arXiv:2407.18885v1 [stat.ME] 26 Jul 2024

Simulation Experiment Design for Calibration via Active Learning

Özge Sürer[1,*]

[1]Farmer School of Business, Miami University, Oxford, OH 45056, USA
[*]Corresponding author can be reached at surero@miamioh.edu

July 29, 2024

Abstract

Simulation models often have parameters as input and return outputs to understand the behavior of complex systems. Calibration is the process of estimating the values of the parameters in a simulation model in light of observed data from the system that is being simulated. When simulation models are expensive, emulators are built with simulation data as a computationally efficient approximation of an expensive model. An emulator then can be used to predict model outputs, instead of repeatedly running an expensive simulation model during the calibration process. Sequential design with an intelligent selection criterion can guide the process of collecting simulation data to build an emulator, making the calibration process more efficient and effective. This article proposes two novel criteria for sequentially acquiring new simulation data in an active learning setting by considering uncertainties on the posterior density of parameters. Analysis of several simulation experiments and real-data simulation experiments from epidemiology demonstrates that proposed approaches result in improved posterior and field predictions.

Keywords: acquisition, Bayesian calibration, emulation, sequential design, uncertainty quantification

1 Introduction

Simulation models are pervasive in many engineering and science disciplines to explain the behavior of complex systems. Examples include the simulation of inventory/supply chain systems (Hong and Nelson 2006), manufacturing systems (Chen et al. 2018), storm surge (Plumlee et al. 2021), and physics (Sürer et al. 2022). In these and many other cases, in addition to controllable inputs (aka design inputs), simulation models take user-specified calibration parameters as input and return outputs that can be used to understand reality. However, these parameters are often unknown and need to be inferred using observed data from a real physical/field experiment. Calibration is a way to infer parameters to ensure that simulation outputs accurately represent the real-world system that is being simulated. Calibration becomes a more challenging problem when a single simulation evaluation requires a significant amount of computational resources and time. Therefore, careful selection of the simulation experiments to run is a critical concern.

This work considers Bayesian calibration, which is a specialized form of calibration that offers a way to quantify uncertainty in both parameters and predictions of quantities of interest. In a standard Bayesian calibration, emulators are built via statistical models such as Gaussian processes (Gramacy 2020, Rasmussen and Williams 2005) or Bayesian tree techniques (Chipman et al. 1998, Gramacy and Lee 2008) to mimic the behaviors of the computationally expensive simulation model. Emulators then provide predictions of simulation outputs at any input configuration without running the expensive simulation model. Emulators are built with a simulation data set consisting of a set of inputs (called design) and corresponding simulation outputs. Once an emulator is constructed, it is used to facilitate the calibration process (Higdon et al. 2004, Kennedy and O'Hagan 2001). Hence, the precision of the calibration process relies on the emulator's accuracy. Common techniques to generate designs include random sampling and space-filling designs such as Latin hypercube sampling (LHS, McKay et al. (1979)), minimax designs (Johnson et al. 1990) and the optimized versions (Joseph et al. 2015b); see Santner et al. (2018) for a detailed survey. One drawback of such designs is that the (unknown) input region of interest may not be adequately explored, especially in high-dimensional spaces, leading to potential gaps in the emulator's predictive ability. Simulation experiment design, the topic of this article, should be selected with care to achieve precise calibration inference with a limited number of simulation runs.

Sequential design or active learning allows adaptive simulation data collection based on the simulation data set that has already been gathered. In a sequential design, the decision to sample additional data points is often based on statistical criteria, called acquisition function. The adaptability of the sequential design allows practitioners to focus on regions of interest or refine the experiment as it progresses. Moreover, sequential designs are often more resource-efficient than one-shot design procedures mentioned above since the iterative data collection can be terminated when a sufficient level of preciseness is achieved (Lam and Notz 2008). Bayesian optimization (BO) (Frazier 2018) is a common sequential approach to optimize a black-box, computationally expensive function. In BO, an emulator is used to approximate the unknown objective function and then an acquisition function guides the selection of a new point to evaluate the objective function next. Expected improvement and probability of improvement are commonly used acquisition functions in BO to select the input with the highest expectation/probability for improvement over the best objective value obtained thus far (Jones et al. 1998).

Our approach stands out due to its distinctive utilization of the emulation strategy employed in modern calibration techniques unlike many existing methods in the literature on sequential design, which often do not directly emulate the simulation model itself but rather focus on emulating objective functions (i.e., goodness-of-fit measures). For instance, in the calibration context, Kandasamy et al. (2015) build an emulator of the log-likelihood to estimate the posterior density of parameters. Joseph et al. (2015a) and Joseph et al. (2019) introduce an energy design criterion to obtain a sample from the probability density function using an emulator of the density itself. Constructing emulators for objective functions can pose challenges compared to directly developing emulators of simulation models. One downside of direct emulation of the objective function is that the internal structure of the complex simulation models is often missed. Moreover, transferring the simulation output to obtain the objective function value brings extra complexity to the inference problem. Successful examples of integrating emulators of simulation models within a sequential framework can be found in Damblin et al. (2018), Koermer et al. (2023), Lartaud et al. (2024), Sürer et al. (2024). This work constructs emulators as proxies for simulation models to make the calibration process more efficient and effective by leveraging the information hidden in the structure of the models.

Sequential design has been used in the literature to build globally accurate emulators of simulation models. For a global emulator, one natural acquisition function is to choose the next input with the highest emulation variance (Sacks et al. 1989, Seo et al. 2000). In parallel to this, MacKay (1992) employs an active learning setting where inputs are selected based on the entropy criterion. This approach demonstrates that selecting inputs with the highest emulation variance approximates a maximum entropy design. However, since accurate predictions are desired across the entire input space, a criterion relying on a single point's uncertainty often leads to suboptimal results. The integrated mean squared prediction error (IMSPE), which considers the aggregated emulator uncertainty across the input space, is one of the most common acquisition functions in this field; for a more thorough review see Gramacy (2020). Since it is theoretically sound and applicable in practice, there are numerous developments of IMSPE; see examples in Binois et al. (2018), Cole et al. (2021), Sauer et al. (2023). Because IMSPE for a general-purpose emulator does not take into account the observed data, it does not bring any additional advantage for calibration inference. Recently, Koermer et al. (2023) propose a novel IMSPE criterion within the Kennedy and O'Hagan calibration framework to improve predictions, and Lartaud et al. (2024) develop a weighted IMSPE criterion for Bayesian inverse problems. In this work, our focus is on the aggregated uncertainty in the estimate of the posterior density of parameters rather than the emulator uncertainty.

In a recent work, Sürer et al. (2024) propose the expected integrated variance (EIVAR) criterion to accurately learn the posterior density of parameters. In their setting, when run at a parameter, the simulation model returns a high-dimensional output consisting of multiple responses collected on a set of fixed design inputs. Consequently, the only simulation inputs involved are parameters, and EIVAR is derived to sequentially acquire a new parameter and its high-dimensional output. However, in many settings, a simulation model is often a function of both a parameter and a design input (see the problem relating to a diaper line from the Procter & Gamble Company in Krishna et al. (2022), examples from nuclear physics in Phillips et al. (2021), and the industrial application involving a chemical process in Koermer et al. (2023)). A design input is shared by both the simulation model and the field experiment. At a set of design inputs, which we call field data design inputs throughout the paper, field experiments are conducted to explore a physical system, and field data measured from the experiments are used to infer the unknown parameters of the simulation model. In this work, we first derive the EIVAR criterion to allow the acquisition of a simulation input consisting of a parameter and a design input simultaneously. The acquisition function encourages the selection of simulation design inputs aligned with field data design inputs around the parameter region of interest, and we demonstrate that such acquisitions lead to improved posterior predictions. However, when the calibration experiments target field predictions at unseen design inputs as well, focusing only on the field data design inputs does not allow exploration of the entire design space. For improved field predictions under novel design inputs, we propose another acquisition function by considering the uncertainties in the posterior obtained with unseen design inputs. The acquisition function prefers matching the simulation design inputs with the field data design inputs as well as exploring the remaining design space. Similar to our findings, Ranjan et al. (2011) suggest the alignment of the simulation design inputs with the field data design inputs for the field prediction. However, only a one-step simulation data collection is empirically conducted in Ranjan et al. (2011) and a systematic way is not proposed to acquire a new input. Our sequential approach can be considered an automated way to find how to allocate the simulation data to both existing and unseen design inputs for improved calibration inference.

The remainder of the paper is organized as follows. Section 2 presents the main steps of the sequential design procedure as well as background on Bayesian calibration and Gaussian processes. Section 3 contains our methodologic contributions. In Section 4, we demonstrate the benefits of our proposed methods by analyzing results from several simulation experiments including synthetic models and a COVID-19 simulation model. Conclusions are presented in Section 5.

2 Background

This section overviews the proposed sequential algorithm, Bayesian calibration, and Gaussian process emulators.

2.1 Sequential Experimental Design

We use the following notations throughout the paper. Boldface characters are used to represent vectors and matrices. We label the length-q vector of design inputs $\mathbf{x}$ and the length-p vector of calibration parameters $\boldsymbol{\theta}$. The simulation model, represented with $\eta(\cdot, \cdot)$, is a function that takes design input $\mathbf{x}$ in space $\mathcal{X} \subset \mathbb{R}^q$ and parameter $\boldsymbol{\theta}$ in space $\Theta \subset \mathbb{R}^p$ as an input and returns one-dimensional simulation output $\eta\left(\mathbf{x}, \boldsymbol{\theta}\right) \in \mathbb{R}$. Our goal is to sequentially collect simulation outputs from n acquired inputs for improved calibration inference.

Algorithm 1: Sequential experimental design

```
1 Input: An initial sample size $n_0$, a total number of acquisitions $n$, a simulation model $\eta(\cdot,\cdot)$, and
    an acquisition function $\mathcal{A}_t(\cdot,\cdot)$
2 Initialize $\mathcal{D}_1 = \{((\mathbf{x}_i, \boldsymbol{\theta}_i), \eta(\mathbf{x}_i, \boldsymbol{\theta}_i)) : i = 1, \ldots, n_0\}$
3 for $t = 1, \ldots, n$ do
4     Fit an emulator with $\mathcal{D}_t$
5     Generate candidate solutions $\mathcal{L}_t$
6     Select $(\mathbf{x}^{\text{new}}, \boldsymbol{\theta}^{\text{new}}) \in \underset{(\mathbf{x}^*, \boldsymbol{\theta}^*) \in \mathcal{L}_t}{\arg\min} \mathcal{A}_t(\mathbf{x}^*, \boldsymbol{\theta}^*)$
7     Evaluate $\eta(\mathbf{x}^{\text{new}}, \boldsymbol{\theta}^{\text{new}})$
8     Update $\mathcal{D}_{t+1} \leftarrow \mathcal{D}_t \cup ((\mathbf{x}^{\text{new}}, \boldsymbol{\theta}^{\text{new}}), \eta(\mathbf{x}^{\text{new}}, \boldsymbol{\theta}^{\text{new}}))$
9 Output: Simulation data $\mathcal{D}_{n+1}$ along with the emulator fitted with $\mathcal{D}_{n+1}$
```

The proposed sequential design is summarized in Algorithm 1. The algorithm starts with an initial set of n_0 simulation inputs and their outputs stored in $\mathcal{D}_1 = \{((\mathbf{x}_i, \boldsymbol{\theta}_i), \eta(\mathbf{x}_i, \boldsymbol{\theta}_i)) : i = 1, \ldots, n_0\}$ (see line 2). The initial simulation data set is typically sampled randomly from a prior distribution or through a space-filling design. During each iteration indexed by t, a new input is acquired and the simulation model is evaluated with the new input. The simulation data set $\mathcal{D}_{t+1} = \{((\mathbf{x}_i, \boldsymbol{\theta}_i), \eta(\mathbf{x}_i, \boldsymbol{\theta}_i)) : i = 1, \ldots, n_t\}$, where $n_t = n_0 + t$, keeps all the simulation data obtained by the end of iteration t (line 8). At the beginning of each iteration, an emulator is fitted to the simulation data set $\mathcal{D}_t$ (see Section 2.3) and it is used to construct the acquisition function $\mathcal{A}_t(\cdot,\cdot)$ (see Section 3). To avoid difficult numerical optimization, the acquisition function $\mathcal{A}_t(\cdot,\cdot)$ is minimized over a discrete set of inputs $\mathcal{L}_t$ to determine the next best input to evaluate the simulation model (see lines 5–7). Although the termination criteria for a sequential design can vary depending on the calibration objective, we use a fixed budget of n acquired simulation outputs for comparison purposes.

2.2 Bayesian Calibration

The purpose of Bayesian calibration is to infer unknown calibration parameters using data from the field experiment and to characterize uncertainties in inferred parameters and associated predictions. Let $\mathbf{x}_i^f$ be the design input where the field experiment is conducted and $y\left(\mathbf{x}_i^f\right)$ denote the observed data from the field experiment for $i = 1, \ldots, d$. We first model the data from the field experiment

$$y\left(\mathbf{x}_i^f\right) = \eta\left(\mathbf{x}_i^f, \boldsymbol{\theta}\right) + \epsilon, \tag{1}$$

where $\epsilon \sim \mathrm{N}\left(0, \sigma^2\right)$ denotes the residual error. Let $\mathbf{\Sigma}$ be a $d \times d$ diagonal error covariance matrix with diagonal elements σ^2. We assume σ^2 is known to account for the uncertainty in the difference between the data and the model and extend our criterion for the case of unknown σ^2 with a discrepancy term in Section 3. Our results are built upon the assumption that the observation noise ϵ follows a normal distribution; we leave the derivations of proposed acquisition functions for different distributions of the residual error as future development.

In the Bayesian calibration framework, we are interested in the quantity $p\left(\boldsymbol{\theta}|\mathbf{y}\right)$, which is the posterior probability density of parameter $\boldsymbol{\theta}$ given field data $\mathbf{y} = \left(y\left(\mathbf{x}_1^f\right), \ldots, y\left(\mathbf{x}_d^f\right)\right)^\top$. The initial knowledge about parameter $\boldsymbol{\theta}$ is represented by the prior probability density $p\left(\boldsymbol{\theta}\right)$, which is typically a known, closed-form function of the parameter. Based on Bayes' rule, the posterior density has the form

$$p\left(\boldsymbol{\theta}|\mathbf{y}\right) = \frac{p\left(\mathbf{y}|\boldsymbol{\theta}\right)p\left(\boldsymbol{\theta}\right)}{\int_\Theta p\left(\mathbf{y}|\boldsymbol{\theta}'\right)p\left(\boldsymbol{\theta}'\right)\mathrm{d}\boldsymbol{\theta}'} \propto \tilde{p}\left(\boldsymbol{\theta}|\mathbf{y}\right) = p\left(\mathbf{y}|\boldsymbol{\theta}\right)p\left(\boldsymbol{\theta}\right), \tag{2}$$

where $p\left(\mathbf{y}|\boldsymbol{\theta}\right)$ is the likelihood function indicating the agreement between the simulation output at $\boldsymbol{\theta}$ and field data $\mathbf{y}$ and $\tilde{p}\left(\boldsymbol{\theta}|\mathbf{y}\right)$ represents the unnormalized posterior. In a typical Bayesian calibration, Markov chain Monte Carlo (MCMC) methods (Gelman et al. 2004) are employed to produce samples from the posterior. Since the posterior is analytically intractable in complex models, the unnormalized posterior is used to represent the posterior up to a constant multiplier within MCMC schemes. This work considers the uncertainty in the estimate of the unnormalized posterior $\tilde{p}(\boldsymbol{\theta}|\mathbf{y})$, which is referred to as the posterior for brevity throughout the remainder of this paper. The differences between the field data and the simulation outputs are assumed to follow a multivariate normal (MVN) distribution due to (1), and the likelihood satisfies

$$p\left(\mathbf{y}|\boldsymbol{\theta}\right) = (2\pi)^{-d/2}|\mathbf{\Sigma}|^{-1/2}\exp\left(-\frac{1}{2}\left(\mathbf{y} - \boldsymbol{\eta}\left(\boldsymbol{\theta}\right)\right)^\top\mathbf{\Sigma}^{-1}\left(\mathbf{y} - \boldsymbol{\eta}\left(\boldsymbol{\theta}\right)\right)\right), \tag{3}$$

in which requires the simulation output $\boldsymbol{\eta}(\boldsymbol{\theta}) = \left(\eta\left(\mathbf{x}_1^f, \boldsymbol{\theta}\right), \ldots, \eta\left(\mathbf{x}_d^f, \boldsymbol{\theta}\right)\right)^\top$ at field data design inputs. To produce posterior samples, MCMC techniques have to evaluate $p\left(\mathbf{y}|\boldsymbol{\theta}\right)$ many times (usually thousands or millions of evaluations) for candidate values of $\boldsymbol{\theta}$. However, the evaluation of a simulation model with any candidate parameter becomes computationally prohibitive when a single simulation run takes a substantial amount of computational time. To solve this problem, the emulator of a simulation model introduced in Section 2.3 can be used to estimate the posterior at any $\boldsymbol{\theta}$.

2.3 Gaussian Process Model

We consider Gaussian process (GP) modeling to build an emulator of the simulation model at each iteration of the sequential procedure based on the simulation data set $\mathcal{D}_t$ as shown in line 4 of Algorithm 1. GP emulators are commonly used for calibrating simulation models since GPs can provide both a predictive mean and variance for quantifying uncertainties. Our contribution involves the integration of the emulator with two novel acquisition functions for improved calibration inference via active learning (see Section 3). For simplicity, we assume a zero-mean GP prior with the covariance defined by a positive definite kernel function $k_t(\cdot, \cdot) = \tau_t^2 c(\cdot, \cdot; \boldsymbol{\zeta}_t)$, a scaling parameter τ_t^2 and a lengthscale parameter $\boldsymbol{\zeta}_t = (\zeta_{t,1}, \ldots, \zeta_{t,q+p})^\top$. A scaling parameter τ_t^2 controls the magnitude of the range of the simulation output represented by the GP to capture variations in the data, whereas a lengthscale parameter $\boldsymbol{\zeta}_t$ controls the smoothness of the output (see Chapter 5.2 of Gramacy (2020) for a detailed survey of GP hyperparameters). The covariance can be parameterized by many different choices of kernel functions such as Gaussian and Matérn (Rasmussen and Williams 2005, Santner et al. 2018). The Gaussian kernel function is one of the most popular kernel functions due to its flexibility and theoretical properties and it typically works well for interpolating smooth functions. However, it is argued that strong smoothness assumptions might be unrealistic for many physical processes. Matérn class of kernel functions is recommended to capture the variability of the underlying function better; see Chapter 4 of Rasmussen and Williams (2005) for a detailed survey on different kernel functions. In this work, we use the separable version of the Matérn correlation function with smoothness parameter 1.5 such that

$$c(\mathbf{z}, \mathbf{z}'; \boldsymbol{\zeta}_t) = \prod_{l=1}^{q+p} \left[(1 + |(z_l - z_l') \exp(\zeta_{t,l})|) \exp\left(- \exp(\zeta_{t,l}) |z_l - z_l'| \right) \right], \tag{4}$$

where $\mathbf{z} = (z_1, \ldots, z_{q+p})^\top = \left(\mathbf{x}^\top, \boldsymbol{\theta}^\top\right)^\top$ denotes a vector of size $q + p$ simulation input for notational simplicity. The choice of correlation function does not impact the rationale of the proposed acquisition functions.

