Scalable Inverse Reinforcement Learning Through Multi-Fidelity Bayesian Optimization

Mahdi Imani, and Seyede Fatemeh Ghoreishi

***Abstract*—Data in many practical problems are acquired according to decisions or actions made by users or experts to achieve specific goals. For instance, policies in the mind of biologists during the intervention process in genomics and metagenomics are often reflected in available data in these domains, or data in cyber-physical systems are often acquired according to actions/decisions made by experts/engineers for purposes such as control or stabilization. Quantification of experts' policies through available data, which is also known as reward function learning, has been discussed extensively in literature in the context of inverse reinforcement learning (IRL). However, most of the available techniques come short to deal with practical problems due to the following main reasons: 1) *Lack of scalability:* arising from incapability or poor performance of existing techniques in dealing with large systems; 2) *Lack of reliability:* coming from the incapability of the existing techniques to properly learn the optimal reward function during the learning process. Toward this, in this paper, we propose a multi-fidelity Bayesian optimization (MFBO) framework that significantly scales the learning process of a wide range of existing inverse reinforcement learning techniques. The proposed framework enables the incorporation of multiple approximators and efficiently takes their uncertainty and computational costs into account to balance exploration and exploitation during the learning process. The proposed framework's high performance is demonstrated through genomics, metagenomics, and sets of random simulated problems.**

***Index Terms*—Reinforcement Learning, Bayesian Optimization, Multi-Fidelity Models.**

I. Introduction

Quantifying the policies in mind of experts or users through some realizations of their behavior is becoming more and more critical in contemporary applications. In fact, the complexity, scale, and uncertainty involved in real problems necessitate maximum extraction of information through available data. For instance, medical data are often acquired according to treatments/remedies prescribed by doctors for disease diagnosis or therapy.

Several techniques have been developed in the context of inverse reinforcement learning (IRL) for learning the policies through realizations of experts/demonstrators [1–5]. These techniques can be categorized as follows:

- **Maximum Margin IRL:** This class of techniques attempts to choose the reward function that separates the optimal policy and the second-best policy the most [1, 2]. Other variations of this class have been focused on learning from suboptimal demonstrations [6], learning for cases with nonlinear reward functions [7], and game-theoretic learning [8].
- **Bayesian IRL:** The Bayesian term in the title of this class reflects their ability to incorporate the prior knowledge during the learning process. The prior is often used to bias the search over reward functions [9–11]. Some variations of this class allow dealing with suboptimal demonstrations [12].
- **Gradient-based IRL:** This class of techniques attempts to optimize a loss function, which often penalizes deviations from the expert's policy [13], in the learning of reward function.
- **Maximum entropy IRL:** This class of techniques looks at the learning problem from the information-theoretic prospective [5, 14–17]. The main idea of these techniques is to find the most likely reward function given the demonstrations and attempt to produce a policy that matches the user's expected performance without making further assumptions on the preference over trajectories.

Notice that there are many other IRL techniques that might not belong to the above categorizations (e.g., IRL with specific (e.g., linear/quadratic) reward functions [18, 19], and modeling the user's reward function via classification [20]). For more information, see [4, 21].

Despite the development of several IRL techniques, the following two limitations often prevent the applications of the existing techniques to a wide range of practical problems:

1) **Lack of Scalability:** The existing IRL techniques often rely on excessive sampling of the parameter space during the learning process to avoid local optimum traps. This reliance on excessive sampling often results in intractability or poor performance of the existing techniques due to:
   1) Huge computational cost of a single evaluation: Evaluation/approximation of the desirability of any single parameter sample point requires performing a dynamic programming based technique. This is due to the fact that each parameter sample corresponds to a known reward function, in which dynamic programming based techniques can be employed to compute/approximate the optimal policy. However, the computational cost of dynamic programming for any given parameter sample increases exponentially with the size of systems, rendering the computation of the existing techniques intractable.
   2) Large Parameter Space: Complexity of the practical problems necessitates considering complex models for the reward function with a large number of parameters. It can

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

be shown that the amount of required search over the parameter space for a proper learning process increases exponentially with its size, resulting in intractability or poor performance of the existing techniques.

2) **Lack of Reliability:** Deterministic estimation/learning of parameters of the model assumed for the reward function is the basis for most of the existing IRL techniques. These techniques often come short to properly estimate the optimal or near-optimal parameters of the reward functions, leading to the learning process's unreliability.

