A. Approximate Backward Induction for Bayesian Optimal Stopping

In this section, we will present a commonly-used approximate backward induction algorithm for solving the BOS problem. The algorithm uses summary statistics to compactly represent the posterior belief $\Pr(\theta_t|\mathbf{y}_{t,n})$ which is computed from the prior belief $\Pr(\theta_t)$ and the noisy outputs $\mathbf{y}_{t,n}$ observed up till epoch n in iteration t.

In the approximate backward induction algorithm of Müller et al. (2007), the entire space of summary statistics is firstly partitioned into a number of discrete intervals in each epoch, which results in a two-dimensional domain with one axis being the number of epochs and the other axis representing the discretized intervals of the summary statistic (i.e., assuming the summary statistic is one-dimensional). In the beginning, a number of sample paths are generated through *forward simulation*: Firstly, a large number of samples are drawn from the prior belief $\Pr(\theta_t)$. Then, for each sample drawn from $\Pr(\theta_t)$, an entire sample path is generated from epochs 1 to N through repeated sampling. In this manner, each sample path leads to a curve in the 2-D domain and fully defines N posterior beliefs with one in each epoch. Starting from the last epoch N, for each interval, the expected loss of a terminal decision d_1 or d_2 is evaluated for every sample path ending in this interval (since each such sample path ends with a particular posterior belief in epoch N), and their empirical average is used to approximate the expected loss of the particular terminal decision for this interval. The minimum of the expected losses among the two terminal decisions is the expected loss for this particular interval, which is equivalent to (2) except that decision d_0 is not available in the last epoch N.

Next, the algorithm proceeds backwards from epoch $n = N-1$ all the way to epoch $n = 1$. In each epoch n, the expected loss of each terminal decision is evaluated in the same way as that in the last epoch N, as described above. To evaluate the expected loss of the continuation decision for an interval, for each sample path passing through this interval, the expected loss for the interval that it passes through in the next epoch $n+1$ is recorded and an average of all the recorded expected losses in the next epoch $n+1$ is summed with the cost c_{d_0} of observing the noisy output $y_{t,n+1}$ to yield the expected loss of the continuation decision d_0 for this particular interval; this is equivalent to approximating the $\mathbb{E}_{y_{t,n+1}|\mathbf{y}_{t,n}}[\rho_{t,n+1}(\mathbf{y}_{t,n+1})]+c_{d_0}$ term in (2) via Monte Carlo sampling of the posterior belief $\Pr(y_{t,n+1}|\mathbf{y}_{t,n})$. Following (2), the minimum of the expected losses among all terminal and continuation decisions is the expected loss for this particular interval and the corresponding decision is recorded as the optimal decision when the summary statistic falls into this interval. Then, the algorithm continues backwards until epoch $n = 1$ is reached. After the algorithm has finished running, the optimal decision computed in every pair of epoch and interval will form the optimal decision rules which serve as the output of the approximate backward induction algorithm.

B. Approximate Backward Induction Algorithm for Solving BOS Problem in BO-BOS

In this section, we will describe the approximate backward induction algorithm for solving the BOS problem (line 5) in each iteration of BO-BOS (Algorithm 1), which is adapted from the algorithm introduced in Appendix A.

To account for Assumption 1b in the approximate backward induction algorithm, we adopt the kernel k introduced in (Swersky et al., 2014) to incorporate the inductive bias that the learning curve (in the form of validation error) of the ML model is approximately exponentially decreasing in the number of training epochs, which can be expressed as

$$k(n,n') \triangleq \int_0^\infty \exp(-\lambda n)\exp(-\lambda n')\ \phi(\lambda)\ \mathrm{d}\lambda = \frac{\beta^\alpha}{(n+n'+\beta)^\alpha} \tag{5}$$

for all epochs $n, n' = 1,\ldots,N$ where ϕ is a probability measure over λ that is chosen to be a Gamma prior with parameters α and β. The above kernel (5) is used to fit a GP model to the validation errors $\mathbf{1}-\mathbf{y}_{t,N_0}$ of the ML model trained using $\mathbf{x}_t$ for a fixed number N_0 of initial epochs (e.g., $N_0 = 8$ in all our experiments when $N = 50$), specifically, by computing the values of parameters α and β in (5) via Bayesian update (i.e., assuming that the validation errors follow the Gamma conjugate prior with respect to an exponential likelihood). Samples are then drawn from the resulting GP posterior belief for forward simulation of sample paths from epochs N_0+1 to N, which are used to estimate the $\Pr(\theta_t|\mathbf{y}_{t,n})$ and $\Pr(y_{t,n+1}|\mathbf{y}_{t,n})$ terms necessary for approximate backward induction. Fig. 5 plots some of such sample paths and demonstrates that the GP kernel in (5) can characterize a monotonic learning curve (Assumption 1b) well.

Following the practices in related applications of BOS (Brockwell & Kadane, 2003; Jiang et al., 2013; Müller et al., 2007), the average validation error (or, equivalently, average validation accuracy) over epochs 1 to n is used as the summary statistics. Firstly, the entire space of summary statistics is partitioned into a number of discrete intervals in each epoch, which results in a two-dimensional domain with one axis being the number of epochs and the other axis representing the

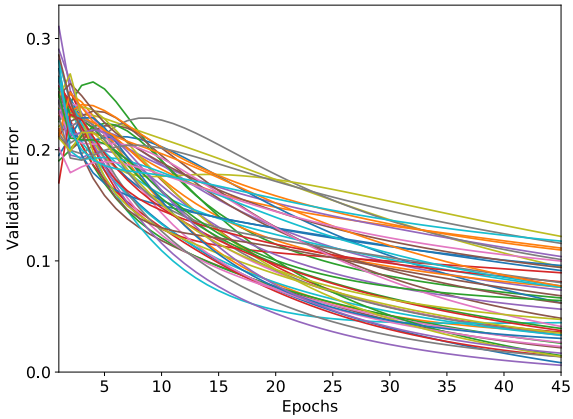


Figure 5. Forward simulation of some sample paths drawn from a GP posterior belief based on the kernel in (5).

discretized intervals of the summary statistic (i.e., average validation error). Next, a forward simulation of a large number (i.e., $100,000$ in all our experiments) of sample paths is performed using the GP kernel in (5), as described above. Each sample path corresponds to a curve in the 2-D domain. Starting from the last epoch N, for each interval, we consider all sample paths ending in this interval and use the proportion of such sample paths with a validation accuracy (from model training for N epochs) larger than the currently found maximum (offset by a noise correction term) to estimate the posterior probability $\Pr(\theta_t = \theta_{t,2}|\mathbf{y}_{t,N}) = \Pr(f([\mathbf{x}_t, N]) > y^*_{t-1} - \xi_t|\mathbf{y}_{t,N})$, which is in turn used to evaluate the expected losses of the terminal decisions d_1 and d_2 for this interval.[4] The minimum of the expected losses among the two terminal decisions is the expected loss for this particular interval.

Next, the algorithm proceeds backwards from epoch $n = N - 1$ all the way to epoch $n = N_0 + 1$. In each epoch n, the expected loss of each terminal decision is evaluated in the same way as that in the last epoch N, as described above. The expected loss of the continuation decision d_0 is evaluated in the same way as that in Appendix A: For each sample path passing through an interval in epoch n, the expected loss for the interval that it passes through in the next epoch $n + 1$ is recorded and an average of all the recorded expected losses in the next epoch $n + 1$ is summed with the cost c_{d_0} of observing the validation accuracy $y_{t,n+1}$ to yield the expected loss of the continuation decision d_0 for this particular interval. Note that this step is equivalent to approximating the $\mathbb{E}_{y_{t,n+1}|\mathbf{y}_{t,n}}[\rho_{t,n+1}(\mathbf{y}_{t,n})]$ term in (2) via Monte Carlo sampling of the posterior belief $\Pr(y_{t,n+1}|\mathbf{y}_{t,n})$. Following (2), the minimum of expected losses among all terminal and continuation decisions is the expected loss for this particular interval and the corresponding decision is recorded as the optimal decision to be recommended when the summary statistic falls into this particular interval. Then, the algorithm continues backwards until epoch $n = N_0 + 1$ is reached. We present in Algorithm 2 the pseudocode for the above-mentioned approximate backward induction algorithm for ease of understanding.

