# SEQUENTIAL DESIGN FOR RANKING RESPONSE SURFACES

RUIMENG HU AND MIKE LUDKOVSKI
DEPARTMENT OF STATISTICS AND APPLIED PROBABILITY UNIVERSITY OF CALIFORNIA, SANTA BARBARA 93106-3110 HU@PSTAT.UCSB.EDU,LUDKOVSKI@PSTAT.UCSB.EDU

Abstract. We propose and analyze sequential design methods for the problem of ranking several response surfaces. Namely, given $L \geq 2$ response surfaces over a continuous input space $\mathcal{X}$, the aim is to efficiently find the index of the minimal response across the entire $\mathcal{X}$. The response surfaces are not known and have to be noisily sampled one-at-a-time. This setting is motivated by stochastic control applications and requires joint experimental design both in space and response-index dimensions. To generate sequential design heuristics we investigate stepwise uncertainty reduction approaches, as well as sampling based on posterior classification complexity. We also make connections between our continuous-input formulation and the discrete framework of pure regret in multi-armed bandits. To model the response surfaces we utilize kriging surrogates. Several numerical examples using both synthetic data and an epidemics control problem are provided to illustrate our approach and the efficacy of respective adaptive designs.

Keywords: sequential design, response surface modeling, kriging models, sequential uncertainty reduction, expected improvement

## 1. Introduction

Let $\mu_\ell : \mathcal{X} \to \mathbb{R}$, $\ell \in \mathfrak{L} \equiv \{1, 2, \ldots, L\}$ be $L$ smooth functions over a subset $\mathcal{X}$ of $\mathbb{R}^d$. We are interested in the problem of learning the resulting *ranking* of $\mu_\ell$ over the input space $\mathcal{X}$, namely finding the classifier

$$\mathcal{C}(x) := \arg\min_{\ell} \{\mu_\ell(x)\} \in \mathfrak{L}. \tag{1.1}$$

The functions $\mu_\ell$ are a priori unknown but can be noisily sampled. That is for any $x \in \mathcal{X}, \ell \in \mathfrak{L}$ we have access to a simulator $Y_\ell(x)$ which generates estimates of $\mu_\ell(x)$:

$$Y_\ell(x) = \mu_\ell(x) + \epsilon_\ell(x), \ \ell \in \mathfrak{L} \tag{1.2}$$

where $\epsilon_\ell$ are independent, mean zero random variables with variance $\sigma^2_\ell(x)$. Intuitively speaking, we have $L$ smooth hyper-surfaces on $\mathcal{X}$ that can be sampled via Monte Carlo (MC).

Our goal is to identify the minimal surface *globally* over the entire input space. More precisely, we seek to assign a label $\hat{\mathcal{C}}(x)$ to any $x \in \mathcal{X}$ that well-approximates the true $\mathcal{C}$ in terms of the loss function

$$\mathcal{L}(\hat{\mathcal{C}}, \mathcal{C}) := \int_{\mathcal{X}} \left\{ \mu_{\hat{\mathcal{C}}(x)}(x) - \mu_{\mathcal{C}(x)}(x) \right\} \, F(\mathrm{d}x), \tag{1.3}$$

where $F(\cdot)$ is a specified weight function on $\mathcal{X}$. Thus, the loss is zero if the ranking is correct $\hat{\mathcal{C}}(x) = \mathcal{C}(x)$, and otherwise is proportional to the (positive) difference between the selected response and the true minimum $\mu_{\hat{\mathcal{C}}} - \mu_{\mathcal{C}}$. The weights $F(x)$ come from the probability distribution that describes the likely inputs $x \in \mathcal{X}$, implying the relative importance of ranking different regions.

*Date*: Sept 1, 2015.

The loss function in (1.3) is a blend of the classical regression and classification objectives. In regression, the loss function is tied to a single response surface $\mu_\ell(x)$ and one seeks to estimate the response marginally. Instead, (1.3) is only about correctly identifying the index of the minimal response. As a result, small estimation errors are tolerated as long as the minimal response does not change, leading to a thresholding behavior in the loss function. In classification the loss function is discrete (typically with fixed mis-classification penalties), whereas (1.3) takes losses *proportional* to the mis-classification distance $\mu_{\hat{\mathcal{C}}(x)}(x) - \mu_{\mathcal{C}(x)}(x)$. A further key distinction is that in classification the sampling space is just $\mathcal{X}$ (returning a noisy label $C(x) \in \mathfrak{L}$), whereas in our context a sampling query consists of the *location-index* pair $(x, \ell) \in \mathcal{X} \times \mathfrak{L}$, sampling one response at a time. The question of which surface to sample requires separate analysis over $\mathfrak{L}$.

We assume that sampling is not inexpensive and sampling efficiency is important. Hence, we focus on the design problem of constructing efficient sampling strategies that can well-estimate $\mathcal{C}(x)$ while optimizing the number of MC samples needed. Because $\mu_\ell(x)$ are unknown, we frame (1.3) as a sequential learning problem of adaptively growing a design $\mathcal{Z}$ that quickly learns $\mathcal{C}(x)$. Classical approaches to design of experiments (DoE), recently reinterpreted in the context of design and analysis of computer experiments (DACE) construct static, i.e. response-independent, designs. However, for problems such as ranking, static designs are inadequate since the whole essence of the problem is to *learn* the structure of the unknown $\mu_\ell$'s and then optimize computational efforts. Intuitively, learning manifests itself in focusing the sampling efforts on the most promising regions in terms of the loss function. This implies discriminating both in input space $\mathcal{X}$ (focus on regions where identifying $\mathcal{C}(x)$ is difficult) and in sampling indices $\mathfrak{L}$ (focus on the surfaces where $\mu_\ell$ is likely to be the smallest response).

Due to the joint design space $\mathcal{X} \times \mathcal{L}$, our problem allows for a dual interpretation. Fixing $\ell$, (1.1) is about reconstructing an unknown response surface $x \mapsto \mu_\ell(x)$ through noisy samples. Aggregating the different response surfaces, sequential design over $\mathcal{X}$ reduces to identifying the boundary areas where the ranking $\mathcal{C}(x)$ changes values. Indeed, we observe that (1.1) is identical to the problem of partitioning $\mathcal{X} = \cup_{i=1}^{L} \mathcal{C}_i$ into the sets

$$\mathcal{C}_i := \{x : \mathcal{C}(x) = i\} = \{x : \mu_{\mathcal{C}(x)}(x) = \min_\ell \mu_\ell(x) = \mu_i(x)\}, \quad i = 1, \dots, L. \tag{1.4}$$

Because in interiors of the partitions $\mathcal{C}_i$ the ranking $\mathcal{C}(x)$ is easier to identify, the main problem is to find the partition boundaries $\partial\mathcal{C}_i$. As a result, (1.1) is related to the problem of contour-finding, for which sequential design was studied in [20, 40, 41]. Standard contour-finding attempts to identify the level set $\{\mu(x) = a\}$ of the response surface which is precisely the case $L = 2$ and known $\mu_2(x) = a$ in (1.1). Hence, the analysis herein can be viewed as a multi-variate extension of contour finding. In turn, contour-finding generalizes the classical objective of minimizing a noisy response, allowing for connections to the Expected Improvement/Efficient Global Optimization approaches in simulation optimization. In particular, we are inspired by the active learning rules of [12, 34] that trade-off the loss criterion against information gain.

Conversely, fixing $x$ the aim of determining the smallest response $\arg\min_\ell \mu_\ell(x)$ corresponds to the setting of multi-armed bandits (MAB). Indeed, we have a bandit with $L$ arms and corresponding payoffs $\mu_\ell(x), \ell \in \mathcal{L}$ with a decision-theoretic objective (1.1) which is known as the pure exploration problem in the bandit literature [6, 7]. Decision policies for which arm to pull are usually expressed in terms of posterior mean and confidence about the respective payoff; this point of view motivates our use of Gap-Upper Confidence Bound (UCB) design strategies [3, 44]. However, compared to

existing literature, (1.3) contains two key differences. First, the loss function is a weighted pure-regret criterion which to our knowledge has never been used in MAB context. Second, instead of a single bandit with independent arms, we face the radical extension to a continuum of bandits indexed by $x \in \mathcal{X}$, and where arm payoffs have a correlation structure imposed by $\mathcal{X}$. Recently, [24, 15] considered multiple bandits which can be viewed as (1.1) with a discrete, non-metrized $\mathcal{X}$. We generalize their setting to a structured continuous $\mathcal{X}$ using kriging response models.

1.1. **Summary of Approach.** To handle response surfaces indexed by the continuous space variable $x \in \mathcal{X}$ we adopt the framework of kriging or Gaussian process (GP) regression. Gaussian processes are a robust algorithmic framework that offers tractable quantification of posterior uncertainty and related sequential metrics. It has already been used extensively for sequential regression designs and allows an intuitive approach to borrow information cross-sectionally across samples to build global estimates of the entire surface $\mu_\ell$. In both contexts of DoE and continuous MAB's, kriging models have emerged as perhaps the most popular framework [45]. Moreover, GPs possess a wealth of analytic structure that allows for analytic evaluation of many Expected Improvement criteria. They also admit a natural transition between modeling of deterministic (noise-free) experiments where data needs to be interpolated, and stochastic simulators where data smoothing is additionally required. Moreover, while classical kriging may not be flexible enough for some challenging problems, there are now several well-developed generalizations, including treed GPs [22], local GPs [18], and particle-based GPs [21], all offering off-the-shelf use through public `R` packages. We stress that kriging is not essential to implementation of our algorithms; for example competitive alternatives are available among tree-based models, such as dynamic trees [23] and Bayesian trees [11].

Following the Efficient Global Optimization approach [27], we define expected improvement scores that blend together the local complexity of the ranking problem and the posterior variance of our estimates. In particular, we rely on the expected *reduction* in posterior variance and borrow from the Stepwise Uncertainty Reduction criteria based on GP regression from [39, 9]. We also utilize UCB-type heuristics [3] to trade-off exploration and exploitation objectives. Based on the above ideas, we obtain a number of fully sequential procedures that specifically target efficient learning of $\mathcal{C}(x)$ over the entire design space $\mathcal{X}$. Extensive numerical experiments are conducted to compare these proposals and identify the most promising solutions.

As explained, our algorithms are driven by the exploration-exploitation paradigm quantified in terms of (empirically estimated) local ranking complexity for $\mathcal{C}(x)$ and confidence in the estimated $\hat{\mathcal{C}}$. To quantify the local ranking complexity, we use the *gaps* $\Delta(x)$ [15, 8, 26]. For any $x \in \mathcal{X}$, denote by $\mu_{(1)}(x) < \mu_{(2)}(x) < \ldots < \mu_{(L)}(x)$ the ranked responses at $x$ and by

$$\Delta(x) := \mu_{(1)}(x) - \mu_{(2)}(x)$$

the gap between the best (smallest) and second-best response. $\Delta(x)$ measures the difficulty in ascertaining $\mathcal{C}(x)$: for locations where $\mu_{(1)} - \mu_{(2)}$ is *big*, we do not need high fidelity, since the respective minimal response surface is easy to identify; conversely for locations where $\mu_{(1)} - \mu_{(2)}$ is small we need more precision. Accordingly, we wish to preferentially sample where $\Delta(x)$ is small. This is operationalized by considering the estimated gaps $\widehat{\Delta}(x)$, which drives the design decisions over $\mathcal{X}$.

In terms of design over $\mathfrak{L}$, exploration suggests to spend the budget on learning the responses offering the biggest information gain. This intuition demonstrates that there will be substantial benefits to discriminating over the sampling indices $\ell$. To wit, one can locally concentrate on the

(two) most promising surfaces $\mu_{(1)}, \mu_{(2)}$, ignoring the rest. As we demonstrate, this is much more efficient than the naive strategy of sampling each $Y_\ell$ equally. In addition, since the noise level in $Y_\ell$ may vary with $\ell$ this must also be taken into account. Summarizing, our expected improvement metrics blend empirical gaps $\widehat{\Delta}$ and empirical posterior uncertainty based on kriging variance $\delta_\ell(x)$, jointly discriminating across $\mathcal{X} \times \mathfrak{L}$.

Our contributions can be traced along three directions. First, we introduce and analyze a novel sequential design problem targeting the loss function (1.3). As explained in Section 1.2 below, this setting is motivated by dynamic programming algorithms where statistical response models have been widely applied since the late 1990s [13, 31]. Here we contribute to this literature by proposing a sequential design framework that generates substantial computational savings. This aspect becomes especially crucial in complex models where simulation is expensive and forms the main computational bottleneck. Second, we generalize the existing literature on Bayesian optimization and contour-finding to the multi-surface setting, which necessitates constructing new EI measures that address joint design in space and index dimensions. In particular, we demonstrate that this joint design allows for a double efficiency gain: both in $\mathcal{X}$ and in $\mathcal{L}$. Third, we extend the multiple bandits problem of [15] to the case of a continuum of bandits, which requires building a full metamodel for the respective arm payoffs. Our construction offers an alternative to the recent work [7] on $\mathcal{X}$-armed bandits (a single bandit with a continuum of arms) and opens new vistas regarding links between MAB and DACE.

Our approach also generalizes Gramacy and Ludkovski [20]. The latter work proposed sequential design for the case $L = 2$ and $\sigma_2(x) = 0$, i.e. the contour-finding case. This allows to reduce the design heuristics to be just over the input space $\mathcal{X}$. In that context [20] introduced several EI heuristics and suggested the use of dynamic trees for the response surface modeling. The framework herein however requires a rather different approach, in particular we emphasize the bandit-inspired tools (such as UCB) that arise with simultaneous modeling of multiple response surfaces.

The rest of the paper is organized as follows. Section 2 describes the kriging response surface methodology that we employ, as well as some analytic formulas helpful in the context of ranking. Section 3 then develops the expected improvement heuristics for (1.1). Sections 4 and 5 illustrate the designed algorithms using synthetic data (where ground truth is known), and a case-study from epidemic management, respectively. Finally, Section 6 concludes.