This paper addresses the aforementioned challenges in the existing inverse reinforcement learning techniques by developing a multi-fidelity Bayesian optimization framework. For any arbitrary model of the reward function (e.g., parametric or non-parametric) with a relatively small number of parameters, the log-likelihood or log-posterior computed by IRL over the parameters of the reward function is represented by a Bayesian surrogate model. This surrogate model enables incorporating multiple reinforcement learning techniques with different episode or epoch numbers as multiple approximators for evaluation of the IRL objective function. Large episode numbers refer to more accurate (high-fidelity) approximators at a higher computational time/cost, whereas small episode numbers correspond to fast but less accurate (low-fidelity) approximators. The surrogate model optimally takes the approximators' uncertainty and computational cost into account and enables simultaneous sequential selection of parameters and approximators. More specifically, the proposed framework selects sequential samples so that the largest single-period expected increase in the maximum of the objective function per unit cost is achieved. The proposed framework's accuracy and speed are demonstrated empirically by applying it to genomics, metagenomics, and simulated problems.

II. Background

The reinforcement learning (RL) framework can be formulated as a Markov decision process (MDP). An MDP is formally defined by a 5-tuple $\langle \mathbb{S}, \mathbb{A}, T, R, \gamma \rangle$, where $\mathbb{S}$ is the *state space*, $\mathbb{A}$ is the *action space*, $T : \mathbb{S} \times \mathbb{A} \times \mathbb{S}$ is the *state transition probability function* such that $T(\mathbf{s}, \mathbf{a}, \mathbf{s}') = p(\mathbf{s}' \mid \mathbf{s}, \mathbf{a})$ represents the probability of moving to state s' after taking action $\mathbf{a}$ in state $\mathbf{s}$, $R : \mathbb{S} \times \mathbb{A} \to \mathbb{R}$ is a bounded *reward function* such that $R(\mathbf{s}, \mathbf{a})$ encodes the reward earned when action $\mathbf{a}$ is taken in state $\mathbf{s}$, and $0 < \gamma < 1$ is a *discount factor*. The schematic diagram of the RL is shown in Fig. 1. There is an agent that interacts with the system/environment: the agent starts at initial state $\mathbf{s}_0 \in \mathbb{S}$ in the environment. At each time step t, the agent takes an action $\mathbf{a}_t \in \mathbb{A}$ and observes an immediate reward $r_t \in \mathbb{R}$ and moves to the new state $\mathbf{s}_{t+1} \in \mathbb{S}$.

The RL aims to find the best action-strategy that maximizes the expected accumulated rewards. More formally, a deterministic stationary policy π for an MDP is a mapping $\pi : \mathbb{S} \to \mathbb{A}$ from states to actions. The expected discounted reward function at state $\mathbf{s} \in \mathbb{S}$ after taking action $\mathbf{a} \in \mathbb{A}$ and following policy π afterward is defined as:

$$Q^{\pi}(\mathbf{s}, \mathbf{a}) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(\mathbf{s}_t, \mathbf{a}_t) \mid \mathbf{s}_0 = \mathbf{s}, \mathbf{a}_0 = \mathbf{a}, \mathbf{a}_{1:\infty} \sim \pi\right]. \tag{1}$$

According to (1), the expected return under the optimal policy π^* can be defined as:

$$Q^{\pi^*}(\mathbf{s}, \mathbf{a}) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(\mathbf{s}_t, \mathbf{a}_t) \mid \mathbf{s}_0 = \mathbf{s}, \mathbf{a}_0 = \mathbf{a}, \mathbf{a}_{1:\infty} \sim \pi^*\right].$$

where $Q^{\pi^*}(\mathbf{s}, \mathbf{a})$ indicates the expected discounted reward after executing action $\mathbf{a}$ in state $\mathbf{s}$ and following optimal policy π^* afterward. An optimal stationary policy π^* attains the maximum expected return for all states as: $\pi^*(\mathbf{s}) = \max_{\mathbf{a} \in \mathbb{A}} Q^{\pi^*}(\mathbf{s}, \mathbf{a})$.

An MDP is said to be known if the 5-tuple $\langle \mathbb{S}, \mathbb{A}, T, R, \gamma \rangle$ is fully specified. For an MDP with known dynamics and finite state and action spaces, planning algorithms such as value iteration or policy iteration [22] can be used to compute the optimal policy offline. For large state and action spaces, approximate dynamic programming (ADP) [23], and reinforcement learning (RL) [24, 25] techniques are often used for approximation of an infinite horizon policy. The ADP techniques rely on simulated or existing batch data for their learning process. An inverse reinforcement learning deals with cases that everything is specified in the MDP except the reward function. As indicated in the term "inverse", some realizations of policy are available, and the goal is to learn reward function through available data. This has been explained in the next section.

