Two-Stage Bayesian Optimization for Scalable Inference in State Space Models

Mahdi Imani and Seyede Fatemeh Ghoreishi

***Abstract*—State-space models (SSMs) are a rich class of dynamical models with a wide range of applications in economics, healthcare, computational biology, robotics, and more. Proper analysis, control, learning, and decision-making in dynamical systems modeled by SSMs depend on the accuracy of the inferred/learned model. Most of the existing inference techniques for SSMs are capable of dealing with very small systems, unable to be applied to most of the large-scale practical problems. Toward this, this paper introduces a two-stage Bayesian optimization framework for scalable and efficient inference in SSMs. The proposed framework maps the original large parameter space to a reduced space, containing a small linear combination of the original space. This reduced space, which captures the most variability in the inference function (e.g., log-likelihood or log-aposteriori), is obtained by eigenvalue decomposition of the covariance of gradients of the inference function approximated by particle filtering scheme. Then, an exponential reduction in the search space of parameters during the inference process is achieved through the proposed two-stage Bayesian optimization (BO) policy, where the solution of the first-stage BO policy in the reduced space specifies the search space of the second-stage BO in the original space. The proposed framework's accuracy and speed are demonstrated through several experiments, including real metagenomics data from a gut microbial community.**

***Index Terms*—State-Space Models, Parameter Estimation, Bayesian Optimization.**

I. Introduction

Modeling, learning, and decision making are becoming critical parts of our everyday's lives, including robotics [1], healthcare [2], smart grids [3], finance [4], self-driving cars [5], and many more. Many practical systems that we are dealing with are dynamic, large, complex, and uncertain, mostly observed through imperfect data acquired from sensors/technologies. State-space models (SSMs), also known as the hidden Markov model (HMM), are perhaps one of the most popular classes of dynamical models [6].

The main step toward modeling a system by SSMs is to infer/estimate a set of unknown parameters given a sequence of observed measurements. Several techniques have been developed for the inference of SSMs. These include: 1) direct gradient-based maximum likelihood techniques [6–11], which try to maximize the log-likelihood function using gradient-ascent or quasi-Newton techniques; 2) expectation-maximization techniques [12–14], which attempt to maximize the "complete" log-likelihood function instead of the "incomplete" one using the fact that maximizing the complete log-likelihood is easier than the incomplete one; 3) Bayesian techniques [15–18], which aim to take the prior knowledge into account during the inference process; and 4) surrogate-based techniques [19–22], which construct a surrogate model for the main objective function during the inference process and try to find its maximizer iteratively.

All aforementioned techniques become intractable or very slow in dealing with big data and large systems. The reason is that the aforementioned techniques require extensive sampling of the parameter space in order to find the maximizer of the inference function (e.g., log-likelihood or log-aposteriori). This results in intractability or poor performance of the existing techniques, primarily due to the following reasons:

1. *Huge computational cost of a single evaluation:* Evaluation/approximation of the log-likelihood or log-posterior at any single parameter sample point requires performing a sequential Monte Carlo (SMC) based technique [6, 23]. The computational complexity of SMC approximation increases exponentially with the system's size, leading to poor performance or intractability of the existing techniques that rely on excessive inference function evaluations.
2. *Large Parameter Space:* Complexity of most practical problems poses a huge uncertainty in the modeling process, which appears in terms of large number of parameters in the modeling process. The amount of search over the parameter space for a proper inference process increases significantly with the size of the parameter space, resulting in intractability or poor performance of the existing techniques.

Toward this, this paper introduces a two-stage Bayesian optimization framework, which consists of dimensionality reduction and sample selection processes. In the reduction process, an eigenvalue decomposition based technique is developed for mapping the original large parameter space to reduced space in which the inference function has the highest variability. Then, in the selection process, an efficient sequential search over the parameter space is achieved by performing two consecutive Bayesian optimization (BO) policies. The first Bayesian optimization selects the best sample in the reduced space, and the inverse map of the selected sample to the original parameter space, which is a hyperplane, specifies the search space for the second Bayesian optimization.

The main advantage of the proposed framework over the existing techniques is the capability of dealing with expensive-to-evaluate inference functions (e.g., likelihood or posterior), which is often the case in large systems and big datasets.

M. Imani is with the Department of Electrical and Computer Engineering at the George Washington University, and S. F. Ghoreishi is with the Institute for Systems Research at the University of Maryland. (e-mails: mimani@gwu.edu, sfg@umd.edu)

In these cases, intelligent selection of the search space for finding the maximizer of the inference function is extremely critical, as the number of inference function evaluations over the parameter space is extremely limited. This is done by the proposed two-stage Bayesian optimization framework by allowing selection of samples in the lower-dimensional space in which much more informed selection can be achieved. The non-parametric Bayesian representation of the inference function by the proposed framework enables risk consideration in the inference process, which is a key in sensitive domains, where deterministic modeling might lead to unreliability of the inferred model. This Bayesian surrogate model allows prediction of the distribution of the inference function and selection of the next sample by considering all potential outcomes instead of a single predicted value. This is a key advantage of the proposed framework compared with the existing inference techniques, which allows handling of expensive-to-evaluate inference functions commonly encountered in large systems.

The article is organized as follows. In Section II, the state-space models and particle filters for their inference are briefly described. In Section III, the proposed two-stage Bayesian optimization framework is introduced. Finally, Section IV and Section V contain numerical examples and concluding remarks, respectively.

II. Background

A. State-Space Models (SSMs)

The general nonlinear state-space models (SSMs) can be represented by the following two main processes:

$$\begin{aligned} \mathbf{x}_k &= \mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{n}_k, \boldsymbol{\theta}) \quad \text{(State Process)}, \\ \mathbf{y}_k &= \mathbf{g}_k(\mathbf{x}_k, \mathbf{v}_k, \boldsymbol{\theta}) \quad \text{(Measurement Process)}, \end{aligned} \tag{1}$$

for $k = 1, 2, \ldots$ where $\mathbf{x}_k \in \mathbb{X}$ is the state variable, $\mathbf{u}_k \in \mathbb{U}$ is the input to the system, and $\mathbf{y}_k \in \mathbb{Y}$ is the output of the system. The nonlinear functions $\mathbf{f}_k(.)$ and $\mathbf{g}_k(.)$ model the state and measurement processes, with a set of parameters denoted by $\boldsymbol{\theta} \in \Theta$, where Θ denotes the parameter space. Finally, $\{\mathbf{n}_k, \mathbf{v}_k; k = 1, 2, \ldots\}$ are mutually independent i.i.d. noise processes, which are also independent of $\mathbf{x}_0$. The parameter vector $\boldsymbol{\theta}$ models the uncertainty in both state and measurement processes. Equivalently, $\mathbf{x}_k \sim p_{\boldsymbol{\theta}}(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{u}_{k-1})$ and $\mathbf{y}_k \sim p_{\boldsymbol{\theta}}(\mathbf{y}_k \mid \mathbf{x}_k)$, where $p_{\boldsymbol{\theta}}(.)$ is a probability density or probability mass function. Without loss of generality and for the sake of simplicity, we will drop the input $\mathbf{u}_{k-1}$ in what follows.

B. Problem Formulation: Inference in SSMs

A key step in modeling systems or processes with SSMs is to estimate the set of unknown parameters of these models according to the available data. Letting $\mathbf{y}_{1:T} = (\mathbf{y}_1, ..., \mathbf{y}_T)$ be the sequence of observed measurements, one needs to find the best estimate of parameter vector $\boldsymbol{\theta}$. Two common estimators for inference in SSMs are the maximum-likelihood (ML) and the maximum a-posteriori (MAP), that can be formulated as:

$$\begin{aligned} \hat{\boldsymbol{\theta}}^{\mathrm{ML}} &= \arg\max_{\boldsymbol{\theta}\in\Theta} \log p_{\boldsymbol{\theta}}(\mathbf{y}_{1:T}), \\ \hat{\boldsymbol{\theta}}^{\mathrm{MAP}} &= \arg\max_{\boldsymbol{\theta}\in\Theta} p(\boldsymbol{\theta} \mid \mathbf{y}_{1:T}) \\ &= \arg\max_{\boldsymbol{\theta}\in\Theta} \left[\log p(\boldsymbol{\theta}) + \log p_{\boldsymbol{\theta}}(\mathbf{y}_{1:T})\right], \end{aligned} \tag{2}$$

where $p(\boldsymbol{\theta})$ denotes the prior distribution of the parameters, and $\log p_{\boldsymbol{\theta}}(\mathbf{y}_{1:T})$ is the data log-likelihood function. It can be seen that the data log-likelihood function appears in both ML and MAP estimators. This term can be further expanded as:

$$\begin{aligned} L_T(\boldsymbol{\theta}) &= \log p_{\boldsymbol{\theta}}(\mathbf{y}_{1:T}) \\ &= \log p_{\boldsymbol{\theta}}(\mathbf{y}_1) + \sum_{k=2}^{T} \log p_{\boldsymbol{\theta}}(\mathbf{y}_k \mid \mathbf{y}_{1:k-1}), \end{aligned} \tag{3}$$

where

$$p_{\boldsymbol{\theta}}(\mathbf{y}_k \mid \mathbf{y}_{1:k-1}) = \int p_{\boldsymbol{\theta}}(\mathbf{y}_k \mid \mathbf{x}_k)\, p_{\boldsymbol{\theta}}(\mathbf{x}_k \mid \mathbf{y}_{1:k-1})\, d\mathbf{x}_k, \tag{4}$$

$$p_{\boldsymbol{\theta}}(\mathbf{x}_k \mid \mathbf{y}_{1:k-1}) = \int p_{\boldsymbol{\theta}}(\mathbf{x}_k \mid \mathbf{x}_{k-1})\, p_{\boldsymbol{\theta}}(\mathbf{x}_{k-1} \mid \mathbf{y}_{1:k-1})\, d\mathbf{x}_{k-1}. \tag{5}$$

The integrals in (4) and (5) need to be replaced by summations in the case of a discrete state space.