1.2. **Motivation.** The setting of (1.1) comes from the stochastic control context, where $\mu_\ell$'s correspond to Q-values for different actions $\ell \in \mathfrak{L}$ and noisy samples $Y_\ell(x)$ are pathwise rewards from simulated scenarios. Consider the control objective of minimizing total costs associated with a controlled state process $X$,

$$c(0; u_{0:T}) = \sum_{t=0}^{T} g(t, X_t, u_t) \tag{1.5}$$

on the horizon $\{0, 1, \dots, T\}$. Above $g(t, x, u)$ encodes the stagewise running costs, $u_{0:T}$ is the control strategy taking values in the finite action space $u_t \in \mathfrak{L}$, and $X_t \equiv X_t^u$ is a stochastic discrete-time Markov state process taking values in the state space $\mathcal{X} \subseteq \mathbb{R}^d$. The dynamics of $X^u$ are assumed to be of the form

$$X_{t+1}^u = F(X_t, u_t, \xi_{t+1})$$

for some map $F : \mathcal{X} \times \mathfrak{L} \times \mathbb{R} \to \mathcal{X}$, where $\xi_{t+1}$ is a random independent centered noise source. Since from the point of view at 0, $c(0; u_{0:T})$ is a random variable, the performance criterion is based on

*expected* rewards and consists of optimizing the value function $V(0,x)$ defined as

$$V(t,x) := \inf_{u_{t:T}\in\mathcal{U}} \mathbb{E}[c(t;u_{t:T})|X_t = x], \qquad t \in \{0,1,\ldots,T\}, x \in \mathcal{X},$$

over all admissible closed-loop Markov strategies $u_{t:T} \in \mathcal{U}$. Thus, at time $t$, the action $u_t \equiv u(t,X_t)$ is specified in feedback form as a function of current state $X_t$. Hence, the control $u$ is determined dynamically as $X$ evolves. The policy map $(t,x) \mapsto u^*(t,x)$ translates system states into actions and is related to the value function via the dynamic programming equation (DPE):

$$V(t,x) = \min_{u\in\mathfrak{L}} \left\{ g(t,x,u) + \mathbb{E}_t \left[ V(t+1, X^u_{t+1}) \right](x) \right\} = \mu_{u^*}(x;t), \tag{1.6}$$

$$\text{with} \qquad \mu_u(x;t) := g(t,x,u) + \mathbb{E}_t[V(t+1,X^u_{t+1})](x). \tag{1.7}$$

The notation $\mathbb{E}_t[\cdot](x) \equiv \mathbb{E}[\cdot|X_t = x]$ is meant to emphasize averaging of the stochastic future at $t+1$ based on time-$t$ information summarized by the system state $X_t = x$. The term $\mu_u(x;t)$ is known as the Q-value associated with action $u$ and provides the expected cost-to-go if one applies $u \in \mathcal{L}$ at $X_t = x$.

To solve the DPE it suffices to compute the Q-values since by (1.6), $V(t,x) = \min_{\ell\in\mathfrak{L}}\{\mu_\ell(x;t)\}$. The above provides the first link to the ranking problem in (1.1). Thus, we identify the $\mathfrak{L}$ response surfaces with expected costs-to-go for each immediate action $\ell \in \mathfrak{L}$ and the ranking problem corresponds to constructing the policy map $x \mapsto u^*(t,x)$, which partitions the state space $\mathcal{X}$ into $L$ action sets $\mathcal{C}_i(t)$. The second link is based on the modified backward induction method: given $u^*(s,\cdot)$ for all $s = t+1,\ldots,T$ and all $x \in \mathcal{X}$ (initialized via $V(T,x) = g(T,x)$), we observe that

$$\mu_u(x;t) = g(t,x,u) + \mathbb{E}_t \left[ \sum_{n=t+1}^{T} g(n, X^{\widetilde{u}}_n, \widetilde{u}_n) \right](x), \tag{1.8}$$

where $(\widetilde{u}_t)$ is a strategy that uses action $u$ at $t$ and $u^*(s,X_s)$ thereafter, $s > t$. This formulation allows pursuit of *policy search* methods by tying the accuracy in (1.7) not to the immediate fidelity of (estimated) Q-values $\mu_u(\cdot;t)$, but to the quality of the policy map $u^*(t,x)$. Namely, one iteratively computes approximate policy maps $\hat{u}(s,\cdot)$ for $s = T-1, T-2, \ldots$, using (1.8) to construct $\hat{u}(t,\cdot)$ based on $\{\hat{u}(s,\cdot) : s > t\}$. As a result, rather than requiring a uniform approximation quality on the entire $\mathcal{X}$ for $\mu_u(\cdot;t)$, it becomes sufficient to optimize the fidelity of the ranking map $x \mapsto \mathcal{C}(x;t)$, i.e the relative ranking of each Q-value across possible actions $\ell \in \mathfrak{L}$. Wrong decisions about $\mathcal{C}$ lead to the loss (1.3), namely the difference between acting optimally as $u^*(t,X_t)$ at $t$, vis-a-vis taking action $\ell$ (and then acting optimally for the rest of the future, $\{t+1,\ldots,T\}$), weighted by the distribution $F(dx)$ of $X_t$. Note that the original objective of finding $V(0,x)$ requires solving $T$ ranking problems of the form (1.1).

The above approach to dynamic programming is especially attractive when the action space $\mathfrak{L}$ is very small. A canonical example are optimal stopping problems where the action space is $\mathfrak{L} = \{\text{stop}, \text{continue}\}$, i.e. $L = 2$. In the classical setting there is a single stopping decision and the immediate reward $\mu_2(x;t)$ is known directly, leading to the case of estimating a single Q-value $\mu_1(x;t)$, see [20]. Multiple stopping problems where both $\mu_1$ and $\mu_2$ need to be estimated arise in the pricing of swing options [36], valuing of real options [1], and optimizing entry-exit trading strategies [46]. The case $L > 2$ was considered for valuation of energy assets, especially gas storage [29], that lead to optimal switching problems. For example, storage decisions are usually modeled in terms of the triple alternative $L = 3$ of $\{inject, do-nothing, withdraw\}$. Small action spaces also arise in many engineering settings, such as target tracking [2, 25], and sensor management [14].

The above references highlight the wide range of control applications of (1.1) and will be further explored in a separate forthcoming article.

## 2. Statistical Model

2.1. **Sequential Design.** Fix a configuration $\{\mu_\ell, \ell = 1, \dots, L\}$ and corresponding classifier $\mathcal{C}(\cdot)$. A design of size $K$ is a collection $\mathcal{Z}^{(K)} := (x, \ell)^{1:K}$, $x \in \mathcal{X}, \ell \in \mathfrak{L}$, with superscripts denoting vectors. Fixing $\mathcal{Z}^{(K)}$, and conditioning on the corresponding samples $Y^{1:K} \equiv (Y_{\ell^k}(x^k))_{k=1}^K$, let $\hat{\mathcal{C}}^{(K)} \equiv \hat{\mathcal{C}}(Y^{1:K}, \mathcal{Z}^{(K)})$ be an estimate of $\mathcal{C}$. We aim to minimize the expected loss $\mathcal{L}(\hat{\mathcal{C}}(\cdot, \mathcal{Z}^{(K)}), \mathcal{C})$ over all designs of size $K$, i.e.

$$\inf_{\mathcal{Z}:|\mathcal{Z}|=K} \mathbb{E}\left[\mathcal{L}(\hat{\mathcal{C}}(Y^{1:K}, \mathcal{Z}), \mathcal{C})\right], \tag{2.1}$$

where the expectation is over the sampled responses $Y^{1:K}$. To tackle (2.1) we are interested in sequential algorithms that iteratively augment the designs $\mathcal{Z}$ as $Y$-samples are collected. The interim designs $\mathcal{Z}^{(k)}$ are accordingly indexed by their size $k$ where $k = K_0, K_0 + 1, \dots, K$. At each step, a new location $(x^{k+1}, \ell^{k+1})$ is added and the estimate $\hat{\mathcal{C}}^{(k+1)}$ is recomputed based on the newly obtained information. The overall procedure is summarized by the following pseudo-code:

(1) Initialize $\mathcal{Z}^{(K_0)}$ and $\hat{\mathcal{C}}^{(K_0)}$
(2) LOOP for $k = K_0, \dots$
  (a) Select a new location $(x^{k+1}, \ell^{k+1})$ and sample corresponding $y^{k+1} := Y_{\ell^{k+1}}(x^{k+1})$
  (b) Augment the design $\mathcal{Z}^{(k+1)} = \mathcal{Z}^{(k)} \cup \{(x^{k+1}, \ell^{k+1})\}$
  (c) Update the classifier $\hat{\mathcal{C}}^{(k+1)} = \hat{\mathcal{C}}(Y^{1:(k+1)}, \mathcal{Z}^{(k+1)})$ by assimilating the new observation
(3) END Loop

The basic greedy sampling algorithm adds locations with the aim of minimizing the myopic expected estimation error. More precisely, at step $k$, given design $\mathcal{Z}^{(k)}$ (and corresponding $Y^{1:k}$), the next pair $\left(x^{k+1}, \ell^{k+1}\right)$ is chosen by

$$\arg\inf_{(x^{k+1}, \ell^{k+1}) \in \mathcal{X} \times \mathfrak{L}} \mathbb{E}\left[\mathcal{L}(\hat{\mathcal{C}}(Y^{1:(k+1)}, \mathcal{Z}^{(k+1)}), \mathcal{C})\right], \tag{2.2}$$

where the expectation is over the next sample $Y_{\ell^{k+1}}(x^{k+1})$. This leads to a simpler one-step-ahead optimization compared to the $K$-dimensional (and typically we are looking at $K \gg 100$) formulation in (2.1). Unfortunately, the optimization in (2.2) is still generally intractable because it requires

- re-computing the full loss function $\mathcal{L}(\cdot, \mathcal{C})$ at each step;
- finding the expected change in $\hat{\mathcal{C}}$ given $Y_{\ell^{k+1}}(x^{k+1})$;
- integrating over the (usually unknown) distribution of $Y_{\ell^{k+1}}(x^{k+1})$;
- optimizing over the full $d+1$-dimensional design space $\mathcal{X} \times \mathfrak{L}$.

In this article we accordingly propose efficient numerical approximations to (2.2). Construction of these approximations relies on the twin ideas of (i) sequential statistical modeling (i.e. computing and updating $\hat{\mathcal{C}}$ as $\mathcal{Z}$ grows), and (ii) stochastic optimization (i.e. identifying promising new design sites $(x, \ell)$).

2.2. **Response Surface Modeling.** A key aspect of sequential design is adaptive assessment of approximation quality in order to maximize information gain from new samples. Consequently, measuring predictive uncertainty is a central tool in picking $(x^{k+1}, \ell^{k+1})$. For that purpose, we use a Bayesian paradigm, treating $\mu_\ell$ as random objects. Hence, we work with a function space $\mathcal{M}$ and assume that $\mu_\ell \in \mathcal{M}$ with some prior distribution $\mathcal{F}_0$. Thus, for each $x$, $\mu_\ell(x)$ is a random

variable whose posterior distribution is updated based on the collected information from samples $(x, \ell, y_\ell(x))$. We assume that the sample noise $\epsilon_\ell$ is independent of $\mu_\ell$. Let $\mathcal{F}_k$ be the information generated by the $k^{th}$ step design $\mathcal{Z}^{(k)}$, i.e. $\mathcal{F}_k = \sigma\left\{Y_\ell(x) : (x,\ell) \in \mathcal{Z}^{(k)}\right\}$. Given the information summarized in $\mathcal{F}_k$, we define the posterior $M_\ell^{(k)}(x) \sim \mu_\ell(x)|\mathcal{F}_k$. The random variable $M_\ell^{(k)}(x)$ is the belief about $\mu_\ell(x)$ conditional on $\mathcal{F}_k$; its two first moments are referred to as the kriging mean and variance respectively,

$$\widehat{\mu}_\ell^{(k)}(x) := \mathbb{E}[\mu_\ell(x)|\mathcal{F}_k], \tag{2.3}$$

$$\delta_\ell^{(k)}(x)^2 := \mathbb{E}[(\mu_\ell(x) - \widehat{\mu}_\ell^{(k)}(x))^2|\mathcal{F}_k]. \tag{2.4}$$

We will use $\widehat{\mu}(x)$ as a point estimate of $\mu_\ell(x)$, and $\delta_\ell(x)$ as a basic measure of respective uncertainty. The overall global map $x \mapsto M_\ell^{(k)}(x)$ is called the $\ell^{th}$ kriging surface. Note that while there is a spatial correlation structure over $\mathcal{X}$, we assume that observations are independent across $\mathfrak{L}$, so that the posteriors $M_\ell^{(k)}(x)$, $\ell = 1, 2, \ldots$ are independent.

**Remark 2.1.** *Note that beyond the kriging surface, many (Bayesian) regression algorithms also generate the full predictive distribution $\widetilde{Y}_\ell(x)$, i.e. the posterior of $Y_\ell(x)$ which requires additionally an estimate of the noise distribution $\widehat{\epsilon}_\ell(x)$,*

$$\widetilde{Y}_\ell(x)|\mathcal{F}_k \stackrel{d}{=} M_\ell^{(k)}(x) + \widehat{\epsilon}_\ell(x),$$

*where $\widehat{\epsilon}_\ell(x)$ has mean zero and posterior variance $\widehat{\sigma}_\ell^2(x)$. Since we are only interested in the mean response $\mu_\ell(x)$ we do not discuss $\widetilde{Y}_\ell(x)$ in the sequel.*

To describe the ranking of the surfaces, we use the order statistics $\widehat{\mu}_{(1)}(x) \le \widehat{\mu}_{(2)}(x) \le \ldots$ to denote the sorted posterior means at a fixed $x$. A natural definition is to announce the minimum estimated surface

$$\hat{\mathcal{C}}(x) := \arg\min_\ell \left\{\widehat{\mu}_\ell(x)\right\}, \tag{2.5}$$

i.e. the estimated classifier $\hat{\mathcal{C}}$ corresponds to the smallest posterior mean, so that $\widehat{\mu}_{\hat{\mathcal{C}}(x)}(x) = \widehat{\mu}_{(1)}(x)$. On the other hand, the uncertainty about $\mathcal{C}(x)$ can be summarized through the expected minimum of the posteriors $M_1, M_2, \ldots, M_L$,

$$m^{(k)}(x) := \mathbb{E}[M_{(1)}^{(k)}] = \mathbb{E}[\min(\mu_1(x), \ldots, \mu_L(x))|\mathcal{F}_k]. \tag{2.6}$$

Observe that $\mathbb{E}[\min_\ell \mu_\ell(x)|\mathcal{F}_k] = m^{(k)}(x) \le \widehat{\mu}_{(1)}^{(k)} = \min_\ell \mathbb{E}[\mu_\ell(x)|\mathcal{F}_k]$, and we accordingly define the M-gap

$$\mathcal{M}(x) := \widehat{\mu}_{(1)}(x) - m(x) \ge 0, \tag{2.7}$$

The M-gap measures the difference between expectation of the minimum and the minimum expected response, which precisely corresponds to the Bayesian expected loss at $x$ in (1.3). This fact offers an empirical analogue $\mathcal{EL}(\hat{\mathcal{C}})$ of the original loss function $\mathcal{L}(\hat{\mathcal{C}}, \mathcal{C})$ in (1.3),

$$\mathcal{EL}(\hat{\mathcal{C}}) := \int_{\mathcal{X}} \mathcal{M}(x) F(dx). \tag{2.8}$$

The above formula translates the local accuracy of the kriging surface into a global measure of fidelity of the resulting classifier $\hat{\mathcal{C}}$ and will be the main performance measure for our algorithms.