III. The proposed Framework

A. Problem Formulation

Let D_T contain all available realizations of a demonstrator, denoted by:

$$D_T = \{(\tilde{\mathbf{s}}_1, \tilde{\mathbf{a}}_1), (\tilde{\mathbf{s}}_2, \tilde{\mathbf{a}}_2), ..., (\tilde{\mathbf{s}}_T, \tilde{\mathbf{a}}_T)\}, \tag{2}$$

where $\tilde{\mathbf{a}}_r$ is the taken action by demonstrator at the state $\tilde{\mathbf{s}}_r$ at time step r. Let also $\boldsymbol{\theta}$ be a vector in the space Θ which denotes the unknown part of reward function (i.e., $R_{\boldsymbol{\theta}}$). It should be noted that we do not make any restrictive assumption on the choice of the reward function, and depending on the problem, the reward function can be a simple parametric model such as linear or polynomial functions, or a non-parametric model such as Gaussian processes [26]. It should be mentioned that despite the appearance of the "non-parametric" term in non-parametric models, these models contain a set of parameters or hyper-parameters similar to parametric models. For instance, the non-parametric Gaussian process model can be expressed by mean and kernel functions, where both functions contain a set of hyper-parameters that need to be learned according to the available data.

IRL techniques attempt to estimate the parameters of the reward function using the observed demonstrations. This can also be interpreted as finding the best estimate for parameters of the reward function $R_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{a})$, whose corresponding policy

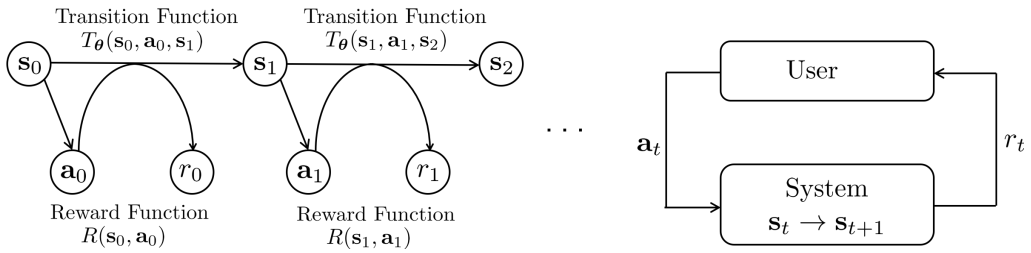


Fig. 1: Illustration of an MDP (left) and a schematic diagram of RL framework.

(i.e., $\pi^*_{\boldsymbol{\theta}}$) matches the demonstration D. Despite some differences in the existing IRL techniques [2, 5, 9, 13] (e.g., the space or size of the state and action spaces or the choice of model for the reward function), the majority of them aims to solve the optimization of the following form:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}\in\Theta}{\operatorname{argmax}}\, f(\boldsymbol{\theta}). \tag{3}$$

where $f(.)$ is called "objective function", which depending on the main goal, can be likelihood, posterior, entropy or any objective function. Two popular instances of them that have led to maximum likelihood (ML) [13] and maximum a posteriori (MAP) [27] IRL techniques are as follows:

$$\begin{aligned} f(\boldsymbol{\theta}) &= \log P(D_T \mid \boldsymbol{\theta}), \ \text{(ML)} \\ f(\boldsymbol{\theta}) &= \log P(\boldsymbol{\theta} \mid D_T) \\ &\propto \log p(\boldsymbol{\theta}) + \log P(D_T \mid \boldsymbol{\theta}), \ \text{(MAP)} \end{aligned} \tag{4}$$

where $p(\boldsymbol{\theta})$ is the prior probability and the second expression in MAP estimator is written according to the Bayes' rule. Evaluation of the objective function at any given sample $\boldsymbol{\theta}$ for both ML and MAP requires computation of the following log-likelihood function:

$$\begin{aligned} &\log P(D_T \mid \boldsymbol{\theta}) \\ &= \underbrace{\sum_{t=2}^{T} \log p(\tilde{\mathbf{s}}_t \mid \tilde{\mathbf{s}}_{t-1}, \tilde{\mathbf{a}}_{t-1}, \boldsymbol{\theta})}_{\text{State-Action Transition Term}} + \underbrace{\sum_{t=1}^{T} \log p(\tilde{\mathbf{a}}_t \mid \tilde{\mathbf{s}}_t, \boldsymbol{\theta})}_{\text{Demonstrator Policy Term}}. \end{aligned} \tag{5}$$

where $p(.)$ is a probability density or probability mass function. While computation of the first term in (5) can be done relatively fast for any arbitrary MDP, computation of the second term can be both computationally and analytically very challenging.