For a general nonlinear state-space model, the exact computation of (4) and (5) is not tractable and techniques such as sequential Monte-Carlo (SMC), also known as particle filtering, are often used for their approximation. SMC methods comprise a general class of techniques for inference of nonlinear state-space models [6, 23]. The idea of these techniques is to approximate the target distribution using a finite set of samples drawn from a proposal distribution, using the fact that sampling from the proposal distribution is easier than from the target.

The basic algorithm to perform particle filtering is called sequential importance resampling (SIR). The auxiliary particle filter (APF) [24] is a successful variation of SIR technique which can efficiently predict the location of particles with high probability at time step k using information up to time step $k-1$ via an auxiliary variable ζ_k. The method first draws a sample of points (particles) from the joint distribution $p_{\boldsymbol{\theta}}(\mathbf{x}_k, \zeta_k \mid \mathbf{y}_{1:k})$, then drops the auxiliary variable to obtain particles from $p_{\boldsymbol{\theta}}(\mathbf{x}_k \mid \mathbf{y}_{1:k})$.

Let $\{\mathbf{x}_{k-1,i}, w_{k-1,i}\}_{i=1}^{N}$ be N particles and their associated weights at time $k-1$ approximating $p_{\boldsymbol{\theta}}(\mathbf{x}_{k-1} \mid \mathbf{y}_{1:k-1})$. New weights and particles can be obtained upon observing the new measurement $\mathbf{y}_k$ as follows:

1) The first stage weights can be computed as:

$$v_{k,i} = p_{\boldsymbol{\theta}}(\mathbf{y}_k \mid \mu_{k,i}) w_{k-1,i}, \tag{6}$$

for $i = 1, \ldots, N$; where $\mu_{k,i}$ is a characteristic of $\mathbf{x}_k$ given $\mathbf{x}_{k-1,i}$, which can be the mean, the mode or even a sample from $p_{\boldsymbol{\theta}}(\mathbf{x}_k \mid \mathbf{x}_{k-1,i})$ [24]. The auxiliary variables $\{\zeta_{k,i}\}_{i=1}^{N}$ are obtained by sampling from a discrete distribution:

$$\{\zeta_{k,i}\}_{i=1}^{N} \sim \mathrm{Cat}(\{\tilde{v}_{k,i}\}_{i=1}^{N}), \tag{7}$$

where $\{\tilde{v}_{k,i}\}_{i=1}^{N}$ are the normalized first-stage weights, and $\mathrm{Cat}(a_1, ..., a_N)$ represents a categorical distribution with probability mass function $f(\zeta = i) = a_i$.

2) The new particles $\{\mathbf{x}_{k,i}\}_{i=1}^{N}$ and the associated second-stage weights $\{w_{k,i}\}_{i=1}^{N}$ can be obtained as follows:

$$\mathbf{x}_{k,i} \sim p_{\boldsymbol{\theta}}(\mathbf{x}_k \mid \mathbf{x}_{k-1,\zeta_{k,i}}),\ w_{k,i} = \frac{p_{\boldsymbol{\theta}}(\mathbf{y}_k \mid \mathbf{x}_{k,i})}{p_{\boldsymbol{\theta}}(\mathbf{y}_k \mid \mu_{k,\zeta_{k,i}})}. \quad (8)$$

Continuing this process iteratively for all measurements, the log-likelihood function can be approximated as [25]:

$$L_T(\boldsymbol{\theta}) \approx \sum_{k=1}^{T} \log\left[\left(\frac{1}{N}\sum_{i=1}^{N} v_{k,i}\right)\left(\frac{1}{N}\sum_{i=1}^{N} w_{k,i}\right)\right]. \quad (9)$$

This quantity is used in Section III when the proposed framework is discussed. See [25], for comprehensive discussions regarding the unbiasedness of the log-likelihood approximation in (9).

C. Previous Work

Particle-Based Maximum-Likelihood (ML) Techniques: The existing particle-based ML techniques for inference of general nonlinear state-space models can be divided into three main categories:

Direct Gradient-Based ML Techniques: The idea here is to maximize the log-likelihood function using gradient-ascent or quasi-Newton techniques [6, 7]. These methods start by drawing an initial sample point from the parameter space, approximating the log-likelihood function and moving to another sample point based on the approximated gradient at the current sample. The computational complexity of successful techniques in this class per each sample is of order $O(N^2(T+1))$ [6, 7], where N is the number of particles and T is the length of the time series data. These techniques require extensive sampling of the parameter space to avoid local optimum traps.

Expectation-Maximization Techniques: Unlike direct ML techniques, which attempt to maximize the "incomplete" log-likelihood function $L_T(\boldsymbol{\theta}) = \log p_{\boldsymbol{\theta}}(\mathbf{y}_{1:T})$, expectation-maximization (EM) considers instead the "complete" log-likelihood function $\log p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}, \mathbf{y}_{1:T})$. The logic behind this is that maximizing the complete log-likelihood is easier than the incomplete one. The EM algorithm thus consists of picking an initial guess $\boldsymbol{\theta} = \boldsymbol{\theta}^{(0)}$ and iterating two steps:

1) **E-Step:** Compute $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(n)})$, where
$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(n)}) = \mathbb{E}_{\mathbf{x}_{0:T}}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}, \mathbf{y}_{1:T}) \mid \mathbf{y}_{1:T}, \boldsymbol{\theta}^{(n)}\right];$$
2) **M-Step:** Find $\boldsymbol{\theta}^{(n+1)} = \operatorname{argmax}_{\boldsymbol{\theta}\in\Theta} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(n)})$.

These steps need to be performed iteratively until a stopping criterion is met. The exact computation of the E-step is not possible for general nonlinear state-space models, and one needs to use particle methods for its approximation. Two popular particle smoothers are the backward simulation smoother [26] and the reweighing particle smoother [27], which have led to two different particle-based EM algorithms for general nonlinear state-space models introduced in [12] and [13] respectively. The computational complexity of both methods are of order $O(N^2(T+1))$. It should be noted that a closed-form solution for the M-step might not be achievable in general, posing another expensive computation. Similar to direct ML techniques, this class of estimators requires several iterations to avoid local optimum traps.

Particle-Based Bayesian Techniques: There are several particle-based Bayesian techniques for the inference of general nonlinear state-space models. An important representative is the particle marginal Metropolis-Hastings (PMMH) method [15]. Given that $\boldsymbol{\theta}$ is the current sample and $p_{\boldsymbol{\theta}}(\mathbf{y}_{1:T})$ is the likelihood associated with $\boldsymbol{\theta}$ approximated by a particle filter (e.g., APF), one needs to draw a new sample parameter $\boldsymbol{\theta}' \sim q(\boldsymbol{\theta}' \mid \boldsymbol{\theta})$ from the proposal distribution and run a particle filter to approximate the likelihood $p_{\boldsymbol{\theta}'}(\mathbf{y}_{1:T})$. Then, the new parameter $\boldsymbol{\theta}'$ gets accepted with probability $\min\{1, p_{\boldsymbol{\theta}'}(\mathbf{y}_{1:k})\, p(\boldsymbol{\theta}')\, q(\boldsymbol{\theta} \mid \boldsymbol{\theta}') / p_{\boldsymbol{\theta}}(\mathbf{y}_{1:k})\, p(\boldsymbol{\theta})\, q(\boldsymbol{\theta}' \mid \boldsymbol{\theta})\}$. This process continues for a large enough (usually prespecified) number of iterations in order to ensure a good inference performance.

All the aforementioned techniques require extensive sampling of the parameter space to approximate the complete/incomplete log-likelihood function. For large systems, which require a large number of particles, and for tall data sets, the computational cost of approximating the log-likelihood function per parameter sample point can be prohibitive.

Surrogate-Based Techniques: This class of techniques has been developed for fast inference in SSMs with an intractable likelihood functions [19–21]. The idea of these techniques is to use Gaussian process regression [28] for log-likelihood approximation and apply Bayesian optimization techniques for efficient exploration of the maximizer of the log-likelihood function using an SMC approximator. These techniques' computational complexity is of order $O(N(T+1))$ for each function evaluation. In addition, a multi-fidelity Bayesian optimization framework is introduced in [22**? ?**], which enables incorporation of various SMC approximators with different fidelities and computational costs during the inference process. Despite these techniques' relative success in reducing the sample inefficiency issue of the aforementioned techniques, their performance is highly impacted by the parameter space's size. In fact, it can be shown that the number of samples for achieving a proper inference process increases exponentially with the size of parameter space, rendering intractable the computation of this class of techniques in large-scale practical problems.

III. Proposed Framework

In this paper, we introduce an adaptive two-stage Bayesian optimization framework for scalable and efficient inference in SSMs. According to (2), the inference process in SSMs consists of solving the optimization in the following form:

$$\boldsymbol{\theta}^* = \operatorname*{argmax}_{\boldsymbol{\theta}\in\Theta} f(\boldsymbol{\theta}), \quad (10)$$

where we refer to $f(.)$ as "inference function". Two common inference functions are the log-likelihood and the log-aposteriori, denoted by $f(\boldsymbol{\theta}) := \log p_{\boldsymbol{\theta}}(\mathbf{y}_{1:T})$ and $f(\boldsymbol{\theta}) := \log p(\boldsymbol{\theta} \mid \mathbf{y}_{1:T})$ respectively (see (2)).

A. Modeling the Inference Function by Gaussian Process Regression

For general SSMs, the inference function can only be *approximated* by a particle filter with N particles at any given sample. The approximation made in the inference function evaluation, indicated by $\hat{f}(\boldsymbol{\theta})$ for any $\boldsymbol{\theta} \in \Theta$, and correlation over the parameter space are accounted by employing the Gaussian process (GP) regression [28] as:

$$\hat{f}(\boldsymbol{\theta}) \approx \mathcal{F}(\boldsymbol{\theta}) + \Delta\hat{f}_N, \tag{11}$$

where $\mathcal{F}(\boldsymbol{\theta})$ indicates the GP over the parameter space Θ, and $\Delta\hat{f}_N$ is a zero-mean Gaussian residual with variance σ_N^2 which models, for all parameters, the uncertainty arising from the use of a particle filter with N particles.