2.3. **Kriging.** The response surfaces are assumed to be smooth in $\mathcal{X}$. As a result, information about $\mu_\ell(x')$ is also revealing about $\mu_\ell(x)$ for $x \neq x'$, coupling observations at different sites. To enforce such conditions without a parametric representation, we use the Reproducing Kernel Hilbert Space (RKHS) approach to view each $\mu_\ell$ as a sample from a Gaussian process (GP). A GP is specified by its trend or mean function $t_\ell(x) = \mathbb{E}[\mu_\ell(x)]$ and a covariance structure $\mathcal{K}_\ell : \mathcal{X}^2 \to \mathbb{R}$, with $\mathcal{K}_\ell(x, x') = \mathbb{E}[(\mu_\ell(x) - t_\ell(x))(\mu_\ell(x') - t_\ell(x'))]$. The resulting function class $\mathrm{span}(\mathcal{K}(\cdot, x'), x' \in \mathcal{X})$ forms a Hilbert space and is denoted by $\mathcal{M}_\mathcal{K}$. By specifying the correlation behavior, the kernel $\mathcal{K}$ encodes the smoothness of the response surfaces drawn from the GP. We postulate that $\mu_\ell \in \mathcal{M}_\mathcal{K}$; it then follows from the reproducing property that both the prior and posterior distributions of $\mu_\ell(\cdot)$ are multivariate Gaussian.

Fix the response surface index $\ell$ and let $\vec{y} = (y(x^1), \dots, y(x^n))^T$ denote the observed noisy samples at locations $\vec{x} = x^{1:n}$. These realizations are modeled as in (1.2) with the response represented as

$$\mu_\ell(x) = t_\ell(x) + Z_\ell(x),$$

where $t_\ell(\cdot)$ is a fixed trend term and $Z_\ell(\cdot)$ is a realization of a Gaussian process. Given the samples $(x, y)^{1:n}$, the posterior of $\mu_\ell$ again forms a GP; in other words any collection $M_\ell^{(n)}(x'_1), \dots, M_\ell^{(n)}(x'_k)$ is multivariate Gaussian with mean $\widehat{\mu}_\ell^{(n)}(x'_i)$, covariance $v_\ell^{(n)}(x'_i, x'_j)$, and variance $\delta_\ell^{(n)}(x'_i)^2$, specified by [45, Sec. 2.7]:

$$\widehat{\mu}_\ell^{(n)}(x'_i) = t_\ell(x'_i) + \vec{k}_\ell^{(n)}(x'_i)^T (\mathbf{K}_\ell + \mathbf{\Sigma}_\ell^{(n)})^{-1} (\vec{y} - \vec{t}_\ell^{(n)}) \tag{2.9}$$

$$v_\ell^{(n)}(x'_i, x'_j) = \mathcal{K}_\ell(x'_i, x'_j) - \vec{k}_\ell^{(n)}(x'_i)^T (\mathbf{K}_\ell + \mathbf{\Sigma}_\ell^{(n)})^{-1} \vec{k}_\ell^{(n)}(x'_j) \tag{2.10}$$

with

$$\delta_\ell^{(n)}(x'_i)^2 = v_\ell^{(n)}(x'_i, x'_i) \qquad \vec{t}_\ell^{(n)} = (t_\ell(x^1), \dots, t_\ell(x^n))^T, \quad \text{and}$$
$$\vec{k}_\ell^{(n)}(x'_i) = (\mathcal{K}_\ell(x^1, x'_i), \dots, \mathcal{K}_\ell(x^n, x'_i))^T$$

and where $\mathbf{\Sigma}_\ell^{(n)} := \mathrm{diag}(\sigma_\ell^2(x^1), \dots, \sigma_\ell^2(x^n))$ and $\mathbf{K}_\ell$ is the $n \times n$ positive definite matrix $(\mathbf{K}_\ell)_{i,j} := \mathcal{K}_\ell(x^i, x^j)$, $1 \le i, j \le n$. Each posterior $M_\ell(x)$ is Gaussian and by independence across $\ell$, the vector of posteriors $\boldsymbol{M(x)}$ at a fixed $x$ satisfies

$$\boldsymbol{M(x)} \sim \mathcal{N}(\widehat{\boldsymbol{\mu}}(x), \mathbf{\Delta}(x)) \quad \text{with } \widehat{\boldsymbol{\mu}}(x) = [\widehat{\mu}_1(x), \dots, \widehat{\mu}_L(x)]^T, \ \mathbf{\Delta}(x) = \mathrm{diag}\left(\delta_1^2(x), \dots, \delta_L^2(x)\right).$$

The role of the covariance kernel is to specify the complexity of $\mu_\ell$ under the RKHS norm. Two main examples we use are the squared exponential kernel

$$\mathcal{K}(x, x'; s, \theta) = s^2 \exp(-\|x - x'\|^2/(2\theta^2))$$

and the (isotropic) Matern-5/2 kernel

$$\mathcal{K}(x, x'; s, \theta) = s^2\big(1 + (\sqrt{5} + 5/3)\|x - x'\|_\theta^2\big) \cdot e^{-\sqrt{5}\|x - x'\|_\theta}, \qquad \|x\|_\theta = \sqrt{x\,\mathrm{diag}\,\theta x^T}. \tag{2.11}$$

The length-scale parameter $\theta$ controls the smoothness of members of $\mathcal{M}_\mathcal{K}$, the smaller the rougher. The variance parameter $s^2$ determines the amplitude of fluctuations in the response. For both of the above cases, members of the function space $\mathcal{M}_\mathcal{K}$ can uniformly approximate any continuous function on any compact subset of $\mathcal{X}$.

A major advantage of kriging for sequential design are *updating formulas* that allow to efficiently assimilate new data points into an existing fit. Namely, if a new sample $(x, y)^{k+1}$ is added to an

existing design $x^{1:k}$, the mean and kriging variance at location $x$ are updated via

$$\widehat{\mu}^{(k+1)}(x) = \widehat{\mu}^{(k)}(x) + \lambda(x, x^{k+1}; x^{1:k})(y^{k+1} - \widehat{\mu}^{(k)}(x^{k+1})); \tag{2.12}$$

$$\delta^{(k+1)}(x)^2 = \delta^{(k)}(x)^2 - \lambda(x, x^{k+1}; x^{1:k})^2[\sigma^2(x^{(k+1)}) - \widehat{\mu}^{(k)}(x^{k+1})], \tag{2.13}$$

where $\lambda(x, x^{k+1}; x^{1:k})$ is a weight function specifying the influence of the new sample at $x^{k+1}$ on $x$ (conditioned on existing design locations $x^{1:k}$). In particular, the local reduction in posterior standard deviation at $x^{k+1}$ is proportional to the current $\delta^{(k)}(x^{k+1})$ with a proportionality factor [10]:

$$\frac{\delta^{(k)}(x^{k+1}) - \delta^{(k+1)}(x^{k+1})}{\delta^{(k)}(x^{k+1})} = 1 - \frac{\sigma(x^{k+1})}{\sqrt{\sigma^2(x^{k+1}) + \delta^{(k)}(x^{k+1})^2}}. \tag{2.14}$$

Note that the updated posterior variance $\delta^{(k+1)}(x)^2$ is a deterministic function of $x^{k+1}$ which is independent of $y^{k+1}$.

Several options are available for modeling the trends $t_\ell(\cdot)$. The zero-trend case $t_\ell(x) = 0$ is called Ordinary Kriging. Universal Kriging assumes a parametric trend $t_\ell(x) = \sum_i \beta^i t_\ell^i(x)$, and estimates the trend coefficients $\beta^i$ based on generalized least squares regression. See [40] or [45, Sec. 2.7] for the corresponding formulas of kriging mean and covariance. The constant trend sub-case $t_\ell(x) = \beta^0$ is known as Simple Kriging. For our examples below, we have used the **DiceKriging** `R` package [43] to compute (2.9). The software takes as input the location-index pairs $(x, \ell)^{1:n}$, the corresponding samples $y_\ell(x)^{1:n}$ and the noise levels $\sigma^2_{\ell^n}(x^n)$, as well as the kernel family (Matern-5/2 (2.11) by default) and trend basis functions $t_\ell^i(x)$ and runs an EM MLE algorithm to train a (Universal) kriging model, i.e. to estimate the kriging kernel $\mathcal{K}_\ell$ hyper-parameters $s, \theta$. One advantage of DiceKriging is support for state-dependent noise levels, as well as a variety of optimization routines to estimate the hyper-parameters.

2.4. **Summary Statistics for Ranking.** Given a fitted kriging surface $M_\ell$ (for notational convenience in this section we omit the indexing by the design size $k$), the respective classifier $\hat{\mathcal{C}}$ is obtained as in (2.5). Note that $\hat{\mathcal{C}}(x)$ is not necessarily the MAP (maximum a posteriori probability) estimator, since the ordering of the posterior probabilities and posterior means need not match for $L > 2$. Two further quantities are of importance for studying the accuracy of $\hat{\mathcal{C}}$: gaps and posterior probabilities. First, the gaps quantify the differences between the posterior means, namely

$$\widehat{\Delta}_\ell(x) := |\widehat{\mu}_\ell(x) - \min_{j \neq \ell} \widehat{\mu}_j(x)|, \tag{2.15}$$

$$\widehat{\Delta}(x) := |\widehat{\mu}_{(1)}(x) - \widehat{\mu}_{(2)}(x)|, \tag{2.16}$$

where recall that $\widehat{\mu}_{(1)} \leq \widehat{\mu}_{(2)} \leq \ldots \leq \widehat{\mu}_{(L)}$ are the ordered posterior means. Note that under $L = 2$, we have $\widehat{\Delta}_1 = \widehat{\Delta}_2 = \widehat{\Delta}$ due to symmetry.

Second, define the posterior probabilities for the minimal rank

$$p_\ell(x) := \mathbb{P}\left(\mu_\ell(x) = \mu_{(1)}(x) | \mathcal{F}_k\right) = \mathbb{P}(M_\ell(x) = \min_j M_j(x)). \tag{2.17}$$

We refer to $p_{(1)}(x) \geq p_{(2)}(x) \geq \ldots \geq p_{(L)}(x)$ as the decreasing ordered values of the vector $\vec{p}(x) := \{p_\ell(x)\}_{\ell=1}^L$, so that the index of $p_{(1)}(x)$ is the MAP estimate of the minimal response surface. The following proposition provides a semi-analytic recursive formula to evaluate $\vec{p}(x)$ in terms of the kriging means and variances $(\widehat{\mu}_\ell(x), \delta_\ell^2(x))$.

**Proposition 2.2** (Azimi *et al.* [4]). *If $\boldsymbol{M}(x) \sim \mathcal{N}(\widehat{\boldsymbol{\mu}}(x), \boldsymbol{\Delta}(x))$, then for any $\ell \in \mathfrak{L}$,*

$$p_\ell(x) = \mathbb{P}\left(M_\ell(x) = \min_j M_j(x)\right) = \prod_{j=1}^{L-1} \Phi\left(-r_j^{(\ell)}\right), \tag{2.18}$$

*where $\Phi(\cdot)$ is standard normal cdf, $\boldsymbol{r}^{(\ell)} = [r_1, r_2, \ldots, r_{L-1}]^T = (A(\ell)\boldsymbol{\Delta}(x)A(\ell)^T)^{-1/2}A(\ell)\widehat{\boldsymbol{\mu}}(x)$, with $A(\ell)$ a $(L-1)\times L$ matrix defined via*

$$A(\ell)_{i,j} = \begin{cases} 1 & \text{if} \quad j = \ell, \\ -1 & \text{if} \quad 1 \le i = j < \ell, \\ -1 & \text{if} \quad \ell < i+1 = j \le L, \\ 0 & \text{otherwise.} \end{cases}$$

**Corollary 2.3.** *For $L = 2$, we have $p_1(x) = \mathbb{P}(M_1(x) \le M_2(x)) = \Phi\left(\frac{\widehat{\mu}_2(x) - \widehat{\mu}_1(x)}{\sqrt{\delta_1^2(x) + \delta_2^2(x)}}\right)$, and $p_2(x) = 1 - p_1(x)$.*

The next proposition provides another semi-analytic formula to evaluate $m(x)$ defined in (2.6).

**Proposition 2.4.** *Suppose that $L = 2$ and let $M_\ell(x) \sim \mathcal{N}(\widehat{\mu}_\ell(x), \delta_\ell^2(x))$, $\ell = 1, 2$ be two independent Gaussians. Define*

$$d_{12} := \sqrt{\delta_1^2(x) + \delta_2^2(x)}, \qquad \text{and} \quad a_{12} := (\widehat{\mu}_1(x) - \widehat{\mu}_2(x))/d_{12}.$$

*Then the first two moments of $M_{(1)}(x) = \min(M_1(x), M_2(x))$ are given by:*

$$m(x) \equiv \mathbb{E}[M_{(1)}(x)] = \widehat{\mu}_1(x)\Phi(-a_{12}) + \widehat{\mu}_2(x)\Phi(a_{12}) - d_{12}\phi(a_{12}), \tag{2.19}$$

$$\begin{aligned} \mathbb{E}\left[M_{(1)}(x)^2\right] = {} & (\widehat{\mu}_1^2(x) + \delta_1^2(x))\Phi(-a_{12}) + (\widehat{\mu}_2^2(x) + \delta_2^2(x))\Phi(a_{12}) \\ & - (\widehat{\mu}_1(x) + \widehat{\mu}_2(x))d_{12}\phi(a_{12}). \end{aligned} \tag{2.20}$$

Equation (2.19) provides a closed-form expression to evaluate $m(x) = \mathbb{E}[M_{(1)}(x)]$ for $L = 2$. In the case $L > 2$, one may evaluate $m(x)$ recursively using a Gaussian approximation. To wit, for $L = 3$, approximate $\widetilde{Y} := M_1(x) \wedge M_2(x)$ by a Gaussian random variable with mean/variance specified by (2.19)-(2.20) respectively (i.e. using $a_{12}$ and $d_{12}$) and then apply Proposition 2.4 once more to $M_{(1)}(x) = \widetilde{Y} \wedge M_3(x)$.

## 3. Expected Improvement

The Bayesian approach to sequential design is based on greedily optimizing an acquisition function. Maximizing the acquisition function is used to select the next design point and is quantified through Expected Improvement (EI) scores. The aim of EI scores is to identify pairs $(x, \ell)$ which are most promising in terms of lowering the global empirical loss function $\mathcal{EL}$ according to (2.2). In our context the EI scores are based on the posterior distributions $M_\ell^{(k)}$ which summarize information learned so far about $\mu_\ell(x)$.