Let $\pi^*_{\boldsymbol{\theta}}$ be the optimal policy under reward function paramterized by $\boldsymbol{\theta}$ (i.e., $R_{\boldsymbol{\theta}}$). This policy can be computed exactly for small and finite state and action spaces by running a dynamic programming technique; whereas for large and arbitrary spaces, the policy $\pi^*_{\boldsymbol{\theta}}$ can be approximated by performing a reinforcement learning technique associated to the reward function parametrized by $\boldsymbol{\theta}$. Assuming that the "imperfect" demonstrator's decisions can be modeled by the well-known Boltzmann softmax policy [24], one can write:

$$p(\mathbf{a} \mid \mathbf{s}, \boldsymbol{\theta}) \ \propto \ \exp\left(-\eta\, Q^{\pi^*_{\boldsymbol{\theta}}}(\mathbf{s}, \mathbf{a})\right), \tag{6}$$

for $\mathbf{s} \in \mathbb{S}$ and $\mathbf{a} \in \mathbb{A}$; where $\eta > 0$ represents our confidence on the expert's decision. The smaller the value of η, the more imperfect the demonstrator is expected to be. Using (6) in (5), the log-likelihood function and as a result the objective function in (4) can be evaluated.

Since most of the realistic problems can be modeled by MDPs with large state and action spaces, one can understand that a good approximation of the objective function for any given sample point $\boldsymbol{\theta} \in \Theta$ is computationally expensive, as it requires approximating $\pi^*_{\boldsymbol{\theta}}$ by running an RL technique in very large spaces. In fact, it can be shown that the number of episodes/epochs required by any RL technique for proper learning of the expected returns increases exponentially with the size of the state and action spaces. This makes the computation of most of the conventional inverse reinforcement learning techniques intractable or very slow, as they rely on excessive evaluations of the objective function over the samples of the parameter space. In the next paragraph, we will introduce a multi-fidelity Bayesian optimization framework for scalable, efficient, and fast inverse reinforcement learning.

B. Modeling the Objective Function by Gaussian Process Regression

In this paper, we assume the reward function is an arbitrary parametric/non-parametric model with a relatively small number of parameters, encoded in a finite-dimensional vector $\boldsymbol{\theta}$. Let also N be the number of episodes or epochs used by a reinforcement learning technique to approximate the objective function at a given sample point $\boldsymbol{\theta} \in \Theta$, which denotes a realization of the reward function. The approximation made in the objective function evaluation, indicated by $\hat{f}(\boldsymbol{\theta})$ for any $\boldsymbol{\theta} \in \Theta$, and correlation over the parameter space are accounted for by employing the Gaussian process (GP) regression [26] as:

$$\hat{f}(\boldsymbol{\theta}) \ \approx \ \mathcal{F}(\boldsymbol{\theta}) + \Delta\hat{f}_N, \tag{7}$$

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^2_N, which models, for all parameters, the uncertainty arising from the use of an RL with episode number N. The noise value, σ^2_N, indicates the variance of the objective function approximated by an RL with N episode number (more information is provided in Section III-C).

The following prior distribution is assumed for the GP:

$$\mathcal{F}(\boldsymbol{\theta}) = \mathcal{G}P\left(\mu(\boldsymbol{\theta}), k(\boldsymbol{\theta}, \boldsymbol{\theta})\right), \tag{8}$$

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 [26**?**].

Let $\mathbf{f}_m = [\hat{f}(\boldsymbol{\theta}^{(1)}), \ldots, \hat{f}(\boldsymbol{\theta}^{(m)})]^T$ be the approximated objective function obtained by performing m RLs with episode numbers $\mathbf{N}_m = (N^{(1)}, ..., N^{(m)})$ at samples $\boldsymbol{\Theta}_m = \left(\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(m)}\right)$. The posterior of the GP can be obtained as [26, 28]:

$$\mathcal{F}(\boldsymbol{\Theta}) \mid \boldsymbol{\Theta}_m, \mathbf{N}_m, \mathbf{f}_m \sim \mathcal{N}\left(\bar{\mathcal{F}}_m(\boldsymbol{\Theta}), \mathrm{cov}_m\left(\boldsymbol{\Theta}, \boldsymbol{\Theta}\right)\right), \quad (9)$$

where

$$\begin{aligned}\bar{\mathcal{F}}_m(\boldsymbol{\Theta}) = \mu(\boldsymbol{\Theta}) + \mathbf{K}_{\boldsymbol{\Theta},\boldsymbol{\Theta}_m}(\mathbf{K}_{\boldsymbol{\Theta}_m,\boldsymbol{\Theta}_m} + \boldsymbol{\Sigma}_{\mathbf{N}_m})^{-1} \\ (\mathbf{f}_m - \mu(\boldsymbol{\Theta}_m)),\end{aligned}$$

