evaluated next. Acquisition functions are used to find the next candidate point with the high promise of finding the optima. An algorithm for the standard Bayesian optimization procedure is provided in Algorithm 2.

Algorithm 2 Standard Bayesian Optimization

Input: Set of observations $\mathcal{D}_{1:t'} = \{\mathbf{x}_{1:t'}, \mathbf{y}_{1:t'}\}$, Sampling budget T

1. **for** $t = t', \ldots, T$ iterations **do**
2. optimize $\Theta^\star = \underset{\Theta}{\text{argmax}}\ \log\ \mathcal{L}$
3. update GP model with optimal kernel hyperparameters $\Theta^\star$
4. find the next query point $\mathbf{x}_{t+1} = \underset{\mathbf{x}\in\mathcal{X}}{\text{argmax}}\, u(\mathbf{x})$
5. query $f(\mathbf{x})$ at $\mathbf{x}_{t+1}$ as $y_{t+1} = f(\mathbf{x}_{t+1}) + \eta_{t+1}$
6. augment the data as $\mathcal{D}_{1:t+1} = \mathcal{D}_{1:t} \cup (\mathbf{x}_{t+1}, y_{t+1})$
7. update GP model
8. **end for**

A.2.1 ACQUISITION FUNCTIONS

The acquisition function guides the optimization by balancing the trade-off between exploration and exploitation. Kushner (1964) proposed Expected Improvement (EI) acquisition function to guide the search by taking into account both the probability and magnitude of improvement over the current known optima. The next candidate point is obtained by maximizing the acquisition function given as:

$$u_{\text{EI}}(\mathbf{x}) = \begin{cases} (\mu(\mathbf{x}) - f(\mathbf{x}^+))\ \Phi(Z) + \sigma(\mathbf{x})\ \phi(Z) & \text{if } \sigma(\mathbf{x}) > 0 \\ 0 & \text{if } \sigma(\mathbf{x}) = 0 \end{cases} \tag{14}$$

$$\mathcal{Z} = \frac{\mu(\mathbf{x}) - f(\mathbf{x}^+)}{\sigma(\mathbf{x})}$$

where $\Phi(\mathcal{Z})$ and $\phi(\mathcal{Z})$ represents the Cumulative Distribution Function (CDF) and Probability Distribution Function (PDF) of the standard normal distribution, respectively and $f(\mathbf{x}^+)$ is the best value observed so far in the optimization.

Gaussian Process-Upper Confidence Bound (GP-UCB) acquisition function is another popular acquisition function defined based on confidence bounds criteria. GP-UCB acquisition function is given as:

$$u_{\text{GP-UCB}}(\mathbf{x}) = \mu(\mathbf{x}) + \sqrt{\beta_t}\ \sigma(\mathbf{x})$$

where β_t is a hyperparameter that balances the exploration-exploitation at iteration t. Srinivas et al. (2012) discussed in detail the possible values for trade-off parameter β_t. Following Brochu et al. (2010), we set the value for trade-off parameter (β_t) at iteration t as:

$$\beta_t = 2\log\left(\frac{t^{\frac{d}{2}+2}\pi^2}{3\delta'}\right)$$

where $\delta' \in (0, 1)$, d is the number of input dimensions.

A.2.2 THOMPSON SAMPLING BASED BAYESIAN OPTIMIZATION

Thompson sampling (Thompson, 1933) is a randomized selection strategy to select the next candidate for function evaluation by maximizing a random sample drawn from the posterior distribution. There have been significant advancements Kaufmann et al. (2012); Shahriari et al. (2014); Bijl et al. (2016); Chowdhury and Gopalan (2017) in Thompson sampling literature that demonstrate the theoretical guarantees of Thompson sampling. Russo and Van Roy (2014) provided a Bayesian regret bound for Thompson sampling using the notion of eluder dimension Li et al. (2021). A complete algorithm of Thompson sampling based Bayesian optimization is provided in Algorithm 3.