### 3.1. Gap-UCB Heuristic.

Motivated by multi-armed bandit literature, our first proposal for EI is based on the exploration-exploitation trade-off. Recall that in MAB exploitation corresponds to sampling the most promising arm, while exploration corresponds to reducing posterior variance of arm payoffs. To measure the relative value of arms, we use the local empirical gap measure [15]

$\widehat{\Delta}_\ell(x)$. To quantify exploration we use kriging variance $\delta^2_\ell(x)$. To merge these criteria, we borrow the 'UCB' (upper confidence bound) concept [44], i.e. construct the linear combination

$$E_k^{Gap-UCB}(x,\ell) := -\widehat{\Delta}_\ell(x) + \gamma_k \delta_\ell(x). \tag{3.1}$$

The above Gap-UCB criterion favors locations with small gaps in posterior means and high kriging variance. The tuning parameter $\gamma_k$, indexed in terms of the current design size $k$, balances exploration (regions with high $\delta_\ell(x)$) and exploitation (regions with small gap). Srinivas et al. [44] proved that in a cumulative regret setting $\gamma_k = O(\sqrt{\log k})$ should grow logarithmically in sample size $k$.

To better understand (3.1) note that it is based on two components: the local complexity of determining the true $\mathcal{C}(x)$ and the information gain from a new sampling location. For the latter, information gain can be related to the kriging variance $\delta^2(x)$ which measures uncertainty regarding the posterior mean and can be related to the usual standard error of a point estimator. For the former, recall that in a frequentist setting with fixed unknown $\mu_i(x)$'s, the gap $\Delta_\ell(x) := \mu_\ell(x) - \mu_{(1)}(x)$ is a basic measure of the hardness for testing whether $\mu_\ell(x) = \min_i \mu_i(x)$; the smaller $\Delta_\ell(x)$ the tougher. Analogously, in our Bayesian framework, the posterior estimated $\widehat{\Delta}(x)$ evaluates the complexity of identifying the minimum surface at $x$. Thus, the idea of Gap-UCB is to mimic a complexity-sampling scheme that selects design sites based on the complexity of the underlying ranking-and-selection problem, i.e. preferring sites with small $\Delta_\ell(x)$. This link was exploited to generate rules on how to choose $\gamma_k$ (for the case of a finite state space $\mathcal{X}$) in [15]. Still, finetuning the schedule of $k \mapsto \gamma_k$ is highly non-trivial in black-box settings. For this reason, usage of the Gap-UCB approach is sensitive to implementation choices.

We further note that when working with the gap $\Delta_\ell(x)$ the UCB approach is necessary to assure convergence. To wit, with $\gamma_k \equiv 0$ the algorithm would not be globally consistent. Indeed, consider $x_1, x_2$ such that $\Delta(x_2) > \Delta(x_1)$, but the estimated gaps based on interim $\mathcal{Z}^{(k)}$ satisfy $\widehat{\Delta}(x_1) > \widehat{\Delta}(x_2) > \Delta(x_2)$. Then at stage $k$ the algorithm will prefer site $x_2$ over $x_1$ (since it has smaller gap $\widehat{\Delta}$) and will then possibly get trapped indefinitely, never realizing that the estimated ordering between $\Delta(x_1)$ and $\Delta(x_2)$ is wrong. Hence without UCB the algorithm is prone to get trapped at local minima of $\widehat{\Delta}$.

3.2. **M-Gap & Uncertainty Reduction Heuristic.** From a wholly different perspective of stochastic optimization, we next design an acquisition function based on the loss function. Recall that we strive to lower the empirical loss $\mathcal{EL}$ in (2.8) which is related to the M-gap in (3.2), $\mathcal{EL} = \int \mathcal{M}(x)F(dx)$. Accordingly, we use $\mathcal{M}(x)$ to guide the adaptive design, by aiming to maximizing its *expected reduction*

$$E_k^{Gap-SUR}(x,\ell) := \mathbb{E}[\mathcal{M}^{(k)}(x) - \mathcal{M}^{(k+1)}(x) | x^{k+1} = x, \ell^{k+1} = \ell, \mathcal{F}_k]. \tag{3.2}$$

Thus the Gap-SUR criterion quantifies the expected reduction in local $\mathcal{M}(x)$ if we add $(x,\ell)$ to the design. To evaluate $\mathbb{E}[\mathcal{M}^{(k+1)}(x)|x^{k+1} = x, \ell^{k+1} = \ell, \mathcal{F}_k]$ recall Proposition 2.4 (which gives an exact formula for $L = 2$) that reduces to computing the expected mean and variance of $M_\ell$ at $x$. Now the updating formula (2.12) implies that (keeping $\mathcal{K}$ fixed) $\mathbb{E}[\widehat{\mu}_\ell^{k+1}(x)|x^{k+1} = x, \ell^{k+1} = \ell, \mathcal{F}_k] = \widehat{\mu}_\ell^k(x)$, while (2.14) yields $\delta_\ell^{(k+1)}(x)$. This allows a straightforward method to obtain (3.2) from the present kriging mean and variance $\widehat{\mu}_\ell(x), \delta^2_\ell(x)$. Stepwise uncertainty reduction (SUR) strategies were introduced in [5, 9] who were the first to take advantage of kriging updating formulas for this purpose.

Importantly, the Gap-SUR heuristic does not require any additional tuning schedules such as the $(\gamma_k)$ in the UCB approach. Moreover, it intrinsically ensures exploration by enforcing $\mathcal{M}(x) \to 0$ everywhere on $\mathcal{X}$. Because $\mathcal{M}(x) \to 0$ if and only if $\widehat{\mu}_{(1)} \to m(x)$, which is equivalent to the kriging variance vanishing $\delta^2_\ell(x) \to 0$ for all $\ell$, this guarantees strong consistency for learning $\mu_\ell$ (and in turn strong consistency for learning the ranks $\mathcal{C}(x)$). Hence, using Gap-SUR as an EI metric produces a globally consistent algorithm; in particular the resulting designs $\mathcal{Z}^{(K)}$ become dense on $\mathcal{X}$ as $K \to \infty$.

**Remark 3.1.** *The mentioned link between kriging variance and $\mathcal{M}$ connects our Gap-SUR criterion to the Active Learning Cohn (ALC) [12] approach to DoE. Namely, in ALC the goal of minimizing integrated posterior variance is achieved by sampling at sites with maximum expected reduction in posterior variance. In our context of minimizing the expected loss $\mathcal{EL}$, a natural criterion is therefore to prefer sites where the present (empirical) local loss $\mathcal{M}(x)$ is high and where $\delta^2(x)$ can be lowered quickly by augmenting the design. The ALC paradigm also suggests an alternative to* (3.1)*, namely $E_k^{Gap-ALC}(x,\ell) = -\widehat{\Delta}_\ell(x) + \gamma_k[\delta_\ell^{(k)}(x) - \delta^{(k+1)}(x)]$, that blends expected decline in kriging variance with the estimated gap.*

3.3. **Selecting the Next Sample Location.** Recall the basic looping procedure for growing the designs $\mathcal{Z}^{(k)}$ over $k = K_0, K_0+1, \ldots$. The EI heuristic indicates promising sampling locations, so that the "best" sites correspond to the highest EI scores $E_k(x,\ell)$. This leads to taking

$$(x_{k+1}, \ell_{k+1}) = \operatorname*{arg\,sup}_{(x,\ell)\in\mathcal{X}\times\mathfrak{L}} E_k(x,\ell). \tag{3.3}$$

Because the above introduces a whole new optimization sub-problem, in cases where this is computationally undesirable we instead replace $\arg\sup_{x\in\mathcal{X}}$ with $\arg\max_{x\in\mathcal{T}}$ where $\mathcal{T}$ is a finite *candidate set*. Optimization over $\mathcal{T}$ is then done by direct inspection. The justification for this procedure is that (i) we expect $E_k(x,\ell)$ to be smooth in $x$ and moreover relatively flat around $x^*$; (ii) $E_k(x,\ell)$ is already an approximation so that it is not required to optimize it precisely; (iii) performance of optimal design should be insensitive to small perturbations of the sampling locations. To construct such candidate sets $\mathcal{T}$ in $\mathcal{X}$, we employ Latin hypercube sampling (LHS) [35]. LHS candidates ensure that new locations are representative, and well spaced out over $\mathcal{X}$. See [19, Sec 3.4] for some discussion on how $\mathcal{T}$ should be designed. In addition, we refresh our *candidate set* $\mathcal{T}$ at each iteration, to enable "jittering".

**Remark 3.2.** *There are other approaches on how to use acquisition functions to select the next design site. For example, $\epsilon$-greedy sampling applies a randomized approach, using* (3.3) *with probability $1-\epsilon$ at each step, and otherwise picking $(x,\ell)$ uniformly in $\mathcal{X}\times\mathfrak{L}$. The latter feature guarantees (and possibly speeds up the rate of) global consistency by ensuring that the designs $\mathcal{Z}^{(k)}$ are dense in $\mathcal{X}$ as $k\to\infty$. Another possibility is to use $E_k(x,\ell)$ as a potential in a probabilistic sampling method, for example sampling according to weights $w(x,\ell) \propto \exp(-\alpha_k E_k(x,\ell))$.*

3.3.1. *Hierarchical Sampling in $\mathcal{X}\times\mathfrak{L}$.* Instead of sampling directly over the pairs $(x,\ell)\in\mathcal{X}\times\mathfrak{L}$, one can consider two-step procedures that first pick $x$ and then $\ell$ (or vice-versa). This strategy allows more intuition by connecting to the standard sequential design over $\mathcal{X}$. Indeed, one can then implement directly the active learning approach of [34, 12] by first picking $x^{k+1}$ using the gap

metrics, and then picking the index $\ell^{k+1}$ based on the kriging variance. One concrete proposal is:

$$\begin{cases} x^{k+1} = \arg\min_{x\in\mathcal{X}} \widehat{\Delta}(x)|\mathcal{F}_k, & \text{cf. (2.16)} \\ \ell^{k+1} = \arg\max_{\ell\in\mathfrak{L}} \delta_\ell^{(k)}(x^{k+1}). \end{cases} \tag{3.4}$$

Conditional on picking $x^{k+1}$, the above choice selects surfaces with large kriging variance $\delta_\ell(x)$, attempting to equalize $\delta_\ell(x)$ across $\ell$. As we mentioned before, to estimate $\mathcal{C}(x)$ well it is necessary to know every $\mu_\ell(x)$ well, so that it is intuitive to require $\delta_\ell(x)$ to be equal. Another choice is to pick $\ell^{k+1}$ to greedily maximize the information gain as in (2.14). Such two-step EI heuristics allow to avoid having to specify the schedule $\gamma_k$ of UCB criteria (3.1).

The two-step procedure in (3.4) also links to the idea of concurrent marginal modeling of each $\mu_\ell$. Concurrent sampling means that after choosing a location $x^{k+1} \equiv x$, one augments the design with the $L$ respective pairs $(x,1), (x,2), \dots (x,L)$. This approach "parallelizes" the learning of all response surfaces while still building an adaptive design over $\mathcal{X}$. The disadvantage of this strategy becomes clear in the extreme situation when the variance of $Y_1(x)$ is zero, $\sigma_1(x) \equiv 0$ while the noise of $Y_2(x)$ is large. In that case, after sampling a given location once for each response, $(x,1)$ and $(x,2)$, we would have $\delta_1(x) = 0$, $\delta_2(x) \gg 0$. Hence, another sample from $Y_1(x)$ would gain no information at all, while substantial information would still be gleaned from sampling $Y_2(x)$, making parallel sampling twice as costly as needed.

At the same time, the idea of concurrent sampling yields an upper bound on the overall performance of the ranking problem. Indeed, in the presence of multiple surfaces one may always learn each of them separately and then compare the fitted models to pick the minimum index according to (2.5), which is equivalent to $L$ marginalized learning problems over $\mathcal{X}$.

**Algorithm 1** Sequential Design for Global Ranking using Kriging

**Require:** $K_0, K$

1: Generate initial design $\mathcal{Z}^{(K_0)} := (x,\ell)^{1:K_0}$ using LHS
2: Sample $y^{1:K_0}$, estimate the GP kernels $\mathcal{K}_\ell$'s and initialize the response surface models $M_\ell$
3: Construct the classifier $\mathcal{C}^{(K_0)}(\cdot)$ using (2.5)
4: $k \leftarrow K_0$
5: **while** $k < K$ **do**
6: Generate a new candidate set $\mathcal{T}^{(k)}$ of size $D$
7: Compute the expected improvement (EI) $E_k(x,\ell)$ for each $x \in \mathcal{T}$, $\ell \in \mathfrak{L}$
8: Pick a new location $(x,\ell)^{k+1} = \arg\max_{(x,\ell)\in\mathcal{T}^{(k)}\times\mathfrak{L}} E_k(x,\ell)$ and sample the corresponding $y^{k+1}$
9: (Optional) Re-estimate the kriging kernel $\mathcal{K}_{\ell^{k+1}}$
10: Update the response surface $M_{\ell^{k+1}}$ using (2.12)-(2.13)
11: Update the classifier $\mathcal{C}^{(k+1)}$ using (2.5)
12: Save the overall grid $\mathcal{Z}^{(k+1)} \leftarrow \mathcal{Z}^{(k)} \cup (x^{k+1}, \ell^{k+1})$
13: k ← k+1
14: **end while**
15: **return** Estimated classifier $\mathcal{C}^{(K)}(\cdot)$.

3.4. **Implementation.** To implement the abstract sequential ranking algorithm outlined in Section 2.1, it remains to specify the initialization step. In the context of a kriging model, the initial

design $\mathcal{Z}^{(K_0)}$ is crucial to ensure convergence, as the algorithm must first train the Gaussian process by learning its covariance structure. One common challenge is to avoid assuming that $\mu_\ell$'s are too flat by missing the shorter-scale fluctuations [40]. Thus, $K_0$ must be large enough to reasonably estimate the covariance structure. In order to maximize exploration at this initial stage, we once again utilize a space-filling LHS design (sampling equally across the $L$ surfaces). The recommendation is that $K_0$ should be about 20% of the eventual design size $K$. In cases where a priori knowledge about the kernels $\mathcal{K}_\ell$ is available, $K_0$ can be much smaller. Algorithm 1 below presents the resulting method in pseudo-code.

Our last remark concerns step 9 in Algorithm 1. Re-estimation of the kriging kernel $\mathcal{K}_\ell$ is computationally expensive and makes the GP framework not sequential. In contrast, if one treats $\mathcal{K}_\ell$ as fixed, updating the kriging model via (2.12)-(2.13) is efficient and takes only $\mathcal{O}(k)$ flops at stage $k$. Since we expect the algorithm to converge as $k \to \infty$, we adopt the practical rule of running the full estimation procedure for $\mathcal{K}$ according to the doubling method [16]. To wit, we re-estimate $\mathcal{K}_\ell$ for $k = 2, 4, 8, \dots$ a power of two, keeping it frozen across other steps.