After solving our BOS problem for early stopping in BO using the approximate backward induction algorithm described above, Bayes-optimal decision rules are obtained in every pair of epoch and interval. Fig. 6 shows an example of optimal decision rules obtained from solving an instance of our BOS problem where the white, yellow, and red regions correspond to recommending optimal continuation decision d_0 and terminal decisions d_1 and d_2, respectively. In particular, after model training under $\mathbf{x}_t$ to yield the validation error $1 - y_{t,n}$ in epoch n, the summary statistic is updated to the average validation error over epochs 1 to n. The updated summary statistic falls into an interval with a corresponding optimal decision to be recommended. For example, Fig. 6 shows that if the summary statistic falls into the yellow region in any epoch n, then the optimal terminal decision d_1 is recommended to early-stop model training under $\mathbf{x}_t$ (assuming that C2 is satisfied). If the summary statistic falls into any other region, then model training continues under $\mathbf{x}_t$ for one more epoch and the above procedure is repeated in epoch $n + 1$ until the last epoch $n = N$ is reached. This procedure, together with C2, constitutes lines 6 to 9 in Algorithm 1.

[4] In contrast to the approximate backward induction algorithm of Müller et al. (2007) (Appendix A), we employ a computationally cheaper way to approximate the expected losses of the terminal decisions for an interval.

Algorithm 2 Approximate Backward Induction Algorithm for Solving BOS Problem in BO-BOS

Partition the domain of summary statistics into M discrete intervals
Train the ML model using $\mathbf{x}_t$ for N_0 epochs
Generate a large number of forward simulation samples using kernel (5)
Let $n = N$
for $m = 1, 2, \ldots, M$ **do**
  Find all sample paths ending in interval m at epoch n, denoted as $\mathcal{S}$
  Estimate $\Pr(\theta_t = \theta_{t,2}|\mathbf{y}_{t,n}) = \Pr(f([\mathbf{x}_t, N]) > y^*_{t-1} - \xi_t|\mathbf{y}_{t,n})$ by the proportion of $\mathcal{S}$ that end up (after N epochs) having larger validation accuracy than $y^*_{t-1} - \xi_t$
  Calculate the expected losses of the terminal decisions d_1 and d_2 using (4)
  Use the minimum of these two expected losses as the expected loss of epoch n and interval m
for $n = N-1, N-2, \ldots, N_0+1$ **do**
  for $m = 1, 2, \ldots, M$ **do**
    Find all sample paths passing through interval m at epoch n, denoted as $\mathcal{S}$
    Estimate $\Pr(\theta_t = \theta_{t,2}|\mathbf{y}_{t,n}) = \Pr(f([\mathbf{x}_t, N]) > y^*_{t-1} - \xi_t|\mathbf{y}_{t,n})$ by the proportion of $\mathcal{S}$ that end up (after N epochs) having larger validation accuracy than $y^*_{t-1} - \xi_t$
    Calculate the expected losses of the terminal decisions d_1 and d_2 using (4)
    $l_{d_0,n+1,m} = 0$
    for each sample path s in $\mathcal{S}$ **do**
      $l_{d_0,n+1,m} = l_{d_0,n+1,m}+$ the expected loss of the interval reached by s at epoch $n+1$
    $l_{d_0,n+1,m} = l_{d_0,n+1,m}/|\mathcal{S}|$
    Calculate the expected loss of the continuation decision d_0 as: $\mathbb{E}_{y_{t,n+1}|\mathbf{y}_{t,n}}[\rho_{t,n+1}(\mathbf{y}_{t,n})] + c_{d_0} = l_{d_0,n+1,m} + c_{d_0}$
    Use the minimum expected losses among d_1, d_2 and d_0 as the expected loss of epoch n and interval m (following (2)), and record the corresponding decision as the optimal decision

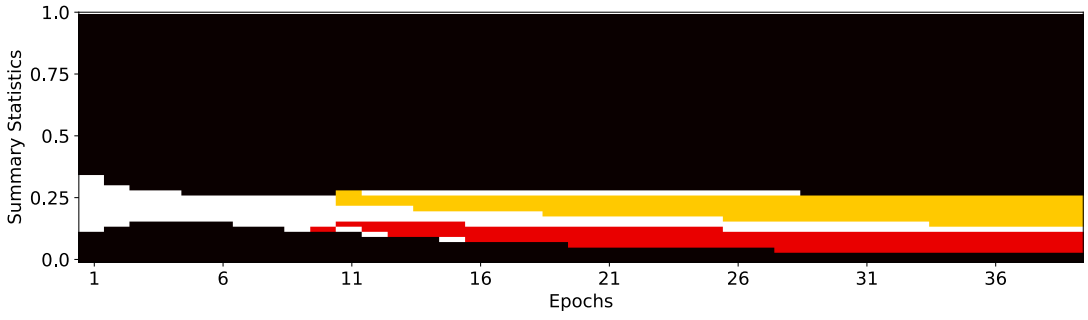


Figure 6. An example of optimal decision rules obtained from solving an instance of our BOS problem: White, yellow, and red regions correspond to recommending optimal continuation decision d_0 and terminal decisions d_1 and d_2, respectively. The sample paths cannot reach the black regions due to the use of the GP kernel in (5) for characterizing a monotonic learning curve (Assumption 1b).

C. Proof of Theorems 1 and 2

In this section, we prove the theoretical results in this paper.

C.1. Regret Decomposition

In this work, it is natural and convenient to define the instantaneous regret at step t as $r_t = f(\mathbf{z}^*) - f_t^*$, in which $\mathbf{z}^*$ is the location of the global maximum: $\mathbf{z}^* = \text{argmax}_{\mathbf{z}} f(\mathbf{z})$, and f_t^* is the maximum observed function value from iterations 1 to t: $f_t^* = \max_{t'=1,\dots,t} f(\mathbf{z}_{t'})$. Subsequently, the cumulative regret and simple regret after T iterations are defined as $R_T = \sum_{t=1}^{T} r_t$ and $S_T = \min_{t=1,\dots,T} r_t$ respectively. As a result, as long as we can show that R_T grows sub-linearly in T, then we can conclude that the average regret $\frac{R_T}{T}$ asymptotically goes to 0; therefore, S_T vanishes asymptotically since it is upper-bounded by the average regret: $S_T \leq \frac{R_T}{T}$. In contrast to the more commonly used definition of instantaneous regret: $r_t = f(\mathbf{z}^*) - f(\mathbf{z}_t)$, the slightly modified definition introduced here is justified in the sense that the induced definition of simple regrets, which is the ultimate goal of the theoretical analysis, obtained in both cases are equivalent, i.e., $\min_{t=1,\dots,T} f(\mathbf{z}^*) - f_t^* = \min_{t=1,\dots,T} f(\mathbf{z}^*) - f(\mathbf{z}_t)$.

The instantaneous regret defined above can be further decomposed as

$$\begin{aligned} r_t =& f(\mathbf{z}^*) - f_t^* = f(\mathbf{z}^*) - \max_{t'=1,\dots,t} f(\mathbf{z}_{t'}) \\ =& f(\mathbf{z}^*) - \max\{f_{t-1}^*, f(\mathbf{z}_t)\} \end{aligned} \tag{6}$$

Note that in our algorithm, the BO iterations can be divided into two types: **1)** t^+ such that $n_{t^+} = N$: those iterations that are not early-stopped; and **2)** t^- such that $n_{t^-} < N$: those that are early-stopped. For all t^+, it follows from Equation 6 that $r_t = f(\mathbf{z}^*) - \max\{f_{t-1}^*, f(\mathbf{z}_t)\} \leq f(\mathbf{z}^*) - f(\mathbf{z}_t) = f(\mathbf{z}^*) - f([\mathbf{x}_t, n_t]) = f(\mathbf{z}^*) - f([\mathbf{z}_t, N]) \triangleq r_{t^+}$; for all t^-, from Equation 6, we have that $r_t = f(\mathbf{z}^*) - \max\{f_{t-1}^*, f(\mathbf{z}_t)\} \leq f(\mathbf{z}^*) - f_{t-1}^* \triangleq r_{t^-}$. In the following, we will focus on the analysis of the sum of all r_{t^+} and all r_{t^-}: $R_T' = \sum_{t^+} r_{t^+} + \sum_{t^-} r_{t^-}$. As a result of the definition, R_T' is an upper bound of R_T, therefore, sub-linear growth of R_T' implies that R_T also grows sub-linearly.