Algorithm 3 Thompson Sampling based Bayesian Optimization

Input: Set of observations $\mathcal{D}_{1:t'} = \{\mathbf{x}_{1:t'}, \mathbf{y}_{1:t'}\}$, Sampling budget T

1. **for** $t = t', \cdots, T$ iterations **do**
2. optimize $\Theta^* = \underset{\Theta}{\text{argmax}}\ \log\ \mathcal{L}$
3. update GP model with Θ^*
4. draw a random sample $\mathfrak{g}_{t+1}$ from the updated GP.
5. find the next query point $\mathbf{x}_{t+1} = \underset{\mathbf{x}\in\mathcal{X}}{\text{argmax}}\ \mathfrak{g}_{t+1}(\mathbf{x})$
6. query $f(\mathbf{x})$ at $\mathbf{x}_{t+1}$ as $y_{t+1} = f(\mathbf{x}_{t+1}) + \eta_{t+1}$
7. augment the data as $\mathcal{D}_{1:t+1} = \mathcal{D}_{1:t} \cup (\mathbf{x}_{t+1}, y_{t+1})$
8. update GP posterior model
9. **end for**

At each iteration t+1, Thompson sampling strategy selects a point $\mathbf{x}_t$ that is highly likely to be the optimum according to the posterior distribution i.e., $\mathbf{x}_t$ is drawn from the posterior distribution

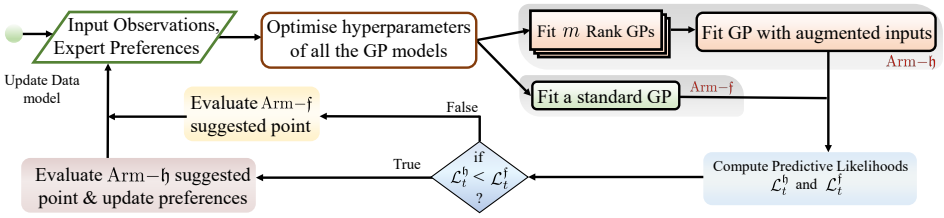

Figure 4: A complete process flowchart of our proposed BOAP framework.

$\mathcal{P}_{\mathbf{x}^\star}(\cdot|\mathcal{D}_{1:t-1})$ conditioned on data $\mathcal{D}_{1:t-1}$. Thompson sampling strategy simplifies if Gaussian Processes (GPs) are used for prior and posterior distributions. For GPs, if $\mathfrak{g}$ is a sample drawn from $\mathcal{GP}(\mu_{\mathcal{D}_{1:t-1}}, k_{\mathcal{D}_{1:t-1}})$ we have:

$$\mathcal{P}_{\mathbf{x}^\star}(\mathbf{x}|\mathcal{D}_{1:t-1}) = \int \mathcal{P}_{\mathbf{x}^\star}(\mathbf{x}|\mathfrak{g})\mathcal{P}(\mathfrak{g}|\mathcal{D}_{1:t-1})\, d\mathfrak{g}$$

The probability $\mathcal{P}_{\mathbf{x}^\star}(\mathbf{x}|\mathfrak{g})$ has its mass at the maximizer:

$$\underset{\mathbf{x}\in\mathcal{X}}{\operatorname{argmax}}\, \mathfrak{g}(\mathbf{x}) \tag{15}$$

Using this utility, at each iteration $t+1$, we draw a sample $\mathfrak{g}_{t+1}$ from $\mathcal{GP}(\mu_{\mathcal{D}_{1:t-1}}, k_{\mathcal{D}_{1:t-1}})$ and then find its maxima as per Eq. (15). The obtained maxima is used as the next candidate for function evaluation.

A.3 Additional Details of BOAP Framework

As discussed in the main paper, our proposed framework uses a model selection based decision making on whether to choose augmented GP ($\mathcal{GP}_h$) built on the expert preferential knowledge on abstract properties or the standard GP ($\mathcal{GP}_h$) for suggesting the next candidate for function evaluation. The arm containing the standard GP model is denoted as Arm-ƒ and the arm containing the augmented GP model is termed as Arm-ɧ. A complete flowchart of our framework is shown in Figure 4.