Let the $n_t \times (q+p)$ matrix $\mathbf{z}_{1:n_t} = (\mathbf{z}_1, \ldots, \mathbf{z}_{n_t})^\top$ represent the inputs where the simulation model has been evaluated. The simulation outputs are stored in $\boldsymbol{\eta}_t = (\eta(\mathbf{x}_1, \boldsymbol{\theta}_1), \ldots, \eta(\mathbf{x}_{n_t}, \boldsymbol{\theta}_{n_t}))^\top$. According to the GP prior, the joint distribution of the simulation outputs $\boldsymbol{\eta}_t$ and the output $\eta(\mathbf{x}, \boldsymbol{\theta})$ at an unseen input $\mathbf{z} = \left(\mathbf{x}^\top, \boldsymbol{\theta}^\top\right)^\top$ is MVN distribution such that

$$\begin{bmatrix} \boldsymbol{\eta}_t \\ \eta(\mathbf{x}, \boldsymbol{\theta}) \end{bmatrix} \sim \text{MVN}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K}_t & \mathbf{k}_t(\mathbf{z}) \\ \mathbf{k}_t(\mathbf{z})^\top & k_t(\mathbf{z}, \mathbf{z}) \end{bmatrix}\right). \tag{5}$$

Here, $\mathbf{k}_t(\mathbf{z}) = (k_t(\mathbf{z}, \mathbf{z}_1), \ldots, k_t(\mathbf{z}, \mathbf{z}_{n_t}))^\top$ is comprised of cross-kernel evaluations between $\mathbf{z}$ and $\mathbf{z}_{1:n_t}$ and $\mathbf{K}_t$ is the $n_t \times n_t$ matrix with ij coordinates $k_t(\mathbf{z}_i, \mathbf{z}_j) + \upsilon\delta_{i=j}$ for $1 \leq i, j \leq n_t$. In addition, $\upsilon > 0$ is a nugget parameter and $\delta_{i=j}$ is the Kronecker delta function with value 1 if $i = j$ and value 0 otherwise. The nugget parameter υ is added to the diagonal elements of $\mathbf{K}_t$ to ensure the positive definiteness of the resulting matrix in the case of two identical inputs $\mathbf{z}_i = \mathbf{z}_j$ for improved numerical stability. Conditioning the joint GP prior distribution on $\boldsymbol{\eta}_t$ (see appendix A.2 of Rasmussen and Williams (2005) for further details) reveals that the predictive distribution $\eta(\mathbf{x}, \boldsymbol{\theta})|\boldsymbol{\eta}_t$ is Gaussian with mean $m_t(\mathbf{z})$ and variance $\varsigma_t^2(\mathbf{z})$ such that $\eta(\mathbf{x}, \boldsymbol{\theta})|\boldsymbol{\eta}_t \sim \text{N}\left(m_t(\mathbf{z}), \varsigma_t^2(\mathbf{z})\right)$ where

$$m_t(\mathbf{z}) = \mathbf{k}_t(\mathbf{z})^\top \mathbf{K}_t^{-1} \boldsymbol{\eta}_t \text{ and } \varsigma_t^2(\mathbf{z}) = k_t(\mathbf{z}, \mathbf{z}) - \mathbf{k}_t(\mathbf{z})^\top \mathbf{K}_t^{-1} \mathbf{k}_t(\mathbf{z}). \tag{6}$$

The emulator provides a probabilistic representation of the simulation output at design inputs with mean $\boldsymbol{\mu}_t(\cdot)$ and covariance matrix $\mathbf{S}_t(\cdot)$ such that

$$\boldsymbol{\eta}(\boldsymbol{\theta})|\mathcal{D}_t \sim \text{MVN}\left(\boldsymbol{\mu}_t(\boldsymbol{\theta}), \mathbf{S}_t(\boldsymbol{\theta})\right), \tag{7}$$

where $\boldsymbol{\mu}_t(\boldsymbol{\theta}) = \left(m_t\left(\mathbf{z}_1^f\right), \ldots, m_t\left(\mathbf{z}_d^f\right)\right)^\top$ and the ith diagonal element of $\mathbf{S}_t(\boldsymbol{\theta})$ is $\varsigma_t^2\left(\mathbf{z}_i^f\right)$ and (i,j)th element of $\mathbf{S}_t(\boldsymbol{\theta})$ is $\text{cov}_t\left(\mathbf{z}_i^f, \mathbf{z}_j^f\right) = k_t\left(\mathbf{z}_i^f, \mathbf{z}_j^f\right) - \mathbf{k}_t\left(\mathbf{z}_i^f\right)^\top \mathbf{K}_t^{-1} \mathbf{k}_t\left(\mathbf{z}_j^f\right)$ where $\mathbf{z}_i^f = \left(\mathbf{x}_i^{f\top}, \boldsymbol{\theta}^\top\right)^\top$ for $i, j = 1, \ldots, d$. The next section uses these results for posterior inference and the derivation of acquisition functions.

3 Acquisition Functions

The acquisition function $\mathcal{A}_t(\cdot, \cdot)$ in Algorithm 1 provides a way to make an informed decision about where to evaluate the simulation model next given data $\mathcal{D}_t$. We propose two acquisition functions, one of which results in improved posterior predictions, while the other provides improved field predictions. The proposed acquisition functions focus on minimizing the aggregated variance of the posterior and use the mean $\mathbb{E}\left[\tilde{p}\left(\boldsymbol{\theta}|\mathbf{y}\right)|\mathcal{D}_t\right]$ and variance $\mathbb{V}\left[\tilde{p}(\boldsymbol{\theta}|\mathbf{y})|\mathcal{D}_t\right]$ of posterior prediction from Lemma 3.1. The proof follows from Sürer et al. (2024) and is given in Appendix A.1.1 for the sake of completeness.

Lemma 3.1. *Assuming that the covariance matrices* $\boldsymbol{\Sigma}$ *and* $\mathbf{S}_t(\boldsymbol{\theta})$ *are positive definite, under the model given by Equations* (2), (3)*, and* (7)*,*

$$\mathbb{E}[\tilde{p}(\boldsymbol{\theta}|\mathbf{y})|\mathcal{D}_t] = f_{\mathcal{N}}\left(\mathbf{y};\, \boldsymbol{\mu}_t\left(\boldsymbol{\theta}\right), \boldsymbol{\Sigma} + \mathbf{S}_t\left(\boldsymbol{\theta}\right)\right) p(\boldsymbol{\theta}), \tag{8}$$

$$\begin{aligned}\mathbb{V}[\tilde{p}(\boldsymbol{\theta}|\mathbf{y})|\mathcal{D}_t] = \Bigg(&\frac{1}{2^d \pi^{d/2} |\boldsymbol{\Sigma}|^{1/2}} f_{\mathcal{N}}\left(\mathbf{y};\, \boldsymbol{\mu}_t\left(\boldsymbol{\theta}\right), \frac{1}{2}\boldsymbol{\Sigma} + \mathbf{S}_t\left(\boldsymbol{\theta}\right)\right) \\ &- \left(f_{\mathcal{N}}\left(\mathbf{y};\, \boldsymbol{\mu}_t\left(\boldsymbol{\theta}\right), \boldsymbol{\Sigma} + \mathbf{S}_t\left(\boldsymbol{\theta}\right)\right)\right)^2\Bigg) p(\boldsymbol{\theta})^2,\end{aligned} \tag{9}$$

where $f_{\mathcal{N}}(\mathbf{a};\, \mathbf{b},\, \mathbf{C})$ *denotes the probability density function of the normal distribution with mean* $\mathbf{b}$ *and covariance* $\mathbf{C}$*, evaluated at the value* $\mathbf{a}$*.*

Recently, Sürer et al. (2024) propose an expected integrated variance (EIVAR) criterion to select new simulation runs for an improved posterior prediction in the case of high-dimensional simulation outputs. Their setting considers that once the simulation model is evaluated with parameter $\boldsymbol{\theta}$, it returns simulation outputs simultaneously at all design inputs $\mathbf{x}_1^f, \ldots, \mathbf{x}_d^f$. In other words, when a new parameter is acquired at iteration t, the associated simulation outputs at all design inputs are included in the simulation data set. The GP-based emulator relying on the basis vector approach (Higdon et al. 2008) is used to emulate the high-dimensional simulation output, and EIVAR is derived using this specific form of an emulator. First, we generalize the EIVAR criterion for a one-dimensional simulation output setting to allow the acquisition of both a design input and a parameter. The proposed acquisition function, denoted by $\mathcal{A}_t^p(\cdot)$, aggregates the variance of the posterior over the parameter space to better learn the posterior and is calculated for any candidate input $\mathbf{z}^* = \left(\mathbf{x}^{*\top}, \boldsymbol{\theta}^{*\top}\right)^\top$ from the discrete set of inputs $\mathcal{L}_t$ introduced in Section 2.1 by

$$\mathcal{A}_t^p(\mathbf{z}^*) = \int_\Theta \mathbb{E}_{\eta^*|\mathcal{D}_t}\left(\mathbb{V}[p(\mathbf{y}|\boldsymbol{\theta})\,|(\mathbf{z}^*, \eta^*) \cup \mathcal{D}_t]\right) p(\boldsymbol{\theta})^2 d\boldsymbol{\theta}. \tag{10}$$

Here, $\eta^* := \eta\left(\mathbf{x}^*, \boldsymbol{\theta}^*\right)$ represents the new simulation output at $\mathbf{z}^*$ and the expectation is taken over the hypothetical simulation output η^* under data $\mathcal{D}_t$.

If accuracy over the entire design space $\mathcal{X}$ is desired then we suggest the following acquisition function represented by $\mathcal{A}_t^y(\cdot)$ for improved field predictions

$$\mathcal{A}_t^y(\mathbf{z}^*) = \int_{\mathcal{X}} \mathbb{E}_{\eta^*|\mathcal{D}_t}\left(\mathbb{V}[p(\mathbf{y}^{\mathbf{x}}|\hat{\boldsymbol{\theta}})\,|(\mathbf{z}^*, \eta^*) \cup \mathcal{D}_t]\right) p(\hat{\boldsymbol{\theta}})^2 d\mathbf{x}, \tag{11}$$

where $\mathbf{y}^{\mathbf{x}} = \left(y\left(\mathbf{x}_1^f\right), \ldots, y\left(\mathbf{x}_d^f\right), y(\mathbf{x})\right)^\top$ and $\hat{\boldsymbol{\theta}}$ represents the estimate of the parameter of interest. $\mathcal{A}_t^y(\cdot)$ integrates the variance of the posterior over the design space $\mathcal{X}$ by considering all design inputs as possible locations to collect field data for a given parameter estimate. Since the field data is available only at design inputs $\mathbf{x}_1^f, \ldots, \mathbf{x}_d^f$, we use the approximation of (11) leveraging the predictions of the emulator to estimate $y(\mathbf{x})$ at any $\mathbf{x}$ as described later in this section. Our focus here is on how to select the new input given the estimate. One can use the same plug-in estimate throughout the sequential procedure or obtain it at each iteration, both of which are illustrated in our experiments. Instead of substituting a single value $\hat{\boldsymbol{\theta}}$ into (11), another way is to integrate over the parameter space Θ as well to consider the total uncertainty on the posterior estimate across all parameters and design inputs. However, due to the computational cost accompanying the evaluation of double integral, we use the plug-in approach in our experiments and leave the computational enhancements as future work.

As an illustration, consider the simulation model $\eta\left(x, \theta\right)$ presented in Figure 1 with a design input x and a parameter θ. Note that the boldface characters are removed from both x and θ since they are scalars. Figure 1 shows field predictions (top panels) and posterior predictions (bottom panels) for three different acquisition functions represented by columns. To initialize, $n_0 = 10$ inputs are sampled uniformly from the prior and then $n = 20$ simulation outputs are collected with each acquisition function using Algorithm 1. The emulator returned by the end of the sequential procedure is employed to produce field and posterior predictions as follows. The field predictions are obtained with the emulator mean and variance in (6) across

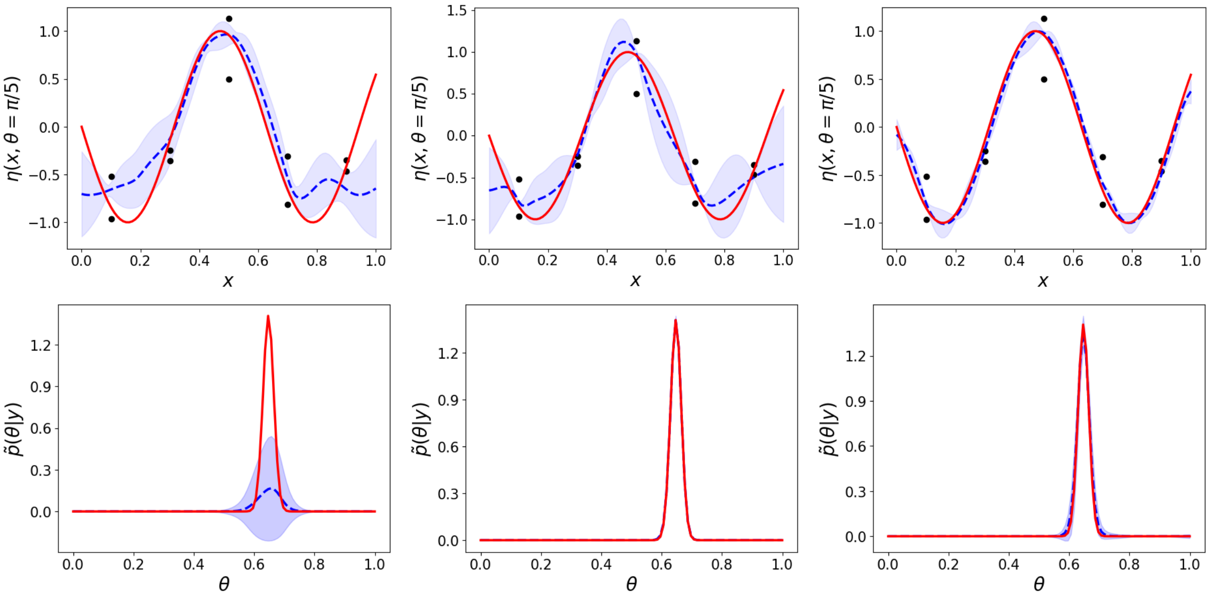

Figure 1: Illustration with a simulation model $\eta(x, \theta) = \sin(10x - 5\theta)$ where $x \in [0, 1]$ and $\theta \in [0, 1]$. Black dots represent the field data replicated twice at the design inputs $(x^f_1, x^f_2, x^f_3, x^f_4, x^f_5) = (0.1, 0.3, 0.5, 0.7, 0.9)$ and generated through $y(x) = \eta\left(x, \theta = \frac{\pi}{5}\right) + \epsilon$, where $\epsilon \sim \mathrm{N}(0, 0.2^2)$. The top and bottom panels show field and posterior predictions, respectively. Predictions in panels are obtained with a GP emulator built with the simulation data set using LHS (left), $\mathcal{A}^p_t(\cdot)$ (middle), and $\mathcal{A}^y_t(\cdot)$ (right). The blue dashed line shows the prediction mean and the shaded area illustrates one predictive standard deviation from the mean. The red line illustrates the simulation model at $\theta = \frac{\pi}{5}$ (top panels) or the posterior (bottom panels).

design inputs paired with $\hat{\theta}$, the estimate of the true value of θ. The parameter estimate $\hat{\theta}$ is iteratively updated to minimize the sum of the squared errors between the field data and its corresponding predictions. The posterior mean and variance are obtained by applying Lemma (3.1) with the associated GP emulator. For example, predictions in the left panels are produced using the simulation data set constructed with LHS and the emulator is not adequate to predict both the field data and posterior truly. In the following, we describe the resulting predictions with the proposed acquisition functions shown in the middle and right panels of Figure 1. Moreover, Figure 2 illustrates the acquisition function value surface for three iterations, presenting different stages for each function to better explain their behavior.