## 4. Simulated Experiments

4.1. **Toy Example.** In this section we consider a simple one-dimensional example with synthetic data which allows a fully controlled setting. Let $L = 2, \mathcal{X} = [0, 1]$. The noisy responses $Y_1(x)$ and $Y_2(x)$ are specified by (cf. the example in [43, Sec 4.4])

$$Y_1(x) = \mu_1(x) + \epsilon_1(x) \equiv \frac{5}{8}\left(\frac{\sin(10x)}{1+x} + 2x^3\cos(5x) + 0.841\right) + \sigma_1(x)Z_1,$$
$$Y_2(x) = \mu_2(x) + \epsilon_2(x) \equiv 0.5 + \sigma_2(x)Z_2.$$

Here $Z_\ell$ are independent standard Gaussian, and the noise strengths are fixed at $\sigma_1(x) \equiv 0.2$ and $\sigma_2(x) \equiv 0.1$, homoscedastic in $x$ but heterogenous in $\ell = 1, 2$. The weights $F(dx) = dx$ in the loss function are uniform on $\mathcal{X}$.

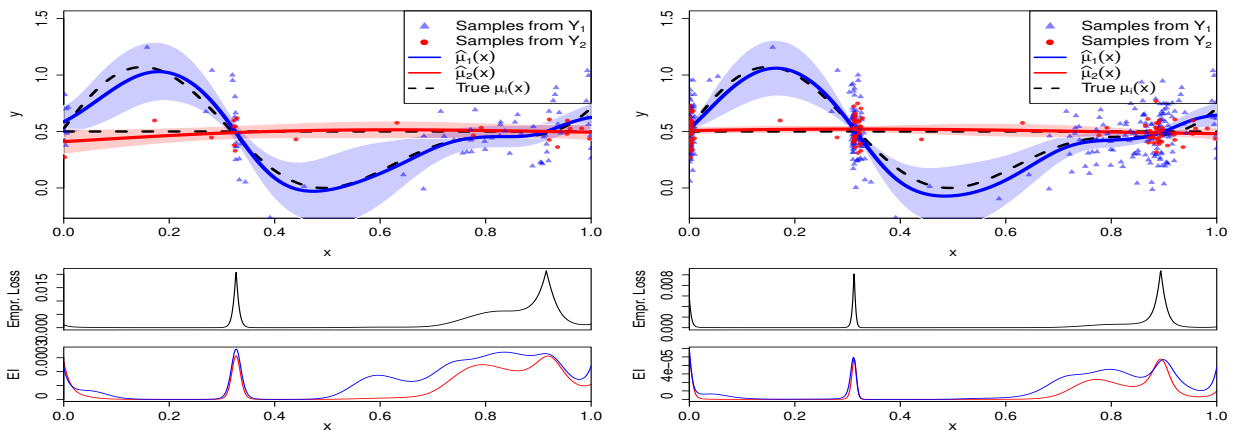
The true ranking classifier $\mathcal{C}(x)$ consists of three pieces and is given by

$$\mathcal{C}(x) = \begin{cases} 2 & \text{for } 0 < x < r_1, \\ 1 & \text{for } r_1 < x < r_2, \\ 2 & \text{for } r_2 < x < 1. \end{cases} \tag{4.1}$$

where $r_1 \approx 0.3193, r_2 \approx 0.9279$.

To focus on the performance of various acquisition functions, we fix the kriging kernels $\mathcal{K}_\ell$ to be of the Matern-5/2 type (2.11) with hyperparameters $s_1 = 0.1, \theta_1 = 0.18$ for $\mathcal{K}_1$ and $s_2 = 0.1, \theta_2 = 1$ for $\mathcal{K}_2$. These hyper-parameters are close to those obtained by training a kriging model for $Y_\ell(x)$ given a dense design on $\mathcal{X}$ and hence capture well the smoothness of the response surfaces above. Moreover, we employ a Simple Kriging model with a given constant trend $t_\ell(x) = 0.5$.

To apply Algorithm 1 we then initialize with $K_0 = 10$ locations $(x, \ell)^{1:K_0}$ (five each from $Y_1(x)$ and $Y_2(x)$), drawn from a LHS design on $[0, 1]$. Note that because the kriging kernels are assumed to be known, $K_0$ is taken to be very small. To grow the designs we employ the Gap-SUR EI criterion and optimize for the next $(x, \ell)^{k+1}$ using a fresh candidate set $\mathcal{T}^{(k)}$ based on a LHS design of size $D = 100$. Figure 1 illustrates the evolution of the posterior response surface models. The two panels show the estimated $M_\ell^{(K)}(x)$ at $K = 100$ and $K = 400$ (namely we plot the posterior means $\widehat{\mu}_\ell^{(K)}(x)$ and the corresponding 90% CI $\widehat{\mu}_\ell^{(K)}(x) \pm 1.645\delta_\ell^{(K)}(x)$). We observe that most of the samples are heavily concentrated around the two classification boundaries $r_1, r_2$, as well as the "false" boundary at $x = 0$. As a result, the kriging variance $\delta_\ell^2(x)$ is much lower in those




$K = 100$ $K = 400$

FIGURE 1. Response surface modeling with the Gap-SUR EI criterion of (3.2). We plot the true surfaces $\mu_\ell(x)$ (black dashed lines), the posterior means $\widehat{\mu}_\ell(x)$ (blue/red solid lines), the 90% posterior credibility intervals (light blue/red areas) of $M_1(x)$ and $M_2(x)$, and the sampling locations $x^{1:K}$ for $Y_1(x)$ (blue triangles) and $Y_2(x)$ (red circles). The middle panel shows the local loss $\mathcal{M}(x)$, cf. (2.7), while the bottom panel shows the Gap-SUR EI metric $E_K(x, \ell)$ (blue: $\ell = 1$, red: $\ell = 2$).

neighborhoods, generating the distinctive "sausage" shape for the posterior credibility intervals of $M_\ell(x)$. In contrast, in regions where the gap $\Delta(x)$ is large (e.g., around $x = 0.5$), ranking the responses is easy so that almost no samples are taken and the kriging variance remains large. Also, because $\sigma_1(x) > \sigma_2(x)$, the credibility intervals of $\mu_2$ are tighter, $\delta_1(x) > \delta_2(x)$, and most of the samples are from the first response $Y_1$. Indeed, we find $D_1(k) \simeq 3D_2(k)$ where

$$D_i(K) := |\{1 \le k \le K : \ell^k = i\}|$$

is the number of samples in the design $\mathcal{Z}^{(K)}$ from the $i$-th surface. The above observations confirm the double efficiency from making the EI scores depend on both the $\mathcal{X}$ and $\mathfrak{L}$ dimensions.

From a different angle, Figure 2 shows the resulting design $\mathcal{Z}^{(400)}$ in this example and the location of sampled sites $x^k$ as a function of sampling order $k = 1, \ldots, 400$. We observe that the algorithm first engages in exploration and then settles into a more targeted mode, alternating between sampling around 0, $r_1$ and $r_2$.

4.2. **Benchmarks.** To judge the efficiency of different sequential designs, we proceed to benchmark the performance of different approaches.

To begin, we provide two non-adaptive designs that are heuristic upper and lower bounds for the achieved (empirical) loss. For the upper bound, we consider the uniform sampling method that relies purely on the law of large numbers to learn $\mu_\ell(x)$. Thus, at each step $k$, we generate a new sampling location $(x, \ell)^k$ uniformly from $\mathcal{X} \times \mathcal{L}$. This generates a roughly equal number of samples $D_1(k) \simeq D_2(k)$ from each response and a kriging variance $\delta^2_\ell(x)$ that is approximately constant in $x$.

For the "lower" bound, we assume that the true $\mu_\ell(\cdot), \sigma_\ell(\cdot)$ are known and use that foresight to generate a design that takes into account the resulting complexity for resolving $\mathcal{C}(x)$. Namely, we plug in the true $\Delta_\ell(x)$ into the constructed EI metrics, for example the Gap-UCB metric in

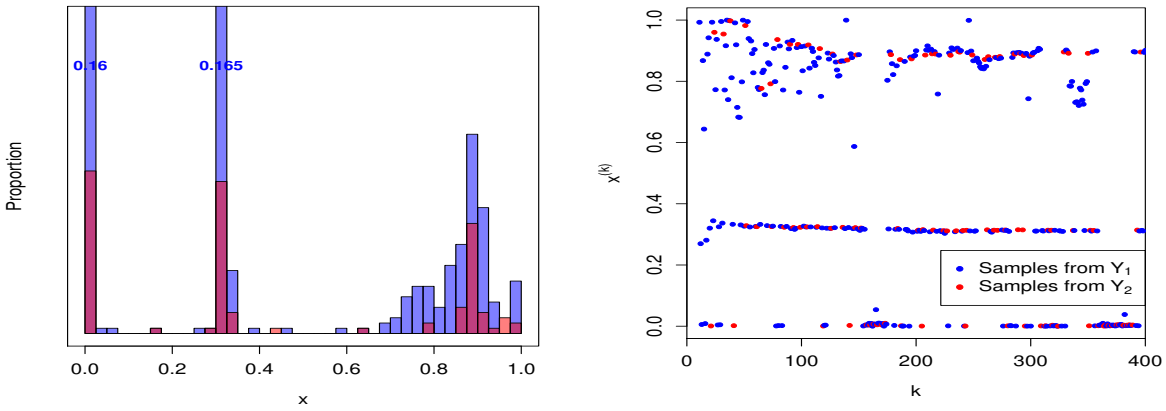



FIGURE 2. Left: the design $\mathcal{Z}^{(400)}$ based on the Gap-SUR EI criterion of (3.2). There were $D_1(400) = 294$ and $D_2(400) = 106$ samples from $Y_1$ and $Y_2$ respectively. Right: sampled locations $x^k$ as a function of $k$ (blue for $\ell^k = 1$, red for $\ell^k = 2$).

(3.1). The resulting design is sequential but deterministic since the sampling depends solely on the given $\Delta_\ell(x)$'s and kriging variances $\delta_\ell^{(k)}(x)^2$ that are iteratively determined by the previous $x^{1:k}$, cf. (2.10). We expect the above design to be non-myopic since it is independent of observed $Y^{1:k}$ (of course, the outputted $\hat{\mathcal{C}}(\cdot)$ is still a function of $Y^{1:K}$). We found that the resulting designs perform comparably across the derived EI-heuristic, and report the representative results for True Gap-UCB approach.

The left panel of Figure 3 shows the convergence rate of our two main algorithms, Gap-UCB (with $\gamma_k = \sqrt{\log k}$) and Gap-SUR against the above benchmarks in terms of the resulting empirical loss $\mathcal{EL}$ which is approximated by

$$\mathcal{EL}(\hat{\mathcal{C}}, \mathcal{C}) = \frac{1}{M} \sum_{j=1}^{M} \left\{ \widehat{\mu}_{(1)}(j\Delta x) - m(j\Delta x) \right\}, \tag{4.2}$$

where we used $M = 1000 = 1/\Delta x$ uniformly spaced gridpoints in $\mathcal{X} = [0, 1]$. We observe that our designs are much more efficient than the naive uniform design, and, perhaps surprisingly, also beat the non-adaptive method that already knows the true gaps. This happens because the empirical loss of the non-adaptive method is in fact rather sensitive to the observed samples $Y^{1:K}$ which can generate erroneous estimates of $\mu_\ell(x)$ and mis-classified $\mathcal{C}(x)$. Consequently the True Gap-UCB design, while properly placing $(x, \ell)^{1:K}$ *on average*, does not allow self-correction so that erroneous beliefs about $\mu_\ell$ can persist for a long time, increasing $\mathcal{EL}$. In contrast, adaptive algorithms add samples to any regions where observations suggest that $\Delta(x)$ is small, precising the estimated means there and lowering both true and empirical loss functions. All methods appear to enjoy a power-law (linear behavior on the log-log plot) for $\mathcal{EL}$ as a function of design size $k$.

The right panel of Figure 3 shows boxplots of $\mathcal{L}(\hat{\mathcal{C}}^{(K)}, \mathcal{C})$, computed analogously to (4.2), across a variety of designs. In addition to the four methods above, we also show the performance of two more sequential designs, Best-UCB and Gap-SUR w/training that are discussed in Section 4.3 below, cf. Table 1.

4.3. **Comparison and Discussion of EI Criteria.** To further explore the role of the acquisition function $E_k$, in this section we compare several heuristic EI definitions. A 100 trials are run for

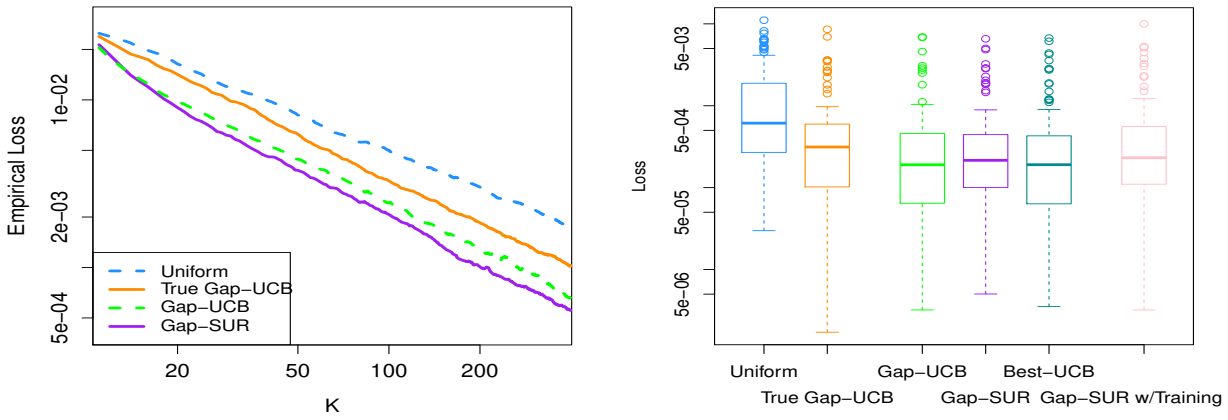



FIGURE 3. Left: Averaged empirical loss $\mathcal{EL}(\hat{\mathcal{C}}^{(k)})$ as a function of design size $k$ (in log-log scale). We compare our adaptive Gap-SUR (3.2) and Gap-UCB methods (3.1) against a uniform sampler and a non-adaptive Gap-UCB based on the true gap $\Delta(\cdot)$. Right: boxplot of $\mathcal{L}(\hat{\mathcal{C}}^{(K)}, \mathcal{C})$ at $K = 400$ across six different methods.

each method to compute the resulting mean and standard deviation of the loss function $\mathcal{L}$ and the empirical loss $\mathcal{EL}$. To isolate the effect of the EI criterion, we continue to work in the setting with a fixed GP covariance structure $\mathcal{K}_\ell$ for the $\mu_\ell$'s (see hyperparameter values in Sec. 4.1).