In BOAP, Arm-ƒ directly models the given objective function $f(\mathbf{x})$, whereas Arm-ɧ models $f(\mathbf{x})$ via a human objective function ($h(\tilde{\mathbf{x}})$) in a search space comprising of inputs that are augmented with the latent abstract properties computed using rank GPs. As discussed in the main paper, the human objective function $h(\tilde{\mathbf{x}})$ incorporates the additional expert feedback from experts, and thus more-informed and a simplified version of $f(\mathbf{x})$. Therefore, at iteration t, if Arm-ɧ is selected and $\tilde{\mathbf{x}}_t = [\mathbf{x}_t, \mu_{\omega_1}(\mathbf{x}_t), \cdots, \mu_{\omega_m}(\mathbf{x}_t)]$ is the candidate suggested, then we observe y_t as:

$$y_t = h([\mathbf{x}_t, \mu_{\omega_1}(\mathbf{x}_t), \cdots, \mu_{\omega_m}(\mathbf{x}_t)]) \approx f(\mathbf{x}_t)$$

A graphical representation of BOAP framework and its components are shown in Figure 5. Nodes highlighted in blue color corresponds to inputs or outputs of a Gaussian process. Rectangular boxes shaded in Grey correspond to the nodes representing rank GPs, whereas the rectangular boxes shaded in orange correspond to the nodes representing conventional GPs. The estimated parameters and the latent variables are highlighted in yellow and green color, respectively.

A.3.1 Derivatives of the Likelihood Function

In rank GP distributions the MAP estimates are computed using Eq. (4) mentioned in the main paper. However, the Newton-Raphson recursion needs access to the first and second order derivatives of the loss function $L = -\ln \Phi\big(z(\mathbf{x}, \mathbf{x}')\big)$ with respect to latent function values $\boldsymbol{\omega}$. The analytical formulation of the first order and second order derivatives are given as:

$$\frac{\partial L}{\partial \omega(\mathbf{x}_i)} = \frac{\Gamma(\mathbf{x}_i)}{\sqrt{2\sigma_\eta^2}} \frac{\phi(z)}{\Phi(z)} \tag{16}$$

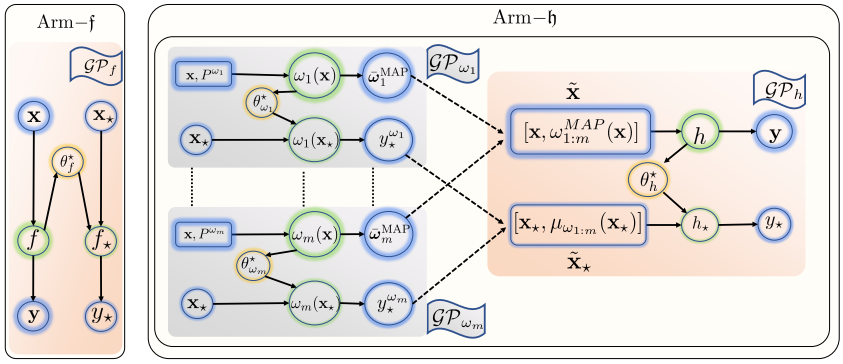

Figure 5: Components of Arm−ƒ and Arm−ɦ in BOAP framework. Nodes highlighted in blue color corresponds to inputs or outputs of a Gaussian process. The estimated parameters and the latent variables are highlighted in yellow and green color, respectively. Rectangular boxes shaded in Grey and Orange correspond to the nodes representing rank GPs and conventional GPs, respectively.

$$\frac{\partial^2 L}{\partial \omega(\mathbf{x}_i)\partial \omega(\mathbf{x}_j)} = \frac{\Gamma(\mathbf{x}_i)\ \Gamma(\mathbf{x}_j)}{2\sigma_\eta^2}\left(\frac{\phi^2(z)}{\Phi^2(z)} + z\frac{\phi(z)}{\Phi(z)}\right) \tag{17}$$

$$\text{where} \qquad \Gamma_{(\mathbf{x},\mathbf{x}')}(\mathbf{x}_i) = \begin{cases} 1 & \text{if } \mathbf{x} = \mathbf{x}_i \\ -1 & \text{if } \mathbf{x}' = \mathbf{x}_i \\ 0 & \text{Otherwise} \end{cases}$$