Next, note that for all t^- such that $n_{t^-} < N$ (when $\mathbf{x}_t$ is early-stopped),

$$\begin{aligned} r_{t^-} = f(\mathbf{z}^*) - f_{t-1}^* &= \underline{f(\mathbf{z}^*) - f([\mathbf{x}_t, N])} + \underline{f([\mathbf{x}_t, N]) - f_{t-1}^*} \\ &\overset{(1)}{\leq} \underline{f(\mathbf{z}^*) - f([\mathbf{x}_t, n_t])} + \underline{f([\mathbf{x}_t, N]) - f_{t-1}^*} \end{aligned} \tag{7}$$

in which (1) results from Assumption 1. As a result, R_T' can be re-written as

$$\begin{aligned} R_T' &\overset{(1)}{=} \sum_{\{t|n_t=N\}} [f(\mathbf{z}^*) - f([\mathbf{x}_t, N])] + \sum_{\{t|n_t<N\}} [f(\mathbf{z}^*) - f_{t-1}^*] \\ &= \sum_{\{t|n_t=N\}} [f(\mathbf{z}^*) - f([\mathbf{x}_t, N])] + \sum_{\{t|n_t<N\}} [f(\mathbf{z}^*) - f([\mathbf{x}_t, N])] + \sum_{\{t|n_t<N\}} [f([\mathbf{x}_t, N]) - f_{t-1}^*] \\ &\overset{(2)}{\leq} \sum_{\{t|n_t=N\}} [f(\mathbf{z}^*) - f([\mathbf{x}_t, N])] + \sum_{\{t|n_t<N\}} [f(\mathbf{z}^*) - f([\mathbf{x}_t, n_t])] + \sum_{\{t|n_t<N\}} [f([\mathbf{x}_t, N]) - f_{t-1}^*] \\ &\overset{(3)}{=} \sum_{t=1}^{T} [f(\mathbf{z}^*) - f([\mathbf{x}_t, n_t])] + \sum_{\{t|n_t<N\}} [f([\mathbf{x}_t, N]) - f_{t-1}^*] \\ &\triangleq \sum_{t=1}^{T} r_{t,1} + \sum_{\{t|n_t<N\}} r_{t,2} \\ &\triangleq R_{T,1} + R_{T,2} \end{aligned} \tag{8}$$

in which (1) makes use of the definition of R_T', (2) results from Equation 7 and (3) follows by combining the first two terms on the previous line. The first term following (3) of Equation 8 is summed over all time steps, whereas the second term is only summed over those time steps that are early-stopped ($n_t < N$). As mentioned earlier, in the sequel, we will attempt to

prove an upper bound on the expected value of R'_T,

$$\mathbb{E}[R'_T] \leq \mathbb{E}[R_{T,1}] + \mathbb{E}[R_{T,2}] \tag{9}$$

in which the expectation is taken with respect to the posterior probabilities used in the BOS problems, corresponding to those iterations that are early-stopped: $\Pi_{t\in\{t'|t'=1,\ldots,T,n_{t'}<N\}}\Pr(f([\mathbf{x}_t,N]) > y^*_{t-1} - \xi_t|\mathbf{y}_{t,n_t})$. Note that the probability distributions are independent across all the early-stopped iterations, therefore, for each early-stopped iteration t, the expectations of both $r_{t,1}$ and $r_{t,2}$ are only taken over the specific distribution: $\Pr(f([\mathbf{x}_t,N]) > y^*_{t-1} - \xi_t|\mathbf{y}_{t,n_t})$; whereas for each not-early-stopped iteration t, $\mathbb{E}[r_{t,1}] = r_{t,1}$ (whereas $r_{t,2}$ is absent). In the next two sections, we will prove upper bounds on $\mathbb{E}[R_{T,1}]$ and $\mathbb{E}[R_{T,2}]$ respectively.

C.2. Upper Bound on $\mathbb{E}[R_{T,1}]$

In this section, we will upper-bound the term $\mathbb{E}[R_{T,1}]$. As mentioned in the main text, for simplicity, we will focus on the case in which the underlying domain $\mathcal{D}$ is discrete, i.e., $|\mathcal{D}| < \infty$. To begin with, we will need a supporting lemma showing a uniform upper bound over the entire domain.

Lemma 1. *Suppose that $\delta \in (0,1)$ and $\beta_t \triangleq 2\log(|\mathcal{D}|t^2\pi^2/6\delta)$. Then, with probability $\geq 1-\delta$*

$$|f(\mathbf{z}) - \mu_{t-1}(\mathbf{z})| \leq \beta_t^{1/2}\sigma_{t-1}(\mathbf{z}) \quad \forall \mathbf{z} \in \mathcal{D}, t \geq 1\ .$$

The proof of lemma 1 makes use of standard Gaussian tail bounds and a number of union bounds, and the proof is identical to the proof of lemma 5.1 in (Srinivas et al., 2010). The next supporting lemma makes use of the Lipschitz continuity of f to bound the differences between function values whose inputs only differ by the dimension corresponding to the number of training epochs.

Lemma 2. *Suppose that Assumption 2 holds and let $\delta' \in (0,1)$. Then, with probability $\geq 1-\delta'$,*

$$|f([\mathbf{x},N]) - f([\mathbf{x},n])| \leq Nb\sqrt{\log\frac{da}{\delta'}} \qquad \forall \mathbf{x}, n = 1,\ldots,N\ .$$

Proof. Let $\mathbf{z} = [\mathbf{x},n]$ denote the input to the objective function f. Assumption 2, together with a union bound over $j = 1,\ldots,d$, implies that with probability $\geq 1 - dae^{-(\frac{L}{b})^2}$,

$$|f(\mathbf{z}) - f(\mathbf{z}')| \leq L||\mathbf{z}-\mathbf{z}'||_1 \qquad \forall \mathbf{z} \in \mathcal{D}$$

Since $[\mathbf{x},N]$ and $[\mathbf{x},n]$ differ only by the dimension corresponding to the number of training epochs, we have that

$$|f([\mathbf{x},N]) - f([\mathbf{x},n])| \leq LN$$

Then, the lemma follows by letting $\delta' = dae^{-(\frac{L}{b})^2}$. □

The next lemma bounds $E[r_{t,1}]$ by the Gaussian process posterior standard deviation with some scaling constants.