$$\begin{aligned}\mathrm{cov}_m(\boldsymbol{\Theta}, \boldsymbol{\Theta}) &= \mathbf{K}(\boldsymbol{\Theta}, \boldsymbol{\Theta}) \\ &- \mathbf{K}_{\boldsymbol{\Theta},\boldsymbol{\Theta}_m}(\mathbf{K}_{\boldsymbol{\Theta}_m,\boldsymbol{\Theta}_m} + \boldsymbol{\Sigma}_{\mathbf{N}_m})^{-1}\mathbf{K}^T_{\boldsymbol{\Theta},\boldsymbol{\Theta}_m},\end{aligned} \quad (10)$$

$\boldsymbol{\Sigma}_{\mathbf{N}_m}$ is a diagonal matrix of size m with ith diagonal element $(\boldsymbol{\Sigma}_{\mathbf{N}_m})_{ii} = \sigma^2_{N^{(i)}}$, and

$$\mathbf{K}_{\boldsymbol{\Theta},\boldsymbol{\Theta}'} = \begin{bmatrix} k(\boldsymbol{\theta}_1, \boldsymbol{\theta}'_1) & \ldots & k(\boldsymbol{\theta}_1, \boldsymbol{\theta}'_n) \\ \vdots & \ddots & \vdots \\ k(\boldsymbol{\theta}_l, \boldsymbol{\theta}'_1) & \ldots & k(\boldsymbol{\theta}_l, \boldsymbol{\theta}'_n) \end{bmatrix}, \quad (11)$$

for $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_l\}$ and $\boldsymbol{\Theta}' = \{\boldsymbol{\theta}'_1, ..., \boldsymbol{\theta}'_n\}$. Using the above formulation, the objective function before observing any data is modeled by a zero-mean Gaussian process with covariance $k(\boldsymbol{\theta}, \boldsymbol{\theta})$, while at iteration m, the objective function is predicted based on the sequence of queried samples $\boldsymbol{\Theta}_m$, the approximate objective function values $\mathbf{f}_m$, and the episode numbers $\mathbf{N}_m$ used for these approximations. The uncertainty in the objective function, which is modeled by the covariance function in equation (9), decreases as more points are sampled from the parameter space and added to the GP.

The hyper-parameters of the Gaussian process, such as the parameters of the kernel function or the mean function, can be estimated at each time point using the marginal likelihood function [26]:

$$\mathbf{f}_m \mid \boldsymbol{\Theta}_m, \mathbf{N}_m \sim \mathcal{N}\left(\mu(\boldsymbol{\Theta}_m), \mathbf{K}_{\boldsymbol{\Theta}_m,\boldsymbol{\Theta}_m} + \boldsymbol{\Sigma}_{\mathbf{N}_m}\right). \quad (12)$$

Notice that, due to the difficulty of choosing a proper model for the mean function and its impact on the learning accuracy, a possible option is to use $\mu(\boldsymbol{\theta}) = \min_{i=1,\ldots,m} \mathbf{f}_m(i)$. This adaptive constant mean avoids the challenging task of picking a proper parametric model for the mean function, and also prevents over-estimation of the objective function over regions that have not been explored well.

It should be noted that the Bayesian representation of the objective function makes the proposed framework different from the conventional Bayesian IRL [9–11, 29]. In fact, the BIRL methods do not represent the objective function by surrogate models and employ Bayesian techniques such as Gibbs sampling and Markov chain Monte Carlo based techniques to find the distribution of the parameters of the reward function. By contrast, the proposed framework provides the Bayesian representation of the objective function over parameters of the reward function, which yields the following benefits: 1) correlation consideration over the parameter space, which provides predicted values of the objective function over the whole parameter space given limited available information; 2) capability of uncertainty analysis through posterior of the Gaussian process representing the objective function, which provides the confidence regarding the predicted values at any given sample in the parameter space.

C. Simultaneous Sequential Selection of Parameter and Approximator:

The uncertainty in the objective function evaluation is modeled by parameter σ^2_N in the GP surrogate model in (7). This value, which indicates the variance of the objective function approximated by an RL, mainly depends on the RL episode number N and can be quantified in one of the following three ways: 1) running multiple RL with a fixed episode number at an arbitrary sample of the parameter space and computing the sample variance of the approximated objective values; 2) using available theoretical upper bounds on the approximation error of RL techniques for any choice of episode number [30, 31]; 3) treating the noise parameters as hyper-parameters and learning them on the fly according to the marginal likelihood in (12).