A.4 Additional Results and Experimental Details

A.4.1 Parameter Selection in BOAP framework

The kernel functions (k) used in fitting a Gaussian process model for the unknown objective function f is associated with its own hyperparameter set θ. The optimal kernel hyperparameters ($\theta^\star$) are estimated by maximizing the marginal likelihood function, given by the equation:

$$\mathcal{P}(\mathbf{y}|\mathbf{X},\theta) = \int p(\mathbf{y}|f)\ p(f|\mathbf{X},\theta)\ df \tag{18}$$

By marginalizing Eq. (18), we get the closed-form for GP log-likelihood as:

$$\mathcal{L} = \log \mathcal{P}(\mathbf{y}|\mathbf{X},\theta) = -\frac{1}{2}(\mathbf{y}^\intercal[\mathbf{K} + \sigma_\eta^2\mathbf{I}]^{-1}\mathbf{y}) - \frac{1}{2}\log|\mathbf{K} + \sigma_\eta^2\mathbf{I}| - \frac{n}{2}\log(2\pi) \tag{19}$$

where n corresponds to the total number of training instances. For a rank GP the closed-form of the GP log-likelihood can be obtained as:

$$\bar{\mathcal{L}} = -\frac{1}{2}\boldsymbol{\omega}_{\mathbf{MAP}}^\intercal[\mathbf{K} + \sigma_\eta^2\mathbf{I}]^{-1}\boldsymbol{\omega}_{\mathbf{MAP}} - \frac{1}{2}\log\mid \mathbf{K} + \sigma_\eta^2\mathbf{I} \mid - \frac{n}{2}\log(2\pi) \tag{20}$$

The only difference in the formulation of log-likelihood of a rank GP and a traditional GP is that the absolute measurements ($\mathbf{y}$) of the objective function is replaced with the latent function values obtained via Maximum A Posteriori (MAP) estimates. The GP log-likelihood mentioned in Eq. (19) and Eq. (20) is now maximized to find the optimal hyperparameters $\theta^\star$.

$$\theta^\star = \underset{\theta}{\operatorname{argmax}}\ \mathcal{L}$$

Further, if we use all training instances for the computation of the log marginal likelihood, there are chances that only Control arm may get selected in majority of the rounds. Therefore to avoid this,

instead of using all the training instances for computing the marginal likelihood, we use only the subset of the original training data for finding the optimal hyperparameter set and then we use the held-out instances from the original training set to compute the (predictive) likelihood.

In this paper, we implement Automatic Relevance Determination (ARD) kernel (Neal, 2012) based on Squared Exponential kernel mentioned in Eq. (9) at all levels of our proposed BOAP framework. ARD kernel is popular in the machine learning community due to its ability to suppress the irrelevant features. In ARD, each input dimension is assigned a different lengthscale parameter to keep track of the relevance of that dimension. Therefore, a GP fitted in a $d-$dimensional input space with ARD kernel has d lengthscale parameters *i.e.,* $\boldsymbol{l} = l_{1:d}$ and optional variance hyperparameters (noise variance σ_η^2 and signal variance σ_f^2).

Specifically, for the human-inspired arm (Arm-ɧ) we use a spatially-varying ARD kernel where we set the lengthscales of the augmented input dimensions in proportion to the rank GP uncertainties via a parametric function of the input $\mathbf{x}$. The lengthscale function for each of the augmented input dimension (corresponding to property ω_i) is set to be $l_{\omega_i}(\mathbf{x}) = \alpha\sigma_{\omega_i}(\mathbf{x})$, where α is the scale parameter and $\sigma_{\omega_i}(\mathbf{x})$ is the normalized standard deviation predicted using the rank GP ($\mathcal{GP}_{\omega_i}$). Therefore, the hyperparameter set (θ_h) of Arm-ɧ consists of the lengthscale parameters ($l_{1:D}$) for the original un-augmented dimensions and the scale factor ($\boldsymbol{\alpha}$) from the augmented input dimensions i.e., $\theta_h = \{l_{1:D}, \boldsymbol{\alpha}\}$.