Lemma 3. *Let $\delta, \delta' \in (0,1)$ and $\kappa \geq 1$ be the constant used in C2 in the BO-BOS algorithm. Then, at iteration t of the BO-BOS algorithm, we have that, with probability $\geq 1-\delta-\delta'$,*

$$\mathbb{E}[r_{t,1}] \leq 2\kappa\beta_t^{1/2}\sigma_{t-1}([\mathbf{x}_t,n_t]) + Nb\sqrt{\log\frac{da}{\delta'}}\mathbb{1}_{n_t<N}\ .$$

Proof. Firstly, with probability $\geq 1-\delta$,

$$f(\mathbf{z}^*) \overset{(1)}{=} f([\mathbf{x}^*,N]) \overset{(2)}{\leq} \mu_{t-1}([\mathbf{x}^*,N]) + \beta_t^{1/2}\sigma_{t-1}([\mathbf{x}^*,N]) \overset{(3)}{\leq} \mu_{t-1}([\mathbf{x}_t,N]) + \beta_t^{1/2}\sigma_{t-1}([\mathbf{x}_t,N]) \tag{10}$$

in which (1) follows from Assumption 1 which states that, for each $\mathbf{x}$, the function value is monotonically non-decreasing in the number of training epochs, which implies that at the (unknown) global maximum $\mathbf{z}^*$, the dimension corresponding to the

number of epochs is equal to N. (2) makes use of Lemma 1, whereas (3) is due to the way $\mathbf{x}_t$ is selected in the algorithm, i.e., $\mathbf{x}_t = \text{argmax}_{\mathbf{x}} \mu_{t-1}([\mathbf{x}, N]) + \sqrt{\beta_t}\sigma_{t-1}([\mathbf{x}, N])$. As a result, we have that with probability $\geq 1 - \delta - \delta'$

$$\begin{aligned}
\mathbb{E}[r_{t,1}] &= \mathbb{E}[f(\mathbf{z}^*) - f([\mathbf{x}_t, n_t])] \overset{(1)}{\leq} \mathbb{E}[\beta_t^{1/2}\sigma_{t-1}([\mathbf{x}_t, N]) + \mu_{t-1}([\mathbf{x}_t, N]) - f([\mathbf{x}_t, n_t])] \\
&= \mathbb{E}[\beta_t^{1/2}\sigma_{t-1}([\mathbf{x}_t, N]) + \mu_{t-1}([\mathbf{x}_t, N]) - f([\mathbf{x}_t, N]) + f([\mathbf{x}_t, N]) - f([\mathbf{x}_t, n_t])] \\
&\overset{(2)}{\leq} \mathbb{E}[2\beta_t^{1/2}\sigma_{t-1}([\mathbf{x}_t, N])] + \mathbb{E}[f([\mathbf{x}_t, N]) - f([\mathbf{x}_t, n_t])] \\
&\overset{(3)}{\leq} \mathbb{E}[2\beta_t^{1/2}\sigma_{t-1}([\mathbf{x}_t, N])] + Nb\sqrt{\log\frac{da}{\delta'}}\mathbb{1}_{n_t<N} \\
&\overset{(4)}{\leq} 2\beta_t^{1/2}\sigma_{t-1}([\mathbf{x}_t, N]) + Nb\sqrt{\log\frac{da}{\delta'}}\mathbb{1}_{n_t<N} \\
&\overset{(5)}{\leq} 2\kappa\beta_t^{1/2}\sigma_{t-1}([\mathbf{x}_t, n_t]) + Nb\sqrt{\log\frac{da}{\delta'}}\mathbb{1}_{n_t<N}
\end{aligned} \tag{11}$$

in which (1) follows from Equation 10, and (2) results from Lemma 1 and the linearity of the expectation operator. $\mathbb{1}_{n_t<N}$ in (3) is the indicator function, which takes the value of 1 if the event $n_t < N$ is true and 0 otherwise. (3) is obtained by analyzing two different cases separately: if $n_t = N$ ($\mathbf{x}_t$ is not early-stopped), then $\mathbb{E}[f([\mathbf{x}_t, N]) - f([\mathbf{x}_t, n_t])] = 0$; if $n_t < N$ (x_t is early-stopped), then $\mathbb{E}[f([\mathbf{x}_t, N]) - f([\mathbf{x}_t, n_t])] \leq \mathbb{E}[Nb\sqrt{\log\frac{da}{\delta'}}] = Nb\sqrt{\log\frac{da}{\delta'}}$ with probability $\geq 1 - \delta'$ following Lemma 2. (4) is due to the fact that $\sigma_{t-1}([\mathbf{x}_t, N])$ only depends on the observations up to step $t-1$ and is not dependent on the probability $\Pr(f([\mathbf{x}_t, N]) > y^*_{t-1} - \xi_t | \mathbf{y}_{t,n_t})$. (5) follows from the design of the algorithm; in particular, if $n_t < N$, then $\kappa\sigma_{t-1}([\mathbf{x}_t, n_t]) \geq \sigma_{t-1}([\mathbf{x}_t, N])$ is guaranteed by C2; otherwise, if $n_t = N$, then $\kappa\sigma_{t-1}([\mathbf{x}_t, n_t]) \geq \sigma_{t-1}([\mathbf{x}_t, n_t]) = \sigma_{t-1}([\mathbf{x}_t, N])$ since $\kappa \geq 1$. □

Subsequently, we can upper bound $\mathbb{E}[R_{T,1}] = \sum_{t=1}^{T}\mathbb{E}[r_{t,1}]$ by extensions of Lemma 5.3 and 5.4 from (Srinivas et al., 2010), which are presented here for completeness. The following lemma connects the information gain about the objective function with the posterior predictive variance, whose proof results from straightforward extension of Lemma 5.3 of (Srinivas et al., 2010).

Lemma 4. *Let $\mathbf{y}_T$ be a set of observations of size T, and let $\mathbf{f}_T$ be the corresponding function values. The information gain about $\mathbf{f}_T$ from observing $\mathbf{y}_T$ is*

$$I(\mathbf{y}_T; \mathbf{f}_T) = \frac{1}{2}\sum_{t=1}^{T}\log[1 + \sigma^{-2}\sigma^2_{t-1}([\mathbf{x}_t, n_t])] \,.$$

Next, we use the following lemma to bound the sum of the first term of the expected instantaneous regret as given in Lemma 3.

Lemma 5. *Let $\delta \in (0,1)$, $C_1 \triangleq \frac{8}{\log(1+\sigma^{-2})}$, $\beta_t \triangleq 2\log(|\mathcal{D}|t^2\pi^2/6\delta)$, and $\gamma_T \triangleq \max_{A\in\mathcal{D},|A|=T} I(\mathbf{y}_A; \mathbf{f}_A)$ is the maximum information gain about f from any subset of size T. Then,*

$$\sum_{t=1}^{T} 2\beta_t^{1/2}\sigma_{t-1}([\mathbf{x}_t, n_t]) \leq \sqrt{TC_1\beta_T I(\mathbf{y}_T; \mathbf{f}_T)} \leq \sqrt{TC_1\beta_T\gamma_T} \,.$$

Proof. Firstly, we have that

$$\begin{aligned}
(2\beta_t^{1/2}\sigma_{t-1}([\mathbf{x}_t, n_t]))^2 = 4\beta_t\sigma^2_{t-1}([\mathbf{x}_t, n_t]) &\overset{(1)}{\leq} 4\beta_T\sigma^2[\sigma^{-2}\sigma^2_{t-1}([\mathbf{x}_t, n_t])] \\
&\overset{(2)}{\leq} 4\beta_T\sigma^2\frac{\sigma^{-2}}{\log(1+\sigma^{-2})}\log[1 + \sigma^{-2}\sigma^2_{t-1}([\mathbf{x}_t, n_t])] \\
&\leq \beta_T\frac{8}{\log(1+\sigma^{-2})}\frac{1}{2}\log[1 + \sigma^{-2}\sigma^2_{t-1}([\mathbf{x}_t, n_t])]
\end{aligned} \tag{12}$$

in which (1) holds since β_t is monotonically increasing in t; (2) results from the fact that $\sigma^{-2}x \leq \frac{\sigma^{-2}}{\log(1+\sigma^{-2})}\log[1+\sigma^{-2}x]$ for $x \in (0,1]$, whereas $0 < \sigma^2_{t-1}([\mathbf{x}_t, n_t]) \leq 1$. Next, summing over $t = 1, \ldots, T$, we get

$$\begin{aligned}\sum_{t=1}^{T}(2\beta_t^{1/2}\sigma_{t-1}([\mathbf{x}_t,n_t]))^2 &\leq \beta_T \frac{8}{\log(1+\sigma^{-2})}\frac{1}{2}\sum_{t=1}^{T}\log[1+\sigma^{-2}\sigma^2_{t-1}([\mathbf{x}_t,n_t])] \\ &\overset{(1)}{=} \beta_T \frac{8}{\log(1+\sigma^{-2})} I(\mathbf{y}_T;\mathbf{f}_T) \overset{(2)}{\leq} C_1\beta_T\gamma_T\end{aligned} \tag{13}$$

in which (1) results from Lemma 4, and (2) follows from the definitions of C_1 and γ_T. Next, making use of the Cauchy-Schwarz inequality, we get

$$\begin{aligned}\sum_{t=1}^{T} 2\beta_t^{1/2}\sigma_{t-1}([\mathbf{x}_t,n_t]) &\leq \sqrt{T}\sqrt{\sum_{t=1}^{T}(2\beta_t^{1/2}\sigma_{t-1}([\mathbf{x}_t,n_t]))^2} \\ &\leq \sqrt{C_1 T\beta_T\gamma_T}\end{aligned} \tag{14}$$

which completes the proof. □

Next, putting everything together, we get the follow lemma on the upper bound on $\mathbb{E}[R_{T,1}]$.