The variance issue is linked to the computational complexity of RL algorithms. The computational complexity, which will be indicated by c_N in this paper, increases linearly with the number of episodes. This means that the large N refers to an approximator with smaller variance (high- fidelity) and high computational complexity, whereas small N leads to an approximator with large uncertainty (low-fidelity) but low computational complexity. This has been clearly shown in Fig. 2, where the main objective function denoted by solid black line over one-dimensional parameter space is modeled by a surrogate model constructed according to 10 evaluations using low-fidelity approximator in the left plot and a surrogate model constructed according to 2 evaluations by high-fidelity approximator in the right plot. While the computational cost of 10 evaluations by low-fidelity approximator is the same as 2 evaluations by high-fidelity approximator, it can be seen that exploration is well performed by the low-fidelity approximator. Notice that Fig. 2 is for illustration purposes, and samples are not computed by performing RL algorithms.

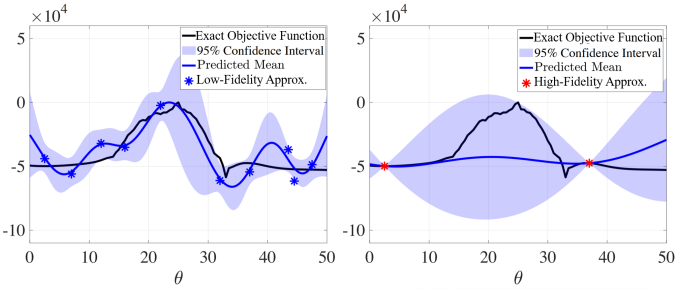


Fig. 2: The GP surrogate models constructed over function evaluations obtained by low-fidelity (left) and high fidelity (right) apprximators.

In order to achieve a solution that is scalable, accurate and fast, this paper introduces a policy that takes all these fidelity approximators into account and simultaneously selects good

sample points in the parameter space and good approximators of the objective function to achieve both accurate and fast learning process in inverse reinforcement learning domain. Assuming that we are at iteration m during the learning process (which means the set of objective functions $\mathbf{f}_m$ are approximated by performing m RLs with episode numbers $\mathbf{N}_m$ at sample set $\mathbf{\Theta}_m$), if we want to stop the learning process at the current iteration, the best estimate of parameters of the reward function would be a sample in the parameter space which has the highest expected mean in the posterior distribution of the GP. This sample point can be selected according to (10) as:

$$\hat{\boldsymbol{\theta}}^{*}_{\mathrm{GP}} = \underset{\boldsymbol{\theta}\in\Theta}{\operatorname{argmax}}\, \bar{\mathcal{F}}_m(\boldsymbol{\theta}), \tag{13}$$

where $\bar{\mathcal{F}}_m(\boldsymbol{\theta}) = \mathbb{E}[\mathcal{F}(\boldsymbol{\theta}) \mid \mathbf{\Theta}_m, \mathbf{N}_m, \mathbf{f}_m]$. If an additional pair of parameter and approximator is to be selected from the parameter space and set of approximators, we suggest choosing the pair with the highest single-period expected increase in the maximum of the surrogate model per unit cost. This can be formulated as:

$$(\boldsymbol{\theta}^{(m+1)}, N^{(m+1)}) = \underset{(\boldsymbol{\theta},N)\in(\Theta,\mathbb{N})}{\operatorname{argmax}} \frac{1}{c_N} \mathbb{E}_m \Big[\max_{\boldsymbol{\theta}'\in\Theta} \mathbb{E}\left[\mathcal{F}(\boldsymbol{\theta}') \mid \mathbf{\Theta}_m, \mathbf{N}_m, \mathbf{f}_m, \boldsymbol{\theta}^{(m+1)} = \boldsymbol{\theta}, N^{(m+1)} = N\right] - \max_{\boldsymbol{\theta}'\in\Theta} \mathbb{E}\left[\mathcal{F}(\boldsymbol{\theta}') \mid \mathbf{\Theta}_m, \mathbf{N}_m, \mathbf{f}_m\right] \Big], \tag{14}$$

where $\mathbb{N}$ denotes a finite set of episode numbers corresponding to a finite number of approximators, and $\mathbb{E}_m$ denotes expectation over the unobserved objective function at point $\boldsymbol{\theta}^{(m+1)}$ approximated by a RL (approximator) with $N^{(m+1)}$ episode number, given all available information up to iteration m.