The following prior distribution is assumed for the GP:

$$\mathcal{F}(\boldsymbol{\theta}) = \mathcal{G}P\left(\mu(\boldsymbol{\theta}), k(\boldsymbol{\theta}, \boldsymbol{\theta})\right), \tag{12}$$

where $\mu(\boldsymbol{\theta})$ and $k(.,.)$ are the mean function and a real-valued kernel function which encodes our prior belief on the correlation in the parameter space. A common kernel choice for a continuous parameter space is the well-known exponential kernel function [28].

Let $\boldsymbol{\Theta}_t = (\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(t)})$ be a sample from the parameter space, with the approximated inference functions $\mathbf{f}_t = [\hat{f}(\boldsymbol{\theta}^{(1)}), \ldots, \hat{f}(\boldsymbol{\theta}^{(t)})]^T$. The posterior distribution of $\mathcal{F}(\boldsymbol{\theta})$ in equation (12) can be obtained as [28]:

$$\mathcal{F}(\boldsymbol{\theta}) \mid \boldsymbol{\Theta}_t, \mathbf{f}_t \sim \mathcal{N}\left(\bar{\mathcal{F}}_t(\boldsymbol{\theta}), \mathrm{cov}_t(\boldsymbol{\theta}, \boldsymbol{\theta})\right), \tag{13}$$

where

$$\begin{aligned} \bar{\mathcal{F}}_t(\boldsymbol{\theta}) &= \mu(\boldsymbol{\theta}) + \mathbf{K}_{\boldsymbol{\theta},\boldsymbol{\Theta}_t}\left(\mathbf{K}_{\boldsymbol{\Theta}_t,\boldsymbol{\Theta}_t} + \boldsymbol{\Sigma}_t\right)^{-1}(\mathbf{f}_t - \mu(\boldsymbol{\Theta}_t)), \\ \mathrm{cov}_t(\boldsymbol{\theta}, \boldsymbol{\theta}) &= k(\boldsymbol{\theta}, \boldsymbol{\theta}) - \mathbf{K}_{\boldsymbol{\theta},\boldsymbol{\Theta}_t}\left(\mathbf{K}_{\boldsymbol{\Theta}_t,\boldsymbol{\Theta}_t} + \boldsymbol{\Sigma}_t\right)^{-1}\mathbf{K}_{\boldsymbol{\theta},\boldsymbol{\Theta}_t}^T, \end{aligned} \tag{14}$$

$\boldsymbol{\Sigma}_t$ is a diagonal matrix of size t with diagonal elements σ_N^2, and

$$\mathbf{K}_{\boldsymbol{\Theta},\boldsymbol{\Theta}'} = \begin{bmatrix} k(\boldsymbol{\theta}_1, \boldsymbol{\theta}'_1) & \ldots & k(\boldsymbol{\theta}_1, \boldsymbol{\theta}'_r) \\ \vdots & \ddots & \vdots \\ k(\boldsymbol{\theta}_l, \boldsymbol{\theta}'_1) & \ldots & k(\boldsymbol{\theta}_l, \boldsymbol{\theta}'_r) \end{bmatrix}, \tag{15}$$

for $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_l\}, \boldsymbol{\Theta}' = \{\boldsymbol{\theta}'_1, ..., \boldsymbol{\theta}'_r\}$.

Using the above formulation, the inference function before observing any data is modeled by a zero-mean Gaussian process with covariance $k(\boldsymbol{\theta}, \boldsymbol{\theta})$, while at iteration t, the inference function is predicted based on the sequence of queried samples $\boldsymbol{\Theta}_t$ and the approximate inference function values $\mathbf{f}_t$.

B. Linear Dimensionality Reduction

Let m be the size of parameter vector, i.e., $\boldsymbol{\theta} \in \Theta \subset \mathbb{R}^m$. This paper aims to find a small subset of linear combinations of the original parameter space, capturing the highest variability of the objective function and mapping the inference function to this reduced space for an informative and efficient search process. It is shown in [29] that this linear subset corresponds to eigenvectors associated with the largest eigenvalues of the covariance of the gradients of the main objective function.

Since the goal of this paper is inference in SSMs, one needs to compute the covariance of the gradients of the inference function as:

$$\mathbf{C} = \mathbb{E}\left[\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta})^\top\right], \tag{16}$$

where the expectation is taken over the prior distribution of the parameter. In maximum aposteriori (MAP) estimator, this prior information is known, but in the maximum likelihood (ML) estimator, the expectation in (16) should be taken over a uniform distribution over the parameter space. Likewise the inference function, the gradients of the inference function can only be approximated at each sample in the parameter space using one of the existing particle filtering techniques, such as direct particle-based gradient approximations [7–10] or particle-based Fishers identity approximations [10, 30–32]. Thus, assuming that the approximate values of gradients are available up to time step t, the covariance in (16) can be approximated as:

$$\mathbf{C} \approx \frac{1}{t}\sum_{i=1}^{t} \nabla_{\boldsymbol{\theta}} \hat{f}(\boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(i)}} \nabla_{\boldsymbol{\theta}} \hat{f}(\boldsymbol{\theta})^T|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(i)}}. \tag{17}$$

The expression in (17) approximates the covarience of the gradients of the inference function in (16). As the number of samples from the inference function becomes more, (17) becomes a better approximation of the covariance function in (16).

The eigenvalue decomposition of covariance of the gradients of the inference function leads to:

$$\mathbf{C} = \mathbf{W}\Lambda\mathbf{W}^\top, \tag{18}$$

where Λ is a diagonal matrix containing eigenvalues with the corresponding eigenvectors $\mathbf{W}$, which is a matrix of size $m \times m$. The eigenvectors need to be placed in a descending order of eigenvalues, followed by partitioning the eigenvectors with normalized eigenvalues greater and smaller than a small pre-specified threshold $0 < \epsilon < 1$ as:

$$[\mathbf{U}_{m\times n} \;\; \mathbf{V}_{m\times m-n}]_{m\times m}, \tag{19}$$

where $\mathbf{U}$ contains n columns of $\mathbf{W}$ (i.e., $n \leq m$) associated to the normalized eigenvalues greater than ϵ. Then, the original parameter space, Θ, can be mapped to the reduced space denoted by Υ, using the following mapping:

$$\boldsymbol{\upsilon} = \mathbf{U}^\top\boldsymbol{\theta}, \tag{20}$$

for any $\boldsymbol{\theta} \in \Theta$ and $\boldsymbol{\upsilon} \in \Upsilon$. The available information up to current step t can be transferred to the reduced space, called also *active subspace*, as:

$$(\boldsymbol{\Theta}_t, \mathbf{f}_t) \rightarrow (\boldsymbol{\Upsilon}_t = \mathbf{U}^T\boldsymbol{\Theta}_t, \mathbf{f}_t). \tag{21}$$

Now, one can construct a Gaussian process over the reduced parameter space as:

$$\mathcal{H}(\boldsymbol{\upsilon}) \mid \boldsymbol{\Upsilon}_t = \mathbf{U}^T\boldsymbol{\Theta}_t, \mathbf{f}_t \sim \mathcal{N}\left(\bar{\mathcal{H}}_t(\boldsymbol{\upsilon}), \mathrm{cov}_t(\boldsymbol{\upsilon}, \boldsymbol{\upsilon})\right), \tag{22}$$

where

$$\begin{aligned} \bar{\mathcal{H}}_t(\boldsymbol{\upsilon}) &= \mu(\boldsymbol{\upsilon}) + \mathbf{K}_{\boldsymbol{\upsilon},\boldsymbol{\Upsilon}_t}\left(\mathbf{K}_{\boldsymbol{\Upsilon}_t,\boldsymbol{\Upsilon}_t} + \boldsymbol{\Sigma}_t\right)^{-1}(\mathbf{f}_t - \mu(\boldsymbol{\Upsilon}_t)), \\ \mathrm{cov}_t(\boldsymbol{\upsilon}, \boldsymbol{\upsilon}) &= k(\boldsymbol{\upsilon}, \boldsymbol{\upsilon}) - \mathbf{K}_{\boldsymbol{\upsilon},\boldsymbol{\Upsilon}_t}\left(\mathbf{K}_{\boldsymbol{\Upsilon}_t,\boldsymbol{\Upsilon}_t} + \boldsymbol{\Sigma}_t\right)^{-1}\mathbf{K}_{\boldsymbol{\upsilon},\boldsymbol{\Upsilon}_t}^T, \end{aligned} \tag{23}$$

with

$$\mathbf{K}_{\boldsymbol{\Upsilon},\boldsymbol{\Upsilon}'} = \begin{bmatrix} k(\boldsymbol{\upsilon}_1, \boldsymbol{\upsilon}'_1) & \dots & k(\boldsymbol{\upsilon}_1, \boldsymbol{\upsilon}'_r) \\ \vdots & \ddots & \vdots \\ k(\boldsymbol{\upsilon}_l, \boldsymbol{\upsilon}'_1) & \dots & k(\boldsymbol{\upsilon}_l, \boldsymbol{\upsilon}'_r) \end{bmatrix}, \tag{24}$$

for $\boldsymbol{\Upsilon} = \{\boldsymbol{\upsilon}_1, ..., \boldsymbol{\upsilon}_l\}, \boldsymbol{\Upsilon}' = \{\boldsymbol{\upsilon}'_1, ..., \boldsymbol{\upsilon}'_r\}$.

A simple example of a reducible inference function is shown in Fig. 1. One can see that the original inference function defined over a two-dimensional parameter space (top-left plot) can be represented in a one-dimensional space (top-right plot) containing a linear combination of the original space. The surrogate models, after three queries over the original space (bottom-left) and the reduced space (bottom-right), are also shown in Fig. 1. One can see that the surrogate model in the reduced space provides a better representation of the inference function compared to the original space.

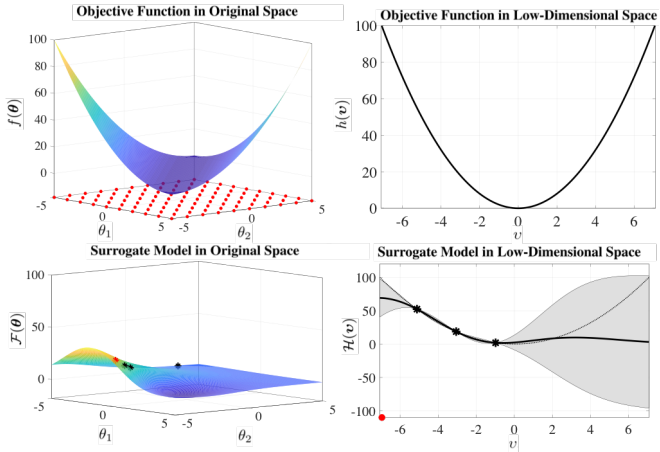

Fig. 1: A simple example in which the inference function can be represented in a lower dimension.