Lemma 6. *Suppose that Assumptions 1 and 2 hold. Let $\delta, \delta' \in (0,1)$, $C_1 = 8/\log(1+\sigma^{-2})$, $\beta_t = 2\log(|\mathcal{D}|t^2\pi^2/6\delta)$, $\kappa \geq 1$ be the constant used in C2 in the BO-BOS algorithm, and $\gamma_T = \max_{A\in\mathcal{D},|A|=T} I(\mathbf{y}_A;\mathbf{f}_A)$. Let $\tau_T = \sum_{t=1}^{T}\mathbb{1}_{n_t<N}$ be the number of BO iterations in which early stopping happens from iterations 1 to T. Assume that f is a sample from a GP, and $y(\mathbf{z}) = f(\mathbf{z}) + \epsilon\, \forall z \in \mathcal{D}$ in which $\epsilon \sim N(0,\sigma^2)$. Then, with probability $\geq 1-\delta-\delta'$,*

$$\mathbb{E}[R_{T,1}] = \sum_{t=1}^{T}\mathbb{E}[r_{t,1}] \leq \kappa\sqrt{TC_1\beta_T\gamma_T} + Nb\sqrt{\log\frac{da}{\delta'}}\tau_T \quad \forall T \geq 1\,.$$

Proof.

$$\begin{aligned}\mathbb{E}[R_{T,1}] \overset{(1)}{=} \sum_{t=1}^{T}\mathbb{E}[r_{t,1}] &\overset{(2)}{\leq} \sum_{t=1}^{T}[2\kappa\beta_t^{1/2}\sigma_{t-1}([\mathbf{x}_t,n_t]) + Nb\sqrt{\log\frac{da}{\delta'}}\mathbb{1}_{n_t<N}] \\ &\overset{(3)}{\leq} \kappa\sqrt{TC_1\beta_T\gamma_T} + \sum_{t=1}^{T} Nb\sqrt{\log\frac{da}{\delta'}}\mathbb{1}_{n_t<N} \\ &= \kappa\sqrt{TC_1\beta_T\gamma_T} + Nb\sqrt{\log\frac{da}{\delta'}}\sum_{t=1}^{T}\mathbb{1}_{n_t<N} \\ &= \kappa\sqrt{TC_1\beta_T\gamma_T} + Nb\sqrt{\log\frac{da}{\delta'}}\tau_T\end{aligned} \tag{15}$$

in which (1) follows from the linearity of the expectation operator, (2) results from Lemma 3, and (3) follows from Lemma 5. □

C.3. Upper Bound on $\mathbb{E}[R_{T,2}]$

In this section, we prove an upper bound on $\mathbb{E}[R_{T,2}]$. A few supporting lemmas will be presented and proved first. To begin with, the next lemma derives the appropriate choice of the incumbent values used in the BOS problems in different iterations of the BO-BOS algorithm.

Lemma 7. *Let the objective function f be a sample from a GP and $y(\mathbf{z}) = f(\mathbf{z}) + \epsilon\, \forall \mathbf{z} \in \mathcal{D}$ in which $\epsilon \sim N(0,\sigma^2)$. Let $\delta'' \in (0,1)$. At iteration $t > 1$, define $f^*_{t-1} \triangleq \max_{t'=1,\ldots,t-1} f(\mathbf{z}_{t'})$ and $y^*_{t-1} \triangleq \max_{t'=1,\ldots,t-1} y_{t'}$; for iteration $t = 1$, define $f^*_0 \triangleq 0$ and $y^*_0 \triangleq 0$. Then with probability $\geq 1-\delta''$,*

$$f^*_{t-1} \geq y^*_{t-1} - \xi_t \quad \forall t \geq 1$$

in which

$$\xi_t = \sqrt{2\sigma^2 \log \frac{\pi^2 t^2 (t-1)}{6\delta''}} \quad \forall t > 1$$

and $\xi_1 = 0$.

Proof. The lemma trivially holds for $t = 1$. Assume we are at iteration $t > 1$ of the BO-BOS algorithm, and let $t' \in \{1, 2, \ldots, t-1\}$. Since $y_{t'} = f(\mathbf{z}_{t'}) + \epsilon$, in which $\epsilon \sim N(0, \sigma^2)$, we have that $y_{t'} \sim N(f(\mathbf{z}_{t'}), \sigma^2)$. Making use of the *upper deviation inequality* for Gaussian distribution and the definition of ξ_t, we get

$$\Pr[y_{t'} \geq f(\mathbf{z}_{t'}) + \xi_t] \leq e^{-\frac{{\xi_t}^2}{2\sigma^2}} = \frac{6\delta''}{\pi^2 t^2 (t-1)} \tag{16}$$

Denote the event that $\{\exists\, t' \in \{1, 2, \ldots, t-1\} \text{ s.t. } y_{t'} \geq f(\mathbf{z}_{t'}) + \xi_t\}$ as $\mathcal{A}_t$. Next, taking a union bound over the entire observation history $t' \in \{1, 2, \ldots, t-1\}$, we get

$$\begin{aligned} \Pr[\mathcal{A}_t] \leq & \sum_{t'=1}^{t-1} \Pr[y_{t'} \geq f(\mathbf{z}_{t'}) + \xi_t] \\ \leq & (t-1) \frac{6\delta''}{\pi^2 t^2 (t-1)} = \frac{6\delta''}{\pi^2 t^2} \end{aligned} \tag{17}$$

which implies that at iteration t, with probability $\geq 1 - \frac{6\delta''}{\pi^2 t^2}$, $y_{t'} - f(\mathbf{z}_{t'}) < \xi_t \, \forall t' \in \{1, 2, \ldots, t-1\}$, which further suggests that $y^*_{t-1} - f^*_{t-1} \leq \xi_t$ at iteration t. Next, taking a union bound over $t \geq 1$, we get

$$\Pr[\exists t \geq 1 \text{ s.t. } \mathcal{A}_t \text{ holds}] \leq \sum_{t \geq 1} \Pr[\mathcal{A}_t] \leq \sum_{t \geq 1} \frac{6\delta''}{\pi^2 t^2} = \delta'' \tag{18}$$

which suggests that, with probability $\geq 1 - \delta''$, $y^*_{t-1} - f^*_{t-1} \leq \xi_t \; \forall t \geq 1$, and thus completes the proof. □

The next lemma shows that, with appropriate choices of the incumbent value, the posterior probability used in Bayesian optimal stopping is upper-bounded.