An exact closed-form solution for computation of (14) is possible through the knowledge gradient algorithm [32], if the parameter space, Θ, be replaced by a finite set of alternative in (14). This finite alternative set should be selected based on the goal of the learning process. In particular, for non-prior based IRL techniques [1, 2, 6, 7, 13**?**] hypercube sampling [33] can be employed for generating this set, whereas for prior-based IRL techniques [9, 12], the alternative set could be provided by samples drawn from the prior distribution. The set of episode numbers, which specifies the number of approximators, needs to be chosen based on the system's size. However, the algorithm is fairly robust against this choice. In our numerical experiments, we observed that "small", "medium," and "large" episode numbers all lead to good learning accuracy and speed.

Let $\mathbf{\Theta}^a$ be the alternative set containing n samples from the parameter space (i.e., $\mathbf{\Theta}^a \subset \Theta$), and $\boldsymbol{\mu}^a_m$ and $\mathbf{\Sigma}^a_m$ be the mean and covariance functions over the alternative samples computed using the expressions in (10) using all available information up to time step m. The posterior distribution of $\mathcal{F}(.)$ at time $m+1$ depends on the selected sample from the alternative set $\boldsymbol{\theta}^{(m+1)}$, episode number $N^{(m+1)}$ and the approximated objective function $f(\boldsymbol{\theta}^{(m+1)})$. For a given ith alternative sample (i.e., $\boldsymbol{\theta}^{(m+1)} = \boldsymbol{\theta}_i \in \mathbf{\Theta}^a$), this posterior distribution may be calculated using standard results for normal sampling with a multivariate normal prior distribution. The knowledge gradient policy selects the best sample from the alternative set before calculating the desirability function. The mean and variance associated to the ith alternative sample are $\mathbb{E}[\boldsymbol{\mu}^a_{m+1}] = \boldsymbol{\mu}^a_m$ and $\tilde{\sigma}(\mathbf{\Sigma}^a_m, i, N)\tilde{\sigma}(\mathbf{\Sigma}^a_m, i, N)^T$, where

$$\tilde{\sigma}(\mathbf{\Sigma}^a_m, i, N) = \frac{\mathbf{\Sigma}^a_m \mathbf{e}_i}{\sqrt{\sigma^2_N + (\mathbf{\Sigma}^a_m)_{ii}}}, \tag{15}$$

and $(\mathbf{\Sigma}^a_m)_{ii}$ refers to the element in the ith row and ith column of matrix $\mathbf{\Sigma}^a_m$. Thus, the conditional distribution of $\boldsymbol{\mu}^a_{m+1}$ upon selecting parameter N and the ith sample from the alternative set is $\boldsymbol{\mu}^a_{m+1} = \boldsymbol{\mu}^a_m + \tilde{\sigma}(\mathbf{\Sigma}^a_m, i, N)\, Z$, where Z is any independent one-dimensional standard normal random variable. This allows us to rewrite (14) as:

$$(\boldsymbol{\theta}^{(m+1)}, N^{(m+1)}) = \underset{(\boldsymbol{\theta},N)\in(\mathbf{\Theta}^a,\mathbb{N})}{\operatorname{argmax}} \frac{1}{c_N} \Big[\mathbb{E}_m \Big[\max_{i\in\{1,\dots,n\}} \left(\boldsymbol{\mu}^a_m(i) + \tilde{\sigma}\left(\mathbf{\Sigma}^a_m, i, N\right) Z \right) \mid \mathbf{\Theta}_m, \mathbf{N}_m, \mathbf{f}_m, \boldsymbol{\theta}^{(m+1)} = \boldsymbol{\theta}_i, N^{(m+1)} = N \Big] - \max_{i\in\{1,\dots,n\}} \boldsymbol{\mu}^a_m(i) \Big].$$

The steps toward the exact solution for the above optimization problem are provided in two algorithms presented in [32]. The features of the knowledge gradient policy to account for the correlation in the parameter space and eliminating the assumptions regarding known-value and noise-free evaluations are the reasons for choosing this acquisition function over other Bayesian optimization techniques, such as expected improvement and entropy search [34]. Also, the method has been shown in [32] to perform very well when faced with highly nonlinear and multimodal objective functions.

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) changes in the maximum of the mean of the constructed GP in consecutive iterations falls below a pre-specified threshold; 2) algorithm is performed for a fixed pre-specified amount of time.