The overall hyperparameter set Θ of our proposed framework consists of hyperparameters from each of the m rank GPs ($\theta_{\omega_{1:m}}$) and two main GPs corresponding to the 2-arms (θ_h and θ_f) *i.e.,* $\Theta = \{\theta_h, \theta_f\} = \{\{l_{1:D}, \boldsymbol{\alpha}\}, \{l_{1:D}\}\}$. At each iteration t, we find the optimal set of hyperparameters $\Theta_t^\star = \{\theta_h^\star, \theta_f^\star\}$ by maximizing the GP (predictive) log-likelihood mentioned in Eq. (19) and Eq. (20).

In all our experiments, we set the signal variance parameter $\sigma_f^2 = 1$ as we standardize the outputs of $\mathcal{GP}_h$ and $\mathcal{GP}_f$. We set the noise variance as $\eta \sim \mathcal{GP}(0, \sigma_\eta^2 = 0.1)$ and $\tilde{\sigma}_\eta^2 = 0.1$. As we normalize the input space of the GP distributions constructed in our BOAP framework, we tune each lengthscale hyperparameter $l \in \Theta$ in the interval $[0, 1]$. Further, we normalize the outputs of each of the auxiliary rank GPs ($\mathcal{GP}_{\omega_i}$) to avoid different scaling levels in their output, that can lead to undesired structures in the augmented input space.

We run all our experiments on an Intel Xeon CPU@ 3.60GHz workstation with 16 GB of RAM capacity. We repeat our experiments with 10 different random initialization. For a $d-$dimensional problem, we consider $t' = d + 3$ initial observations. The evaluation budget is set as $T = 10 \times d + 5$. For the real-world battery design experiments, due to their expensive nature, we have restricted the evaluation budget to 50 iterations even though $d \gg 5$. In all our experiments, we start with $p = \binom{t'}{2}$ preferences in P, that gets updated in every iteration of the optimization process.

A.4.2 Ablation Studies of BOAP Framework

To demonstrate the robustness of our approach we have conducted additional experiments by accounting for the inaccuracy or poor choices in expert preferential knowledge. Here, we show the performance of our BOAP approach in two scenarios. First, we show the performance of our proposed approach when the higher-order abstract properties are poorly selected. Second, we incorporate noise in the expert preferential feedback by flipping the expert preference between two inputs (designs) with probability $\delta = 0.3$. We now discuss in detail the aforementioned two variations of our proposed method.

Inaccurate Abstract Properties (BOAP-IA) In the first variation, we assume that the expert poorly selects the human abstraction features. Table 2 depicts the synthetic functions considered and the corresponding (poorly chosen) human abstraction features. BOAP-IA uses such inaccurate human abstract features while augmenting the original input space.

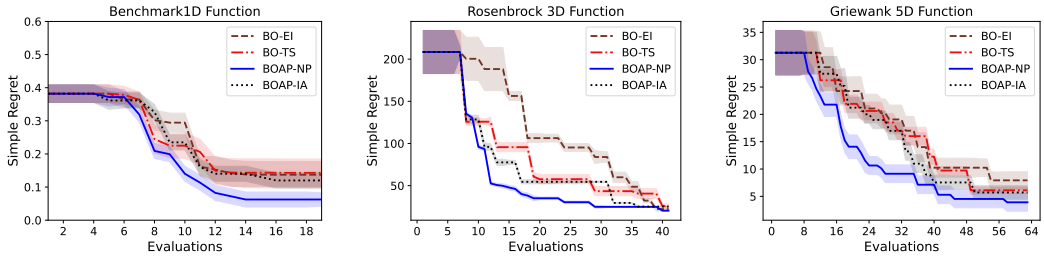

Figure 6: Simple regret vs iterations for the synthetic multi-dimensional benchmark functions. We plot the average regret along with its standard error obtained after 10 random repeated runs.

Table 2: Selection of abstraction features by a simulated human expert. The human abstraction (high level) features shown in the 3rd column are deliberately selected to be uninformative.