The middle panels in Figure 1 demonstrate field (top) and posterior (bottom) predictions obtained through the emulator fitted to the simulation data collected with $\mathcal{A}^p_t(\cdot)$. The top panels in Figure 2 illustrate three iterations of the sequential procedure using $\mathcal{A}^p_t(\cdot)$. In the calibration space, acquired inputs are concentrated around the (unknown) parameter region of interest, whereas in the design space, $\mathcal{A}^p_t(\cdot)$ encour-

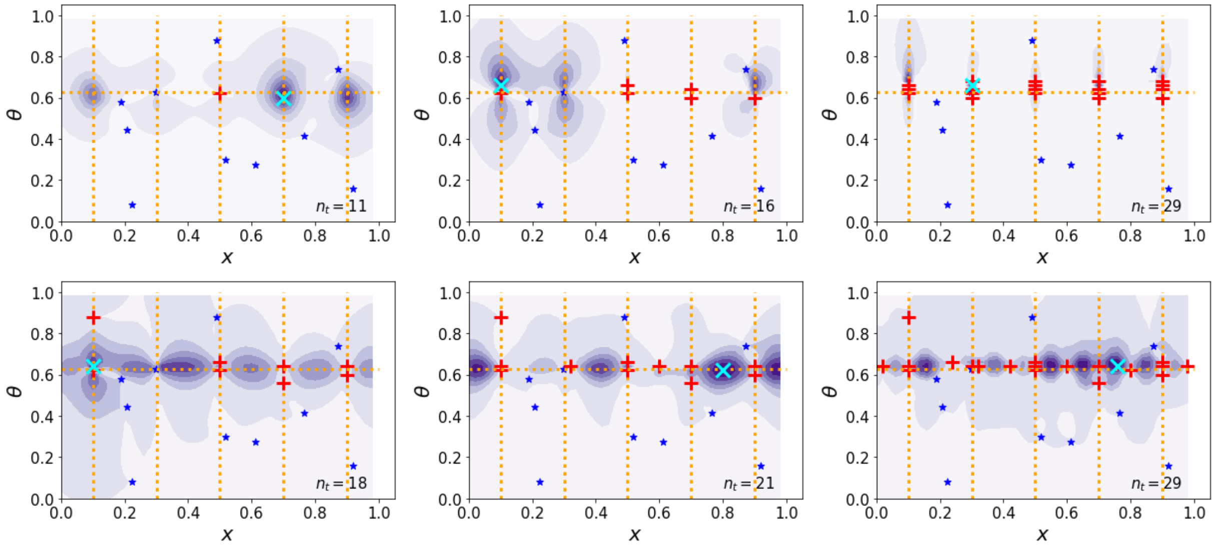

Figure 2: Acquisition function value surface for three iterations of the sequential procedure via $\mathcal{A}_t^p(\cdot)$ (top panels) and $\mathcal{A}_t^y(\cdot)$ (bottom panels) using Figure 1 example. Dark purple indicates lower values and light purple indicates larger values. Markers demonstrate the initial sample (blue star), previously acquired inputs (red plus), and the input minimizing the acquisition function at the current iteration (cyan cross). The orange dotted lines show $\theta = \frac{\pi}{5}$ (horizontal line) and field data design inputs (vertical lines).

ages the selection of design inputs where the field data is collected. The acquisition function $\mathcal{A}_t^p(\cdot)$ places the points around the parameter region of interest since the volume of the high-posterior region is small and parameters with zero-posterior density provide almost no information for the behavior of the simulation model near the parameter of interest. Moreover, as demonstrated in Figure 1, focusing only on the regions where the field data is collected improves the posterior prediction dramatically as compared to the one with LHS. Notice that the emulator uncertainty is shrunk towards zero around the field data design inputs since the targeted data collection results in a refined estimate of the field data, with reduced uncertainty in regions where the simulation outputs have been observed. These results are promising especially for high-dimensional design spaces. If one is only interested in accurate and precise inference of calibration parameters through estimating the posterior density, then building independent GP emulators for each design input is much more efficient than building one large GP emulator in high-dimensional spaces (see examples of such emulators in Gu and Berger (2016), Huang et al. (2020)). The acquisition function $\mathcal{A}_t^p(\cdot)$ can still be used with these efficient emulators to construct the simulation data set by restricting the design space to the ones where the physical experiment is conducted. However, since $\mathcal{A}_t^p(\cdot)$ does not encourage the exploration of the design space, it does not improve field predictions at unseen design inputs.

The emulator built with the simulation data set collected with $\mathcal{A}_t^y(\cdot)$ generates superior field predictions as compared to other approaches (see the right panel in Figure 1). As illustrated in Figure 2 (bottom panels), early in the sequential procedure, $\mathcal{A}_t^y(\cdot)$ encourages the selection of inputs around the design points where the field data is collected to minimize the total uncertainty. Then, it focuses on the entire design space for improved predictions around the parameter of interest. Thus, the emulator uncertainty is reduced not only around the field data design inputs but also over the remaining design space when predicting the field data. Although the posterior predictions obtained with $\mathcal{A}_t^p(\cdot)$ match almost perfectly with the ground truth, $\mathcal{A}_t^y(\cdot)$ also performs well in terms of predicting the posterior since the simulation data set includes points aligned with field data design inputs as well. In this sense, $\mathcal{A}_t^y(\cdot)$ is a way to obtain accurate posterior and field predictions by balancing the exploration of unseen design inputs and exploitation of existing design inputs around the parameter region of interest.

At iteration t, acquisition functions $\mathcal{A}_t^p(\cdot)$ and $\mathcal{A}_t^y(\cdot)$ choose the next input point as the one that minimizes the total uncertainty over the entire parameter and design space, respectively. To evaluate each acquisition function with any candidate input, the expected variance of the posterior is obtained as follows and the derivation is provided in Appendix A.1.2.

Lemma 3.2. *Under the conditions of Lemma 3.1,*

$$\begin{aligned}&\mathbb{E}_{\eta^*|\mathcal{D}_t}\left(\mathbb{V}[p(\mathbf{y}|\boldsymbol{\theta})\,|(\mathbf{z}^*,\eta^*)\cup\mathcal{D}_t]\right)\\&\qquad=\frac{f_{\mathcal{N}}\left(\mathbf{y};\,\boldsymbol{\mu}_t\left(\boldsymbol{\theta}\right),\,\frac{1}{2}\boldsymbol{\Sigma}+\mathbf{S}_t\left(\boldsymbol{\theta}\right)\right)}{2^d\pi^{d/2}|\boldsymbol{\Sigma}|^{1/2}}-\frac{f_{\mathcal{N}}\left(\mathbf{y};\,\boldsymbol{\mu}_t\left(\boldsymbol{\theta}\right),\,\frac{1}{2}\left(\boldsymbol{\Sigma}+\mathbf{S}_t\left(\boldsymbol{\theta}\right)+\boldsymbol{\phi}_t\left(\boldsymbol{\theta},\mathbf{z}^*\right)\right)\right)}{2^d\pi^{d/2}|\boldsymbol{\Sigma}+\mathbf{S}_t\left(\boldsymbol{\theta}\right)-\boldsymbol{\phi}_t\left(\boldsymbol{\theta},\mathbf{z}^*\right)|^{1/2}},\end{aligned}\tag{12}$$

where the (i,j)th element of $\boldsymbol{\phi}_t\left(\boldsymbol{\theta},\mathbf{z}^*\right)$ is $\frac{\mathrm{cov}_t\left(\mathbf{z}_i^f,\mathbf{z}^*\right)\mathrm{cov}_t\left(\mathbf{z}_j^f,\mathbf{z}^*\right)}{\varsigma_t^2(\mathbf{z}^*)+\upsilon}$ with $\mathbf{z}_i^f=\left(\mathbf{x}_i^{f\top},\boldsymbol{\theta}^\top\right)^\top$ for $i,j=1,\ldots,d$.

The expected variance of the posterior needs to be integrated over a multi-dimensional parameter and design spaces to obtain Equations (10)–(11). Since the integrals are analytically difficult to compute, they are approximated with a sum over uniformly distributed reference grids Θ_{ref} and $\mathcal{X}_{\mathrm{ref}}$ within the spaces Θ and $\mathcal{X}$, respectively. In higher-dimensional inputs, the size of a grid covering the input space becomes very large, and one can consider sparse grids, smart sampling techniques, or quadrature schemes for estimating higher dimensional integrals. Alternatively, the reference set could follow a space-filling construction and one can regenerate the space-filling reference set at each iteration to encourage diversity in search.

Notice that the initial term in (12) does not depend on $\mathbf{z}^*$ and we drop this term to efficiently approximate the acquisition criteria. Thus, minimizing $\mathcal{A}_t^p(\cdot)$ in (10) is efficiently approximated by maximizing

$$\frac{1}{|\Theta_{\text{ref}}|}\sum_{\boldsymbol{\theta}\in\Theta_{\text{ref}}} p(\boldsymbol{\theta})^2\left(\frac{f_{\mathcal{N}}\left(\mathbf{y};\boldsymbol{\mu}_t\left(\boldsymbol{\theta}\right), \frac{1}{2}\left(\boldsymbol{\Sigma}+\mathbf{S}_t\left(\boldsymbol{\theta}\right)+\boldsymbol{\phi}_t\left(\boldsymbol{\theta},\mathbf{z}^*\right)\right)\right)}{2^d\pi^{d/2}|\boldsymbol{\Sigma}+\mathbf{S}_t\left(\boldsymbol{\theta}\right)-\boldsymbol{\phi}_t\left(\boldsymbol{\theta},\mathbf{z}^*\right)|^{1/2}}\right). \tag{13}$$

The acquisition function $\mathcal{A}_t^y(\cdot)$ considers the expected posterior variance at the estimate $\hat{\boldsymbol{\theta}}$ integrated over hypothetical design inputs. To do that, the length-$(d+1)$ vectors $\mathbf{y}^{\mathbf{x}}$ and $\boldsymbol{\mu}_t^{\mathbf{x}}(\hat{\boldsymbol{\theta}})$ and $(d+1)\times(d+1)$ matrices $\boldsymbol{\Sigma}^{\mathbf{x}}$, $\mathbf{S}_t^{\mathbf{x}}(\hat{\boldsymbol{\theta}})$, and $\boldsymbol{\phi}_t^{\mathbf{x}}\left(\hat{\boldsymbol{\theta}},\mathbf{z}^*\right)$ are constructed by augmenting a possible design input $\mathbf{x}\in\mathcal{X}_{\text{ref}}$ onto the existing design inputs $\mathbf{x}_1^f,\ldots,\mathbf{x}_d^f$. Then, approximating the minimization of $\mathcal{A}_t^y(\cdot)$ in (11) involves maximizing

$$\frac{1}{|\mathcal{X}_{\text{ref}}|}\sum_{\mathbf{x}\in\mathcal{X}_{\text{ref}}} p(\hat{\boldsymbol{\theta}})^2\left(\frac{f_{\mathcal{N}}\left(\mathbf{y}^{\mathbf{x}};\boldsymbol{\mu}_t^{\mathbf{x}}(\hat{\boldsymbol{\theta}}), \frac{1}{2}\left(\boldsymbol{\Sigma}^{\mathbf{x}}+\mathbf{S}_t^{\mathbf{x}}(\hat{\boldsymbol{\theta}})+\boldsymbol{\phi}_t^{\mathbf{x}}\left(\hat{\boldsymbol{\theta}},\mathbf{z}^*\right)\right)\right)}{2^{(d+1)}\pi^{(d+1)/2}|\boldsymbol{\Sigma}^{\mathbf{x}}+\mathbf{S}_t^{\mathbf{x}}(\hat{\boldsymbol{\theta}})-\boldsymbol{\phi}_t^{\mathbf{x}}\left(\hat{\boldsymbol{\theta}},\mathbf{z}^*\right)|^{1/2}}\right). \tag{14}$$

Since $y(\mathbf{x})$ is unknown at any $\mathbf{x}\in\mathcal{X}_{\text{ref}}$, the emulator mean at $\left(\mathbf{x}^\top,\hat{\boldsymbol{\theta}}^\top\right)^\top$ is used as a plug-in estimator. Recall that we replace the difficult numerical optimization with a discrete search by evaluating the acquisition function $\mathcal{A}_t^p(\cdot)$ $(\mathcal{A}_t^y(\cdot))$ on a candidate set of inputs $\mathcal{L}_t$. Thus, at each iteration, (13) ((14)) is computed for each $\mathbf{z}^*\in\mathcal{L}_t$, and then the next input point included in the simulation data set is the maximizer from the discrete set of inputs $\mathcal{L}_t$. As an alternative to optimizing over a discrete set of inputs, once the sum-based approximation of the integrals is obtained, one can employ numerical optimization with the off-the-shelf solvers using closed-form derivatives. We note that as inputs are selected with (13) ((14)), the covariance matrix $\mathbf{S}_t(\boldsymbol{\theta})$ $(\mathbf{S}_t^{\mathbf{x}}(\hat{\boldsymbol{\theta}}))$ becomes smaller, promoting exploration in regions with higher uncertainty to minimize the aggregated posterior variance.

Our derivations so far assume that the variance σ^2 of the residual error is known. However, in some cases, the variance term might not be available in advance. Moreover, another component of uncertainty can exist in the form of a model discrepancy. Kennedy and O'Hagan (KOH, Kennedy and O'Hagan (2001)) model the field data as the function of a simulation output plus an additional discrepancy term such that $y\left(\mathbf{x}_i^f\right)=\eta\left(\mathbf{x}_i^f,\boldsymbol{\theta}\right)+b\left(\mathbf{x}_i^f\right)+\epsilon$, where $b\left(\mathbf{x}_i^f\right)$ represents the discrepancy or the bias term at the design input $\mathbf{x}_i^f$. Although the discrepancy term can create an identifiability problem (Bayarri et al. 2007, Brynjarsdóttir and O'Hagan 2014, Gu and Wang 2018, Plumlee 2017, Tuo and Wu 2015), the KOH framework has been widely used for calibrating simulation models. One can still use the proposed sequential approach when the error variance is unknown and/or in the case of a discrepancy between the simulation output and field data as follows.

KOH framework assigns a GP emulator as the prior distribution of both the simulation model and discrepancy term. The modular approach (Bayarri et al. 2007) determines unknown hyperparameters of an emulator of the simulation model utilizing only the simulation data. Then, the unknown hyperparameters of an emulator of the discrepancy term are obtained by utilizing discrepancy observations generated as the difference of field data and emulator means of a simulation model at the same input values (Bayarri et al. 2009). In parallel to the modular approach, as described in Section 2.3, an emulator of the simulation model is built using the simulation data set $\mathcal{D}_t$ at each iteration. Then, the unknown discrepancy covariance hyperparameters, denoted by $\boldsymbol{\theta}^e$, are estimated by maximizing the likelihood. Let $\mathbf{\Sigma}^e$ denote the $d \times d$ covariance matrix of the discrepancy term and the noise term. Hyperparameters $\boldsymbol{\theta}^e$ determine the structure of the covariance matrix $\mathbf{\Sigma}^e$. For example, when the discrepancy is negligible, σ^2 is the only hyperparameter that needs to be estimated (e.g., $\theta^e := \sigma^2$ and $\mathbf{\Sigma}^e = \sigma^2\mathbf{I}$). Once the estimates of $\boldsymbol{\theta}^e$ are obtained at each iteration, Equation (13) can be computed by replacing $\mathbf{\Sigma}$ with the estimate of $\mathbf{\Sigma}^e$. Similarly, in Equation (14), the cross-covariance values between any $\mathbf{x} \in \mathcal{X}_{\text{ref}}$ and the existing design inputs are computed using the estimates of $\boldsymbol{\theta}^e$ to replace $\mathbf{\Sigma}^{\mathbf{x}}$ with the estimated covariance matrix.

4 Experiments

Section 4.1 investigates the performance using two synthetic simulation models without and with discrepancy terms. Section 4.2 examines the performance with higher dimensional input spaces. Section 4.3 demonstrates the proposed acquisition functions with the COVID-19 epidemiological simulation model.

4.1 Benchmark with Two Synthetic Simulation Models

We examine the performance of the proposed acquisition functions $\mathcal{A}_t^p(\cdot)$ and $\mathcal{A}_t^y(\cdot)$ abbreviated by $\mathcal{A}^p$ and $\mathcal{A}^y$ using two synthetic simulation models. For comparison, we sample inputs from two space-filling alternatives: uniformly random from a prior and LHS. These approaches are labeled as $\mathcal{A}^{rnd}$ and $\mathcal{A}^{lhs}$ in the experiments. The proposed sequential procedure is implemented under the Python package Parallel Uncertainty Quantification (PUQ) at [BLINDED FOR REVIEW] and the code scripts are provided to replicate the examples.

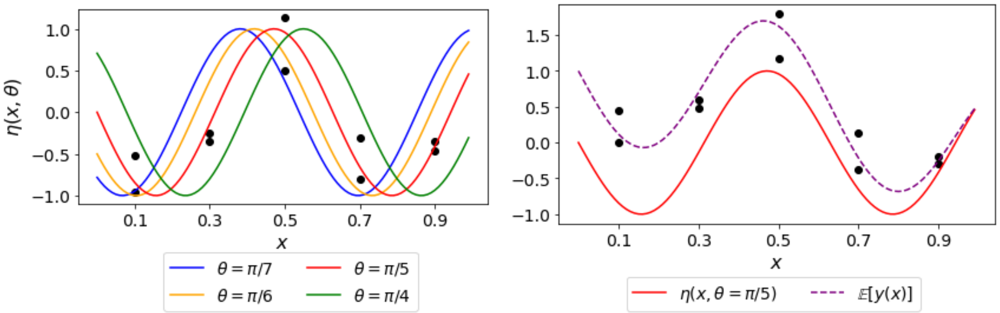

Figure 3: Illustration of the simulation model with two-dimensional $[\mathcal{X},\Theta]$ space introduced in Figure 1. In the left panel, lines demonstrate the simulation outputs at $\theta \in \{\frac{\pi}{4},\frac{\pi}{5},\frac{\pi}{6},\frac{\pi}{7}\}$. The red line represents the mean of the field data $\mathbb{E}[y(x)] = \eta\left(x,\theta=\frac{\pi}{5}\right)$ without a discrepancy term. In the right panel, the dashed purple line corresponds to the mean of the field data $\mathbb{E}[y(x)] = \eta\left(x,\theta=\frac{\pi}{5}\right)+b(x)$ with a discrepancy term $b(x) = 1-\frac{1}{3}x-\frac{2}{3}x^2$. In both panels, black dots represent the field data at equally spaced, five design inputs generated with $y(x) = \mathbb{E}[y(x)]+\epsilon$ using $\epsilon \sim \mathrm{N}(0, 0.2^2)$.

First, the performance comparison is demonstrated with a sinusoidal simulation model similar to the one used by Koermer et al. (2023). Throughout the paper, the model has been used for illustration and it includes a one-dimensional design input x and calibration parameter θ ($q = p = 1$) where $x \in [0,1]$ and $\theta \in [0,1]$. The left panel in Figure 3 shows simulation outputs across design space for different parameter values when the discrepancy is negligible. In the right panel, the mean of the field data is demonstrated in the case of discrepancy. For both cases, the data is collected at five unique locations on an equally spaced grid and each observation is replicated twice. The second example comes from Ranjan et al. (2011) with a two-dimensional design input $\mathbf{x} = (x_1,x_2)^\top \in [0,1]^2$ ($q = 2$) and a one-dimensional calibration parameter $\theta \in [0,1]$ ($p = 1$). Figure 4 demonstrates the contour plots of the expected field data without and with discrepancy terms and the locations of nine unique design inputs where the field data is collected. The data generation mechanisms are detailed in the captions of Figures 3–4.