C. Two-Stage Bayesian Optimization Policy for Next Sample Selection

The computational complexity of the inference function approximation at any sample of parameters can be extremely expensive. For instance, the computational complexity of approximation of an inference function with a particle filter with N particles for a time series of length T is of order $O(N(T+1))$, where N should be chosen very large for large systems and big/tall datasets (see [6], for more information). In addition, as shown in Fig. 1, under limited evaluation scenarios, the surrogate model in the reduced space provides much better insight about an objective function in comparison to the surrogate model in the original parameter space. Thus, what needs to be achieved is to use this reduced space to make an informed decision about the selection in the original space for the next inference function approximation. Toward this, we introduce a two-stage Bayesian optimization policy, described below:

First-Stage Bayesian Optimization: The first-stage Bayesian optimization aims to take advantage of the better representation of the objective function over the reduced space by selecting the best sample for evaluation of the objective function in this low-dimensional parameter space. Using the GP surrogate model constructed over the reduced space, the best sample for making the balance between exploration and exploitation trade-off can be obtained by performing an arbitrary Bayesian optimization policy (e.g., see [33]). For a particular choice of knowledge gradient policy [34], the next selection is the sample in the reduced space with the highest single-period expected increase in the maximum of the inference function:

$$\boldsymbol{\upsilon}^{(t+1)} = \underset{\boldsymbol{\upsilon}\in\boldsymbol{\Upsilon}}{\operatorname{argmax}} \mathbb{E}_t \Big[\max_{\boldsymbol{\upsilon}'\in\boldsymbol{\Upsilon}} \mathbb{E}[\mathcal{H}(\boldsymbol{\upsilon}') \mid \boldsymbol{\Upsilon}_t, \mathbf{f}_t, \boldsymbol{\upsilon}^{(t+1)} = \boldsymbol{\upsilon}] - \max_{\boldsymbol{\upsilon}'\in\boldsymbol{\Upsilon}} \mathbb{E}[\mathcal{H}(\boldsymbol{\upsilon}') \mid \boldsymbol{\Upsilon}_t, \mathbf{f}_t] \Big], \tag{25}$$

where $\mathbb{E}_t$ denotes expectation over the unobserved inference function at point $\boldsymbol{\upsilon}^{(t+1)}$ given all available information up to iteration t, and the inner expectations are with respect to the posterior distribution of $\mathcal{H}(\boldsymbol{\upsilon}')$. An example of the selected sample is indicated by the red point in the x-axis of the left plot in Fig. 2.

Inverse Mapping: It is easy to show that the selected sample in the reduced space corresponds to a hyperplane in the original space. The reason is that the inverse map of the selected sample in the reduced space, $\boldsymbol{\upsilon}^{(t+1)}$, to the original space according to mapping $\mathbf{U}$ leads to infinite solutions, denoted by $\Theta^{\boldsymbol{\upsilon}^{(t+1)}}$. To better understand this, let $\boldsymbol{\upsilon}^{(t+1)}$ be the selected sample in the low dimensional space. This sample needs to be mapped back to the original space. This can be done through the following transformation:

$$\underbrace{\begin{bmatrix} \boldsymbol{\upsilon}_1^{(t+1)} \\ \vdots \\ \boldsymbol{\upsilon}_n^{(t+1)} \end{bmatrix}}_{\boldsymbol{\upsilon}^{(t+1)}} = \underbrace{\begin{bmatrix} \mathbf{u}_{11} & \mathbf{u}_{12} & \cdots & \mathbf{u}_{1n} & \cdots & \mathbf{u}_{1m} \\ & & \vdots & & & \\ \mathbf{u}_{n1} & \mathbf{u}_{n2} & \cdots & \mathbf{u}_{nn} & \cdots & \mathbf{u}_{nm} \end{bmatrix}}_{\mathbf{U}^T} \underbrace{\begin{bmatrix} \boldsymbol{\theta}_1^{(t+1)} \\ \vdots \\ \boldsymbol{\theta}_n^{(t+1)} \\ \vdots \\ \boldsymbol{\theta}_m^{(t+1)} \end{bmatrix}}_{\boldsymbol{\theta}^{\boldsymbol{\upsilon}^{(t+1)}}}, \tag{26}$$

where $\boldsymbol{\theta}^{\boldsymbol{\upsilon}^{(t+1)}} \in \Theta$ is the solution to the above linear equations. It is easy to verify that the solution to the inverse mapping in (26) is not unique and the transformation will thus lead to infinite set of samples in the original space, denoted by: $\Theta^{\boldsymbol{\upsilon}^{(t+1)}} = \{\boldsymbol{\theta} : \boldsymbol{\upsilon}^{(t+1)} = \mathbf{U}^T \boldsymbol{\theta}, \boldsymbol{\theta} \in \Theta\}$.

The inverse map consists of transferring from n dimensions to m dimensions ($n < m$). This can be done through discretizing $m-n$ dimensions of the original space and plugging in these discretized values in the equations and solving the n equations to find the remaining n values of $\boldsymbol{\theta}^{\boldsymbol{\upsilon}^{(t+1)}}$. For simplicity and without loss of generality, N_{inv} samples from the last $m-n$ dimensions are generated using techniques such as Latin Hypercube sampling [35]. Then, the other dimensions of these N_{inv} samples are obtained by solving (26). This leads to an unbiased set of alternatives from this hyperplane. An example of these samples represented over a line in the original space is shown in the right plot of Fig. 2.

Second-Stage Bayesian Optimization: The hyperplane obtained by the inverse mapping denotes the prescribed solution according to the surrogate model in the reduced space.

Therefore, as the surrogate model in the reduced space is often a better representation of the objective function, this hyperplane would be the best region in the original space during the search. Therefore, to achieve a proper balance between exploration and exploitation trade-off, one needs to select a sample in this hyperplane for the next inference function approximation. This requires performing another Bayesian optimization over this hyperplane. For a particular choice of knowledge gradient policy [34], this can be achieved as:

$$\boldsymbol{\theta}^{(t+1)} = \underset{\boldsymbol{\theta}\in\Theta^{\boldsymbol{\upsilon}_{t+1}}}{\operatorname{argmax}} \mathbb{E}_t \Bigg[\max_{\boldsymbol{\theta}'\in\Theta^{\boldsymbol{\upsilon}^{(t+1)}}} \mathbb{E}\Big[\mathcal{F}(\boldsymbol{\theta}') \mid \boldsymbol{\Theta}_t, \mathbf{f}_t, \boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}\Big] - \max_{\boldsymbol{\theta}'\in\Theta^{\boldsymbol{\upsilon}_{t+1}}} \mathbb{E}\Big[\mathcal{F}(\boldsymbol{\theta}') \mid \boldsymbol{\Theta}_t, \mathbf{f}_t\Big] \Bigg], \tag{27}$$

where $\mathbb{E}_t$ denotes expectation over the unobserved inference function at sample $\boldsymbol{\theta}^{(t+1)}$ approximated by a particle filter (approximator) with N particles, given all available information up to iteration t, and the inner expectations are with respect to the posterior distribution of $\mathcal{F}(\boldsymbol{\theta}')$. The intuition behind the proposed two-stage Bayesian optimization framework is very interesting. For a regular Bayesian optimization policy, the search space would be the whole original space, Θ, whereas the search space in the proposed framework is a hyperplane in the original parameter space, i.e., $\Theta^{\boldsymbol{\upsilon}^{(t+1)}}$, which is in an exponentially lower-dimensional space. An example of this reduction can be seen in Fig. 2, where instead of conducting the Bayesian optimization over the whole two-dimensional space in the right plot, the search space of the Bayesian optimization is narrowed to the red line in this two-dimensional space, which is the solution of the Bayesian optimization in the reduced space. The schematic diagram and the detailed procedure of the proposed framework are presented in Fig. 3 and Algorithm 1 respectively. Two possible stopping criteria for the proposed framework could be: 1) the changes in the maximum of the mean of the constructed GP over the original space fall below a pre-specified threshold in consecutive iterations; 2) the algorithm is performed for a fixed pre-specified amount of time or number of iterations.

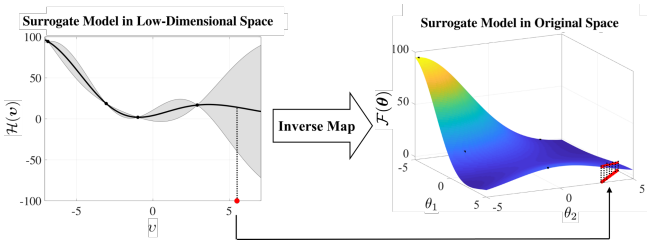

Fig. 2: Schematic diagram of inverse mapping in the proposed framework.