Functions	$f(\mathbf{x})$	Human Abstraction Features
Benchmark	$\exp^{(2-\mathbf{x})^2} + \exp^{\frac{(6-\mathbf{x})^2}{10}} + \frac{1}{\mathbf{x}^2+1}$	$\omega_1 = \sin \mathbf{x}, \omega_2 = \cos \mathbf{x}$
Rosenbrock	$\sum_{i=1}^{d-1} [100 \times (x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$\omega_1 = \sin \mathbf{x}, \omega_2 = \cos \mathbf{x}$
Griewank	$\sum_{i=1}^{d} \left[\frac{x_i^2}{4000} - \prod_{i=1}^{d} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \right]$	$\omega_1 = \sin \mathbf{x}, \omega_2 = \mathbf{x}^3$

Noisy Expert Preferences (BOAP-NP) In the second variation, we account for the inaccurate expert preferential knowledge by introducing an error in human expert preferential feedback. To do this, we flip the preference ordering with a probability δ (we used $\delta = 0.3$) i.e., $P^{\omega,\delta} = \{(\mathbf{x}_i \succ \mathbf{x}_j) \,|\, \mathbf{x}_i, \mathbf{x}_j \in \mathbf{x}_{1:n}, \nu_{ij}\, \omega(\mathbf{x}_i) > \nu_{ij}\, \omega(\mathbf{x}_j)\}$, where ν_{ij} is drawn from a random distribution such that it is $+1$ with probability $1-\delta$, -1 with probability δ.

We evaluate the performance by computing the simple regret after $10d$ iterations. The experimental settings are retained as mentioned in the experiments section (refer to Section 5.1 in the main paper). The empirical results for BOAP with inaccurate features (BOAP-IA) and BOAP framework with noisy preferences (BOAP-NP) are presented in Figure 6. It is significant from the results that our proposed BOAP framework outperforms standard baselines due to the model selection based 2-arm scheme employed that intelligently chooses the arm with maximum predictive likelihood to suggest the next sample.

A.4.3 Details of Real-world Experiments

As mentioned in the main paper, we evaluate the performance of BOAP framework in two real-world optimization paradigms in the Lithium-ion battery manufacturing. The advent of cheap Li-ion battery technology has significantly transformed range of industries including healthcare (Schmidt and Skarstad, 2001), telecommunication (Brunarie et al., 2011), automobiles Harper et al. (2019), and many more due to its ability to efficiently store the electrochemical energy. However, the process for manufacturing of Li-ion batteries is very complex and expensive in nature. Thus, there is a wide scope for optimizing the battery manufacturing process to reduce its CapEx and OpEx. We now provide a brief discussion on battery manufacturing experiments considered in our main paper. For real-world experiments, we use the same set of algorithmic parameters that we used in the synthetic experiments.

Optimization of Electrode Calendering Process
Duquesnoy et al. (2020) discussed the effect of calendering process on electrode properties that significantly contribute to the underlying electrochemical performance of a battery. Authors have implemented a data-driven stochastic electrode structure generator, based on which they construct electrodes and analyze in terms of *Tortuosity* (both in solid phase τ_{sol} and liquid phase τ_{liq}), percentage

of *Current Collector* (CC) surface covered by the active material and percentage of *Active Surface* (AS) covered by the electrolyte. The manufacturing process parameters considered are calendering pressure, Carbon-Binder Domain (CBD), initial electrode porosity and electrode composition. A pictorial representation of the inter dependencies between the input process variables and output electrode properties is provided in Figure 8 of Duquesnoy et al. (2020).

Duquesnoy et al. (2020) published a dataset reporting the input manufacturing process parameters and the output characteristics of 8800 electrode structures. Each of the manufacturing setting has been evaluated for 10 times, therefore we have averaged the results to obtain a refined dataset consisting of $n = 880$ instances with $d = 8$ process variables.