Several further alternatives for evaluating expected improvement can be designed based on classification frameworks. For classification, the main posterior statistic is the probabilities $p_\ell(x)$ of $\mu_\ell(x)$ being the smallest response. One can then use the vector $\vec{p}(x)$ to measure the complexity of the resulting local classification at $x$. Note that such measures intrinsically aggregate across $\ell$ and hence only depend on $x$. This suggests either using a two-step sampling procedure as in Section 3.3.1 or building a UCB-like criterion as in (3.1). We employ the latter method; to wit, a criterion $\Gamma(x)$ is used to discriminate among $x$-locations (larger scores are preferred) and is blended with UCB according to the cooling schedule $(\gamma_k)$, $E_k(x, \ell) = -\Gamma^{(k)}(x) + \gamma_k \delta_\ell(x)$.

A standard way to measure classification complexity is the posterior entropy, namely

$$\Gamma^{ENT}(x) := -\sum_{\ell=1}^{L} p_\ell(x) \log p_\ell(x). \tag{4.3}$$

High entropy indicates more spread in $\vec{p}(x)$ and hence more uncertainty about which is the smallest component of $\vec{\mu}(x)$. A well-known drawback to (4.3) for large $L$ (bigger than 3) arises from undesired effects on $\Gamma^{ENT}$ from the components with small $p_\ell(x)$. In other words, the responses that are very unlikely to be the minimum still strongly affect the overall entropy, leading to non-intuitive shape of the EI scores. To counteract this effect, [28] proposed the Best-versus-Second-Best (BvSB) approach, which uses

$$\Gamma^{BvSB}(x) = -\left[p_{Best}(x) - p_{SB}(x)\right], \tag{4.4}$$

where $p_{Best}(x) := \mathbb{P}\left(\hat{\mathcal{C}}(x) = \mathcal{C}(x) | \mathcal{F}_k\right)$ is the posterior probability the lowest posterior mean is indeed the smallest response, and $p_{SB}$ is the probability that the second-lowest posterior mean is the smallest response. Thus, only the lowest two posterior means are compared, and $\Gamma^{BvSB}$ is quite

similar to the gap measure $\widehat{\Delta}(x)$ (while also taking into account the relative posterior variances). Small differences between $p_{Best}$ and $p_{SB}$ indicate large uncertainty in identifying the minimum response. The BvSB approach can break down however if $\delta_\ell(x)$'s are highly unequal, whereby the ordering between the posterior means and posterior probabilities is not the same. Another alternative to (4.3) is

$$\Gamma^{Best}(x) = -p_{Best}(x). \tag{4.5}$$

The $\Gamma^{Best}$ measure strongly prefers locations where $p_{Best}(x) \simeq 0.5$, i.e. those close to classification boundaries of $\hat{\mathcal{C}}(x)$ and in our experience does not explore enough. $\Gamma^{Best}$ and $\Gamma^{BvSB}$ give the same preferences over $\mathcal{X}$ in the case $L = 2$, whereby $\Gamma^{BvSB}(x) = 1 - 2p_{Best}(x)$.

Yet another alternative is a so-called pure M-Gap heuristic that uses (3.2) via

$$x^{k+1} = \arg\max_{x \in \mathcal{T}^{(k)}} \mathcal{M}(x), \qquad \ell^{k+1} = \arg\max_{\ell} \delta_\ell^2(x^{k+1}).$$

This hierarchical sampling strategy can be viewed as generalizing the Efficient Global Optimization (EGO) criterion of [27] to the ranking problem, compare for example to the classification variant of EGO in [22].

TABLE 1. True loss *vs.* empirical loss with $\mathcal{Z}^{(200)}$ for the 1-D example. For UCB heuristics the cooling schedule is of the form $\gamma_k = c\sqrt{\log k}$ with $c$ as listed below. The error probability $ErrProb$ measures the mean of $1 - p^{(200)}_{Best}(x)$ over the test set. $D_1 = D_1(200)$ is the number of samples out of 200 total from $Y_1$.

| Method | Emp Loss | (SE) | True Loss | (SE) | ErrProb | (SE) | $D_1$ |
|---|---|---|---|---|---|---|---|
| Uniform Sampling | 2.89E-3 | (1.24E-4) | 2.64E-3 | (2.67E-4) | 6.87% | (0.25%) | 100 |
| True Gap-UCB $c = 4$ | 1.77E-3 | (8.35E-5) | 1.43E-3 | (1.91E-4) | 5.61% | (0.23%) | 174 |
| Gap-SUR | 9.56E-4 | (4.98E-5) | 1.19E-3 | (1.84E-4) | 3.82% | (0.17%) | 146 |
| Pure M-Gap | 1.20E-3 | (5.39E-5) | 1.81E-3 | (2.33E-4) | 4.28% | (0.15%) | 172 |
| Concurrent M-Gap | 1.36E-3 | (8.33E-5) | 1.52E-3 | (1.97E-4) | 4.78% | (0.24%) | 100 |
| Gap-UCB, $c = 4$ | 1.54E-3 | (7.12E-5) | 2.00E-3 | (2.85E-4) | 5.15% | (0.20%) | 176 |
| Gap-UCB, $c = 1$ | 1.27E-3 | (5.61E-5) | 1.50E-3 | (1.98E-4) | 4.39% | (0.16%) | 167 |
| $\Gamma^{ENT}$-UCB $c = 100$ | 1.22E-3 | (5.48E-5) | 1.31E-3 | (1.87E-4) | 4.31% | (0.16%) | 146 |
| $\Gamma^{Best}$-UCB $c = 100$ | 1.28E-3 | (5.03E-5) | 1.49E-3 | (2.05E-4) | 4.56% | (0.16%) | 145 |
| Gap-SUR w/training $\mathcal{K}_1$ | 1.20E-3 | (5.87E-5) | 1.69E-3 | (3.24E-4) | 4.34% | (0.37%) | 146 |

Table 1 compares the performance of several EI acquisition functions. Beyond the three methods already mentioned in the previous section, we also examine (i) pure M-gap, (ii) concurrent sampling with M-Gap, (iii) Gap-UCB (for two different $\gamma_k$-schedules), (iv) $\Gamma^{ENT}$-UCB entropy criterion based on (4.3) , (v) $\Gamma^{Best}$-UCB criterion based on (4.5). The first two alternatives compare the benefit of taking into account uncertainty reduction and the benefit of discriminating in $\ell \in \mathfrak{L}$. The latter alternatives compare to UCB-type methods, including those based on the posterior probabilities $\vec{p}(x)$. To construct the summary statistics in Table 1 we ran each algorithm 100 times, initializing with a random LHS design of size $K_0 = 10$ and augmenting it until $K = 200$ sites. Throughout, we compute both the true loss in this synthetic example where $\mu_\ell(x)$ are known, as well as $\mathcal{EL}$ as in (4.2). A further metric reported is the error probability $1 - p^{(K)}_{Best}(x)$ which measures the posterior probability that the identified minimum response is incorrect.

The Gap-SUR algorithm appears to be the most efficient, in particular performing better than Gap-UCB or the pure M-Gap methods. It dominates across all metrics, and moreover also has the smallest fluctuations across algorithm runs, indicating more stable behavior. Nevertheless, the UCB methods are nearly as good, in particular the $\Gamma^{ENT}$-UCB approach is competitive. However, as discussed these methods are sensitive to the choice of the $\gamma_k$-schedule; the table shows that a poorly chosen $\gamma_k$ can materially worsen performance (see rows in Table 1 comparing $\gamma = 4\sqrt{\log k}$ vis-a-vis $\gamma = \sqrt{\log k}$ for Gap-UCB). At the same time, a limitation of Gap-SUR is that it requires knowing the noise variances $\sigma^2_\ell(\cdot)$ when optimizing the EI acquisition function.

Table 1 also highlights the gain from discriminating among the response surfaces, as the Concurrent M-Gap algorithm is notably worse (with losses of about 30% higher) relative to Gap-SUR. The only difference between these methods is that Gap-SUR sampled $Y_1$ 146 times out of 200, while the concurrent method was constrained to sample each response exactly 100 times. All approaches that optimize over the full $\mathcal{X} \times \mathfrak{L}$ focus on fitting the noisier $Y_1$, sampling it 70–85% of the rounds (see the $D_1$ column). The Table also confirms that our designs are much more efficient than the naive uniform sampler, or non-adaptive methods, even those that have access to the true $\mathcal{C}(\cdot)$ (the True Gap method).

As a final comparison, the last row of Table 1 reports the performance of the Gap-SUR method in the practical context where one must also *train* the GP kernels $\mathcal{K}_\ell$'s. Since training introduces additional noise into the fitted response surfaces, algorithm performance is necessarily degraded. We observe that this effect, especially in terms of variation across algorithm runs, is non-negligible which could indicate that Simple Kriging is not ideal for this particular example.

Table 1 also shows that the empirical $\mathcal{EL}(\hat{\mathcal{C}}^{(K)})$ and actual loss $\mathcal{L}(\hat{\mathcal{C}}^{(K)}, \mathcal{C})$ metrics are consistent, so that the former can be used as an internal online assessment tool to monitor accuracy of the estimated classifier. Model mis-specification is the apparent cause of the observed mismatch between the two quantities, as incorrectly inferred covariance structure of $\mu_1(x)$ leads to over-optimism: $\mathcal{EL} < \mathcal{L}$. Note that this issue would arise across all our algorithms and pertains more to the kriging framework than to EI acquisition functions.

4.4. **2D Example.** Our next example treats a more complex setting with $L = 3$ surfaces and a 2-dimensional input space $\mathcal{X} = [-2, 2]^2$. We take

$$Y_\ell(x_1, x_2) = \mu_\ell(x_1, x_2) + \epsilon_\ell(x_1, x_2) \quad \text{where} \quad \begin{cases} \mu_1(x_1, x_2) = 2 - x_1^2 - 0.5x_2^2; \\ \mu_2(x_1, x_2) = 2(x_1 - 1)^2 + 2x_2^2 - 2; \\ \mu_3(x_1, x_2) = 2\sin(2x_1) + 2, \end{cases} \tag{4.6}$$

and $\epsilon_\ell(x_1, x_2) \sim N(0, \sigma^2_\ell(x_1, x_2))$ have equal noise variance $\sigma^2_\ell(x_1, x_2) = 0.1(x_2+2)$ for all $\ell = 1, 2, 3$. Thus, the noise level is heteroscedastic in $\mathcal{X}$, being larger for large $x_2$-values. The resulting classifier $\mathcal{C}(x_1, x_2)$ consists of three connected pieces and is shown in Figure 4. This example is inspired by a GP classification problem in [21].

To illustrate a slightly different design strategy, we generated $\mathcal{Z}$'s using a kriging response surface model and Gap-UCB expected improvement criterion based on estimated gaps $\widehat{\Delta}_\ell(x)$, cf. (3.1) and $\gamma_k = 2\sqrt{\log k}$. The model was initialized at $K_0 = 15$ by generating 5 LHS samples from each $Y_\ell(x_1, x_2)$; at each step the sampling locations were selected from a LHS candidate set $\mathcal{T}$ of size $D = 100$ using $\epsilon$-greedy method with $\epsilon = 0.1$, cf. Remark 3.2.

Figure 4 summarizes the design $\mathcal{Z}^{(K)}$ and resulting classifier $\hat{\mathcal{C}}^{(K)}$ after $K = 150$ samples in total. Among 150 samples, $D_1(K) = 56$ were chosen from $Y_1$, $D_2(K) = 48$ were chosen from $Y_2$ and the

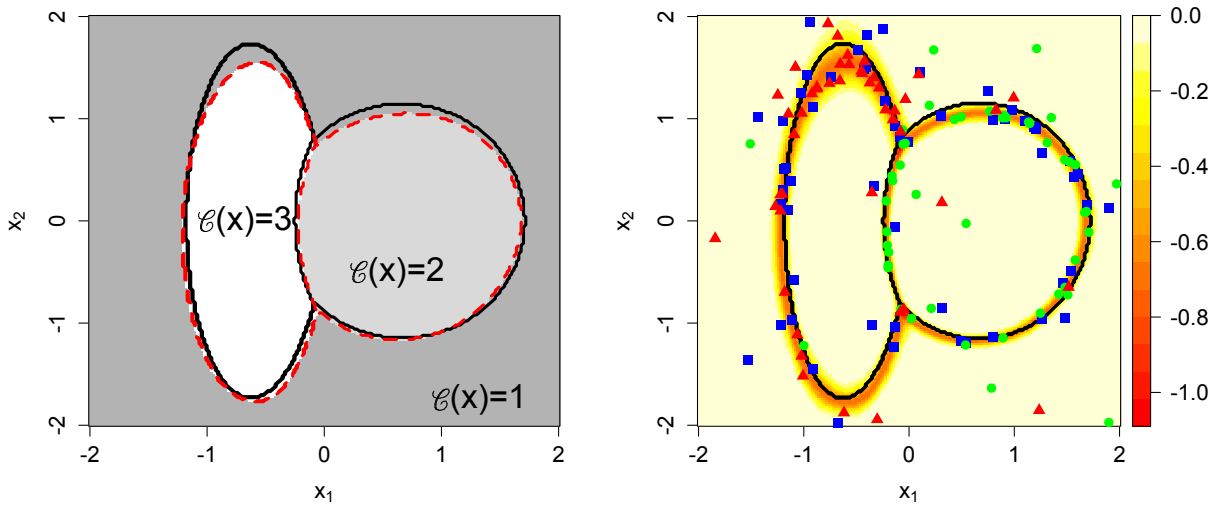


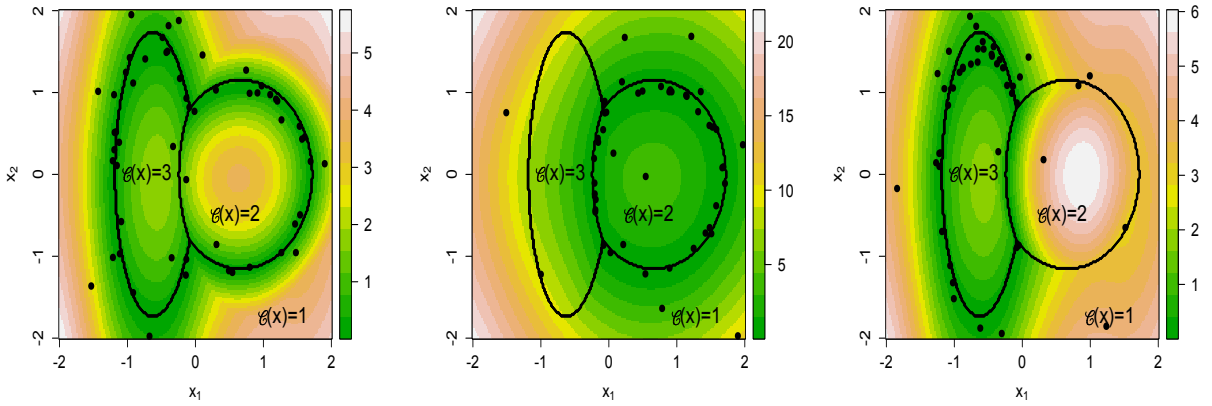

FIGURE 4. 2-D Ranking on $\mathcal{X} = [-2, 2] \times [-2, 2]$. Left panel: estimated classifier $\hat{\mathcal{C}}^{(K)}(x_1, x_2)$ for $K = 150$ (dashed red lines) and the true $\mathcal{C}(x_1, x_2)$ (solid black lines). Right panel: scatterplot of the sampled locations $(x_1, x_2)^{1:K}$ color-coded by the respective sampling surface: $Y_1(x_1, x_2)$ (blue squares), $Y_2(x_1, x_2)$ (green circles) and $Y_3(x_1, x_2)$ (red triangles). Shading indicates the posterior entropy $\Gamma^{(ENT)}(x_1, x_2)$ from (4.3) based on the above design.

rest were from $Y_3$, i.e. each surface took roughly one third of the budget. This is consistent with the identical $\sigma_\ell$'s and the roughly equal volume of each $\mathcal{C}_\ell$. We observe a close match between $\hat{\mathcal{C}}^{(K)}$ and the true $\mathcal{C}$. The fact that most of the samples were around the boundaries, and the rest scattered on $[-2, 2]^2$, indicates that the algorithm successfully balanced exploration and exploitation. Moreover, as expected the higher noise variance for larger $x_2$-values causes most of the sampling locations to concentrate towards the upper half-plane where the ranking complexity is higher.