Lemma 8. *If in iteration t of the BO-BOS algorithm, the BOS algorithm is run with the incumbent value $y^*_{t-1} - \gamma_t$ and the corresponding cost parameters K_1, K_2 and c_{d_0}, and the algorithm early-stops after $n_t < N$ epochs, then with probability $\geq 1 - \delta''$,*

$$Pr(f([\mathbf{x}_t, N]) > f^*_{t-1} | \mathbf{y}_{t,n_t}) \leq \frac{K_2 + c_{d_0}}{K_1} \quad \forall t \geq 1 \,. \tag{19}$$

Proof. Recall that when running the Bayesian optimal stopping algorithm in iteration t of BO-BOS, we only early-stop the experiment ($n_t < N$) when we can safely conclude that the performance of the currently evaluated hyperparameter $\mathbf{x}_t$ will end up having smaller (or equal) validation accuracy than the currently observed optimum offset by a noise correction term: $y^*_{t-1} - \xi_t$; i.e., when the expected loss of decision d_1 is the smallest among all decisions. Therefore, when the evaluation of $\mathbf{x}_t$ is early-stopped after $n_t < N$ epochs, we can conclude that

$$\begin{aligned} & K_1 \Pr(f([\mathbf{x}_t, N]) > y^*_{t-1} - \xi_t | \mathbf{y}_{t,n_t}) \\ \leq & \mathbb{E}_{y_{t,n_t+1} | \mathbf{y}_{t,n_t}} \Big[\min\{K_1 \Pr(f([\mathbf{x}_t, N]) > y^*_{t-1} - \xi_t | \mathbf{y}_{t,n_t+1}), K_2 \Pr(f([\mathbf{x}_t, N]) \leq y^*_{t-1} - \xi_t | \mathbf{y}_{t,n_t+1}), \\ & \mathbb{E}_{y_{t,n_t+2} | \mathbf{y}_{t,n_t+1}} [\rho_{t,n_t+2}(\mathbf{y}_{t,n_t+2})] + c_{d_0} \} \Big] + c_{d_0} \\ \leq & \mathbb{E}_{y_{t,n_t+1} | \mathbf{y}_{t,n_t}} [K_2 \Pr(f([\mathbf{x}_t, N]) \leq y^*_{t-1} - \xi_t | \mathbf{y}_{t,n_t+1})] + c_{d_0} \\ \leq & K_2 \mathbb{E}_{y_{t,n_t+1} | \mathbf{y}_{t,n_t}} [\Pr(f([\mathbf{x}_t, N]) \leq y^*_{t-1} - \xi_t | \mathbf{y}_{t,n_t+1})] + c_{d_0} \\ \leq & K_2 + c_{d_0} \end{aligned} \tag{20}$$

Equation 20, together with Lemma 7, implies that

$$\begin{aligned} \Pr(f([\mathbf{x}_t, N]) > f^*_{t-1} | \mathbf{y}_{t,n_t}) \leq & \Pr(f([\mathbf{x}_t, N]) > y^*_{t-1} - \xi_t | \mathbf{y}_{t,n_t}) \\ \leq & \frac{K_2 + c_{d_0}}{K_1} \end{aligned} \tag{21}$$

which holds uniformly for all $t \geq 1$ with probability $\geq 1 - \delta''$. □

Subsequently, we use the next Lemma to upper-bound $\mathbb{E}[R_T^2]$ by the BOS cost parameters. We set K_2 and c_{d_0} as constants, and use different values of K_1 in different iterations t of the BO-BOS algorithm, which is represented by $K_{1,t}$.

Lemma 9. *In iteration t of the BO-BOS algorithm, define $\frac{K_2+c_{d_0}}{K_{1,t}} \triangleq \eta_t$. Then, with probability $\geq 1 - \delta''$,*

$$\mathbb{E}[R_{T,2}] \leq \sum_{t=1}^{T} \eta_t \quad \forall T \geq 1 \,.$$

Proof. Recall that according to Assumption 1, the value of the objective function f is bounded in the range $[0, 1]$. In iteration t, assume we early-stop the evaluation of $\mathbf{x}_t$ after $n_t < N$ epochs, then

$$\mathbb{E}[f([\mathbf{x}_t, N]) - f^*_{t-1} | \mathbf{y}_{t,n_t}] \overset{(1)}{\leq} \mathbb{E}[\mathbb{1}_{f([\mathbf{x}_t,N])-f^*_{t-1}>0} | \mathbf{y}_{t,n_t}] = \Pr(f([\mathbf{x}_t, N]) > f^*_{t-1} | \mathbf{y}_{t,n_t}) \tag{22}$$

Step (1) in Equation 22 is because $x \leq \mathbb{1}_{x>0}\ \forall x \in [-1, 1]$ and substituting $x = f([\mathbf{x}_t, N]) - f^*_{t-1}$. As a result, with probability $\geq 1 - \delta''$

$$\begin{aligned} \mathbb{E}[R_{T,2}] \overset{(1)}{=} \sum_{\{t|n_t<N\}} \mathbb{E}[r_{t,2}] \overset{(2)}{=} \sum_{\{t|n_t<N\}} \mathbb{E}[f([\mathbf{x}_t, N]) - f^*_{t-1} | \mathbf{y}_{t,n_t}] \overset{(3)}{\leq} \sum_{\{t|n_t<N\}} \Pr(f([\mathbf{x}_t, N]) > f^*_{t-1} | \mathbf{y}_{t,n_t}) \\ \overset{(4)}{\leq} \sum_{\{t|n_t<N\}} \eta_t \leq \sum_{t=1}^{T} \eta_t \end{aligned} \tag{23}$$

in which (1) follows from the linearity of expectation, (2) holds because the Expectation of $r_{t,2}$ is taken over $\Pr(f([\mathbf{x}_t, N]) > y^*_{t-1} - \xi_t | \mathbf{y}_{t,n_t})$, (3) results from Equation 22, and (4) follows from Lemma 8. This completes the proof. □

C.4. Putting Things Together

In this section, we put everything from the previous two sections together to prove the main theorems.

C.4.1. Proof of Theorem 1

Theorem 1 can be proven by combining Lemmas 6 and 9, and making use of the fact that $S_T \leq \frac{R_T}{T}$.

C.4.2. Proof of Theorem 2

Below we analyze the asymptotic behavior of each of the three terms in the upper bound of $\mathbb{E}[S_T]$ in Theorem 1, which is re-presented here for ease of reference.

$$\mathbb{E}[S_T] \leq \frac{\kappa\sqrt{TC_1\beta_T\gamma_T}}{T} + \frac{\sum_{t=1}^{T}\eta_t}{T} + \frac{1}{T}Nb\sqrt{\log\frac{da}{\delta'}}\tau_T \,. \tag{24}$$

The first term in the upper bound of $\mathbb{E}[S_T]$ Firstly, the first term in the upper bound matches the upper bound on the simple regret of the GP-UCB algorithm (Srinivas et al., 2010) (up to the constant κ). The maximum information gain, γ_T, has been analyzed for a few of the commonly used kernels in GP (Srinivas et al., 2010). For example, for the Square Exponential kernel, $\gamma_T = O((\log T)^{d+1})$, whereas for the Matérn kernel with $\nu > 1$, $\gamma_T = O(T^{d(d+1)/(2\nu+d(d+1))}\log T)$. Plugging both expressions of γ_T into Theorem 1, together with the expression of β_T as given in Theorem 1, shows that both kernels lead to sub-linear growth of the term $\sqrt{TC_1\beta_T\gamma_T}$, which implies that the first term in the upper bound of $\mathbb{E}[S_T]$ asymptotically goes to 0.

The second term in the upper bound of $\mathbb{E}[S_T]$ Given that $K_{1,t}$ is an increasing sequence with $K_{1,1} \geq K_2 + c_{d_0}$, the series $\sum_{t=1}^{T}\eta_t = \sum_{t=1}^{T}\frac{K_2+c_{d_0}}{K_{1,t}}$ grows sub-linearly, thus making the second term in the upper bound of $\mathbb{E}[S_T]$ given in Theorem 1, $\frac{\sum_{t=1}^{T}\eta_t}{T}$, asymptotically go to 0.

The third term in the upper bound of $\mathbb{E}[S_T]$ Next, suppose that $K_{1,t}$ becomes $+\infty$ for the first time at iteration T_0. Since $K_{1,t}$ is a non-decreasing sequence, $K_{1,t} = +\infty$ for all $t \geq T_0$. Therefore, for $t \geq T_0$, decision d_1 will never be taken and the algorithm will never early-stop. In other words, $n_t = N$ for all $t \geq T_0$.