To better understand the selection in (25) and (27), let $\boldsymbol{\Theta}^a = \{\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_n\} \subset \Theta$ be a fixed set of alternative samples in which the maximization in (27) at the $(t+1)$th selection is planned to be taken over. Using (14), the mean and covariance of the alternative samples can be computed as:

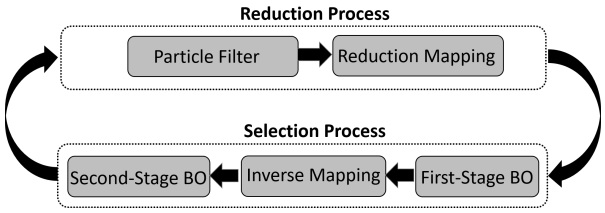

Fig. 3: Schematic diagram of the proposed framework.

$$\begin{aligned} \boldsymbol{\mu}_t^a &= \mu(\boldsymbol{\Theta}^a) + \mathbf{K}_{\boldsymbol{\Theta}^a,\boldsymbol{\Theta}_t}\left(\mathbf{K}_{\boldsymbol{\Theta}_t,\boldsymbol{\Theta}_t} + \boldsymbol{\Sigma}_t\right)^{-1}(\mathbf{f}_t - \mu(\boldsymbol{\Theta}_t)), \\ \boldsymbol{\Sigma}_t^a &= k(\boldsymbol{\Theta}^a,\boldsymbol{\Theta}^a) - \mathbf{K}_{\boldsymbol{\Theta}^a,\boldsymbol{\Theta}_t}\left(\mathbf{K}_{\boldsymbol{\Theta}_t,\boldsymbol{\Theta}_t} + \boldsymbol{\Sigma}_t\right)^{-1}\mathbf{K}^T_{\boldsymbol{\Theta}^a,\boldsymbol{\Theta}_t}. \end{aligned} \tag{28}$$

Using the mean and covariance of the alternative set, the policy in (27) can be expressed as:

$$\begin{aligned} \boldsymbol{\theta}^{(t+1)} &= \underset{\boldsymbol{\theta}_i\in\boldsymbol{\Theta}^a}{\operatorname{argmax}} \mathbb{E}_t \Big[\max_{j\in\{1,...,n\}} \boldsymbol{\mu}_{t+1}^a(j) - \max_{j\in\{1,...,n\}} \boldsymbol{\mu}_t^a(j) \mid \boldsymbol{\Theta}_t, \mathbf{f}_t, \boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_i \Big] \\ &= \underset{\boldsymbol{\theta}_i\in\boldsymbol{\Theta}^a}{\operatorname{argmax}} \mathbb{E}_t \Big[\max_{j\in\{1,...,n\}} \boldsymbol{\mu}_{t+1}^a(j) \mid \boldsymbol{\Theta}_t, \mathbf{f}_t, \boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_i \Big], \end{aligned} \tag{29}$$

where the expectation in (29) is with respect to the unknown inference function at $\boldsymbol{\theta}_{t+1}$, and the last line is derived due to the independency of $\boldsymbol{\mu}_t^a$ to the outcome of the inference function at $\boldsymbol{\theta}_{t+1}$. The knowledge gradient policy allows picking the next sample from the alternative set without the need for inference function evaluation. Given that $\boldsymbol{\theta}_i \in \boldsymbol{\Theta}^a$ is chosen for the next selection and the outcome is unobserved, it can be shown that $\boldsymbol{\mu}_{t+1}^a$ is a normal random variable, denoted by:

$$\boldsymbol{\mu}_{t+1}^a \sim \mathcal{N}\Big(\boldsymbol{\mu}_t^a, \tilde{\sigma}(\boldsymbol{\Sigma}_t^a, i)\tilde{\sigma}(\boldsymbol{\Sigma}_t^a, i)^T\Big), \tag{30}$$

where

$$\tilde{\sigma}(\boldsymbol{\Sigma}_t^a, i) = \frac{\boldsymbol{\Sigma}_t^a\,\mathbf{e}_i}{\sqrt{\sigma_N^2 + \left(\boldsymbol{\Sigma}_t^a\right)_{ii}}}, \tag{31}$$

and the term $\mathbf{e}_i$ is a column vector of size t with a single one at index i and the rest zeros, and $\left(\boldsymbol{\Sigma}_t^a\right)_{ii}$ refers to the element in the ith row and ith column of matrix $\boldsymbol{\Sigma}_t^a$.

Thus, replacing (30) into (29), the selection can be simplified as:

$$\boldsymbol{\theta}^{(t+1)} = \underset{\boldsymbol{\theta}_i\in\boldsymbol{\Theta}^a}{\operatorname{argmax}} \mathbb{E}_t \Big[\max_{j\in\{1,...,n\}} \left(\boldsymbol{\mu}_t^a + \tilde{\sigma}(\boldsymbol{\Sigma}_t^a, i)\, Z\right)_j \mid \boldsymbol{\Theta}_t, \mathbf{f}_t, \boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_i \Big]. \tag{32}$$

where $(.)_j$ indicates the jth element of the vector. The exact solution for the above optimization is provided in [34].

It should be noted that other Bayesian optimization acquisition functions, such as expected improvement [36, 37] and entropy search [38], can be used instead of the knowledge gradient policy in selection processes in (25) and (27). The

Algorithm 1 Two-Stage Bayesian Optimization for Scalable Inference in SSMs (Two-Stage BO)

Set the particle size N, and the reduction threshold ϵ, the stopping threshold δ.

Construct a GP over parameter space $\Theta \subset \mathbb{R}^m$.

$t = -1$, $\boldsymbol{\Theta}_0 = \{\}$, $\mathbf{f}_0 = \{\}$, $\mathbf{C} = \mathbf{I}_{m\times m}$.

Do

$t = t + 1$.

Put the eigenvectors of $\mathbf{C}$ corresponding to normalized eigenvalues greater than ϵ in $\mathbf{U}$ (column-wise).

Map the latest information to the reduced space: $(\boldsymbol{\Upsilon}_t = \mathbf{U}^T\boldsymbol{\Theta}_t, \mathbf{f}_t)$.

Construct a GP over the reduced space using (22) and (23).

Run BO over Υ to select $\boldsymbol{\upsilon}^{(t+1)}$ using (25).

Inverse map $\boldsymbol{\upsilon}^{(t+1)}$ to the original space to obtain $\Theta^{\boldsymbol{\upsilon}^{(t+1)}}$.

Perform BO over $\Theta^{\boldsymbol{\upsilon}^{(t+1)}}$ to select $\boldsymbol{\theta}^{(t+1)}$ using (27).

Run a particle filter tuned to $\boldsymbol{\theta}^{(t+1)}$ to get $\hat{f}(\boldsymbol{\theta}^{(t+1)})$ and $\nabla_{\boldsymbol{\theta}}\hat{f}(\boldsymbol{\theta})|_{\boldsymbol{\theta}^{(t+1)}}$.

$\boldsymbol{\theta}_{t+1} = \{\boldsymbol{\theta}_t, \boldsymbol{\theta}^{(t+1)}\}$, $\mathbf{f}_{t+1} = \{\mathbf{f}_t, \hat{f}(\boldsymbol{\theta}^{(t+1)})\}$.

$\mathbf{C} = \left[t\mathbf{C} + \nabla_{\boldsymbol{\theta}}\hat{f}(\boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(t+1)}}\nabla_{\boldsymbol{\theta}}\hat{f}(\boldsymbol{\theta})^T|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(t+1)}}\right]/(t+1)$.

Update the GP in the original space according to $(\boldsymbol{\Theta}_{t+1}, \mathbf{f}_{t+1})$.

Until $|\operatorname{argmax}_{\boldsymbol{\theta}\in\Theta}\bar{\mathcal{F}}_{t+1}(\boldsymbol{\theta}) - \operatorname{argmax}_{\boldsymbol{\theta}\in\Theta}\bar{\mathcal{F}}_t(\boldsymbol{\theta})| < \delta$

$\hat{\boldsymbol{\theta}}^*_{\mathrm{GP}} = \operatorname{argmax}_{\boldsymbol{\theta}\in\Theta}\bar{\mathcal{F}}_t(\boldsymbol{\theta})$, where $\bar{\mathcal{F}}_t(\boldsymbol{\theta})$ is the mean of the final GP in the original space.

reasons for using the knowledge gradient policy in this paper are its unique features in accounting for the uncertainty in the log-likelihood function approximation and efficient correlation consideration in the selection process.

The future work includes investigating the theoretical analysis of the proposed two-stage Bayesian optimization framework. The analysis requires the integration of the theoretical analysis of the dimensionality reduction process and the selection in lower and higher dimensional spaces. The theoretical analysis of the reduction process relies on the error bounds of the active subspace framework presented in [29]. The theoretical analysis of the selection process depends on the Bayesian optimization in use; for instance, for the knowledge gradient policy employed in this manuscript, the analysis is provided in [34].

D. Complexity

The complexity of the proposed framework is discussed in this section. Two main components of the proposed framework are the dimensionality reduction and the selection process. Assuming that T is the length of data and N is the number of particles, the complexity of the dimensionality reduction process at iteration t is of order $O(t^3)$, coming from the posterior update of the GP model. The complexity of the selection process is of order $O(\max\{N(T+1), t^3, |\boldsymbol{\Theta}^a|\})$, where $N(T+1)$ comes from the inference function approximation, t^3 is the complexity of computation of the posterior of GP model, and $|\boldsymbol{\Theta}^a|$ is the number of samples in the alternative set. Combining the complexity of these two processes, the complexity of the proposed framework is of order $O(\max\{|\boldsymbol{\Theta}^a|, t^3, N(T+1)\})$, which is approximately $O(N(T+1))$. The reason for dominancy of the term $N(T+1)$ over t^3 and $|\boldsymbol{\Theta}^a|$ is the large number of particles, N, required in large-scale systems for properly capturing the system dynamics and approximating the inference function. The number of iterations, t, (i.e., number of samples acquired from the parameter space), is often small due to the correlation consideration over the parameter space and intelligent selection, making t^3 the non-dominant term in complexity of the proposed framework. Finally, comparing the complexity of $O(N(T+1))$ in the proposed framework with the complexity of $O(N^2(T+1))$ in the particle-based EM and ML techniques [7, 12, 13], a significant reduction in computational complexity by the proposed framework can be understood in large systems. The aforementioned complexities correspond to a single iteration of various frameworks; however, the number of iterations in the proposed framework is also significantly smaller than existing surrogate-based and non-surrogate based techniques. The smaller number of iterations by the proposed framework comes from the selections that are made in lower dimensions, as opposed to the large original space.