Figure 5 further shows that our algorithm is highly discriminating in sampling jointly on $\mathcal{X} \times \mathcal{L}$. At any given classification boundary, the algorithm effectively only sampled two out of the three responses, endogenously recovering the concept of Best-versus-Second Best testing. Thus, samples from $Y_\ell$ are mostly located around the boundaries of surface $\mu_\ell$ and other surfaces. These contours, where $\Delta_\ell = \mu_\ell - \min_i \mu_i = 0$, are precisely the regions targeted by the Gap EI metrics.

## 5. Case Study in Epidemics Management

Our last example is based on control problems in the context of infectious epidemics [30, 32, 33, 37]. Consider the stochastic SIR model which is a compartmental state-space model that partitions a population pool into the three classes of Susceptible counts $S_t$, Infecteds $I_t$ and Recovereds $R_t$. We assume a fixed population size $M = S_t + I_t + R_t$ so that the state space is the two-dimensional simplex $\mathcal{X} = \{(s, i) \in \mathbb{Z}_+^2 : s + i \leq M\}$. In a typical setting, $M \in [10^3, 10^5]$, so that $\mathcal{X}$ is discrete but too large to be explicitly enumerated (on the order of $|\mathcal{X}| \simeq 10^6$). The dynamics of $(S_t, I_t)$ are time-stationary and will be specified below in (5.3).




$\ell = 1$ $\ell = 2$ $\ell = 3$

FIGURE 5. Marginal designs $(x_1, x_2)^{1:D_\ell(K)}$ at $K = 150$ for each of the 3 response surfaces constructed using the Gap-UCB heuristic. Shading indicates the estimated empirical gaps $\widehat{\Delta}_\ell(x_1, x_2)$, $\ell = 1, 2, 3$. We observe that most samples gravitate towards regions where $\widehat{\Delta}_\ell \simeq 0$. Solid curves indicate boundaries of the true classifier $\mathcal{C}(x_1, x_2)$.

The goal of the controller is to mitigate epidemic impact through timely intervention, such as social distancing measures that lower the infectivity rate by reducing individuals' contact rates; mathematically this corresponds to modifying the dynamics of $(S_t, I_t)$. To conduct cost-benefit optimization, we introduce on the one hand epidemic costs, here taken to be proportional to the number of cumulative infecteds, and on the other hand intervention costs, that are proportional to the current number of remaining susceptibles $C^I S_t$. Intervention protocol can then be (myopically) optimized by comparing the expected cost of no-action $\mu_0(s, i)$ (conditional on the present state $(s, i)$) against the expected cost of immediate action, $\mu_A(s, i)$. More precisely, let

$$\mu_0(s, i) := \mathbb{E}^0[S_0 - S_T | I_0 = i, S_0 = s] \quad \text{and} \tag{5.1}$$

$$\mu_A(s, i) := \mathbb{E}^A[S_0 - S_T | I_0 = i, S_0 = s] + C^I s. \tag{5.2}$$

Above, $T = \inf\{t : I_t = 0\}$ is the random end date of the outbreak; due to the fixed population and posited immunity from disease after being infected, the epidemic is guaranteed to have a finite lifetime. The difference $S_0 - S_T$ thus precisely measures the total number of original susceptibles who got infected at some point during the outbreak.

The overall goal is then to *rank* $\mu_0$ and $\mu_A$, with the intervention region corresponding to $\{(s, i) : \mu_A(s, i) > \mu_0(s, i)\}$. Because no analytic formulas are available for $\mu_\ell$'s, the only feasible procedure (which is also preferred due to the ease with which it can handle numerous extensions of SIR models) is a Monte Carlo sampler that given an initial condition $S_0 = s, I_0 = i$ and the regime $\ell \in \{0, A\}$ generates a trajectory $(S_t, I_t)(\omega)$ and uses it to evaluate the pathwise $S_T(\omega)$, which maps this problem into the framework of (1.1).

From the policy perspective, the trade-off in (5.1)-(5.2) revolves around doing nothing and letting the outbreak run its course, which carries a unit cost for each individual that is eventually infected, or implementing preventive social distancing measures which costs $C^I$ for each *susceptible*, but

lowers the expected number of future infecteds. Typical countermeasures might be public ad campaigns, school closures, or distribution of prophylactic agents. In general, intervention is needed as soon as there is a threat of a big enough outbreak. However, if $I_t$ is low, the cost of intervention is too high relative to its benefit because the epidemic might end on its own. Similarly, if $S_t$ is low, the susceptible pool is naturally exhausted, again making intervention irrelevant (due to being "too late"). Quantifying these scenarios requires a precise probabilistic model.

The dynamics of $(S_t, I_t)$ under the respective laws $\mathbb{P}^0$ and $\mathbb{P}^A$ follow continuous-time Markov chains with the following two transition channels:

$$\left\{\begin{array}{ll} \text{Infection } : S+I \to 2I & \text{with rate } \beta^j S_t I_t/M, \quad j=0,A; \\ \text{Recovery } : I \to R & \text{with rate } \gamma I_t. \end{array}\right\} \tag{5.3}$$

Above, $\beta^A < \beta^0$ is interpreted as lowered contact rate among Infecteds and Susceptibles in the intervention regime, which lowers the attack rate of the disease agent and hence reduces outbreak impact. The Markov chain $(S_t, I_t)$ described in (5.3) is readily simulatable using the Gillespie time-stepping algorithm [17], utilizing the fact that the sojourn times between state transitions have (state-dependent) Exponential distributions, and are independent of the next transition type. These simulations are however rather time-consuming, requiring $\mathcal{O}(M)$ Uniform draws. Consequently, efficient ranking of expected costs is important in applications.

**Remark 5.1.** *Since* (5.3) *implies that each individual infected period has an independent* $Exp(\gamma)$ *distribution it follows that*

$$\mathbb{E}[S_0 - S_T] = \gamma \mathbb{E}\big[\int_0^T I_t \, dt\big],$$

*so that* (5.1) *can also be interpreted as proportional to total expected infected-days.*

We note that in this example the input space $\mathcal{X}$ is discrete, which however requires minimal changes to our implementation of Algorithm 1. The biggest adjustment is the fact that the noise variances $\sigma^2_\ell(x)$ in (1.2) are unknown. Knowledge of $\sigma^2_\ell(x)$'s is crucial for training the GP covariance kernel $\mathcal{K}_\ell$, see e.g. (2.9). Indeed, while it is possible to simultaneously train $\mathcal{K}_\ell$ and a constant observation noise $\sigma$ (the latter is known as the "nugget" in GP literature), with state-dependent noise $\mathcal{K}$ is not identifiable. We resolve this issue through a batching procedure to estimate $\sigma^2_\ell(x)$ on-the-go. Namely, we re-use the same site $x \equiv (s,i)$ $r$-times, to obtain independent samples $y^{(1)}_\ell(x), \dots, y^{(r)}_\ell(x)$ from the corresponding $Y_\ell(x)$. This allows to estimate the conditional variance

$$\widetilde{\sigma}^2_\ell(x) := \frac{1}{r-1}\sum_{i=1}^r (y^{(i)}(x) - \bar{y}_\ell(x))^2, \qquad \text{where} \qquad \bar{y}_\ell(x) = \frac{1}{r}\sum_{i=1}^r y^{(i)}_\ell(x)$$

is the sample mean. Moreover, as shown in [38, Sec 4.4.2] we can treat the $r$ samples at $x$ as the single design entry $(x, \bar{y}_\ell(x))$ with noise variance $\widetilde{\sigma}^2_\ell(x)/r$. The resulting reduction in post-averaged design size by a factor of $r$ offers substantial computational speed-up in fitting and updating the kriging model. Formally, the EI step in Algorithm 1 is replaced with using $(x^{k+1}, \ell^{k+1}) = (x^{k+2}, \ell^{k+2}) = \dots = (x^{k+r}, \ell^{k+r})$ and re-computing the EI score once every $r$ ground-level iterations.

For our study we set $M = 2000$, $\beta^0 = 0.75, \beta^A = 0.5$, $\gamma = 0.5$. We further assume that the cost of intervention is $C^I = 0.25$ per susceptible. Figure 6 shows the resulting decision boundary $\partial\mathcal{C}$. In the light region, the relative cost of intervention is lower, and hence it is preferred to undertake action. For example, starting at $I_0 = 10, S_0 = 1800$, without any action the outbreak would affect more than 40% of the susceptible population (expected cost of about 800), while under social

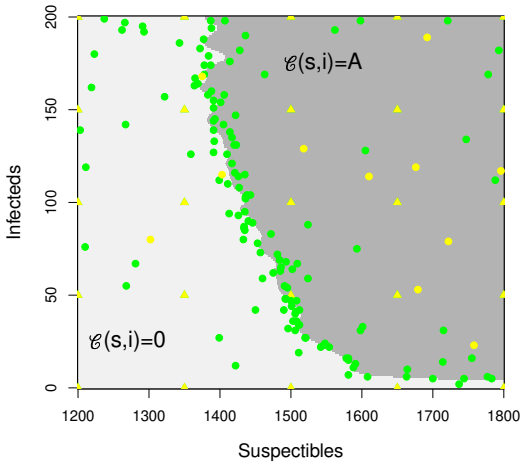


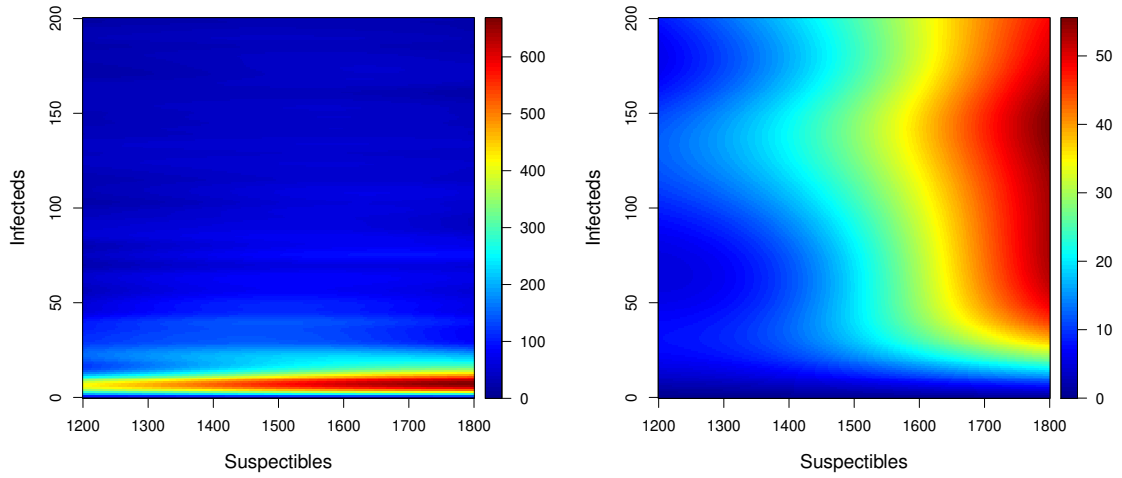

FIGURE 6. Fitted response boundary $\partial\mathcal{C}$ for the epidemic response example using the Gap-SUR expected improvement metric. The scatterplot indicates the design $\mathcal{Z}^{(K)}$ for $K = 200$; triangles indicate the initial design $\mathcal{Z}^{(K_0)}$, and circles the adaptively placed $(s,i)^{K_0:K}$ (green: $Y_0$; yellow: $Y_A$).

distancing, the impact would be about 60 infecteds (leading to much lower total expected cost of $60+C^I S_0 \simeq 510$). In the light region, wait-and-see approach has lower expected costs. For example at $I_0 = 50, S_0 = 1400$, the expected number of new infecteds without any action is 385 while the cost of countermeasures is bigger at $0.25 \times 1400 + 102 = 452$. Overall, Figure 6 shows that the optimal decision is very sensitive to the current number of susceptibles $S_0$. This feature is due to the fact that outbreaks are created when the infection rate dominates the recovery (reproductive ratio $\mathcal{R}_0 := (\beta^0/\gamma)(S_0/M)$ above 1). Hence, for a pool with more than 85% susceptibles ($S_0 > 1700$), the initial growth rate satisfies $\beta^0 S_0/M > \gamma$ and is likely to trigger an outbreak. However, as $S$ is lowered, the region where $\beta^0 S_0/M \simeq \gamma$ is approached, which makes social distancing unnecessary, as outbreak likelihood and severity diminishes. In particular, Figure 6 shows than no action is undertaken for $S_0 < 1350$. In the intermediate region, there is a nontrivial classifier boundary for determining $\mathcal{C}(s,i)$.