Therefore, we can conclude that $\tau_T \leq T_0$ for all $T \geq 1$. As a result, the last term in the upper bound on $\mathbb{E}[S_T]$ in Theorem 1 can be upper-bounded by

$$\frac{\tau_T}{T} Nb\sqrt{\log \frac{da}{\delta'}} \leq \frac{T_0 Nb\sqrt{\log \frac{da}{\delta'}}}{T} = O\left(\frac{1}{T}\right) \tag{25}$$

which asymptotically goes to 0 as T goes to $+\infty$, because the numerator term is a constant. Therefore, this term also asymptotically vanishes in the upper bound.

To summarize, if the BOS parameters are selected according to Theorem 2, we have that

$$\mathbb{E}[S_T] = O\left(\frac{\sqrt{T\beta_T\gamma_T}}{T} + \frac{\sum_{t=1}^{T}\eta_t}{T} + \frac{1}{T}\right) \tag{26}$$

and $\mathbb{E}[S_T]$ goes to zero asymptotically.

D. Additional Experimental Details

In each experiment, the same initializations (6 initial points if not further specified) are used for all BO-based methods: GP-UCB, BOCA, LC Prediction, and BO-BOS. The Square Exponential kernel is used for BOCA since the algorithm is only given for this kernel (Kandasamy et al., 2017), the other BO-based algorithms use the Matérn kernel; the kernel hyperparameters are updated by maximizing the Gaussian process marginal likelihood after every 10 BO iterations. In the BO-BOS algorithm, since the number of training epochs is an input to the GP surrogate function, some of the intermediate observations ($n < N$) can be used as additional input to GP to improve the modeling of the objective function. However, using the observation after every epoch as input leads to poor scalability. Therefore, for all experiments with $N = 50$ (which include most of the experiments), we use the observations after first, 10-th, 20-th, 30-th and 40-th epochs as additional inputs to the GP surrogate function; whereas for the RL experiment with $N = 100$ in section 5.3.1, we use the 1-th, 20-th, 40-th, 60-th and 80-th intermediate observations as additional inputs. 100,000 forward simulation samples are used for each BOS algorithm; the grid size of the discretized summary statistics is set to 100; for simplicity, the incumbent value at iteration t is chosen as $y^*_{t-1} = \max_{t'=1,\ldots,t-1} y_{t'}$, thus ignoring the observation noise. In the LC Prediction algorithm (Domhan et al., 2015), learning curve prediction is performed after every 2 epochs. In Hyperband (Li et al., 2017), the successive halving parameter η is set to 3 as recommended by the original authors, and the maximum number of epochs is set to $N = 80$ (we observed that setting $N = 80$ led to better performance than $N = 50$ since it allows the Hyperband algorithm to run for more epochs overall).

D.1. Hyperparameter Tuning for Logistic Regression

In the first set of experiments, we perform hyperparameter tuning for a simple ML model, logistic regression (LR). The LR model is trained using the MNIST image dataset, which consists of 70,000 images of the 10 digits, corresponding to a 10-class classification problem. Three hyperparameters are tuned: the batch size (20 to 500), L2 regularization parameter (10^{-6} to 1.0), and learning rate (10^{-3} to 0.1). We use 80% of the images as the training set and the remaining 20% as the validation set.

Some of the learning curves during a particular run of the BO-BOS algorithm is shown in Fig. 7. It can be observed that the learning curves that show minimal potential in achieving small validation errors are early-stopped, whereas the promising hyperparameter settings are run for larger number of epochs. The reliability of the early stopping achieved by the BO-BOS algorithm is demonstrated in Fig. 8. In this figure, the green triangles correspond to the learning curves that are not early-stopped ($n_t = N$), and the red circles represent the final validation errors (after training for the maximum number of epochs N) that *could have been* reached by the early-stopped learning curves ($n_t < N$). Note that the red circles are shown only for the purpose of illustration and are not observed in practice. As displayed in the figure, the early stopping decisions made during the BO-BOS algorithm are reliable, since those early-stopped learning curves all end up having large validation errors.

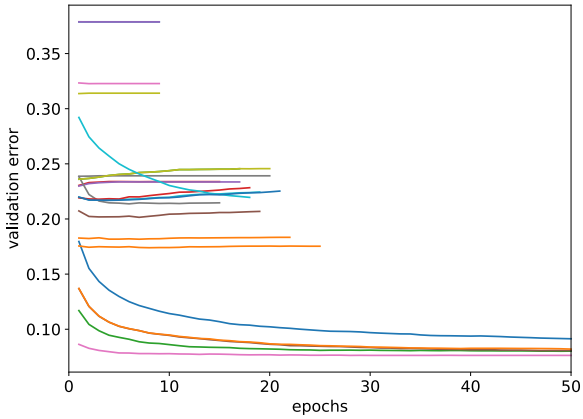


Figure 7. Some learning curves during the BO-BOS algorithm.

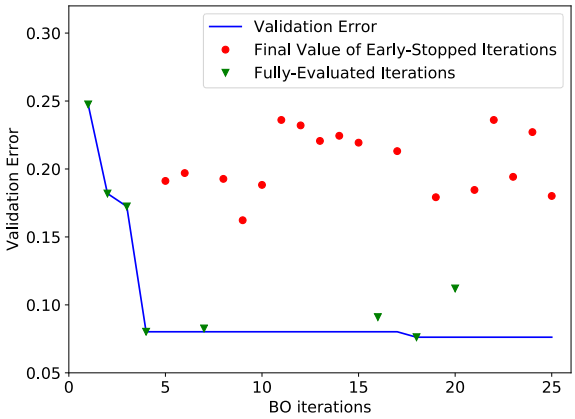


Figure 8. Illustration of the effectiveness of the early stopping decisions made during the BO-BOS algorithm.

D.2. Hyperparameter Tuning for Convolutional Neural Networks

Next, we tune the hyperparameters of convolutional neural networks (CNN) using the CIFAR-10 (Krizhevsky, 2009) and Street View House Numbers (SVHN) dataset (Netzer et al., 2011). Both tasks correspond to 10-class classification problems. For CIFAR-10, $50,000$ images are used as the training set and 10,000 images are used as the validation set; for SVHN, 73,257 and 26032 images are used as the training and validation sets respectively following the original dataset partition. The CNN model consists of three convolutional layers (each followed by a max-pooling layer) followed by one fully-connected layer. We tune six hyperparameters in both experiments: the batch size (32 to 512), learning rate (10^{-7} to 0.1), learning rate decay (10^{-7} to 10^{-3}), L2 regularization parameter (10^{-7} to 10^{-3}), the number of convolutional filters in each layer (128 to 256), and the number of units in the fully-connected layer (256 to 512). In addition to the results in Figure 2, the corresponding figures with standard error is presented below in Figures 9 and 10, which demonstrate the robustness of the performance advantages of BO-BOS.

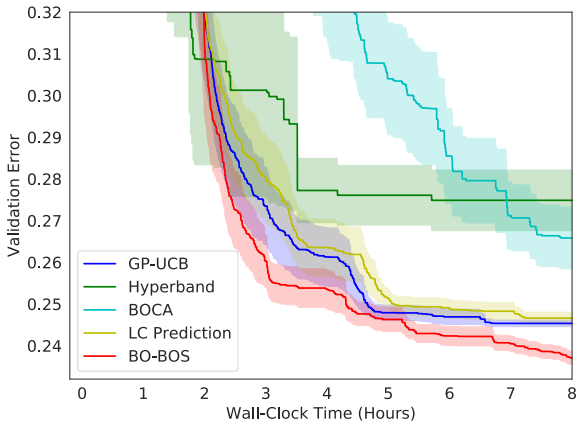


Figure 9. Best-found validation error of CNN v.s. run-time using the CIFAR-10 dataset, with standard error (averaged over 30 random initializations).