To initialize the sequential procedure, the sample of size n_0 is taken from a uniform prior. For the two- and three-dimensional simulation models presented in Figures 3–4, we set $n_0 = 10$ and $n_0 = 30$, respectively, in $[\mathcal{X},\Theta]$ space. The initial sample size is selected to enable the emulator to sufficiently learn the response surface in the early stages while also permitting enhancements in predictions through subsequent acquisitions. Additional analysis on the initial sample size is provided in Appendix A.2. The sequential

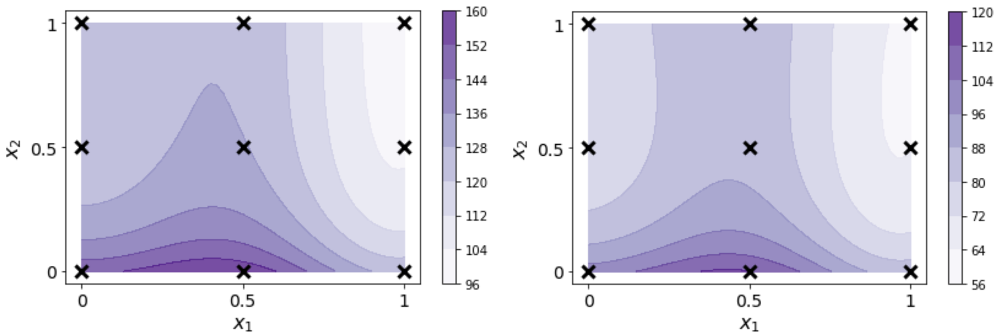

Figure 4: Illustration of the simulation model with three-dimensional $[\mathcal{X}, \Theta]$ space such that $\eta\left(\mathbf{x}, \theta\right) = \left(30 + 5x_1 \sin\left(5x_1\right)\right)\left(6\theta + 1 + \exp\left(-5x_2\right)\right)$. The left and right panels illustrate the contour plots of $\mathbb{E}[y(\mathbf{x})] = \eta\left(\mathbf{x}, \theta = 0.5\right)$ and $\mathbb{E}[y(\mathbf{x})] = \eta\left(\mathbf{x}, \theta = 0.5\right) + b(\mathbf{x})$ with $b(\mathbf{x}) = -50\exp\left(-0.2x_1 - 0.1x_2\right)$, respectively. Cross markers represent nine design inputs where the field data is collected as two replicates of nine points using $y(\mathbf{x}) = \mathbb{E}[y(\mathbf{x})] + \epsilon$ with $\epsilon \sim \mathrm{N}\left(0, 0.5^2\right)$.

approach is terminated once simulation outputs are collected from $n = 90$ and $n = 150$ acquired inputs for the two- and three-dimensional functions, respectively. The fixed budget termination criterion is determined based on the point where the best method starts to converge. At each iteration, the candidate list $\mathcal{L}_t$ introduced in lines 5–6 of Algorithm 1 is generated to combine exploration using a sample from the prior with exploitation using a sample of parameters paired with field data design inputs. First, we sample 100 parameters from a uniform prior in Θ space. Then, to account for the influence of the number of field data design inputs, each unique field data design input is paired with each parameter to construct 500 ($= 5 \times 100$) and 900 ($= 9 \times 100$) sets of candidate inputs for the first and second models, respectively. These candidate points are generated to facilitate the optimization procedure for exploiting existing field data design inputs. In addition, another 500 and 900 inputs are randomly sampled from the prior in $[\mathcal{X}, \Theta]$ space to allow exploration for the first and second functions, respectively. We opt for a 50%-50% split to ensure equal representation for exploration and exploitation. Therefore, at iteration t, the size of the candidate list is $|\mathcal{L}_t| = 1000$ and $|\mathcal{L}_t| = 1800$ for the first and second functions. The size and construction of the candidate list depend on various experimental characteristics, such as the number of field data design inputs and the dimension of the input space. As dimensionality increases, practitioners can adjust the candidate list based on their computational budget and experimental setup.

The performance comparison is summarized over 30 replications. At each replication, the initial design of size n_0 is randomly chosen from a uniform prior. The same initial sample is used for all methods $\mathcal{A}^p$, $\mathcal{A}^y$, $\mathcal{A}^{rnd}$, and $\mathcal{A}^{lhs}$ for a fair comparison. Similarly, the field data is rerandomized at each replication and the same field data is used across different methods within each replication. As a performance metric, we compute the mean absolute difference MAD^p between the estimated posterior and the true posterior $\tilde{p}(\boldsymbol{\theta}|\mathbf{y})$ at unseen calibration parameters and the mean absolute difference MAD^y between the estimated field data and the true field data at unseen design inputs. To do that, we generate a set of reference calibration parameters Θ_{ref} and a set of reference design inputs $\mathcal{X}_{\text{ref}}$ and compute the performance metrics via $\mathrm{MAD}_t^p = \frac{1}{|\Theta_{\text{ref}}|}\sum_{\boldsymbol{\theta}\in\Theta_{\text{ref}}} |\tilde{p}(\boldsymbol{\theta}|\mathbf{y}) - \hat{p}_t(\boldsymbol{\theta}|\mathbf{y})|$ and $\mathrm{MAD}_t^y = \frac{1}{|\mathcal{X}_{\text{ref}}|}\sum_{\mathbf{x}\in\mathcal{X}_{\text{ref}}} |y(\mathbf{x}) - \hat{y}_t(\mathbf{x})|$ at each iteration t. Here, $\hat{p}_t(\boldsymbol{\theta}|\mathbf{y})$ is the posterior prediction obtained with Equation (8) and $\hat{y}_t(\mathbf{x})$ is the field prediction obtained using the emulator mean in Equation (6) at a given design input $\mathbf{x}$ paired with $\hat{\boldsymbol{\theta}}$. For the two-dimensional function, both Θ_{ref} and $\mathcal{X}_{\text{ref}}$ are generated from 100 equally spaced points in $[0, 1]$. For the three-dimensional example, Θ_{ref} is generated from 100 equally spaced points in $[0, 1]$ and $\mathcal{X}_{\text{ref}}$ is generated in a two-dimensional grid of 20^2 points. In addition, the reference sets Θ_{ref} and $\mathcal{X}_{\text{ref}}$ are used to approximate the acquisition function values in Equations (13)–(14). For all the experiments presented in this section, the calibration parameter estimate, $\hat{\boldsymbol{\theta}}$, is updated at each iteration by minimizing the sum of the squared errors between the field data and field prediction, as it is one of the most common parameter estimation techniques. Alternative techniques, such as maximum likelihood estimation and χ^2 minimization, can also be employed. Additionally, for real-world examples with highly nonlinear response surfaces, field scientists may be aware of specific optimization techniques better suited to parameter estimation in their applications. For the experiments with a discrepancy term, in addition to $\hat{\boldsymbol{\theta}}$, GP hyperparameters $\boldsymbol{\theta}^e$ for the discrepancy term are also estimated. We use a covariance form $\boldsymbol{\Sigma}^e$ defined by $\boldsymbol{\Sigma}^e_{i,j} = \sigma^2_\varepsilon \delta_{i=j} + \sigma^2_b \exp\left(-\lambda \left|\left|\mathbf{x}^f_i - \mathbf{x}^f_j\right|\right|_1\right)$ for $i, j = 1, \ldots, d$ and the maximum likelihood estimates of hyperparameters $\boldsymbol{\theta}^e = (\sigma^2_\varepsilon, \sigma^2_b, \lambda)^\top$ are obtained at each iteration. Moreover, for the examples with a discrepancy term, we investigate the quality of acquired parameters via the interval score, instead of comparing the methods with the MAD^p metric due to unknown $\boldsymbol{\Sigma}^e$. To do this, we compute the interval score $S_\alpha(l, u; a)$ (Gneiting and Raftery 2007) for each method and replicate via

$$S_\alpha(l, u; a) = (u - l) + \frac{2}{\alpha}(l - a)\mathbb{1}\{a < l\} + \frac{2}{\alpha}(a - u)\mathbb{1}\{a > u\} \tag{15}$$

where l and u represent the quantiles of acquired parameters at level $\frac{\alpha}{2}$ and $(1 - \frac{\alpha}{2})$, respectively, and $\mathbb{1}$ is an indicator function. To assess whether the best-fit parameter falls within the range of acquired parameters, we substitute the least-squares fit parameter value in place of a and use $\alpha = 0.10$ in the experiments. The interval score helps address the width of acquired parameters and the coverage of the best-fit parameterization.

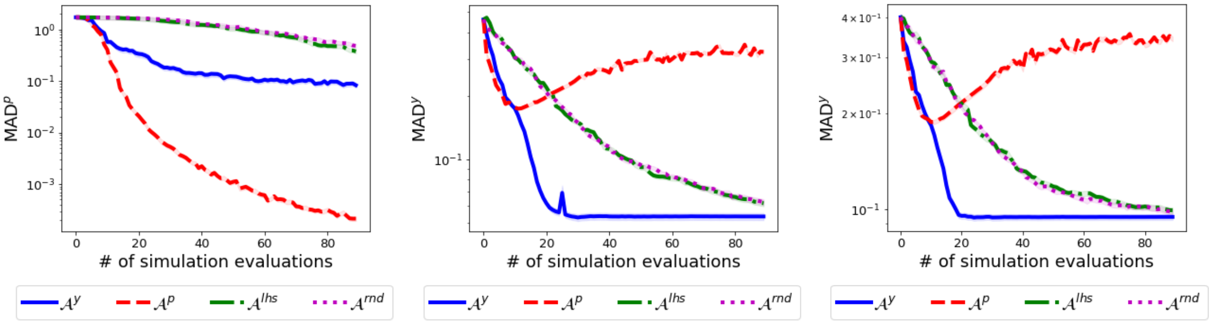

Figure 5: Comparison of different acquisition functions using the two-dimensional simulation model in Figure 3. The left panel compares the accuracy of posterior predictions (without a discrepancy term). The middle (without a discrepancy term) and the right panels (with a discrepancy term) compare the accuracy of field predictions.

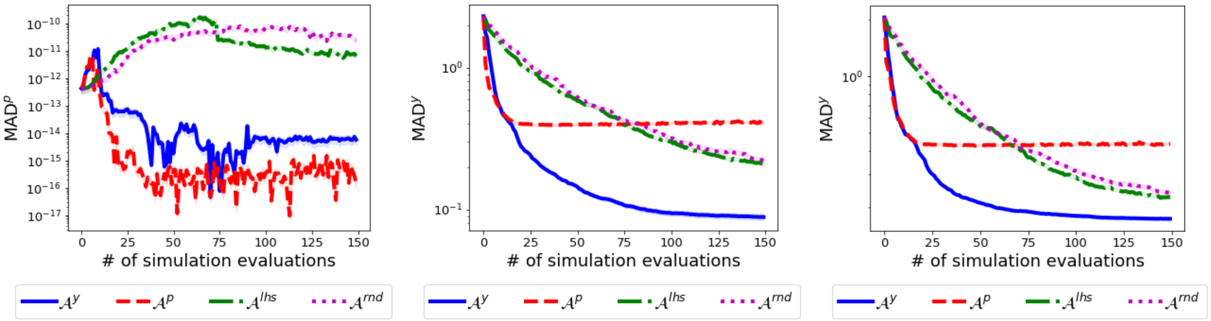

Figure 6: Comparison of different acquisition functions using the three-dimensional simulation model in Figure 4. The left panel compares the accuracy of posterior predictions (without a discrepancy term). The middle (without a discrepancy term) and the right panels (with a discrepancy term) compare the accuracy of field predictions.

Figures 5–6 summarize the performance metrics MAD^p and MAD^y for the two- and three-dimensional simulation models, respectively. The acquisition function $\mathcal{A}^p$ is superior to its competitors in terms of predicting the posterior as shown in the left panels. However, especially for the first function, $\mathcal{A}^p$ behaves poorly in comparison to other acquisition functions in terms of field predictions. The acquisition function $\mathcal{A}^p$ focuses on the exploitation of regions around existing design inputs and it does not encourage the exploration of the design space $\mathcal{X}$ since unseen design points do not provide any additional value for posterior estimation. As a result, the simulation data set is dominated by the field data design inputs (especially during

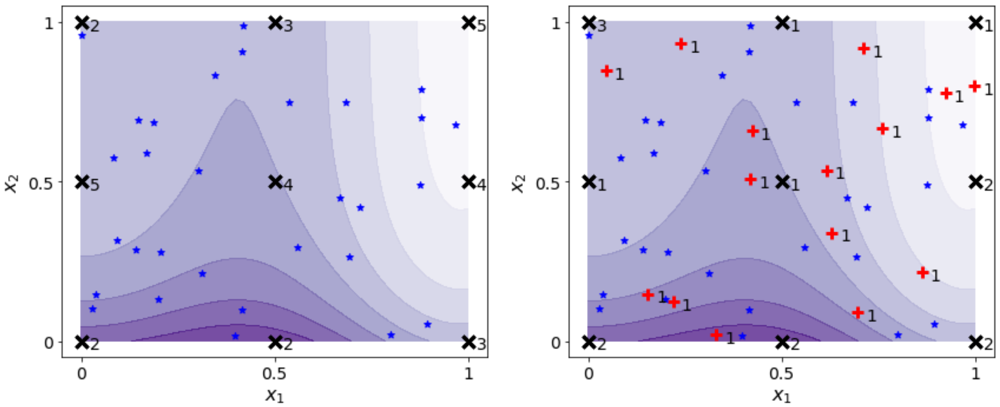

Figure 7: Illustration of design inputs for 30 acquired inputs collected with $\mathcal{A}^p$ (left panel) and $\mathcal{A}^y$ (right panel) using the three-dimensional simulation model. Blue stars illustrate 30 samples used to initiate the proposed procedure. The cross markers (field data design inputs) and red plus markers (unseen design inputs) demonstrate acquired design inputs. The numbers next to the markers show the number of times each design input is included in the simulation data set.

later stages of the sequential procedure), and the emulator's hyperparameters are tuned based on this data, increasing field prediction error (see middle and right panels in Figure 5). On the other hand, the acquisition function $\mathcal{A}^y$ encourages the selection from the entire space $\mathcal{X}$ around the calibration parameter of interest for improved field predictions. Figure 7 demonstrates the design inputs included in the simulation data set constructed by $\mathcal{A}^p$ and $\mathcal{A}^y$ for a single replicate of the second simulation model. $\mathcal{A}^p$ explores the calibration space located on the field data design inputs to minimize the aggregated uncertainty of the posterior over the parameter space. As a result, $\mathcal{A}^p$ distributes the parameters around both high and low posterior regions to better learn the posterior and covers the parameter region of interest well enough by not wasting any computational resources for simulation evaluations outside of the region of interest. On the other hand, $\mathcal{A}^y$ encourages filling the design space to accurately predict the field data while exploiting the field data design inputs to reduce the variance of the posterior at the parameter estimate. Therefore, $\mathcal{A}^y$ achieves not only the best field prediction accuracy in both examples but also more accurate posterior predictions than the space-filling alternatives.

In Figure 6, the posterior prediction error increases with $\mathcal{A}^{lhs}$ and $\mathcal{A}^{rnd}$ during the early stages of the sequential process since the posterior is constantly predicted with a large positive bias. To see the number of samples required for $\mathcal{A}^{lhs}$ and $\mathcal{A}^{rnd}$ to reach the same posterior prediction accuracy level with $\mathcal{A}^p$, we increased the sample size and found that about 500 additional inputs are needed with $\mathcal{A}^{lhs}$ and $\mathcal{A}^{rnd}$. In

		Example in Figure 3				Example in Figure 4			
	Method Metric	$\mathcal{A}^y$	$\mathcal{A}^p$	$\mathcal{A}^{lhs}$	$\mathcal{A}^{rnd}$	$\mathcal{A}^y$	$\mathcal{A}^p$	$\mathcal{A}^{lhs}$	$\mathcal{A}^{rnd}$
without discrepancy	MAD^p	14	11	81	90	38	19	NA	NA
	MAD^y	21	NA	89	90	29	NA	138	150
with discrepancy	MAD^y	20	NA	90	88	37	NA	129	150

Table 1: Number of acquired inputs required to achieve a certain accuracy level. "NA" means the associated method is not able to attain the desired accuracy level.

addition, Table 1 presents the number of simulation evaluations required to achieve a specific accuracy level for two- and three-dimensional examples. For the example presented in Figure 3, for instance, to attain the particular MAD^p level achieved by $\mathcal{A}^{rnd}$ with 90 acquired inputs, $\mathcal{A}^y$, $\mathcal{A}^p$, and $\mathcal{A}^{lhs}$ require 14, 11, and 81 acquisitions, respectively. Therefore, the targeted sampling approach with $\mathcal{A}^p$ and $\mathcal{A}^y$ can be useful especially when working with expensive simulation models since $\mathcal{A}^p$ and $\mathcal{A}^y$ require fewer number of simulation evaluations than the space-filling approaches to achieve the same level of calibration objective. The goal of $\mathcal{A}^p$ is to better infer the calibration parameters through learning the posterior density. To illustrate this in the presence of discrepancy as well, Table 2 examines the quality of acquired parameters by averaging the interval score across 30 replicates of each method. In both examples, the small interval score indicates that $\mathcal{A}^p$ is able to target the high posterior region and collect parameters concentrated around the best-fit parameterization. Therefore, if the goal is to obtain an accurate and precise estimate of the posterior rather than field predictions, we recommend using the acquisition function $\mathcal{A}^p$. However, in some situations, $\mathcal{A}^y$ may be more advantageous if the objective is to exploit the region around the best-fit parameterization rather than to learn the entire posterior, especially when the high posterior region is large. Moreover, deciding whether the discrepancy is negligible or not when employing $\mathcal{A}^p$ or $\mathcal{A}^y$ is an important question similar to the other Bayesian calibration procedures. In such a case, a practitioner can consult with the domain scientist's opinion to determine whether known discrepancies exist between the model and the field data. Additionally, one can perform sensitivity analysis techniques to identify whether including potential discrepancies improves

	Example in Figure 3				Example in Figure 4			
Method Metric	$\mathcal{A}^y$	$\mathcal{A}^p$	$\mathcal{A}^{lhs}$	$\mathcal{A}^{rnd}$	$\mathcal{A}^y$	$\mathcal{A}^p$	$\mathcal{A}^{lhs}$	$\mathcal{A}^{rnd}$
Average Interval Score	0.11	0.08	0.89	0.89	0.07	0.05	0.89	0.90

Table 2: Comparison of different acquisition functions using the examples in Figures 3–4 in the case of discrepancy. The average interval score is computed across 30 replicates for each acquisition function.

the model's performance; see Sung and Tuo (2024) for a recent review on calibration and an extensive discussion on the discrepancy.