IV. Experiments

All experiments have been conducted on a PC with an Intel Core i7-4790 CPU@3.60-GHz clock and 16 GB of RAM. To assess the performance of the proposed framework, the results are averaged over $1,000$ independent runs. The comparison has been made through the following algorithms:

1) The proposed Two-Stage Bayesian optimization method (Two-Stage BO);

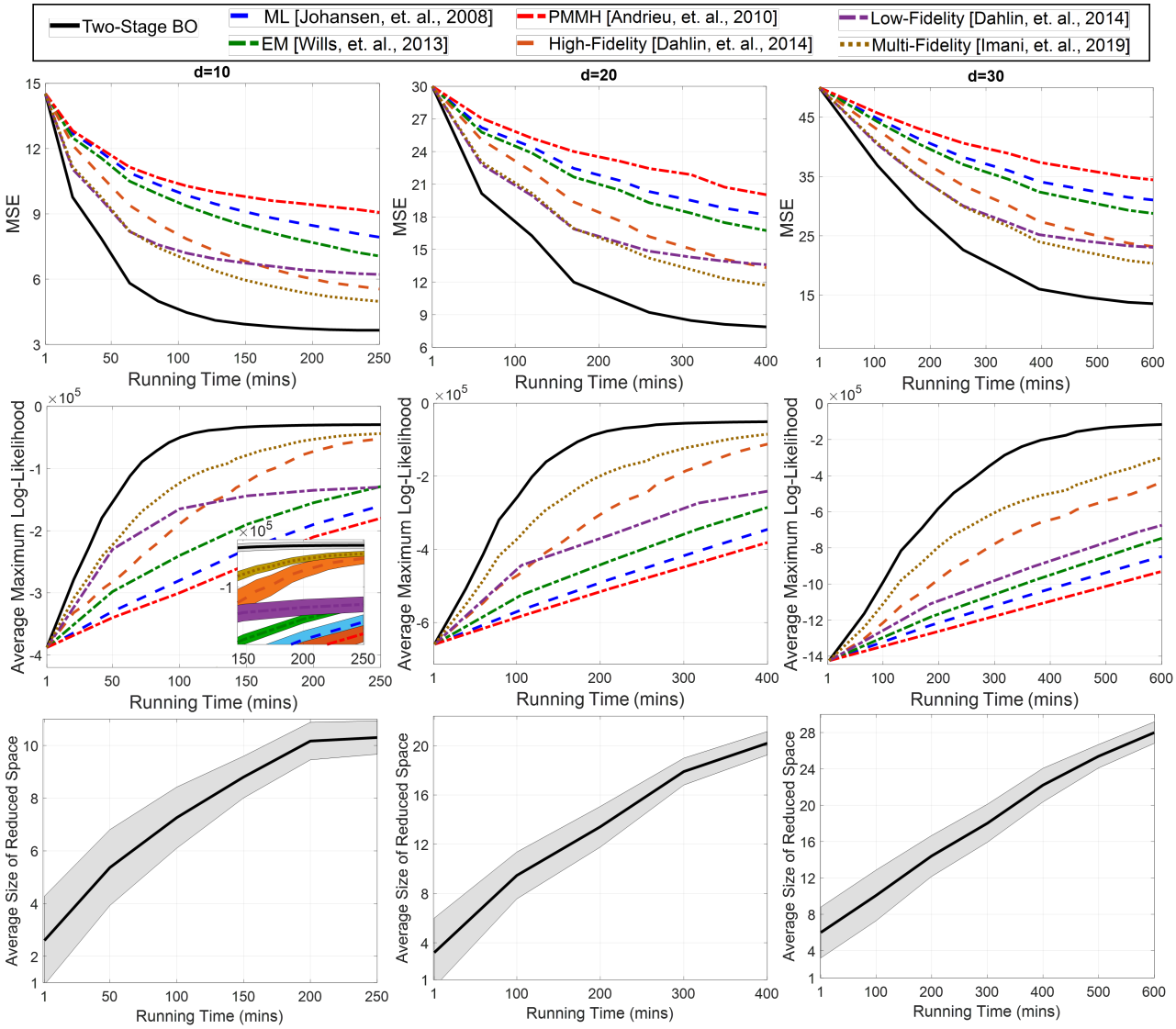

Fig. 4: Average results for Experiment 1.

2) Multi-Fidelity Bayesian optimization method [22] (Multi-Fidelity BO);
3) High-Fidelity Bayesian optimization method [20] (High-Fidelity BO);
4) Low-Fidelity Bayesian optimization method [20] (Low-Fidelity BO);
5) Particle marginal Metropolis-Hastings algorithm [15] (PMMH);
6) Particle-based ML algorithm [7] (ML);
7) Particle-based EM algorithm [13] (EM).

All methods stop when the changes in the estimated values of all parameters over a window of 20 consecutive iterations fall below 5% of their ranges, whereas the PMMH algorithm runs over a fixed number of $10,000$ iterations. The confidence intervals represent 68% deviations from the means, which is equivalent to one standard deviation from the means. For the PMMH, which is a distribution estimator, the mean of the estimated distribution is used for the comparison purpose. Finally, in all experiments, the "mean" is used as a characteristic index in auxiliary particle filter implementation.

Experiment 1 — Large-Scale Simulated State Space Model

We consider the following dynamical system:

$$\mathbf{x}_k(i) = \left| \sum_{j=1}^{d} \exp\left(\frac{\mathbf{x}_{k-1}^2(j)\mathbf{n}_k(i)}{l_{f,j}\mathbf{x}_{k-1}(j)+1} \right) + \mathbf{n}_k(i) \right|, \mathbf{n}_k(i) \sim \mathcal{N}(0, \sigma_{f,i}^2),$$
$$\mathbf{y}_k(i) \sim \mathcal{N}\left(\mathbf{x}_k(i), \sigma_{g,i}^2\right), \tag{33}$$

for $i = 1, ..., d$, where $|.|$ denotes the absolute value. The prior distribution of state is assumed to be $\mathbf{x}_0(i) \sim$ Uniform$[0\ 1]$, $i = 1, ..., d$.

The inference in this process consists of estimating the parameters of the state and measurement processes. These parameters can be encoded in a single parameter vector as:

$$\boldsymbol{\theta} = [\sigma_{f,1}, ..., \sigma_{f,d}, l_{f,1}, ..., l_{f,d}, \sigma_{g,1}, ..., \sigma_{g,d}]^T. \tag{34}$$

We assume the following uniform distributions for the prior and true distributions of parameters: $\sigma_{f,i} \sim$ Uniform$[1\ 5]$,

$l_{f,i} \sim$ Uniform$[0.1\ 1]$, $\sigma_{g,i} \sim$ Uniform$[0.2\ 2]$, $i = 1, ..., d$. Accordingly, we assume the following parameter space: $\Theta = [1\ 5]^d \times [0.1\ 1]^d \times [0.2\ 2]^d$. All numerical experiments are based on the fixed set of values for the system parameters displayed in Table I. The high-fidelity particle size $N = 10,000 \times d$ is used for all methods expect for low-fidelity BO and multi-fidelity BO algorithms.

TABLE I: Parameter values for experiment 1.

Parameter	**Value**
Number of state variables d	10, 20, 30
Number of parameters $\|\theta\|$	$3 \times d$
Low-Fidelity particle number N	1,000$\times d$
High-Fidelity particle number N	10,000$\times d$
PMMH iterations	10,000
Dimensionality reduction rate ϵ	0.1

Fig. 4 represents the following three sets of results, for 1,000 time series of length $T = 1{,}000$, with respect to running time: 1) the average maximum of the log-likelihood values; 2) the mean square errors (MSE) of the estimated parameters; 3) the average size of reduced space in the proposed Two-Stage BO method. Notice that the running time is equivalent to the computational cost of performing particle filters by each algorithm during the inference process.

On the top plots of Fig. 4, one can observe that the proposed framework achieves much lower MSE with respect to the running time in comparison to others. The same trend can be seen in the middle plots regarding the average maximum log-likelihood function, as the proposed two-stage BO method achieved much faster maximization of the log-likelihood function relative to other methods. The second best results correspond to the Multi-Fidelity BO method. This method, which is capable of intelligent switch between particle filters with different particle sizes, has outperformed both Low-Fidelity and High-Fidelity BO methods. The ML, EM, and PMMH are all less efficient than the methods built on Bayesian optimization. The reason is the reliance of these techniques on excessive search over the parameter space in the system defined in (33), in which approximation of the log-likelihood function is computationally expensive.

The bottom plots represent the average size of the reduced spaces for the proposed Two-Stage BO algorithm. The original parameter sizes from left to right are $30, 60$, and 90, respectively. One can see that the proposed framework has reduced the size of the parameter spaces significantly at early iterations. The reason for this significant reduction is the approximation made in the computation of the covariance function in (16) due to the few numbers of inference function evaluations. However, at the end of iterations, the reduced spaces' size is on average less than one-third of the original number of parameters (i.e., $|\boldsymbol{\theta}|/3$).

The results presented in different columns of Fig. 4 explore the effect of system size (i.e., d) on the performance of different methods. One can see that the larger the size of the system is, the more difficult the inference process becomes (i.e., larger MSE and longer running time). This is due to the fact that the log-likelihood function evaluation becomes more expensive for large systems and an efficient search over a large parameter space becomes extremely critical. However, the superiority of the proposed framework compared to other techniques is clear in all system sizes.

Finally, the confidence intervals in the log-likelihood values are represented in the left middle plot of Fig. 4. The robustness of the results obtained by the proposed Two-Stage BO method can be seen in relatively small confidence intervals compared to other methods. The largest confidence interval is associated with the ML method due to the ML method's sensitivity to initialization and local optimum traps.

The effect of particle size on the performance of the inference process by the proposed method is presented in Fig. 5. 1000 independent datasets simulated from the system with $d = 10$ are used in this part of the experiment. The running time is set to 100 minutes for all cases. One can see that the MSE is large for small and very large particle sizes. The reason is that the inference process becomes less accurate for very small particle sizes due to the approximation made in the log-likelihood evaluation, whereas for very large particle sizes, the huge computation associated with log-likelihood approximation prevents a proper search over large parameter spaces. Selecting the best particle size depends on the size of the system and the length of datasets and has been studied extensively in the past. For more information, see [39, 40].

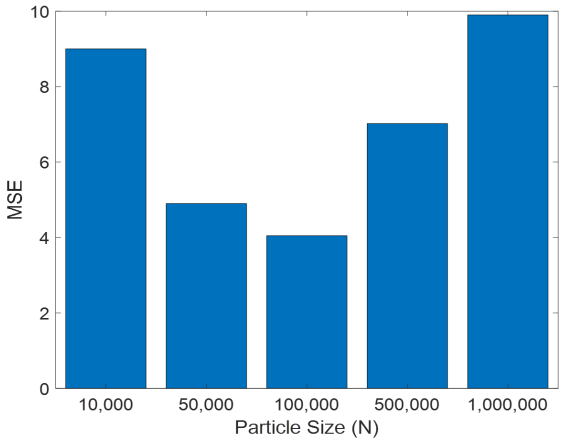

Fig. 5: Effect of particle size on performance of the proposed framework.