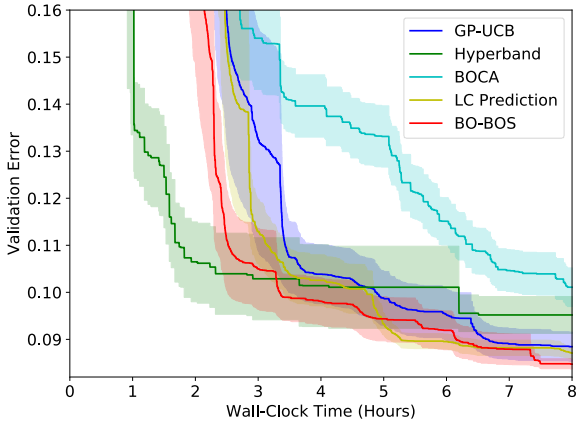
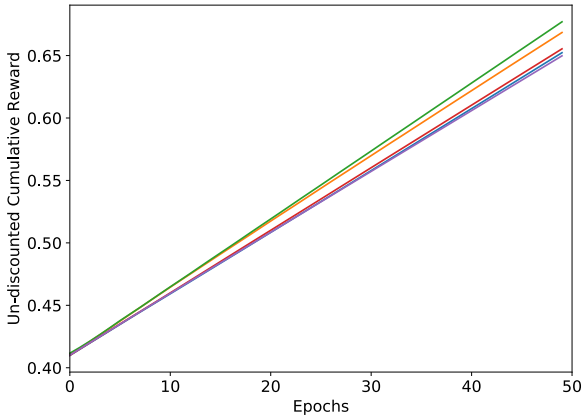


Figure 10. Best-found validation error of CNN v.s. run-time using the SVHN dataset, with standard error (averaged over 30 random initializations).

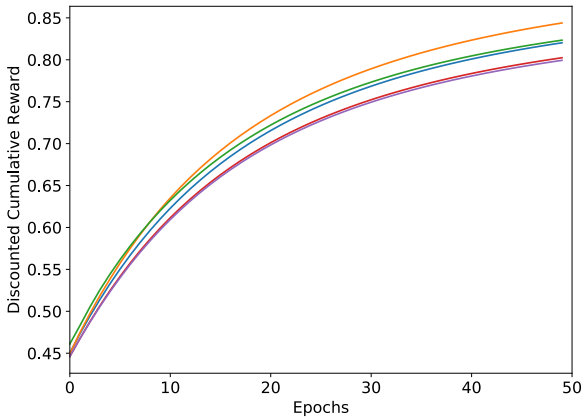
D.3. Policy Search for Reinforcement Learning

We apply our algorithm to a continuous control task: the Swimmer-v2 environment from OpenAI Gym, MuJoCo (Brockman et al., 2016; Todorov et al., 2012). The task involves controlling two joints of a swimming robot to make it swim forward as fast as possible. The state of the robot is represented by an 8-dimensional feature vector, and the action space is 2-dimensional corresponding to the two joints. We use a linear policy, in which the policy is represented by an 8×2 matrix that maps each state vector to the corresponding action vector. In this setting, the input parameters, $\mathbf{x}$, to the GP-UCB and BO-BOS algorithms are the 16 parameters of the policy matrix, and the objective function is the discounted cumulative rewards in an episode. Each episode of the task consists of 1,000 steps. We set N (the maximum number of epochs) to be smaller than $1,000$ by treating a fixed number of consecutive steps as one single epoch. E.g., we can set $N = 50$ or $N = 100$ by treating every 20 or 10 consecutive steps as one epoch respectively. The rewards are clipped, scaled, and normalized such that the discounted cumulative rewards of each episode is bounded in the range $[0, 1]$; for each evaluated policy, we also record the un-discounted and un-scaled cumulative rewards, which are the ultimate objective to be maximized and reported in Fig. 3 in the main text. Each policy evaluation consists of running 5 independent episodes with the given policy, and returning the average discounted cumulative rewards, i.e., average return, as the observed function value.

As mentioned in the main text, the rewards are discounted in order to make the objective function, the discounted cumulative rewards, resemble the learning curves of ML models, such that the BO-BOS algorithm can be naturally applied. This rationale is illustrated in Fig. 11, which plots some example un-discounted ($\gamma = 1.0$) and discounted ($\gamma = 0.9$) cumulative rewards respectively. The figures indicate that, compared with un-discounted cumulative rewards, discounted cumulative rewards bear significantly closer resemblance to the learning curves of ML models, thus supporting the claim made in the main text motivating the use of discounted rewards, as well as the experimental results shown in Fig. 3 (specifically, the poor performance of the curve corresponding to $N = 50$ and $\gamma = 1.0$). In addition to the results presented in the main text in section 5.3.1, we further present the results with standard errors in Fig. 12, to emphasize the significant performance advantage offered by BO-BOS compared with GP-UCB. To avoid clutter, we only present the results with error bar for GP-UCB with $\gamma = 1.0$ and BO-BOS with $N = 50$ and $\gamma = 0.9$, which are best-performing settings for GP-UCB and BO-BOS respectively.



(a) Un-discounted ($\gamma = 1.0$).



(b) Discounted ($\gamma = 0.9$).

Figure 11. Example curves of un-discounted and discounted cumulative rewards

D.4. Joint Hyperparameter Tuning and Feature Selection

In this set of experiments, we use the gradient boosting model (XGBoost (Chen & Guestrin, 2016)), tuning four hyperparameters: the learning rate (10^{-3} to 0.5), maximum depth of each decision tree (2 to 15), feature sub-sampling ratio for each tree (0.3 to 1.0), and L1 regularization parameter (0.0 to 5.0). We use the email spam dataset from the UCI Machine Learning

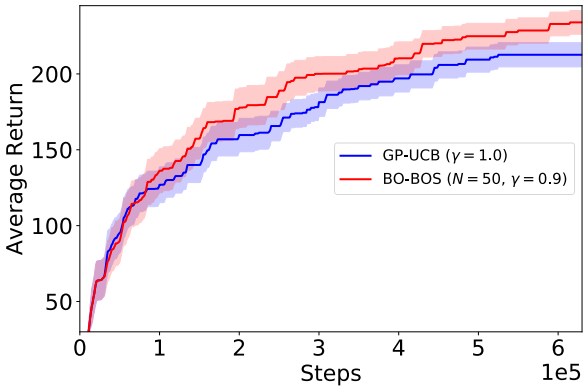


Figure 12. Best-found return (averaged over 5 episodes) v.s. the total number of steps of the robot in the environment (averaged over 30 random initializations) using the Swimmer-v2 task, with standard error.

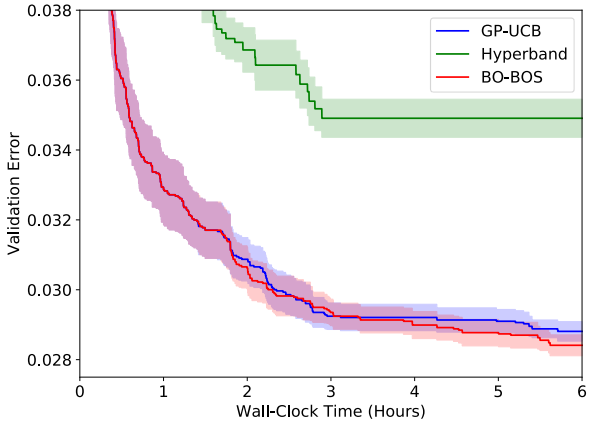


Figure 13. Best-found validation error of XGBoost v.s. run-time with standard error (averaged over 30 random initializations), obtained using joint hyperparameter tuning and feature selection.

Repository (Dheeru & Karra Taniskidou, 2017), which represents a binary classification problem: whether the email is a spam or not. We use 3065 emails as the training set and the remaining 1536 emails as the validation set; each email consists of 57 features. The maximum number of features for each hyperparameter setting is set as $N = 50$. In addition to Figure 4 in the main text, the same plot with error bar (standard error) is shown in Figure 13.