4.2 Benchmark with High Dimensional Inputs

This section investigates the performance of the proposed acquisition functions using higher-dimensional input spaces. To understand the effectiveness of the proposed approaches across different configurations of q-dimensional design input $\mathbf{x}$ and p-dimensional calibration parameter $\boldsymbol{\theta}$, we maintain the input dimension at $q+p = 12$ and consider three different scenarios of q and p: $q = 2, p = 10$; $q = 6, p = 6$; and $q = 10, p = 2$. Details of the data generation mechanism are given in Appendix A.3. In addition to the methods outlined in Section 4.1, we include two common acquisition functions, namely $\mathcal{A}^{var}$ and $\mathcal{A}^{imspe}$, to highlight their differences from the proposed approaches tailored for calibration. While $\mathcal{A}^{var}$ selects the input with the highest emulation variance, $\mathcal{A}^{imspe}$ acquires an input to minimize the aggregated emulation variance. Both functions are typically employed to build globally accurate emulators of simulation models. For the sake of completeness, the implementation details of $\mathcal{A}^{var}$ and $\mathcal{A}^{imspe}$ are provided in Appendix A.3. We note that $\mathcal{A}^{rnd}$ is excluded from the results since it performs comparably to $\mathcal{A}^{lhs}$. For all the examples, the sequential procedure terminates once $n = 150$ inputs and the associated simulation outputs are collected. Figure 8 summarizes the performance metrics MAD^p and MAD^y across 10 replications, similar to the experiments in Section 4.1. To gain insights into the differences between the inputs acquired by each method, we compute the width of the interval between the 5% and 95% quantiles of each design input x_i, $i = 1, \ldots, q$, at each replicate. We also measure the interval score for each calibration parameter θ_i, $i = 1, \ldots, p$, via (15) to see both the coverage of the best-fit parameterization and the width of the interval. Table 3 provides the width of design inputs and the interval score of parameters averaged across 10 replicates.

While $\mathcal{A}^y$ results in the lowest interval score across all calibration parameters, $\mathcal{A}^p$ consistently attains the narrowest width across all design inputs. Since $\mathcal{A}^y$ tightly acquires around the plausible parameter region and explores the design space well, it achieves the most precise predictions of field data. We find that the width of design inputs acquired with $\mathcal{A}^p$ is zero in many cases, indicating that $\mathcal{A}^p$ acquires design points from which the field data is collected. Moreover, compared to $\mathcal{A}^{lhs}$, $\mathcal{A}^{var}$, and $\mathcal{A}^{imspe}$, $\mathcal{A}^p$ better constraints the parameter region of interest. As a result, $\mathcal{A}^p$ has a far superior posterior predictive performance. $\mathcal{A}^{var}$ tends to select points at the boundary where the emulation uncertainty is higher; thus, it has the highest

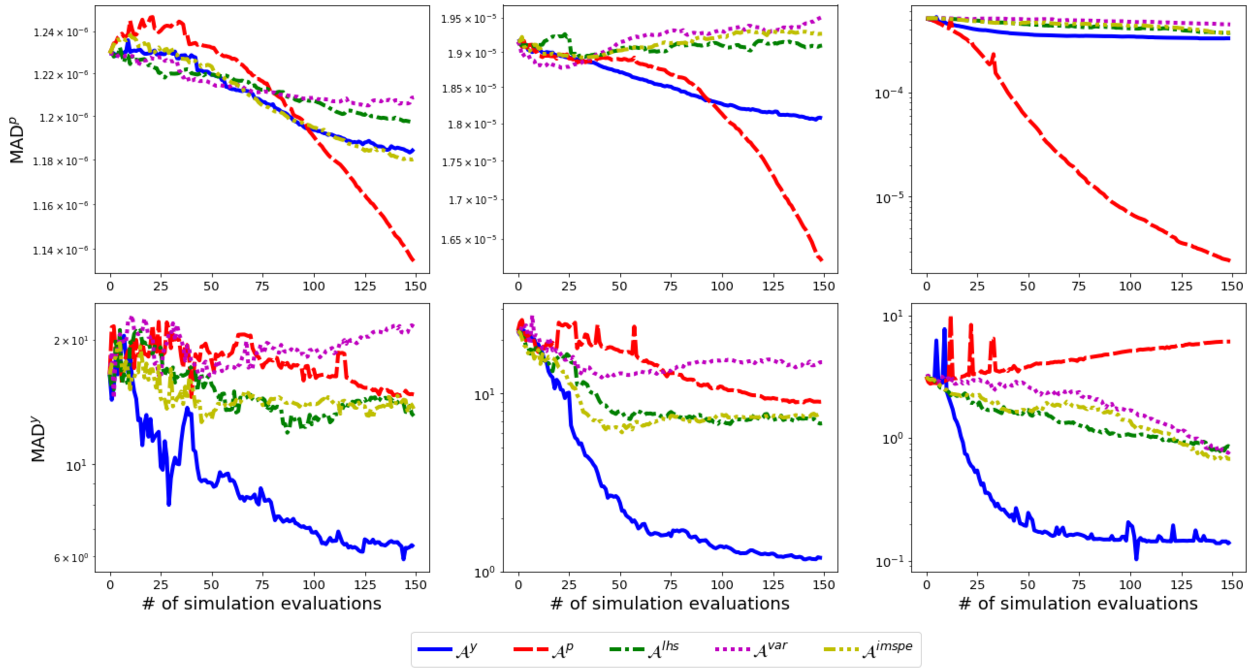

Figure 8: Comparison of different acquisition functions using the higher-dimensional input settings with $q = 2, p = 10$ (left), $q = 6, p = 6$ (middle), $q = 10, p = 2$ (right). The top panel compares the accuracy of posterior predictions and the bottom panel compares the accuracy of field predictions.

interval score and does not perform well in all cases. On the other hand, $\mathcal{A}^{imspe}$ performs relatively better since it fills in interior regions. $\mathcal{A}^{imspe}$ has an overall performance that is comparable to $\mathcal{A}^{lhs}$ for calibration since both $\mathcal{A}^{imspe}$ and $\mathcal{A}^{lhs}$ do not perform a targeted sampling. $\mathcal{A}^{imspe}$ and $\mathcal{A}^{lhs}$ place the inputs far from the region of interest and these inputs provide little information about the behavior of the simulation model near the calibration region of interest. We note that although increasing the dimension of the design and parameter space does not affect the proposed methodology, it results in additional computational costs due to higher-dimensional integrals. One can use alternative sampling methods mentioned in Section 3 to address the curse of dimensionality for approximating higher-dimensional integrals. Additionally, since the proposed approaches require fewer simulation evaluations than space-filling approaches to achieve the same level of calibration goal, the computational expense is less of a concern, especially for expensive simulation models.

	$\mathcal{A}^y$	$\mathcal{A}^p$	$\mathcal{A}^{lhs}$	$\mathcal{A}^{var}$	$\mathcal{A}^{imspe}$		$\mathcal{A}^y$	$\mathcal{A}^p$	$\mathcal{A}^{lhs}$	$\mathcal{A}^{var}$	$\mathcal{A}^{imspe}$		$\mathcal{A}^y$	$\mathcal{A}^p$	$\mathcal{A}^{lhs}$	$\mathcal{A}^{var}$	$\mathcal{A}^{imspe}$
x_1	0.87	0.09	0.89	0.94	0.81	x_1	0.85	0.00	0.89	0.95	0.85	x_1	0.78	0.00	0.89	0.97	0.87
x_2	0.90	0.09	0.89	0.95	0.82	x_2	0.86	0.00	0.89	0.95	0.85	x_2	0.79	0.00	0.89	0.96	0.87
θ_1	0.69	0.78	0.89	0.98	0.88	x_3	0.86	0.00	0.89	0.95	0.84	x_3	0.78	0.00	0.89	0.96	0.87
θ_2	0.63	0.79	0.89	0.98	0.88	x_4	0.83	0.00	0.89	0.95	0.86	x_4	0.78	0.00	0.89	0.96	0.86
θ_3	0.69	0.80	0.89	0.97	0.87	x_5	0.86	0.00	0.89	0.95	0.86	x_5	0.79	0.00	0.89	0.96	0.86
θ_4	0.64	0.79	0.89	0.98	0.87	x_6	0.85	0.00	0.89	0.95	0.84	x_6	0.79	0.00	0.89	0.96	0.86
θ_5	0.70	0.79	0.89	0.98	0.85	θ_1	0.45	0.76	0.89	0.98	0.91	x_7	0.78	0.00	0.89	0.96	0.87
θ_6	0.65	0.80	0.89	0.97	0.87	θ_2	0.45	0.80	0.89	0.98	0.91	x_8	0.80	0.00	0.89	0.96	0.87
θ_7	0.61	0.78	0.89	0.98	0.86	θ_3	0.48	0.77	0.89	0.98	0.91	x_9	0.79	0.00	0.89	0.95	0.88
θ_8	0.67	0.77	0.89	0.98	0.88	θ_4	0.45	0.79	0.89	0.98	0.90	x_{10}	0.80	0.00	0.89	0.96	0.86
θ_9	0.65	0.79	0.89	0.98	0.87	θ_5	0.46	0.72	0.89	0.98	0.90	θ_1	0.14	0.86	0.89	0.99	0.93
θ_{10}	0.64	0.77	0.89	0.98	0.87	θ_6	0.51	0.77	0.89	0.98	0.90	θ_2	0.14	0.88	0.89	0.99	0.93

Table 3: The width of design inputs and the interval score of parameters selected with different acquisition functions using the examples with 12-dimensional inputs: $q = 2$, $p = 10$ (left), $q = 6$, $p = 6$ (middle), $q = 10$, $p = 2$ (right).

4.3 Application to an Epidemiological Simulation Model

We illustrate the proposed design strategy on a real data example of the COVID-19 epidemiological simulation model. The simulation outputs are generated by the COVID-19 differential equation-based simulation model presented in Yang et al. (2021). Yang et al. demonstrate how simulation outputs (e.g., forecasted daily admissions and census hospitalizations, daily and census ICU hospitalizations, confirmed cases, and deaths) helped decision makers to decide on whether the community mitigation measures should be enhanced or relaxed and to guide public policies throughout the COVID-19 epidemic in a large US city Austin, Texas. In their simulation-based optimization model, Yang et al. discard the simulation outputs that are inconsistent with observed data using the coefficient of determination (i.e., r^2) as a metric to evaluate the quality of simulation outputs, and then the optimization model is built with the filtered simulation data. In parallel to this, Sürer and Plumlee (2021) propose a filtering approach to remove unrealistic simulation outputs from the simulation data and show that the calibration with the filtered simulation data reduces the uncertainty in the parameters and the resulting predictions of the COVID-19 model. One downside of such a filtering approach in both procedures is that deciding on plausible simulation outputs based on a priori threshold value may result in oversampling or undersampling. In this sense, our approach can be considered as a systematic way of selecting simulation outputs that are consistent with observations. Therefore, the output of the proposed sequential approach (either the simulation data or the emulator) can serve as a substitute for the filtering procedures in various settings.

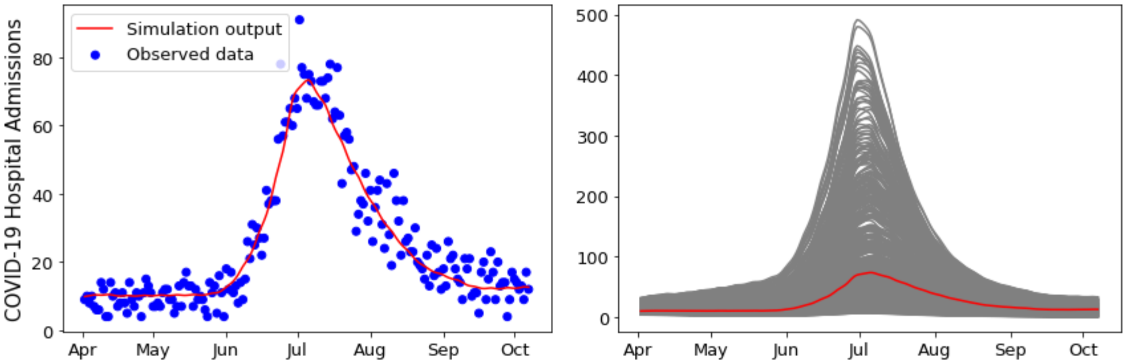

Figure 9: Illustration of the COVID-19 data from April 1 through October 6, 2020 in Austin. The left panel shows the daily COVID-19 hospital admissions. The red line corresponds to the simulation output at the best-fit parameter $\breve{\boldsymbol{\theta}}$ during the calibration period. In the right panel, gray lines are simulation outputs across design points paired with 500 different settings of the calibration parameter sampled using LHS.

Figure 9 shows the COVID-19 hospital admissions from April 1 through October 6, 2020 in Austin. Since this period is used to calibrate the simulation model in Yang et al. (2021) to initiate their projection period afterward, we consider the simulation outputs during the same period in our case study. The epidemiological simulation model is an enhanced Susceptible-Exposed-Infectious-Removed (SEIR)–style model comprising ten compartments provided in Figure 10 and the IH (hospitalized) compartment represents the quantity of interest in this case study. We consider the uncertain parameter $\boldsymbol{\theta} = (1/\sigma_I, \omega_A, 1/\gamma_Y, 1/\gamma_A)^\top$ that affects epidemiological transition dynamics between and within compartments, and the definitions are provided in Table 4. Rather than using the rates σ_I, γ_Y, and γ_A, we consider the inverse $1/\sigma_I$, $1/\gamma_Y$, and $1/\gamma_A$, which correspond to duration in days, since the prior information is provided by the experts for the distributions of durations. In Yang et al. (2021), the best-fit parameterization for the epidemiological parameter $\boldsymbol{\theta}$ is obtained at $\breve{\boldsymbol{\theta}} = (2.9, 0.66, 4, 4)^\top$ via the least-squares estimation. In our setting, each day from April 1 through October 6, 2020 (a total of 189 days) corresponds to a design input x. The red line in Figure 9 shows the simulation output across design inputs paired with $\breve{\boldsymbol{\theta}}$. Since it shows an agreement with the observed hospitalizations, we use this model output as the "true" model in our case study. In addition, we consider $\breve{\boldsymbol{\theta}}$ as a plug-in estimator when selecting inputs with $\mathcal{A}^y$ to evaluate the performance when an accurate estimate of the parameter of interest is available to the practitioner (i.e., $\hat{\boldsymbol{\theta}} = \breve{\boldsymbol{\theta}}$ in (14) throughout the procedure). Moreover, we allow $\hat{\boldsymbol{\theta}}$ to be updated after each active learning acquisition similar to the experiments presented in Sections 4.1–4.2, denoting the corresponding result as $\hat{\mathcal{A}}^y$. In addition to $\mathcal{A}^y$,

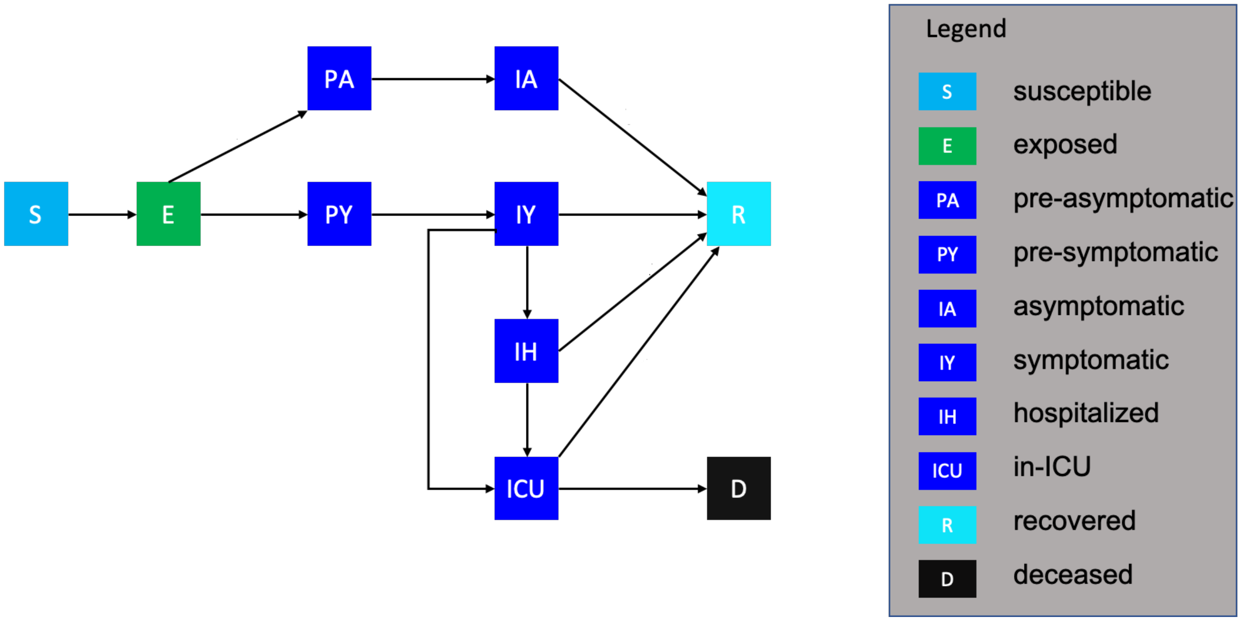

Figure 10: Diagram of the enhanced SEIR-style model comprising ten compartments (Yang et al. 2021).

Parameter	Definition	Prior
σ_I	rate at which exposed individuals become infectious (inverse)	$[2.4, 3.4]$
ω_A	infectiousness of asymptomatic individual relative to infectious individual	$[0.33, 0.99]$
γ_Y	recovery rate from symptomatic compartment (inverse)	$[3.9, 4.1]$
γ_A	recovery rate from asymptomatic compartment (inverse)	$[3.9, 4.1]$

Table 4: Epidemiological parameters and their prior ranges.

$\hat{\mathcal{A}}^y$, and $\mathcal{A}^p$, we include $\mathcal{A}^{lhs}$, $\mathcal{A}^{var}$, and $\mathcal{A}^{imspe}$ in the benchmark, as in the experiments in Section 4.2, and exclude $\mathcal{A}^{rnd}$ since its performance is similar to $\mathcal{A}^{lhs}$. The field data is observed at 13 equally spaced days starting from April 1, 2020, with 15-day intervals (i.e., $d = 13$) since a 15-day interval is enough to capture the changes in hospital admissions. Each design input and parameter is scaled to $[0, 1]$ to simplify the integrals. For each of the design inputs, the observed data is simulated as $y(x) = \eta\left(x, \boldsymbol{\theta} = \breve{\boldsymbol{\theta}}\right) + \epsilon$ where $\epsilon \sim \mathrm{N}(0, 5^2)$. We perform 30 replications with $n_0 = 50$ and $n = 150$. The set Θ_{ref} is constructed with 500 points using LHS since the grid size of the parameter space is very large. $\mathcal{X}_{\text{ref}}$ includes 189 points corresponding to each day in the time. At each iteration, a new input is acquired from a candidate list $\mathcal{L}_t$ of size 2600 constructed to simplify the optimization process. To generate $\mathcal{L}_t$, 100 parameters are sampled uniformly from the prior each of which is augmented with existing design inputs ($= 13 \times 100$) and randomly selected 13 days from the calibration period are paired with another 100 random parameters ($= 13 \times 100$).