Experiment 2 — Gut Microbial Community

In the second experiment, the results of the proposed method is investigated through real datasets from the gut microbial community. The collection of 16S rRNA sequences from the experiments in [41] are used for our analysis. The regulatory relationship representing the relationship among microbes in the gut microbial community is presented in Fig. 6. The normal arrows represent activating regulations and dash arrows represent suppressive regulations. The network contains information of 10 microbes, and their values are represented in a single state vector as: $\mathbf{x}_k = [\text{Lachnospiraceae}_k$, Other_k, $\text{Lachnospiraceae\_Other}_k$, Barnesiela_k, C. Difficile_k,

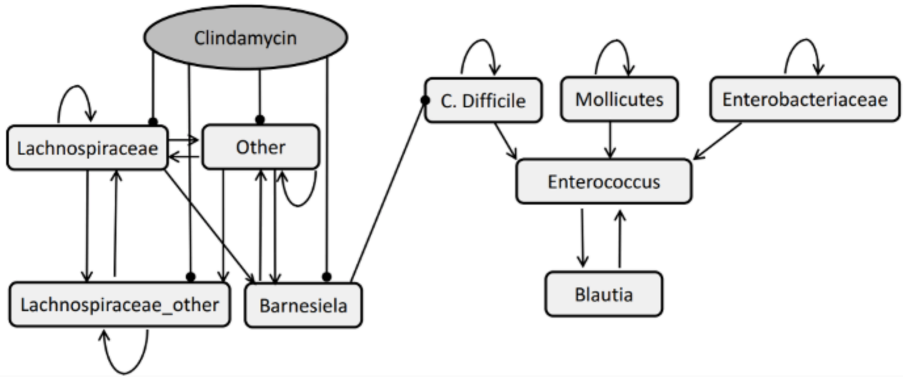

Fig. 6: Dynamical pathway for gut microbial communities.

Mollicutes_k, Entero_k, $\text{Enterobacteriaceae}_k$, Enterococcus_k, $\text{Blautia}_k]^T$. The activation/inactivation of these microbes can be modeled through the following Boolean state process:

$$\mathbf{x}_k = \overline{A\,\mathbf{x}_{k-1}} \oplus \mathbf{n}_k\,, \tag{35}$$

where $\overline{\mathbf{v}}$ maps the positive elements of vector $\mathbf{v}$ to 1 and others to 0, $\oplus$ is the module-2 addition, $\mathbf{n}_k$ is the process noise, and $A = [a_{ij}]$ is the connectivity matrix. a_{ij} specifies the type of regulation from component j to component i: it is equal to $+1$ for the activating regulation, -1 for the inactivating regulation and 0 for no regulation. This connectivity matrix for the gut microbial community shown in Fig. 6 is as:

$$A = \begin{bmatrix} +1 & +1 & +1 & 0 & 0 & 0 & 0 & 0 & 0 \\ +1 & +1 & 0 & +1 & 0 & 0 & 0 & 0 & 0 \\ +1 & +1 & +1 & 0 & 0 & 0 & 0 & 0 & 0 \\ +1 & +1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & +1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & +1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & +1 & 0 & 0 \\ 0 & 0 & 0 & 0 & +1 & +1 & +1 & 0 & +1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & 0 \end{bmatrix}.$$

The process noise $\mathbf{n}_k$ is assumed to have independent components distributed as Bernoulli(p), where the noise parameter p specifies the amount of "perturbation" in the Boolean state process.

Since the available data are the 16S rRNA sequence, we consider the following Gaussian linear observation model [42–44]:

$$\mathbf{y}_k = \boldsymbol{\mu} + \text{diag}(\boldsymbol{\delta})\mathbf{x}_k + \mathbf{v}_k\,,\ \ k = 1, 2, \ldots \tag{36}$$

where $\mathbf{v}_k \sim \mathcal{N}(0, \sigma^2 I)$ is an uncorrelated zero-mean Gaussian noise vector, $\boldsymbol{\mu}$ is a vector of baseline expressions (corresponding to the "zero" state for each microbe), "diag" puts elements of its input vector into diagonal elements of a matrix, and $\boldsymbol{\delta}$ is a vector containing differential expression values for each microbe along the diagonal elements (these indicate by how much the activated state of each microbe is over-expressed over the inactivated state).

Here, we assume that $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_{10}]^T$ and $\boldsymbol{\delta} = [\delta_1, \ldots, \delta_{10}]^T$. Since the value of the Bernoulli state noise process, i.e., p, is also unknown, the goal is to infer the following unknown parameter vector:

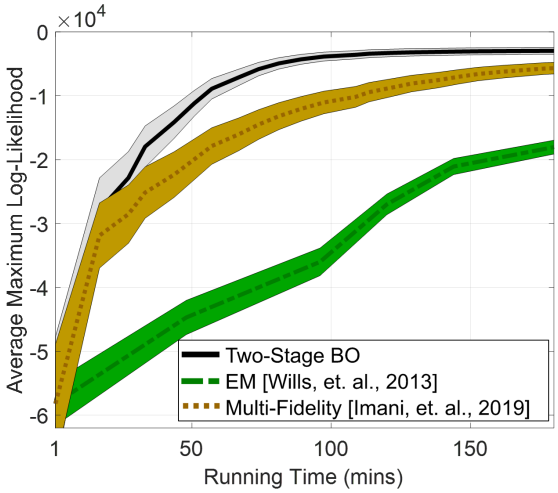

Fig. 7: Average results for Experiment 2.

$\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\delta}, \sigma, p)$. The parameter space is assumed to be $\Theta_i = [5, 40]^{10} \times [5, 40]^{10} \times [5, 20] \times [0.01, 0.3]$. The prior distribution is uniform over Θ. For Low-Fidelity BO method and EM, the particle size is set to $N = 10{,}000$; whereas for Multi-Fidelity BO method, the combination of $N = 1{,}000$, $N = 5{,}000$ and $N = 10{,}000$ is employed.

The average maximum of the log-likelihood value overrunning time is displayed in Fig. 7. One can observe that the log-likelihood value is maximized faster by the proposed framework than the other methods. It can be seen that the standard deviation of results obtained by the proposed framework is smaller than other methods, which indicates the robustness of the inference process by the proposed framework.

Experiment 3 — Cell-Cycle Gene Regulatory Network: In the third experiment, we consider the cell cycle gene regulatory network model in [45]. This is a Boolean network consisting of 14 genes, where each gene can be activated or inactivated. Hence, there are $2^{14} = 16384$ different possible system states. The pathway diagram for this gene regulatory network is displayed in Fig. 8. The state vector $\mathbf{x}_k$ contains the expression state of all 14 genes at time k and process noise $p = 0.01$ is used in our simulation.

The model presented in (35) and (36) is used for modeling

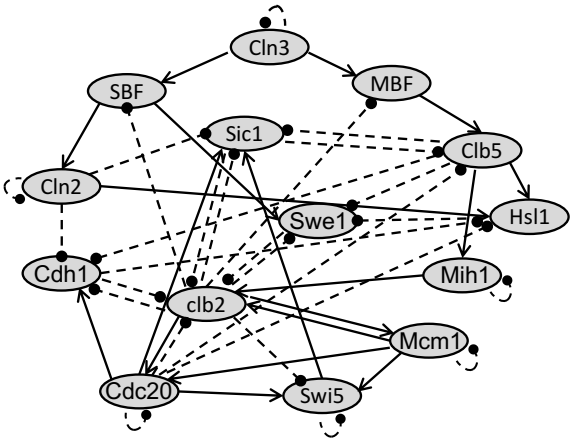

Fig. 8: Pathway diagram for the cell-cycle gene regulatory network model.

the cell-cycle gene regulatory network. Thus, the parameter vector is $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\delta}, \sigma)$. The true values of parameters are randomly generated from $\mu_i^* \sim [25, 35]$, $\delta_i^* \sim [15, 25]$ and $\sigma^* \sim [5, 10]$, for $i = 1, ..., 14$. The parameter space is assumed to be $\Theta_i = [5, 40]^{14} \times [5, 40]^{14} \times [5, 20]$. The prior distribution is uniform over Θ. Given 100 time series of length 100, the MSE of the estimated parameters and running time for different algorithms are displayed in Table II. The particle size is set to be $N = 10,000$.

TABLE II: MSE (running time in minutes) for the cell-cycle gene regulatory network.

Two-Stage BO	($\epsilon = 0.05$) 6.714 (42.59)
	($\epsilon = 0.10$) 7.928 (29.59)
High-Fidelity BO (Dahlin, et. al. 2014)	9.566 (91.15)
EM (Wills, et. al. 2013)	9.499 (168.01)
ML (Johansen, et. al. 2008)	11.031 (200.13)
PMMH (Andrieu, et. al. 2010)	11.311 (369.12)

In Table II, we can observe that the proposed Two-Stage BO method achieves more accurate results with smaller running time than other algorithms. Indeed, the proposed Two-Stage BO is 2 times faster than the fastest competitor. The main reason of this boost in the performance can be justified through the average sizes of the reduced space, which are 12.34 and 7.09 for $\epsilon = 0.05$ and $\epsilon = 0.10$ respectively.

V. Conclusion

This paper introduced a two-stage Bayesian optimization framework for scalable inference in general nonlinear state-space models. The proposed DRBO-SSM framework maps the original parameter space to a reduced space containing a small subset of linear combinations of the original parameter space. Then, the proposed framework sequentially selects the sample according to the proposed two-stage Bayesian optimization framework. First, a sample in the reduced space is selected by performing a Bayesian optimization policy. Then, the inverse map of the selected sample to the original parameter space, which is a hyperplane, is used as a search space for performing the second-stage Bayesian optimization policy. In numerical experiments, using real and synthetic data, the proposed algorithm performed significantly faster than the competing algorithms, at similar or better accuracy levels.