Figure 6 was generated by building an adaptive design using the Gap-SUR acquisition function and a total of $K = 200$ design sites, with $r = 100$ batched samples at each site. The input space was restricted to $\mathcal{X} = \{s \in \{1200,\ldots,1800\}, i \in \{0,200\}\}$. The initial design $\mathcal{Z}^{(K_0)}$ included $50 = 25 \times 2$ sites on the same rectangular $5 \times 5$ lattice for each of $Y_0, Y_A$. In this example, the noise levels $\sigma^2_\ell(s,i)$ are highly state-dependent, see Figure 7. The $\mu_0$ surface has much higher noise, with largest $\sigma^2_0(s,i)$ for $(s,i) \simeq (1800,5))$, whereas $\mu_A$ has largest noise in the top right corner. As a result, $\mathcal{Z}^{(K)}$ contains mostly samples from $Y_0$ and is denser towards the bottom of the Figure.

## 6. CONCLUSION

In this article we have constructed several efficient sequential design strategies for the problem of determining the minimum among $L \geq 2$ response surfaces. Our Gap-SUR heuristic connects (1.1)




$\widetilde{\sigma}_0(s,i)$ $\widetilde{\sigma}_A(s,i)$

FIGURE 7. Estimated noise standard deviations $\widetilde{\sigma}_\ell(s,i)$ for the epidemic response example in the no-countermeasures (left panel, $\ell = 0$) and action (right panel, $\ell = A$) regimes. Note the different color scales of the two panels, with $\sigma_0(\cdot) \gg \sigma_A(\cdot)$ for all $(s,i)$.

to contour-finding and Bayesian optimization, providing a new application of the stepwise uncertainty reduction framework [9]. Our Gap-UCB heuristic mimics multi-armed bandits by treating all possible sampling pairs in $\mathcal{X} \times \mathfrak{L}$ as arms, and trying to balance arm exploration and exploitation.

Our approach is based on the kriging framework, but this is primarily for convenience and is not crucial. To this end, instead of a Bayesian formulation, one could use a maximum-likelihood method to fit $\widehat{\mu}_\ell(\cdot)$, replacing the posterior $M_\ell(x)$ with the point estimator and its standard error. Hence, many other regression frameworks could be selected. However, computational efficiency and the sequential framework place several efficiency restrictions on possible ways to modeling $\mu_\ell(\cdot)$. On the one hand, we need strong consistency, i.e. the convergence of the respective classifier $\hat{\mathcal{C}}^{(K)} \to \mathcal{C}$ as $K \to \infty$. In particular, the regression method must be nonparametric and localized. On the other hand, we wish for a sequential procedure that allows for efficient updating rules in moving from $\hat{\mathcal{C}}^{(k)}$ to $\hat{\mathcal{C}}^{(k+1)}$. Lastly, in practical settings further challenges such as heteroscedasticity, non-Gaussian samplers $Y_\ell$, and heterogenous structure of the response surface are important.

One suitable alternative to GP's is local regression or Loess [42], which is a nonparametric regression framework that fits pointwise linear regression models for $\mu_\ell(x)$. Loess is efficient and well-suited for heteroscedastic contexts with unknown noise distributions as in Section 5. It also automatically generates the posterior mean and variance of the fit (allowing to use the derived formulas based on $\widehat{\mu}_\ell(x)$ and $\delta_\ell(x)$). However, Loess is not updatable, creating computational bottlenecks if many design augmentation iterations are to be used. At the same time fitting is extremely fast, so depending on the implementation it might still be competitive with more sophisticated methods. In this spirit, piecewise linear regression (which first partitions $\mathcal{X}$ into several cells and then carries out least-squares regression in each cell) is updatable via the Sherman-Morrison-Woodbury formulas and could be employed if there is a clear partitioning strategy available.

We further note that GP kriging is just a convenient interim surrogate for building the experimental design. Consequently, once a good design $\mathcal{Z}$ is generated, one could switch to a different response surface model to build a final estimate of the $\mu_\ell$'s and hence $\hat{\mathcal{C}}$. For example, the treed GP approach [22] allows for a higher-fidelity fit for the response surfaces, especially in cases where the underlying smoothness (specified by the covariance kernel) strongly varies across $\mathcal{X}$. Because treed GP models are expensive to fit, one could compromise by using vanilla GP during DoE and treed GP for the final estimate of $\hat{\mathcal{C}}$.

Another fruitful extension would be to investigate ranking algorithms in the fixed confidence setting. As presented, the sequential ranking algorithm is in the fixed budget setting, augmenting the design until a pre-specified size $K$. Practically, it is often desirable to prescribe adaptive, data-driven termination by targeting a pre-set confidence level. A good termination criterion should take both accuracy and efficiency into account, ensuring the accuracy of $\widehat{\mu}_\ell(x)$ and also anticipating low information gain from further sampling steps. One proposed termination criterion is to keep track of the evolution of the empirical loss $\mathcal{EL}(\hat{\mathcal{C}}^{(k)})$, and terminate once $\mathcal{EL}(\hat{\mathcal{C}}^{(k)}) - \mathcal{EL}(\hat{\mathcal{C}}^{(k+1)})$ is small enough. This is equivalent to minimizing $L_k := \mathcal{EL}(\hat{\mathcal{C}}^{(k)}) + \underline{\epsilon} k$, where $\underline{\epsilon} > 0$ is a parameter for cost of simulations; the more we care about efficiency, the larger the $\underline{\epsilon}$ is. When the design size $k$ is small, the first term will dominate, so $L_k$ is expected to first decrease in $k$. As $k \to \infty$, the rate of improvement in the loss function shrinks so that eventually $L_k$ will be increasing. However, we find that $\mathcal{EL}(\hat{\mathcal{C}}^{(k)})$ is quite noisy, especially if the kriging models are re-trained across stages. In that sense, the termination criterion needs to be robust enough to generate sufficiently strong (ad hoc) guarantees that a certain tolerance threshold has truly been achieved.

## References


[1] R Aid, L Campi, N Langrené, and H Pham. "A probabilistic numerical method for optimal multiple switching problem and application to investments in electricity generation". In: *SIAM Journal of Financial Mathematics* 5.1 (2014), pp. 191–231.

[2] R. Anderson and D. Milutinovic. "A stochastic approach to Dubins feedback control for target tracking". In: *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE. 2011, pp. 3917–3922.

[3] P. Auer, N. Cesa-Bianchi, and P. Fischer. "Finite-time analysis of the multiarmed bandit problem". In: *Machine learning* 47.2-3 (2002), pp. 235–256.

[4] J. Azimi, A. Fern, and X. Z. Fern. "Batch Bayesian optimization via simulation matching". In: *Advances in Neural Information Processing Systems 23*. Ed. by J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta. Curran Associates, Inc., 2010, pp. 109–117.

[5] J. Bect, D. Ginsbourger, L. Li, V. Picheny, and E. Vazquez. "Sequential design of computer experiments for the estimation of a probability of failure". In: *Statistics and Computing* 22.3 (2012), pp. 773–793.

[6] S. Bubeck, R. Munos, and G. Stoltz. "Pure exploration in finitely-armed and continuous-armed bandits". In: *Theoretical Computer Science* 412.19 (2011), pp. 1832–1852.

[7] S. Bubeck, R. Munos, G. Stoltz, and C. Szepesvari. "X-armed bandits". In: *The Journal of Machine Learning Research* 12 (2011), pp. 1655–1695.

[8] A. Carpentier, A. Lazaric, M. Ghavamzadeh, R. Munos, and P. Auer. "Upper-confidence-bound algorithms for active learning in multi-armed bandits". In: *Algorithmic Learning Theory*. Springer. 2011, pp. 189–203.

[9] C. Chevalier, J. Bect, D. Ginsbourger, E. Vazquez, V. Picheny, and Y. Richet. "Fast parallel kriging-based stepwise uncertainty reduction with application to the identification of an excursion set". In: *Technometrics* 56.4 (2014), pp. 455–465.

[10] C. Chevalier, D. Ginsbourger, and X. Emery. "Corrected kriging update formulae for batch-sequential data assimilation". In: *Mathematics of Planet Earth*. Springer, 2014, pp. 119–122.

[11] H. A. Chipman, E. I. George, and R. E. McCulloch. "BART: Bayesian additive regression trees". In: *The Annals of Applied Statistics* 4.1 (2010), pp. 266–298.

[12] D. A. Cohn. "Neural network exploration using optimal experiment design". In: *Neural networks* 9.6 (1996), pp. 1071–1083.

[13] D. Egloff. "Monte Carlo algorithms for optimal stopping and statistical learning". In: *Ann. Appl. Probab.* 15.2 (2005), pp. 1396–1432.

[14] J. A. Fuemmeler and V. V. Veeravalli. "Smart sleeping policies for energy efficient tracking in sensor networks". In: *IEEE Trans. Signal Process.* 56.5 (2008), pp. 2091–2101.

[15] V. Gabillon, M. Ghavamzadeh, A. Lazaric, and S. Bubeck. "Multi-bandit best arm identification". In: *Advances in Neural Information Processing Systems*. 2011, pp. 2222–2230.

[16] S. E. Gano, J. E. Renaud, J. D. Martin, and T. W. Simpson. "Update strategies for kriging models used in variable fidelity optimization". In: *Structural and Multidisciplinary Optimization* 32.4 (2006), pp. 287–298.

[17] D. T. Gillespie. "Exact stochastic simulation of coupled chemical reactions". In: *Journal of Physical Chemistry* 81.25 (1977), pp. 2340–2361.

[18] R. B. Gramacy and D. W. Apley. "Local Gaussian process approximation for large computer experiments". In: *Journal of Computational and Graphical Statistics* 24.2 (2015), pp. 561–578.

[19] R. B. Gramacy and H. K. H. Lee. "Adaptive design and analysis of supercomputer experiments". In: *Technometrics* 51.2 (2009), pp. 130–145.

[20] R. B. Gramacy and M. Ludkovski. "Sequential design for optimal stopping problems". In: *SIAM Journal on Financial Mathematics* 6.1 (2015), pp. 748–775.

[21] R. B. Gramacy and N. Polson. "Particle learning of Gaussian process models for sequential design and optimization". In: *Journal of Computational and Graphical Statistics* 20.1 (2011), pp. 102–118.

[22] R. B. Gramacy and M. Taddy. "tgp, an R package for treed Gaussian process models". In: *Journal of Statistical Software* 33 (2012), pp. 1–48.

[23] R. B. Gramacy, M. Taddy, and N. Polson. "Dynamic trees for learning and design". In: *Journal of the American Statistical Association* 106.493 (2011), pp. 109–123.

[24] S. Grünewälder, J.-Y. Audibert, M. Opper, and J. Shawe-Taylor. "Regret bounds for Gaussian process bandit problems". In: *International Conference on Artificial Intelligence and Statistics*. 2010, pp. 273–280.

[25] J. Hespanha, M. Ludkovski, and S. Quintero. *Stochastic optimal coordination of small UAVs for target tracking using regression-based dynamic programming*. Available at `www.pstat.ucsb.edu/faculty/ludkovski/HLQ15.pdf`. 2015.

[26] M. W. Hoffman, B. Shahriari, and N. de Freitas. *Exploiting correlation and budget constraints in Bayesian multi-armed bandit optimization*. Tech. rep. arXiv preprint arXiv:1303.6746, 2013.

[27] D. Jones, M. Schonlau, and W. Welch. "Efficient global optimization of expensive black-box functions". In: *Journal of Global optimization* 13.4 (1998), pp. 455–492.

[28] A. Joshi, F. Porikli, and N. Papanikolopoulos. "Multi-class active learning for image classification". In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE. 2009, pp. 2372–2379.

[29] G. Lai, M. X. Wang, S. Kekre, A. Scheller-Wolf, and N. Secomandi. "Valuation of storage at a liquefied natural gas terminal". In: *Operations Research* 59.3 (2011), pp. 602–616.

[30] J. Lin and M. Ludkovski. "Sequential Bayesian inference in hidden Markov stochastic kinetic models with application to detection and response to seasonal epidemics". In: *Statistics and Computing* 24.6 (2014), pp. 1047–1062.

[31] F. Longstaff and E. Schwartz. "Valuing American options by simulations: a simple least squares approach". In: *The Review of Financial Studies* 14 (2001), pp. 113–148.

[32] M. Ludkovski and J. Niemi. "Optimal dynamic policies for influenza management". In: *Statistical Communications in Infectious Diseases* 2(1) (2010), article 5 (electronic).

[33] M. Ludkovski and J. Niemi. "Optimal disease outbreak decisions using stochastic simulation". In: *Simulation Conference (WSC), Proceedings of the 2011 Winter*. IEEE. 2011, pp. 3844–3853.

[34] D. MacKay. "Information-based objective functions for active data selection". In: *Neural computation* 4.4 (1992), pp. 590–604.

[35] M. McKay, R. Beckman, and W. Conover. "Comparison of three methods for selecting values of input variables in the analysis of output from a computer code". In: *Technometrics* 21 (1979), pp. 239–245.

[36] N. Meinshausen and B. Hambly. "Monte Carlo methods for the valuation of multiple-exercise options". In: *Mathematical Finance* 14.4 (2004), pp. 557–583.

[37] D. Merl, R. Johnson, R. Gramacy, and M. Mangel. "A statistical framework for the adaptive management of epidemiological interventions". In: *PLoS ONE* 4(6) (2009), e5087.

[38] V. Picheny and D. Ginsbourger. "A Nonstationary space-time Gaussian Process model for partially converged simulations". In: *SIAM/ASA Journal on Uncertainty Quantification* 1.1 (2013), pp. 57–78.

[39] V. Picheny, D. Ginsbourger, Y. Richet, and G. Caplin. "Quantile-based optimization of noisy computer experiments with tunable precision". In: *Technometrics* 55.1 (2013), pp. 2–13.

[40] V. Picheny, D. Ginsbourger, O. Roustant, R. T. Haftka, and N.-H. Kim. "Adaptive designs of experiments for accurate approximation of a target region". In: *Journal of Mechanical Design* 132 (2010), p. 071008.

[41] P. Ranjan, D. Bingham, and G. Michailidis. "Sequential experiment design for contour estimation from complex computer codes". In: *Technometrics* 50.4 (2008), pp. 527–541.

[42] B. Ripley. *Loess{stats}: Local Polynomial Regression Fitting*. R package verion 3.0.1.

[43] O. Roustant, D. Ginsbourger, and Y. Deville. "DiceKriging, DiceOptim: Two R packages for the analysis of computer experiments by kriging-based metamodeling and optimization". In: *Journal of Statistical Software* 51.1 (2012), pp. 1–51.

[44] N. Srinivas, A. Krause, S. Kakade, and M. Seeger. "Information-theoretic regret bounds for Gaussian Process optimization in the bandit setting". In: *IEEE Transactions on Information Theory* 58.5 (2012), pp. 3250 –3265.

[45] C. K. Williams and C. E. Rasmussen. *Gaussian Processes for machine learning*. MIT Press, 2006.

[46] M. Zervos, T. C. Johnson, and F. Alazemi. "Buy-low and sell-high investment strategies". In: *Mathematical Finance* 23.3 (2013), pp. 560–578.