Figure 11 summarizes the results from 30 replications. As can be seen in the left panel, the acquisition

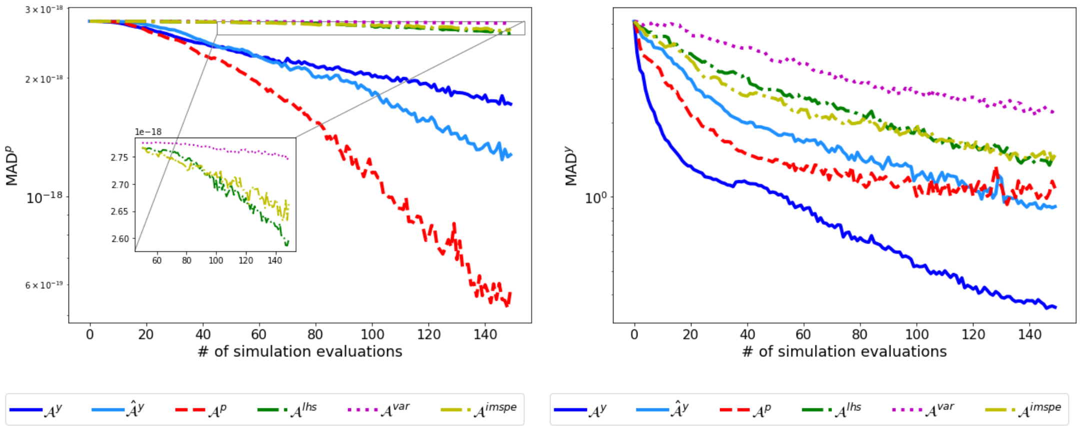

Figure 11: Comparison of different acquisition functions with the COVID-19 simulation model. The left and right panels compare the accuracy of posterior and field predictions, respectively.

function $\mathcal{A}^p$ outperforms the other functions for predicting the posterior. Even though the posterior predictions improve with $\mathcal{A}^{lhs}$, $\mathcal{A}^{var}$, and $\mathcal{A}^{imspe}$ (see the inset zoom), the improvement is negligible as compared to the proposed approaches. The right panel shows that the most accurate field predictions are obtained with $\mathcal{A}^y$. We observe the pairwise scatterplots of acquired parameters for a single replicate of $\mathcal{A}^p$ and $\mathcal{A}^y$ in Figure 12. Additionally, Table 5 provides the average interval score for calibration parameters acquired with each method across 30 replicates. Using the proposed acquisition functions, parameters $1/\sigma_I$ and ω_A are well-constrained around the best-fit parameters, which are also found as the most influential parameters for calibration in Sürer and Plumlee (2021). Moreover, the acquisition function $\mathcal{A}^y$ considers densely the regions around the best-fit parameters, whereas $\mathcal{A}^p$ focuses on a wider region since it considers the total uncertainty across the parameter space. Similarly, $\hat{\mathcal{A}}^y$ focuses on a wider parameter region of interest since varying $\hat{\boldsymbol{\theta}}$ values from one iteration to another lead to exploration of the high posterior region. Consequently, while $\hat{\mathcal{A}}^y$ achieves lower MAD^p values through exploration of the parameter region of interest, $\mathcal{A}^y$ obtains the lowest MAD^y value through exploitation of the region around $\breve{\boldsymbol{\theta}}$. Additionally, $\mathcal{A}^{imspe}$ performs similarly to $\mathcal{A}^{lhs}$, as both approaches sample from the entire input space. In contrast, $\mathcal{A}^{var}$ shows the poorest performance due to its tendency to sample primarily from the boundaries of the input space.

Figure 13 illustrates $n = 150$ simulation outputs collected with $\mathcal{A}^p$ and $\mathcal{A}^y$ for a single replicate. Both $\mathcal{A}^p$ and $\mathcal{A}^y$ select simulation outputs concentrated around 13 field data design inputs, leading to improved posterior predictions. The hospital admissions peak around July 2020 in Austin, and the simulation outputs obtained at the time of the peak using LHS range from zero to 500 in the right panel of Figure 9. In Figure 13,

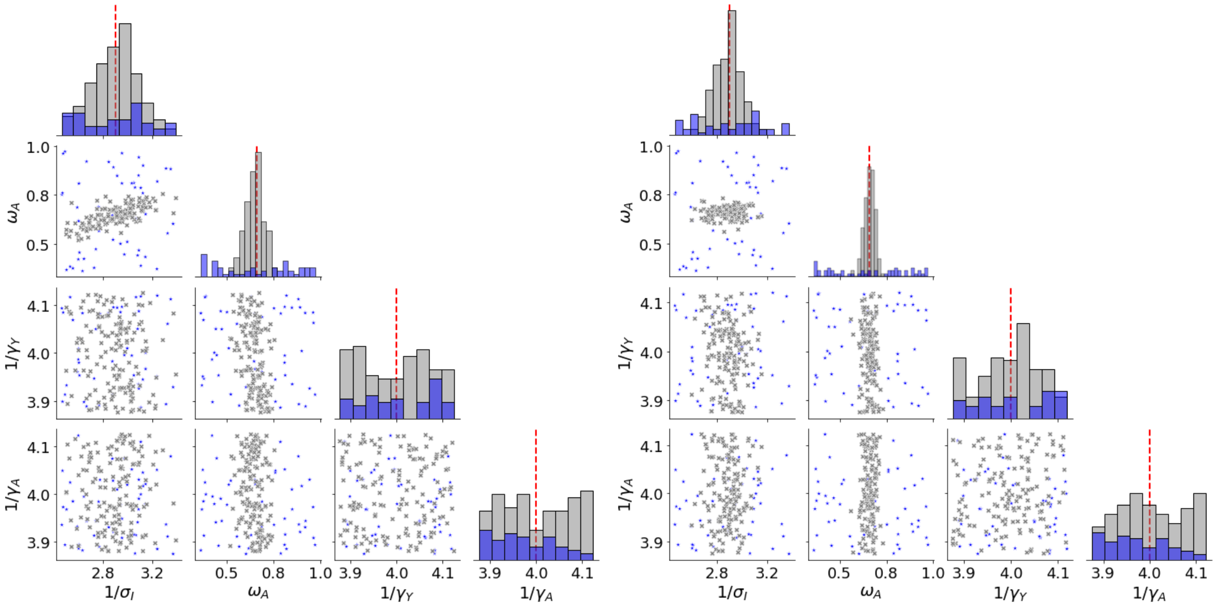

Figure 12: Pairwise plots for acquired parameters (gray cross markers) via acquisition functions $\mathcal{A}^p$ (left) and $\mathcal{A}^y$ (right). Blue stars are the initial sample obtained randomly from the prior. The red dashed line corresponds to the best-fit parameter $\breve{\boldsymbol{\theta}}$.

	$\mathcal{A}^y$	$\hat{\mathcal{A}}^y$	$\mathcal{A}^p$	$\mathcal{A}^{lhs}$	$\mathcal{A}^{var}$	$\mathcal{A}^{imspe}$
$1/\sigma_I$	0.37	0.60	0.53	0.89	0.99	0.92
ω_A	0.15	0.34	0.24	0.89	0.99	0.94
$1/\gamma_Y$	0.77	0.85	0.85	0.89	0.97	0.87
$1/\gamma_A$	0.84	0.89	0.88	0.89	0.97	0.85

Table 5: The average interval score for each calibration parameter across 30 replicates of each acquisition function.

both $\mathcal{A}^p$ and $\mathcal{A}^y$ select outputs around the region of interest (represented by the red line), and no simulation output is selected from the zero-posterior regions where the peak value is very small (i.e., less than 40 daily admissions) or very large (i.e., larger than 200 daily admissions). Overall, the acquisition function $\mathcal{A}^p$ is advantageous when the goal is to better estimate the parameters and understand their relationship through estimating the posterior density. Although $\hat{\mathcal{A}}^y$ explores the design space similar to $\mathcal{A}^y$, it provides slightly better field predictions than $\mathcal{A}^p$ through the end of the procedure since acquiring around 13 field data design inputs is adequate for $\mathcal{A}^p$ to successfully predict the field data due to the characteristics of the response surface. On the other hand, $\mathcal{A}^y$ takes advantage of targeted and consistent acquisitions around $\breve{\boldsymbol{\theta}}$ while exploring the design space for improved field predictions without losing too much of the accuracy of posterior predictions.

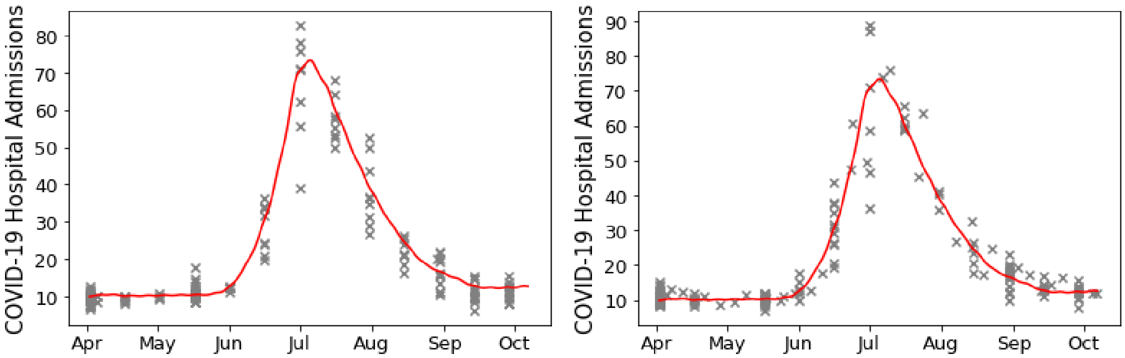

Figure 13: Simulation outputs (gray cross markers) obtained with the sequential strategy via acquisition functions $\mathcal{A}^p$ (left) and $\mathcal{A}^y$ (right). The red line illustrates the simulation outputs at $\breve{\boldsymbol{\theta}}$ across design inputs.

5 Conclusion

We propose two novel acquisition functions for improved calibration inference in an active learning setting. Our results suggest that exploitation of existing field data design points improves posterior prediction, whereas exploration of the design space along with exploitation improves field predictions. Moreover, the proposed acquisition functions encourage selecting parameters in regions of high posterior density, which is essential to accurately learn both the posterior and field data. This work can be expanded in many directions. One area is to modify the proposed acquisition functions for a field experiment design rather than a simulation experiment design to efficiently infer the calibration parameters. Following that, we further examine both one-shot and active learning settings (see, for example, Krishna et al. (2022) and Williams et al. (2011)) to find new design points to collect the field data for improved calibration of simulation models. Given the cost of performing a real physical experiment, this research would play a transformative role in optimizing the investment by guiding the optimal physical experiment design. In parallel to this, integrating these acquisition functions into the combined field and simulation experiments is the subject of another ongoing work (see comparisons in Leatherman et al. (2017) for the selection of the initial design) since both field and simulation experiments must be carefully designed to effectively use the limited resources. Moreover, sequential design can further benefit from using the proposed acquisition functions at different iterations of the sequential process to adapt to their evolving characteristics in a hybrid way. Alternatively, the performance can be explored with other design criteria such as space-filling designs in conjunction with the proposed acquisition functions. This work implements the sequential procedure in a one-at-a-time fashion. In the case of multiple processors, evaluating the simulation model in parallel with a batch of inputs is computationally more effective than the one-at-a-time procedure. Although our code implementation allows parallel runs, investigating the effect of batch size on the predictive quality and computational savings will be of great interest to practitioners. Similarly, physical experimentalists may prefer conducting a batch of experiments simultaneously due to the complexity of arranging experimental setups, and in such a case, field experiment designs with extensions allowing batch updates would be another line of future development. In this work, we utilize stationary GPs to emulate simulation models. However, it is important to note that simulation models often feature non-stationary response surfaces. Addressing non-stationary properties in GPs can vary depending on the specific model employed, necessitating tailored extensions for the proposed acquisition functions. We leave the exploration and development of these extensions for future work.

Acknowledgement

The author is grateful for support from the National Science Foundation (NSF) grant OAC 2004601. We gratefully acknowledge the computing resources provided on Bebop, a high-performance computing cluster operated by the Laboratory Computing Resource Center at Argonne National Laboratory.

References

Bayarri, M. J., Berger, J. O., and Liu, F. (2009). Modularization in Bayesian analysis, with emphasis on analysis of computer models. *Bayesian Analysis*, 4(1):119–150.

Bayarri, M. J., Berger, J. O., Paulo, R., Sacks, J., Cafeo, J. A., Cavendish, J., Lin, C.-H., and Tu, J. (2007). A framework for validation of computer models. *Technometrics*, 49(2):138–154.

Binois, M., Huang, J., Gramacy, R. B., and Ludkovski, M. (2018). Replication or exploration? Sequential design for stochastic simulation experiments. *Technometrics*, 61(1):7–23.

Brynjarsdóttir, J. and O'Hagan, A. (2014). Learning about physical parameters: the importance of model discrepancy. *Inverse Problems*, 30(11):114007.

Chen, P.-H. A., Villarreal-Marroquín, M. G., Dean, A. M., Santner, T. J., Mulyana, R., and Castro, J. M.

(2018). Sequential design of an injection molding process using a calibrated predictor. *Journal of Quality Technology*, 50(3):309–326.

Chipman, H. A., George, E. I., and McCulloch, R. E. (1998). Bayesian CART model search. *Journal of the American Statistical Association*, 93(443):935–948.

Cole, D. A., Christianson, R. B., and Gramacy, R. B. (2021). Locally induced Gaussian processes for large-scale simulation experiments. *Statistics and Computing*, 31(3):33.

Damblin, G., Barbillon, P., Keller, M., Pasanisi, A., and Parent, E. (2018). Adaptive numerical designs for the calibration of computer codes. *SIAM/ASA Journal on Uncertainty Quantification*, 6(1):151–179.

Frazier, P. I. (2018). A tutorial on Bayesian optimization. *https://arxiv.org/abs/1807.02811*.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2004). *Bayesian Data Analysis*. Chapman and Hall/CRC, second edition.

Gneiting, T. and Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378.

Gramacy, R. B. (2020). *Surrogates: Gaussian Process Modeling, Design, and Optimization for the Applied Sciences*. CRC Press; Taylor & Francis Group, New York, NY.

Gramacy, R. B. and Lee, H. K. H. (2008). Bayesian treed Gaussian process models with an application to computer modeling. *Journal of the American Statistical Association*, 103(483):1119–1130.

Gu, M. and Berger, J. O. (2016). Parallel partial Gaussian process emulation for computer models with massive output. *The Annals of Applied Statistics*, 10(3):1317–1347.

Gu, M. and Wang, L. (2018). Scaled Gaussian stochastic process for computer model calibration and prediction. *SIAM/ASA Journal on Uncertainty Quantification*, 6(4):1555–1583.

Higdon, D., Gattiker, J., Williams, B., and Rightley, M. (2008). Computer model calibration using high-dimensional output. *Journal of the American Statistical Association*, 103(482):570–583.

Higdon, D., Kennedy, M., Cavendish, J. C., Cafeo, J. A., and Ryne, R. D. (2004). Combining field data and computer simulations for calibration and prediction. *SIAM Journal on Scientific Computing*, 26(2):448–466.

Hong, L. J. and Nelson, B. L. (2006). Discrete optimization via simulation using COMPASS. *Operations Research*, 54(1):115–129.

Huang, J., Gramacy, R. B., Binois, M., and Libraschi, M. (2020). On-site surrogates for large-scale calibration. *Applied Stochastic Models in Business and Industry*, 36(2):283–304.

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

Jones, D. R., Schonlau, M., and Welch, W. J. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492.

Joseph, V. R., Dasgupta, T., Tuo, R., and Wu, C. F. J. (2015a). Sequential exploration of complex surfaces using minimum energy designs. *Technometrics*, 57(1):64–74.

Joseph, V. R., Gul, E., and Ba, S. (2015b). Maximum projection designs for computer experiments. *Biometrika*, 102(2):371–380.

Joseph, V. R., Wang, D., Gu, L., Lyu, S., and Tuo, R. (2019). Deterministic sampling of expensive posteriors using minimum energy designs. *Technometrics*, 61(3):297–308.

Kandasamy, K., Schneider, J., and Póczos, B. (2015). Bayesian active learning for posterior estimation. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 3605–3611. AAAI Press.

Kennedy, M. C. and O'Hagan, A. (2001). Bayesian calibration of computer models. *Journal of the Royal Statistical Society. Series B*, 63(3):425–464.

Koermer, S., Loda, J., Noble, A., and Gramacy, R. B. (2023). Active learning for simulator calibration. *https://arxiv.org/abs/2301.10228*.

Krishna, A., Joseph, V. R., Ba, S., Brenneman, W. A., and Myers, W. R. (2022). Robust experimental designs for model calibration. *Journal of Quality Technology*, 54(4):441–452.

Lam, C. Q. and Notz, W. I. (2008). Sequential adaptive designs in computer experiments for response surface model fit. *Statistics and Applications*, 6(1):207–233.

Lartaud, P., Humbert, P., and Garnier, J. (2024). Sequential design for surrogate modeling in Bayesian inverse problems. *https://doi.org/10.48550/arXiv.2402.16520*.

Leatherman, E. R., Dean, A. M., and Santner, T. J. (2017). Designing combined physical and computer experiments to maximize prediction accuracy. *Computational Statistics & Data Analysis*, 113:346–362.

MacKay, D. J. C. (1992). Information-based objective functions for active data selection. *Neural Computation*, 4(4):590–604.

McKay, M. D., Beckman, R. J., and Conover, W. J. (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245.

Phillips, D. R., Furnstahl, R. J., Heinz, U., Maiti, T., Nazarewicz, W., Nunes, F. M., Plumlee, M., Pratola, M. T., Pratt, S., Viens, F. G., and Wild, S. M. (2021). Get on the BAND Wagon: a Bayesian framework for quantifying model uncertainties in nuclear dynamics. *Journal of Physics G: Nuclear and Particle Physics*, page 072001.