We optimize the calendering process by maximizing the *Active Surface* of an electrode (overall objective) by modeling two abstract properties: **(i)** Property 1 (ω_1): *Tortuosity in liquid phase* τ_{liq}, and **(ii)** Property 2 (ω_2): *Output Porosity* (OP). As discussed in the main paper, the abstract properties $\{\omega_{\tau_{\text{liq}}}, \omega_{\text{OP}}\}$ can only be qualitatively measured, however to simulate the expert pairwise preferential inputs $\{P^{\omega_{\tau_{\text{liq}}}}, P^{\omega_{\text{OP}}}\}$, we use the empirical measurements for τ_{liq} and OP in the published dataset.

$$P^{\omega_{\tau_{\text{liq}}}} = \{(\mathbf{x} \succ \mathbf{x}')_i \text{ if } \tau_{\text{liq}}(\mathbf{x}) > \tau_{\text{liq}}(\mathbf{x}') \mid \mathbf{x}, \mathbf{x}' \in \mathbf{x}_{1:n} \ \ \forall i \in \mathbb{N}_p\}$$

$$P^{\omega_{\text{OP}}} = \{(\mathbf{x} \succ \mathbf{x}')_i \text{ if } \text{OP}(\mathbf{x}) > \text{OP}(\mathbf{x}') \mid \mathbf{x}, \mathbf{x}' \in \mathbf{x}_{1:n} \ \ \forall i \in \mathbb{N}_p\}$$

We obtain the values $\tau_{\text{liq}}(\mathbf{x})$ and $\text{OP}(\mathbf{x})$ by referring to the dataset published. Based on these preference lists $\{P^{\omega_{\tau_{\text{liq}}}}, P^{\omega_{\text{OP}}}\}$, we fit two auxiliary rank GP distributions $\{\mathcal{GP}_{\omega_{\tau_{\text{liq}}}}, \mathcal{GP}_{\omega_{\text{OP}}}\}$. Then, we use these auxiliary rank GPs to augment the input space of the main GP surrogate $\mathcal{GP}_h$ modeling the overall objective i.e., active surface of the electrode.

Optimization of Electrode Manufacturing Process
In a similar case study (Drakopoulos et al., 2021), the authors have analyzed the manufacturing of Lithium-ion graphite based electrodes. The main aim of Drakopoulos et al. (2021) is to optimize the formulation and manufacturing process of Lithium-ion electrodes using machine learning. Authors have established a relationship between the process parameters at different stages of manufacturing such as mixing, coating, drying, and calendering. The published dataset records all the process parameters in manufacturing 256 coin cells, as well as the associated results showing the charge capacity of each coin cell measured after certain charge-discharge cycles. The refined dataset consists of 12 process variables, out of which two process variables: **(i)** *Anode Thickness* (AT), and **(ii)** *Active Mass* (AM) are treated as abstract properties that can be only qualitatively measured. The overall objective here is to maximize the battery endurance $E = \frac{D_{50}}{D_5}$, where D_{50} and D_5 are the discharge capacities of the cell at 50^{th} cycle and 5^{th} cycle, respectively.

We simulate the expert pairwise preferential inputs $\{P^{\omega_{\text{AT}}}, P^{\omega_{\text{AM}}}\}$ by comparing the empirical values recorded for these variables in the given dataset.

$$P^{\omega_{\text{AT}}} = \{(\mathbf{x} \succ \mathbf{x}')_i \text{ if } \text{AT}(\mathbf{x}) > \text{AT}(\mathbf{x}') \mid \mathbf{x}, \mathbf{x}' \in \mathbf{x}_{1:n} \ \ \forall i \in \mathbb{N}_p\}$$

$$P^{\omega_{\text{AM}}} = \{(\mathbf{x} \succ \mathbf{x}')_i \text{ if } \text{AM}(\mathbf{x}) > \text{AM}(\mathbf{x}') \mid \mathbf{x}, \mathbf{x}' \in \mathbf{x}_{1:n} \ \ \forall i \in \mathbb{N}_p\}$$

We model abstract properties $\{\omega_{\text{AT}}, \omega_{\text{AT}}\}$ by fitting rank GPs $\{\mathcal{GP}_{\omega_{\text{AM}}}, \mathcal{GP}_{\omega_{\text{AT}}}\}$ using preference lists $\{P^{\omega_{\text{AM}}}, P^{\omega_{\text{AM}}}\}$. We use rank GPs to estimate the abstract properties and then use those estimations to augment the input space of the main GP modeling the overall objective i.e., maximizing battery endurance E.