Acknowledgment

The authors acknowledge the support of the National Science Foundation through NSF award IIS-1946999.

References

[1] P. Kormushev, S. Calinon, and D. Caldwell, "Reinforcement learning in robotics: Applications and real-world challenges," *Robotics*, vol. 2, no. 3, pp. 122–148, 2013.

[2] M. L. Littman, "Reinforcement learning improves behaviour from evaluative feedback," *Nature*, vol. 521, no. 7553, p. 445, 2015.

[3] M. Peters, W. Ketter, M. Saar-Tsechansky, and J. Collins, "A reinforcement learning approach to autonomous decision-making in smart electricity markets," *Machine learning*, vol. 92, no. 1, pp. 5–39, 2013.

[4] Y. Deng, F. Bao, Y. Kong, Z. Ren, and Q. Dai, "Deep direct reinforcement learning for financial signal representation and trading," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 3, pp. 653–664, 2017.

[5] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

[6] N. Kantas, A. Doucet, S. S. Singh, J. Maciejowski, and N. Chopin, "On particle methods for parameter estimation in state-space models," *Statistical science*, vol. 30, no. 3, pp. 328–351, 2015.

[7] A. M. Johansen, A. Doucet, and M. Davy, "Particle methods for maximum likelihood estimation in latent variable models," *Statistics and Computing*, vol. 18, no. 1, pp. 47–57, 2008.

[8] D. N. DeJong, R. Liesenfeld, G. V. Moura, J.-F. Richard, and H. Dharmarajan, "Efficient likelihood evaluation of state-space representations," *Review of Economic Studies*, vol. 80, no. 2, pp. 538–567, 2012.

[9] S. Malik and M. K. Pitt, "Particle filters for continuous likelihood evaluation and maximisation," *Journal of Econometrics*, vol. 165, no. 2, pp. 190–209, 2011.

[10] E. L. Ionides, C. Bretó, and A. A. King, "Inference for nonlinear dynamical systems," *Proceedings of the National Academy of Sciences*, vol. 103, no. 49, pp. 18438–18443, 2006.

[11] R. G. Krishnan, U. Shalit, and D. Sontag, "Structured inference networks for nonlinear state space models," *arXiv preprint arXiv:1609.09869*, 2016.

[12] T. B. Schön, A. Wills, and B. Ninness, "System identification of nonlinear state-space models," *Automatica*, vol. 47, no. 1, pp. 39–49, 2011.

[13] A. Wills, T. B. Schön, L. Ljung, and B. Ninness, "Identification of hammerstein–wiener models," *Automatica*, vol. 49, no. 1, pp. 70–81, 2013.

[14] U. Picchini and A. Samson, "Coupling stochastic EM and approximate Bayesian computation for parameter inference in state-space models," *Computational Statistics*, vol. 33, no. 1, pp. 179–212, 2018.

[15] C. Andrieu, A. Doucet, and R. Holenstein, "Particle Markov chain Monte Carlo methods," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 72, no. 3, pp. 269–342, 2010.

[16] G. Mingas, L. Bottolo, and C.-S. Bouganis, "Particle MCMC algorithms and architectures for accelerating inference in state-space models," *International Journal of Approximate Reasoning*, vol. 83, pp. 413–433, 2017.

[17] A. Beskos, D. Crisan, A. Jasra, K. Kamatani, and Y. Zhou, "A stable particle filter for a class of high-dimensional state-space models," *Advances in Applied Probability*, vol. 49, no. 1, pp. 24–48, 2017.

[18] M. Hirt and P. Dellaportas, "Scalable Bayesian learning for state space models using variational inference with SMC samplers," in *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 76–86, PMLR, 2019.

[19] J. Dahlin and F. Lindsten, "Particle filter-based Gaussian process optimisation for parameter inference," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 8675–8680, 2014.

[20] J. Dahlin, M. Villani, and T. B. Schön, "Bayesian optimisation for fast approximate inference in state-space models with intractable likelihoods," *arXiv preprint arXiv:1506.06975*, 2015.

[21] M. U. Gutmann and J. Corander, "Bayesian optimization for likelihood-free inference of simulator-based statistical models," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 4256–4302, 2016.

[22] M. Imani, S. F. Ghoreishi, D. Allaire, and U. Braga-Neto, "MFBO-SSM: Multi-fidelity Bayesian optimization for fast inference in state-space models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 7858–7865, 2019.

[23] D. Arnaud, N. de Freitas, and N. Gordon, *Sequential Monte Carlo methods in practice*. New York: Springer Science+ Business Media, Inc. LLC, 2001.

[24] M. K. Pitt and N. Shephard, "Filtering via simulation: Auxiliary particle filters," *Journal of the American statistical association*, vol. 94, no. 446, pp. 590–599, 1999.

[25] M. K. Pitt, "Smooth particle filters for likelihood evaluation and maximisation," tech. rep., 2002.

[26] S. J. Godsill, A. Doucet, and M. West, "Monte Carlo smoothing for nonlinear time series," *Journal of the american statistical association*, vol. 99, no. 465, pp. 156–168, 2004.

[27] M. Hürzeler and H. R. Künsch, "Monte Carlo approximations for general state-space models," *Journal of Computational and Graphical Statistics*, vol. 7, no. 2, pp. 175–193, 1998.

[28] C. E. Rasmussen and C. Williams, *Gaussian processes for machine learning*. MIT Press, 2006.

[29] P. G. Constantine, E. Dow, and Q. Wang, "Active subspace methods in theory and practice: applications to kriging surfaces," *SIAM Journal on Scientific Computing*, vol. 36, no. 4, pp. A1500–A1524, 2014.

[30] J. Dahlin, F. Lindsten, and T. B. Schön, "Particle Metropolis–Hastings using gradient and Hessian information," *Statistics and computing*, vol. 25, no. 1, pp. 81–92, 2015.

[31] P.-A. Coquelin, R. Deguest, and R. Munos, "Sensitivity analysis in HMMs with application to likelihood maximization," in *Advances in Neural Information Processing Systems*, pp. 387–395, 2009.

[32] E. L. Ionides, A. Bhadra, *et al.*, "Iterated filtering," *The Annals of Statistics*, vol. 39, no. 3, pp. 1776–1802, 2011.

[33] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, "Taking the human out of the loop: A review of Bayesian optimization," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2015.

[34] P. Frazier, W. Powell, and S. Dayanik, "The knowledge-gradient policy for correlated normal beliefs," *INFORMS journal on Computing*, vol. 21, no. 4, pp. 599–613, 2009.

[35] M. D. McKay, R. J. Beckman, and W. J. Conover, "Comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 21, no. 2, pp. 239–245, 1979.

[36] J. Mockus, V. Tiesis, and A. Zilinskas, "The application of Bayesian methods for seeking the extremum," *Towards global optimization*, vol. 2, no. 117-129, p. 2, 1978.

[37] D. R. Jones, "A taxonomy of global optimization methods based on response surfaces," *Journal of global optimization*, vol. 21, no. 4, pp. 345–383, 2001.

[38] J. M. Hernández-Lobato, M. W. Hoffman, and Z. Ghahramani, "Predictive entropy search for efficient global optimization of black-box functions," in *Advances in neural information processing systems*, pp. 918–926, 2014.

[39] N. Whiteley *et al.*, "Stability properties of some particle filters," *The Annals of Applied Probability*, vol. 23, no. 6, pp. 2500–2537, 2013.

[40] J. Olsson and T. Rydén, "Asymptotic properties of particle filter-based maximum likelihood estimators for state space models," *Stochastic Processes and their Applications*, vol. 118, no. 4, pp. 649–680, 2008.

[41] S. N. Steinway, M. B. Biggs, T. P. Loughran Jr, J. A. Papin, and R. Albert, "Inference of network dynamics and metabolic interactions in the gut microbiome," *PLoS computational biology*, vol. 11, no. 6, p. e1004338, 2015.

[42] Y. Chen, E. R. Dougherty, and M. L. Bittner, "Ratio-based decisions and the quantitative analysis of cDNA microarray images," *Journal of Biomedical optics*, vol. 2, no. 4, pp. 364–374, 1997.

[43] M. Imani and U. Braga-Neto, "Particle filters for partially-observed Boolean dynamical systems," *Automatica*, vol. 87, pp. 238–250, 2018.

[44] M. Imani and U. Braga-Neto, "Control of gene regulatory networks using Bayesian inverse reinforcement learning," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 16, no. 4, pp. 1250–1261, 2019.

[45] E. Radmaneshfar and M. Thiel, "Recovery from stress–a cell cycle perspective," *Journal of computational interdisciplinary sciences*, vol. 3, no. 1-2, p. 33, 2012.

Mahdi Imani is an Assistant Professor in the Department of Electrical and Computer Engineering at George Washington University, Washington, DC, USA. He received his Ph.D. degree in Electrical and Computer Engineering from Texas A&M University, College Station, TX in 2019. His research interests include Machine Learning, Bayesian Statistics, and Decision Theory, with a wide range of applications from computational biology to cyber-physical systems. He is the recipient of the Association of Former Students Distinguished Graduate Student Award for Excellence in Research-Doctoral in 2019, the Best Ph.D. Student Award in the ECE department at Texas A&M University in 2015, and a single finalist nominee of the ECE department for the Outstanding Graduate Student Award in the college of engineering at Texas A&M University in 2018. He is also the recipient of the best paper finalist award from the 49th Asilomar Conference on Signals, Systems, and Computers, 2015.

Seyede Fatemeh Ghoreishi is a postdoctoral research fellow at the Institute for Systems Research (ISR) at the University of Maryland. She received her Ph.D. and M.Sc. degrees both in Mechanical Engineering from Texas A&M University in 2019 and 2016, respectively. She holds a minor in Applied Statistics from the Department of Statistics at Texas A&M University. She also received an M.Sc. degree in Biomedical Engineering from Iran University of Science and Technology in 2014 and a B.Sc. degree in Mechanical Engineering from the University of Tehran in 2012. She was selected as Rising Stars in Mechanical Engineering at UC Berkeley in 2020 and in Computational and Data Sciences at the Oden Institute for Computational Engineering and Sciences at the University of Texas at Austin in 2019.