Plumlee, M. (2017). Bayesian calibration of inexact computer models. *Journal of the American Statistical Association*, 112(519):1274–1285.

Plumlee, M., Asher, T. G., Chang, W., and Bilskie, M. V. (2021). High-fidelity hurricane surge forecasting using emulation and sequential experiments. *The Annals of Applied Statistics*, pages 460–480.

Ranjan, P., Lu, W., Bingham, D., Reese, S., Williams, B. J., Chou, C.-C., Doss, F., Grosskopf, M., and Holloway, J. P. (2011). Follow-up experimental designs for computer models and physical processes. *Journal of Statistical Theory and Practice*, 5(1):119–136.

Rasmussen, C. E. and Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press.

Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989). Statistical science. *Design and Analysis of Computer Experiments*, 4(4):409–423.

Santner, T. J., Williams, B. J., and Notz, W. I. (2018). *The Design and Analysis of Computer Experiments*. Springer Series in Statistics. Springer, New York, second edition.

Sauer, A., Gramacy, R. B., and Higdon, D. (2023). Active learning for deep Gaussian process surrogates. *Technometrics*, 65(1):4–18.

Seo, S., Wallat, M., Graepel, T., and Obermayer, K. (2000). Gaussian process regression: Active data selection and test point rejection. *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, 3:241–246.

Sung, C.-L. and Tuo, R. (2024). A review on computer model calibration. *WIREs Computational Statistics*, 16(1):e1645.

Sürer, O., Nunes, F. M., Plumlee, M., and Wild, S. M. (2022). Uncertainty quantification in breakup reactions. *Physical Review C*, 106:024607.

Sürer, O. and Plumlee, M. (2021). Calibration using emulation of filtered simulation results. In *2021 Winter Simulation Conference (WSC)*, pages 1–12.

Sürer, O., Plumlee, M., and Wild, S. M. (2024). Sequential Bayesian experimental design for calibration of expensive simulation models. *Technometrics*, 66(2):157–171.

Tuo, R. and Wu, C. F. J. (2015). Efficient calibration for imperfect computer models. *The Annals of Statistics*, 43(6):2331 – 2352.

Williams, B. J., Loeppky, J. L., Moore, L. M., and Macklem, M. S. (2011). Batch sequential design to achieve predictive maturity with calibrated computer models. *Reliability Engineering & System Safety*, 96(9):1208–1219.

Yang, H., Sürer, O., Duque, D., Morton, D. P., Singh, B., Fox, S. J., Pasco, R., Pierce, K., Rathouz, P., Valencia, V., Du, Z., Pignone, M., Escott, M. E., Adler, S. I., Johnston, S. C., and Meyers, L. A. (2021). Design of COVID-19 staged alert systems to ensure healthcare capacity with minimal closures. *Nature Communications*, 12(1):3767.

A Supplementary Material

A.1 Proofs

A.1.1 Proof of Lemma 3.1

Using Equation (2), we obtain $\mathbb{E}[\tilde{p}(\boldsymbol{\theta}|\mathbf{y})|\mathcal{D}_t] = \mathbb{E}[p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})|\mathcal{D}_t] = \mathbb{E}[p(\mathbf{y}|\boldsymbol{\theta})|\mathcal{D}_t]p(\boldsymbol{\theta})$. Thus, to complete the proof, it is enough to show $\mathbb{E}[p(\mathbf{y}|\boldsymbol{\theta})|\mathcal{D}_t] = f_{\mathcal{N}}\left(\mathbf{y}; \boldsymbol{\mu}_t\left(\boldsymbol{\theta}\right), \boldsymbol{\Sigma} + \mathbf{S}_t\left(\boldsymbol{\theta}\right)\right)$. We drop $\boldsymbol{\theta}$ from $\boldsymbol{\eta}(\boldsymbol{\theta})$, $\boldsymbol{\mu}_t\left(\boldsymbol{\theta}\right)$, and $\mathbf{S}_t\left(\boldsymbol{\theta}\right)$ for the remainder of the proof for brevity. Using Equations (3) and (7), $\mathbb{E}[p(\mathbf{y}|\boldsymbol{\theta})|\mathcal{D}_t] =$

$\int f_{\mathcal{N}}\left(\mathbf{y};\, \boldsymbol{\eta},\, \boldsymbol{\Sigma}\right) f_{\mathcal{N}}\left(\boldsymbol{\eta};\, \boldsymbol{\mu}_t,\, \mathbf{S}_t\right) d\boldsymbol{\eta}$, which is equivalent to

$$(2\pi)^{-d}|\boldsymbol{\Sigma}|^{-1/2}|\mathbf{S}_t|^{-1/2}\int \exp\left\{-\frac{1}{2}(\mathbf{y}-\boldsymbol{\eta})^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{y}-\boldsymbol{\eta})-\frac{1}{2}(\boldsymbol{\eta}-\boldsymbol{\mu}_t)^{\mathsf{T}}\mathbf{S}_t^{-1}(\boldsymbol{\eta}-\boldsymbol{\mu}_t)\right\}d\boldsymbol{\eta}. \tag{16}$$

Equation (16) can be expressed in an equivalent form

$$(2\pi)^{-d}|\boldsymbol{\Sigma}\mathbf{S}_t|^{-1/2}\int \exp\left\{-\frac{1}{2}(\mathbf{v}+\mathbf{z})^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{v}+\mathbf{z})-\frac{1}{2}\mathbf{v}^{\mathsf{T}}\mathbf{S}_t^{-1}\mathbf{v}\right\}d\mathbf{v}, \tag{17}$$

where $\mathbf{v} := \boldsymbol{\mu}_t - \boldsymbol{\eta}$ and $\mathbf{z} := \mathbf{y} - \boldsymbol{\mu}_t$. Equation (17) can be represented in matrix notation as

$$\begin{aligned}\mathbb{E}[p(\mathbf{y}|\boldsymbol{\theta})|\mathcal{D}_t] &= (2\pi)^{-d}|\boldsymbol{\Sigma}\mathbf{S}_t|^{-1/2}\int \exp\left\{-\frac{1}{2}\begin{bmatrix}\mathbf{v}\\ \mathbf{z}\end{bmatrix}^{\mathsf{T}}\begin{bmatrix}\boldsymbol{\Sigma}^{-1}+\mathbf{S}_t^{-1} & \boldsymbol{\Sigma}^{-1}\\ \boldsymbol{\Sigma}^{-1} & \boldsymbol{\Sigma}^{-1}\end{bmatrix}\begin{bmatrix}\mathbf{v}\\ \mathbf{z}\end{bmatrix}\right\}d\mathbf{v}\\ &= \int f_{\mathcal{N}}\left(\begin{bmatrix}\mathbf{v}\\ \mathbf{z}\end{bmatrix};\mathbf{0},\begin{bmatrix}\mathbf{S}_t & -\mathbf{S}_t\\ -\mathbf{S}_t & \boldsymbol{\Sigma}+\mathbf{S}_t\end{bmatrix}\right)d\mathbf{v}.\end{aligned}$$

Following the Gaussian identities in the appendix of (Rasmussen and Williams 2005, Equation (A.6)) completes the proof.

Similarly, we obtain $\mathbb{V}[\tilde{p}(\boldsymbol{\theta}|\mathbf{y})|\mathcal{D}_t] = \mathbb{V}[p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})|\mathcal{D}_t] = \mathbb{V}[p(\mathbf{y}|\boldsymbol{\theta})|\mathcal{D}_t]p(\boldsymbol{\theta})^2$. By definition, $\mathbb{V}[p(\mathbf{y}|\boldsymbol{\theta})|\mathcal{D}_t] = \mathbb{E}[p(\mathbf{y}|\boldsymbol{\theta})^2|\mathcal{D}_t] - \mathbb{E}[p(\mathbf{y}|\boldsymbol{\theta})|\mathcal{D}_t]^2$. From the first proof, we have $\mathbb{E}[p(\mathbf{y}|\boldsymbol{\theta})|\mathcal{D}_t]^2 = f_{\mathcal{N}}\left(\mathbf{y};\, \boldsymbol{\mu}_t,\, \boldsymbol{\Sigma}+\mathbf{S}_t\right)^2$. Thus, it suffices to show that $\mathbb{E}[p(\mathbf{y}|\boldsymbol{\theta})^2|\mathcal{D}_t] = \frac{1}{2^d\pi^{d/2}|\boldsymbol{\Sigma}|^{1/2}}f_{\mathcal{N}}\left(\mathbf{y};\, \boldsymbol{\mu}_t,\, \frac{1}{2}\boldsymbol{\Sigma}+\mathbf{S}_t\right)$. We obtain $\mathbb{E}[p(\mathbf{y}|\boldsymbol{\theta})^2|\mathcal{D}_t] = \int \left(f_{\mathcal{N}}\left(\mathbf{y};\, \boldsymbol{\eta},\, \boldsymbol{\Sigma}\right)\right)^2 f_{\mathcal{N}}\left(\boldsymbol{\eta};\, \boldsymbol{\mu}_t,\, \mathbf{S}_t\right) d\boldsymbol{\eta}$, which is equivalent to

$$= \frac{1}{(2\pi)^{3d/2}|\boldsymbol{\Sigma}\mathbf{S}_t\boldsymbol{\Sigma}|^{1/2}}\int \exp\left\{-\frac{1}{2}\left(2(\mathbf{y}-\boldsymbol{\eta})^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{y}-\boldsymbol{\eta})+(\boldsymbol{\eta}-\boldsymbol{\mu}_t)^{\mathsf{T}}\mathbf{S}_t^{-1}(\boldsymbol{\eta}-\boldsymbol{\mu}_t)\right)\right\}d\boldsymbol{\eta}. \tag{18}$$

Defining again $\mathbf{v} := \boldsymbol{\mu}_t - \boldsymbol{\eta}$ and $\mathbf{z} := \mathbf{y} - \boldsymbol{\mu}_t$, Equation (18) becomes

$$\begin{aligned}&= \frac{1}{(2\pi)^{3d/2}|\boldsymbol{\Sigma}|^{1/2}2^{d/2}\left|\frac{1}{2}\boldsymbol{\Sigma}\mathbf{S}_t\right|^{1/2}}\int \exp\left\{-\frac{1}{2}\begin{bmatrix}\mathbf{v}\\ \mathbf{z}\end{bmatrix}^{\mathsf{T}}\begin{bmatrix}2\boldsymbol{\Sigma}^{-1}+\mathbf{S}_t^{-1} & 2\boldsymbol{\Sigma}^{-1}\\ 2\boldsymbol{\Sigma}^{-1} & 2\boldsymbol{\Sigma}^{-1}\end{bmatrix}\begin{bmatrix}\mathbf{v}\\ \mathbf{z}\end{bmatrix}\right\}d\mathbf{v}\\ &= \frac{1}{2^d\pi^{d/2}|\boldsymbol{\Sigma}|^{1/2}}\int f_{\mathcal{N}}\left(\begin{bmatrix}\mathbf{v}\\ \mathbf{z}\end{bmatrix};\mathbf{0},\begin{bmatrix}\mathbf{S}_t & -\mathbf{S}_t\\ -\mathbf{S}_t & \frac{1}{2}\boldsymbol{\Sigma}+\mathbf{S}_t\end{bmatrix}\right)d\mathbf{v}.\end{aligned}$$

Marginalizing over $\mathbf{v}$ completes the proof.

A.1.2 Proof of Lemma 3.2

For any input $\mathbf{z} = \left(\mathbf{x}^\top, \boldsymbol{\theta}^\top\right)^\top$, recall that $m_t(\mathbf{z})$ and $\varsigma_t^2(\mathbf{z})$ are the emulator mean and variance at iteration t. Suppose that we observe the hypothetical output $\eta^* := \eta(\mathbf{x}^*, \boldsymbol{\theta}^*)$ for any $\mathbf{z}^* = \left(\mathbf{x}^{*\top}, \boldsymbol{\theta}^{*\top}\right)^\top$. After seeing the simulation data set $\mathcal{D}_{t+1}$ that includes $(\mathbf{z}^*, \eta^*)$ (i.e., $\mathcal{D}_{t+1} = (\mathbf{z}^*, \eta^*) \cup \mathcal{D}_t$), we obtain

$$\begin{aligned} m_{t+1}(\mathbf{z}) &= \left[\mathbf{k}_t(\mathbf{z})^\top,\ k_t(\mathbf{z}, \mathbf{z}^*)\right] \begin{bmatrix} \mathbf{K}_t & \mathbf{k}_t(\mathbf{z}^*) \\ \mathbf{k}_t(\mathbf{z}^*)^\top & k_t(\mathbf{z}^*, \mathbf{z}^*) + \upsilon \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{\eta}_t \\ \eta^* \end{bmatrix} \\ &= m_t(\mathbf{z}) + \frac{\mathrm{cov}_t(\mathbf{z}, \mathbf{z}^*)}{\varsigma_t^2(\mathbf{z}^*) + \upsilon}(\eta^* - m_t(\mathbf{z}^*)). \end{aligned} \tag{19}$$

Using a similar line of reasoning for both variance and covariance functions, we have

$$\varsigma_{t+1}^2(\mathbf{z}) = \varsigma_t^2(\mathbf{z}) - \frac{\mathrm{cov}_t(\mathbf{z}, \mathbf{z}^*)^2}{\varsigma_t^2(\mathbf{z}^*) + \upsilon} \qquad \text{and} \qquad \mathrm{cov}_{t+1}(\mathbf{z}, \mathbf{z}') = \mathrm{cov}_t(\mathbf{z}, \mathbf{z}') - \frac{\mathrm{cov}_t(\mathbf{z}, \mathbf{z}^*)\mathrm{cov}_t(\mathbf{z}', \mathbf{z}^*)}{\varsigma_t^2(\mathbf{z}^*) + \upsilon}. \tag{20}$$

Taking the expected value, variance, and covariance of Equation (19), respectively, provides

$$\begin{aligned} \mathbb{E}_{\eta^*|\mathcal{D}_t}\left[m_{t+1}(\mathbf{z})\right] = m_t(\mathbf{z}), \quad \mathbb{V}_{\eta^*|\mathcal{D}_t}\left[m_{t+1}(\mathbf{z})\right] = \frac{\mathrm{cov}_t(\mathbf{z}, \mathbf{z}^*)^2}{\varsigma_t^2(\mathbf{z}^*) + \upsilon}, \quad \text{and} \\ \mathbb{C}_{\eta^*|\mathcal{D}_t}\left[m_{t+1}(\mathbf{z}), m_{t+1}(\mathbf{z}')\right] = \frac{\mathrm{cov}_t(\mathbf{z}, \mathbf{z}^*)\mathrm{cov}_t(\mathbf{z}', \mathbf{z}^*)}{\varsigma_t^2(\mathbf{z}^*) + \upsilon}. \end{aligned} \tag{21}$$

Using $\eta^*|\boldsymbol{\eta}_t \sim \mathrm{N}\left(m_t(\mathbf{z}^*),\ \varsigma_t^2(\mathbf{z}^*) + \upsilon\right)$ and the transformation in Equation (19), we have

$$m_{t+1}(\mathbf{z})|\mathcal{D}_t \sim \mathrm{N}\left(m_t(\mathbf{z}),\ \frac{\mathrm{cov}_t(\mathbf{z}, \mathbf{z}^*)^2}{\varsigma_t^2(\mathbf{z}^*) + \upsilon}\right). \tag{22}$$

Recall that $\boldsymbol{\mu}_t\left(\boldsymbol{\theta}\right)$ represents the mean vector of simulation outputs at field data design inputs paired with $\boldsymbol{\theta}$ at iteration t. Then, Equation (22) implies

$$\boldsymbol{\mu}_{t+1}\left(\boldsymbol{\theta}\right)|\mathcal{D}_t \sim \mathrm{MVN}(\boldsymbol{\mu}_t\left(\boldsymbol{\theta}\right), \boldsymbol{\phi}_t\left(\boldsymbol{\theta}, \mathbf{z}^*\right)), \tag{23}$$

where the ith diagonal element of $\boldsymbol{\phi}_t\left(\boldsymbol{\theta}, \mathbf{z}^*\right)$ is $\frac{\operatorname{cov}_t\left(\mathbf{z}_i^f, \mathbf{z}^*\right)^2}{\varsigma_t^2(\mathbf{z}^*)+\upsilon}$ and (i,j)th element of $\boldsymbol{\phi}_t\left(\boldsymbol{\theta}, \mathbf{z}^*\right)$ is $\frac{\operatorname{cov}_t\left(\mathbf{z}_i^f, \mathbf{z}^*\right)\operatorname{cov}_t\left(\mathbf{z}_j^f, \mathbf{z}^*\right)}{\varsigma_t^2(\mathbf{z}^*)+\upsilon}$ with $\mathbf{z}_i^f = \left(\mathbf{x}_i^{f\top}, \boldsymbol{\theta}^\top\right)^\top$ for $i, j = 1, \ldots, d$. In addition, Equation (20) implies $\mathbf{S}_{t+1}(\boldsymbol{\theta}) = \mathbf{S}_t\left(\boldsymbol{\theta}\right) - \boldsymbol{\phi}_t\left(\boldsymbol{\theta}, \mathbf{z}^*\right)$ and notice that $\mathbf{S}_{t+1}(\boldsymbol{\theta})$ does not depend on η^*.