A.5 LIMITATIONS

Firstly, as discussed in Convergence Remarks section in the main paper, if the expert preferential data is inaccurate or irrelevant, it may mislead the model and increase eluder dimension, thus impeding convergence. Although the ablation studies show the robustness of BOAP against inaccurate/noisy expert preferences, our framework may suffer at least in the initial rounds of the optimization until the Control arm identifies and dominates to suppress the inaccurate/noisy human-inspired model

(Arm-$\mathfrak{h}$). Secondly, BOAP in its vanilla version may not be directly scalable because of the scalability issues of the underlying range of GPs used in the framework. One of the well-known weaknesses of GP is that it poorly scales and suffers from a cubic time complexity $O(n^3)$ due to the inversion of the gram matrix $\mathbf{K}$. This limits the scalability of GP and thus our BOAP framework to use with large-scale datasets. In the future line of work, we aspire to overcome these limitations by employing suitable GP techniques that can be easily scaled.

REFERENCES

N. Aronszajn. Theory of reproducing kernels. *Transactions of the American mathematical society*, 68(3):337–404, 1950.

H. Bijl, T. B. Schön, J.-W. van Wingerden, and M. Verhaegen. A sequential Monte Carlo approach to Thompson sampling for Bayesian optimization. *arXiv preprint arXiv:1604.00169*, 2016.

E. Brochu, V. M. Cora, and N. De Freitas. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.

J. Brunarie, A.-M. Billard, S. Lansburg, and M. Belle. Lithium-ion (Li-ion) battery technology evolves to serve an extended range of telecom applications. In *2011 IEEE 33rd International Telecommunications Energy Conference (INTELEC)*, pages 1–9. IEEE, 2011.

S. R. Chowdhury and A. Gopalan. On kernelized multi-armed bandits. In *International Conference on Machine Learning*, pages 844–853. PMLR, 2017.

S. X. Drakopoulos, A. Gholamipour-Shirazi, P. MacDonald, R. C. Parini, C. D. Reynolds, D. L. Burnett, B. Pye, K. B. O'Regan, G. Wang, T. M. Whitehead, et al. Formulation and manufacturing optimization of Lithium-ion graphite-based electrodes via machine learning. *Cell Reports Physical Science*, 2(12):100683, 2021.

M. Duquesnoy, T. Lombardo, M. Chouchane, E. N. Primo, and A. A. Franco. Data-driven assessment of electrode calendering process by combining experimental results, in silico mesostructures generation and machine learning. *Journal of Power Sources*, 480:229103, 2020.

G. Harper, R. Sommerville, E. Kendrick, L. Driscoll, P. Slater, R. Stolkin, A. Walton, P. Christensen, O. Heidrich, S. Lambert, et al. Recycling Lithium-ion batteries from electric vehicles. *nature*, 575 (7781):75–86, 2019.

E. Kaufmann, N. Korda, and R. Munos. Thompson sampling: An asymptotically optimal finite-time analysis. In *Algorithmic Learning Theory: 23rd International Conference, ALT 2012, Lyon, France, October 29-31, 2012. Proceedings 23*, pages 199–213. Springer, 2012.

H. J. Kushner. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Basic Engineering*, 86(1):97–106, 03 1964. ISSN 0021-9223. doi: 10.1115/1.3653121.

G. Li, P. Kamath, D. J. Foster, and N. Srebro. Understanding the eluder dimension. In *Advances in Neural Information Processing Systems*, 2021.

R. M. Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

D. Russo and B. Van Roy. Learning to optimize via posterior sampling. *Mathematics of Operations Research*, 39(4):1221–1243, 2014.

C. L. Schmidt and P. M. Skarstad. The future of Lithium and Lithium-ion batteries in implantable medical devices. *Journal of power sources*, 97:742–746, 2001.

B. Shahriari, Z. Wang, M. W. Hoffman, A. Bouchard-Côté, and N. de Freitas. An entropy search portfolio for Bayesian optimization. *arXiv preprint arXiv:1406.4625*, 2014.

N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger. Information-theoretic regret bounds for Gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58 (5):3250–3265, 2012.

W. R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3-4):285–294, 1933.