For the rest of the proof, we omit $\boldsymbol{\theta}$ from $\boldsymbol{\mu}_t\left(\boldsymbol{\theta}\right)$, $\mathbf{S}_t\left(\boldsymbol{\theta}\right)$, $\boldsymbol{\mu}_{t+1}\left(\boldsymbol{\theta}\right)$, and $\mathbf{S}_{t+1}(\boldsymbol{\theta})$ for brevity. From Lemma 3.1, we have

$$\mathbb{V}[p(\mathbf{y}|\boldsymbol{\theta})|(\mathbf{z}^*, \eta^*) \cup \mathcal{D}_t] = \frac{1}{2^d \pi^{d/2} |\boldsymbol{\Sigma}|^{1/2}} f_{\mathcal{N}}\left(\mathbf{y};\, \boldsymbol{\mu}_{t+1},\, \frac{1}{2}\boldsymbol{\Sigma} + \mathbf{S}_{t+1}\right) - \left(f_{\mathcal{N}}\left(\mathbf{y};\, \boldsymbol{\mu}_{t+1},\, \boldsymbol{\Sigma} + \mathbf{S}_{t+1}\right)\right)^2.$$

Using Equation (23) and replacing $\mathbf{S}_{t+1}$ with $\mathbf{S}_t - \boldsymbol{\phi}_t\left(\boldsymbol{\theta}, \mathbf{z}^*\right)$, we obtain $\mathbb{E}_{\eta^*|\mathcal{D}_t}\left(\mathbb{V}[p(\mathbf{y}|\boldsymbol{\theta})\,|(\mathbf{z}^*, \eta^*) \cup \mathcal{D}_t]\right)$ as

$$\begin{aligned} &= \int \frac{1}{2^d \pi^{d/2} |\boldsymbol{\Sigma}|^{1/2}} f_{\mathcal{N}}\left(\mathbf{y};\, \boldsymbol{\mu}_{t+1},\, \frac{1}{2}\boldsymbol{\Sigma} + \mathbf{S}_t - \boldsymbol{\phi}_t\left(\boldsymbol{\theta}, \mathbf{z}^*\right)\right) f_{\mathcal{N}}\left(\boldsymbol{\mu}_{t+1};\, \boldsymbol{\mu}_t,\, \boldsymbol{\phi}_t\left(\boldsymbol{\theta}, \mathbf{z}^*\right)\right) d\boldsymbol{\mu}_{t+1} \\ &- \int \left(f_{\mathcal{N}}\left(\mathbf{y};\, \boldsymbol{\mu}_{t+1},\, \boldsymbol{\Sigma} + \mathbf{S}_t - \boldsymbol{\phi}_t\left(\boldsymbol{\theta}, \mathbf{z}^*\right)\right)\right)^2 f_{\mathcal{N}}\left(\boldsymbol{\mu}_{t+1};\, \boldsymbol{\mu}_t,\, \boldsymbol{\phi}_t\left(\boldsymbol{\theta}, \mathbf{z}^*\right)\right) d\boldsymbol{\mu}_{t+1}. \end{aligned} \tag{24}$$

The rest of the proof follows from Sürer et al. (2024), and we provide the remainder for the sake of completeness. Defining $\mathbf{L} \coloneqq \frac{1}{2}\boldsymbol{\Sigma} + \mathbf{S}_t - \boldsymbol{\phi}_t\left(\boldsymbol{\theta}, \mathbf{z}^*\right)$, $\mathbf{M} \coloneqq \boldsymbol{\Sigma} + \mathbf{S}_t - \boldsymbol{\phi}_t\left(\boldsymbol{\theta}, \mathbf{z}^*\right)$, and $a_1 \coloneqq \frac{2^{-d}\pi^{-d/2}|\boldsymbol{\Sigma}|^{-1/2}}{(2\pi)^d |\mathbf{L}\boldsymbol{\phi}_t(\boldsymbol{\theta}, \mathbf{z}^*)|^{1/2}}$, $a_2 \coloneqq \frac{(2\pi)^{-3d/2}}{|\mathbf{M}\boldsymbol{\phi}_t(\boldsymbol{\theta}, \mathbf{z}^*)\mathbf{M}|^{1/2}}$, and assuming $\mathbf{L}$ and $\mathbf{M}$ are invertible, (24) is equivalently written as

$$\begin{aligned} &a_1 \int \exp\left\{-\frac{1}{2}\left(\left(\mathbf{y} - \boldsymbol{\mu}_{t+1}\right)^\top \mathbf{L}^{-1}\left(\mathbf{y} - \boldsymbol{\mu}_{t+1}\right) + \left(\boldsymbol{\mu}_{t+1} - \boldsymbol{\mu}_t\right)^\top \boldsymbol{\phi}_t\left(\boldsymbol{\theta}, \mathbf{z}^*\right)^{-1}\left(\boldsymbol{\mu}_{t+1} - \boldsymbol{\mu}_t\right)\right)\right\} d\boldsymbol{\mu}_{t+1} \\ &- a_2 \int \exp\left\{-\frac{1}{2}\left(2\left(\mathbf{y} - \boldsymbol{\mu}_{t+1}\right)^\top \mathbf{M}^{-1}\left(\mathbf{y} - \boldsymbol{\mu}_{t+1}\right) + \left(\boldsymbol{\mu}_{t+1} - \boldsymbol{\mu}_t\right)^\top \boldsymbol{\phi}_t\left(\boldsymbol{\theta}, \mathbf{z}^*\right)^{-1}\left(\boldsymbol{\mu}_{t+1} - \boldsymbol{\mu}_t\right)\right)\right\} d\boldsymbol{\mu}_{t+1}. \end{aligned} \tag{25}$$

Letting $\mathbf{v} \coloneqq \boldsymbol{\mu}_t - \boldsymbol{\mu}_{t+1}$ and $\mathbf{z} \coloneqq \mathbf{y} - \boldsymbol{\mu}_t$, and writing Equation (25) in matrix notation yields

$$\begin{aligned} &= \frac{1}{2^d \pi^{d/2} |\boldsymbol{\Sigma}|^{1/2}} \int f_{\mathcal{N}}\left(\begin{bmatrix} \mathbf{v} \\ \mathbf{z} \end{bmatrix};\, \mathbf{0},\, \begin{bmatrix} \boldsymbol{\phi}_t\left(\boldsymbol{\theta}, \mathbf{z}^*\right) & -\boldsymbol{\phi}_t\left(\boldsymbol{\theta}, \mathbf{z}^*\right) \\ -\boldsymbol{\phi}_t\left(\boldsymbol{\theta}, \mathbf{z}^*\right) & \mathbf{L} + \boldsymbol{\phi}_t\left(\boldsymbol{\theta}, \mathbf{z}^*\right) \end{bmatrix}\right) d\mathbf{v} \\ &- \frac{1}{2^d \pi^{d/2} |\mathbf{M}|^{1/2}} \int f_{\mathcal{N}}\left(\begin{bmatrix} \mathbf{v} \\ \mathbf{z} \end{bmatrix};\, \mathbf{0},\, \begin{bmatrix} \boldsymbol{\phi}_t\left(\boldsymbol{\theta}, \mathbf{z}^*\right) & -\boldsymbol{\phi}_t\left(\boldsymbol{\theta}, \mathbf{z}^*\right) \\ -\boldsymbol{\phi}_t\left(\boldsymbol{\theta}, \mathbf{z}^*\right) & \frac{1}{2}\mathbf{M} + \boldsymbol{\phi}_t\left(\boldsymbol{\theta}, \mathbf{z}^*\right) \end{bmatrix}\right) d\mathbf{v}. \end{aligned} \tag{26}$$

Marginalizing over $\mathbf{v}$ proves that

$$\mathbb{E}_{\eta^*|\mathcal{D}_t}\left(\mathbb{V}[p(\mathbf{y}|\boldsymbol{\theta})\,|(\mathbf{z}^*,\eta^*)\cup\mathcal{D}_t]\right) = \frac{f_{\mathcal{N}}\left(\mathbf{y};\,\boldsymbol{\mu}_t,\,\frac{1}{2}\boldsymbol{\Sigma}+\mathbf{S}_t\right)}{2^d\pi^{d/2}|\boldsymbol{\Sigma}|^{1/2}} - \frac{f_{\mathcal{N}}\left(\mathbf{y};\,\boldsymbol{\mu}_t,\,\frac{1}{2}\left(\boldsymbol{\Sigma}+\mathbf{S}_t+\boldsymbol{\phi}_t\left(\boldsymbol{\theta},\mathbf{z}^*\right)\right)\right)}{2^d\pi^{d/2}|\boldsymbol{\Sigma}+\mathbf{S}_t-\boldsymbol{\phi}_t\left(\boldsymbol{\theta},\mathbf{z}^*\right)|^{1/2}}.$$

A.2 Analysis on Initial Sample

We investigate the effect of the initial sample size n_0 on the performance of $\mathcal{A}^p$ and $\mathcal{A}^y$ using the two- and three-dimensional simulation models presented in Section 4.1. We vary the number of observations as $n_0 \in \{5, 10, 20, 40\}$ and $n_0 \in \{15, 30, 60, 120\}$ for the two- and three-dimensional functions, respectively, and summarize the results over a single replicate to illustrate the effect of n_0. The algorithm terminates upon reaching a total of 100 simulation evaluations (i.e., $n + n_0 = 100$) for the first example and 180 evaluations (i.e., $n + n_0 = 180$) for the second example. The results are shown in Figure 14 for different values of n_0. In both examples, $\mathcal{A}^y$ has larger errors for smaller values of n_0 early in the algorithm. However, for the two-dimensional model, $\mathcal{A}^y$ with smaller n_0 values (i.e., $n_0 = 5$ and $n_0 = 10$) achieves convergence with fewer simulation evaluations compared to those with larger n_0 values (i.e., $n_0 = 20$ and $n_0 = 40$) thanks to the fast convergence rate of $\mathcal{A}^y$. In such a case, if the algorithm terminates upon achieving the desired accuracy level, larger n_0 values would result in wasted computational resources, especially for simulation models with long run times. On the other hand, for the three-dimensional model, the initial sample size $n_0 = 30$ allows a more thorough exploration of the complex response surface, which in turn improves the overall performance of $\mathcal{A}^y$ as compared to $n_0 = 15$. In this example, exploration with $n_0 = 30$ prevents $\mathcal{A}^y$ from getting stuck on local optimal regions. On the other hand, $\mathcal{A}^p$ takes advantage of its fast convergence rate with smaller initial sample sizes in both examples. Overall, the initial sample size plays an important role in the algorithm's performance and an appropriate initial sample size depends on the complexity of the response surface, computational resources, and convergence rate of the acquisition function for a particular application.

A.3 Details for Experiments with High Dimensional Inputs

We test the proposed approaches using varying values for the dimensions of the design and parameter spaces. We maintain the input space dimension at $q + p = 12$ and generate three scenarios similar to those in Sürer

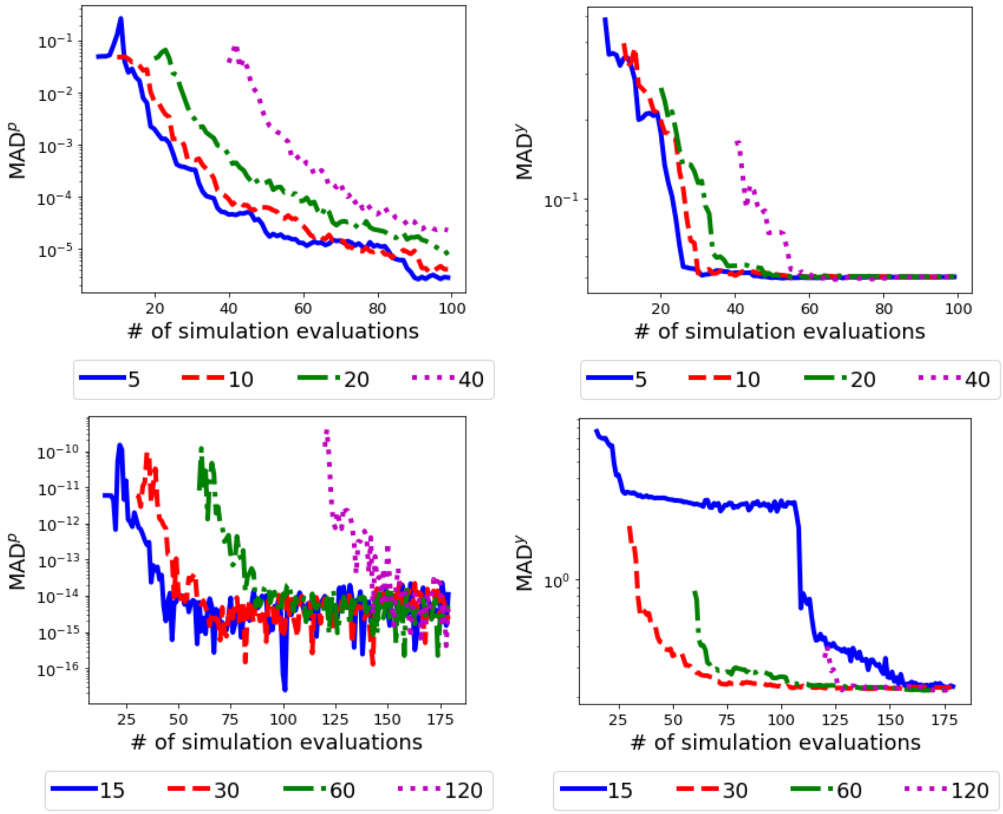

Figure 14: Experiment results for varying values of initial sample $n_0 \in \{5, 10, 20, 40\}$ ($n_0 \in \{15, 30, 60, 120\}$) using the two-dimensional (three-dimensional) simulation model in Figure 3 (Figure 4). The top and bottom panels illustrate results for two- and three-dimensional models, respectively. The left and right panels compare the accuracy of posterior and field predictions using $\mathcal{A}^p$ and $\mathcal{A}^y$, respectively.

et al. (2024). The data generation mechanism for examples with higher dimensional input spaces is provided below.

- For the example with $q = 2$ and $p = 10$, we consider $\mathbf{x} = (x_1, x_2) \in [0, 1]^2$, $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_{10}) \in [-5, 5]^{10}$, and $\eta(\mathbf{x}, \boldsymbol{\theta}) = \sqrt{x_1 + x_2}(\theta_1 + \cdots + \theta_{10})^2$. The field data is generated through $y\left(\mathbf{x}_i^f\right) = \eta\left(\mathbf{x}_i^f, \boldsymbol{\theta} = \breve{\boldsymbol{\theta}}\right) + \epsilon$ with $\epsilon \sim \mathrm{N}(0, 25)$, $\mathbf{x}_i^f = \mathbf{0.5}_2$, $i = 1, \ldots, 4$, and $\breve{\boldsymbol{\theta}} = \mathbf{0}_{10}$.

- For the example with $q = 6$ and $p = 6$, we consider $\mathbf{x} = (x_1, \ldots, x_6) \in [0, 1]^6$, $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_6) \in [-5, 5]^6$, and $\eta(\mathbf{x}, \boldsymbol{\theta}) = \sqrt{x_1 + \cdots + x_6}(\theta_1 + \cdots + \theta_6)^2$. The field data is generated through $y\left(\mathbf{x}_i^f\right) = \eta\left(\mathbf{x}_i^f, \boldsymbol{\theta} = \breve{\boldsymbol{\theta}}\right) + \epsilon$ with $\epsilon \sim \mathrm{N}(0, 5)$, $\mathbf{x}_i^f = \mathbf{0.5}_6$, $i = 1, \ldots, 4$, and $\breve{\boldsymbol{\theta}} = \mathbf{0}_6$.

- For the example with $q = 10$ and $p = 2$, we consider $\mathbf{x} = (x_1, \ldots, x_{10}) \in [0, 1]^{10}$, $\boldsymbol{\theta} = (\theta_1, \theta_2) \in [-5, 5]^2$, and $\eta(\mathbf{x}, \boldsymbol{\theta}) = \sqrt{x_1 + \cdots + x_{10}}(\theta_1 + \theta_2)^2$. The field data is generated through $y\left(\mathbf{x}_i^f\right) =$

$\eta\left(\mathbf{x}_i^f, \boldsymbol{\theta} = \breve{\boldsymbol{\theta}}\right) + \epsilon$ with $\epsilon \sim \mathrm{N}(0,1)$, $\mathbf{x}_i^f = \mathbf{0.5}_{10}$, $i = 1, \ldots, 4$, and $\breve{\boldsymbol{\theta}} = \mathbf{0}_2$.

In addition to the space-filling design $\mathcal{A}^{lhs}$, we investigate the difference between the proposed acquisition functions $\mathcal{A}^p$ and $\mathcal{A}^y$ and the two other common acquisition functions using the examples with high dimensional input spaces. As mentioned in the introduction, one common acquisition strategy is to select the next point where the emulator uncertainty is highest (Seo et al. 2000). In the experiments, the corresponding method abbreviated by $\mathcal{A}^{var}$ uses

$$\mathbf{z}^{\text{new}} \in \underset{\mathbf{z}^* \in \mathcal{L}_t}{\arg\max}\, \varsigma_t^2(\mathbf{z}^*) \tag{27}$$

in place of line 6 of Algorithm 1. One drawback of $\mathcal{A}^{var}$ is that it tends to choose inputs from the boundaries. As an alternative, the integrated mean squared prediction error (IMSPE) considers the emulator uncertainty integrated over the input space to avoid inputs at the boundary locations. The associated method labelled $\mathcal{A}^{imspe}$ replaces line 6 of Algorithm 1 with

$$\mathbf{z}^{\text{new}} \in \underset{\mathbf{z}^* \in \mathcal{L}_t}{\arg\min} \sum_{\mathbf{z} \in \mathcal{Z}_{\text{ref}}} \varsigma_{t+1}^2(\mathbf{z}). \tag{28}$$

Here, $\varsigma_{t+1}^2(\mathbf{z})$ is obtained via (20) and depends on the candidate input $\mathbf{z}^*$ and $\mathcal{Z}_{\text{ref}}$ is a reference set within the $[\mathcal{X}, \Theta]$ space.

At each replication, the initial design of size n_0 is randomly selected from a uniform distribution, and all methods $\mathcal{A}^p$, $\mathcal{A}^y$, $\mathcal{A}^{lhs}$, $\mathcal{A}^{var}$, and $\mathcal{A}^{imspe}$ utilize the identical initial sample to ensure a fair comparison. We set $n_0 = 30$ for the examples with $q = 6$, $p = 6$ and $q = 10$, $p = 2$. For the large p case, due to large variability in the performance metrics during the earlier stages of all methods, we set $n_0 = 50$ when $q = 2$, $p = 10$. The field data is rerandomized at each replication, and the same field data is employed across different methods within each replication. To construct the discrete set of inputs $\mathcal{L}_t$, first, each unique field data design input is paired with each of 500 parameters sampled from a uniform prior in Θ space to facilitate the exploitation of field data design inputs. Then, another 1000 inputs are randomly sampled from the prior in $[\mathcal{X}, \Theta]$ space to allow exploration. The reference sets Θ_{ref}, $\mathcal{X}_{\text{ref}}$, and $\mathcal{Z}_{\text{ref}}$ are constructed with 1500 points generated from LHS. Similar to the experiments in Section 4.1, we compute the performance metrics MAD_t^p and MAD_t^y at each iteration.

A.4 Code and Data Availability

The sequential procedure is implemented in the Python software package Parallel Uncertainty Quantification (PUQ). For practical purposes, the implementation allows users to run a simulation model in a parallel mode as well. PUQ is an open-source software package at https://github.com/parallelUQ/PUQ/tree/dev/jqt_paper. The COVID-19 simulation model is also made publicly available under our repository. The README file contains instructions to install the package and provides a guideline to replicate illustrative examples and a prominent result from the paper.