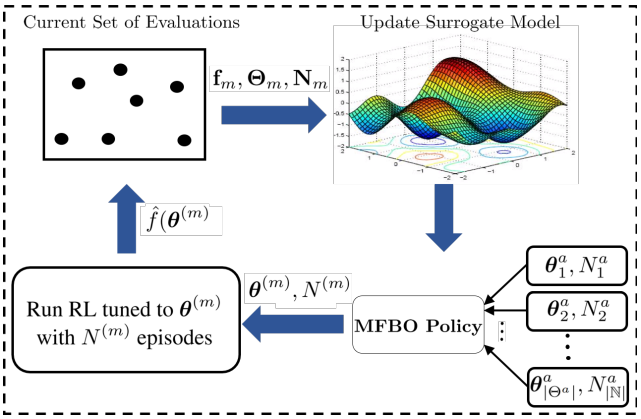
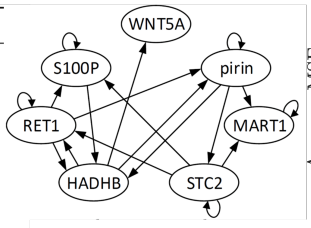


Fig. 3: Schematic diagram of the proposed framework.

IV. Numerical 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. The

Genes	Input Gene(s)	Output
WNT5A	HADHB	10
pirin	prin, RET1,HADHB	00010111
S100P	S100P,RET1,STC2	10101010
RET1	RET1,HADHB,STC2	00001111
MART1	pirin,MART1,STC2	10101111
HADHB	pirin,S100P,RET1	01110111
STC2	pirin,STC2	1101



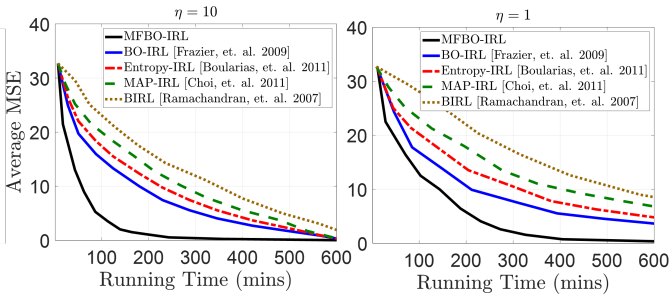


Fig. 4: Results of Melanoma Boolean regulatory network.

Algorithm 1 MFBO-IRL Algorithm

1: Pick a desired reward function $R_{\boldsymbol{\theta}}$ with $\boldsymbol{\theta} \in \Theta$; set the finite alternative set Θ^a; specify the set of episode numbers $\mathbb{N}$; set the cost c_N and variance parameter σ_N^2 for any $N \in \mathbb{N}$.

2: Construct a GP over parameter space Θ.

3: $m = -1$, $\boldsymbol{\Theta}_0 = \{\}$, $\mathbf{N}_0 = \{\}$, $\mathbf{f}_0 = \{\}$.

4: **while** stopping criterion is not met **do**

5: $m = m + 1$.

6: Select $(\boldsymbol{\theta}^{(m+1)}, N^{(m+1)})$ according to (14).

7: Run an RL tuned to $\boldsymbol{\theta}^{(m+1)}$ with episode number $N^{(m+1)}$ to get $\hat{f}(\boldsymbol{\theta}^{(m+1)})$.

8: $\boldsymbol{\Theta}_{m+1} = \{\boldsymbol{\Theta}_m, \boldsymbol{\theta}^{(m+1)}\}$, $\mathbf{N}_{m+1} = \{\mathbf{N}_m, N^{(m+1)}\}$, $\mathbf{f}_{m+1} = \{\mathbf{f}_m, \hat{f}(\boldsymbol{\theta}^{(m+1)})\}$.

9: Update GP according to $(\boldsymbol{\Theta}_{m+1}, \mathbf{N}_{m+1}, \mathbf{f}_{m+1})$.

10: **end while**

11: $\hat{\boldsymbol{\theta}}^*_{\text{GP}} = \text{argmax}_{\boldsymbol{\theta}\in\Theta} \bar{\mathcal{F}}_m(\boldsymbol{\theta})$, where $\bar{\mathcal{F}}_m(\boldsymbol{\theta})$ is the mean of the final GP.

performance of the proposed framework in terms of accuracy and speed is examined through three problems, described below.

Melanoma Gene Regulatory Network: We consider a gene regulatory network corresponding to the most dangerous form of skin cancer, called Melanoma [35]. The dynamical behavior of Melanoma is represented through the Boolean activities of 7 genes displayed in Fig. 4. Each gene expression can be 0 or 1, corresponding to gene inactivation or activation, respectively. The gene states are assumed to be updated at each discrete time through the following nonlinear signal model:

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}) \oplus \boldsymbol{\beta}_k(\mathbf{a}_{k-1}) \oplus \boldsymbol{n}_k \,, \tag{16}$$

where $\mathbf{x}_k = [\text{WNT5A}_k, \text{pirin}_k, \text{S100P}_k, \text{RET1}_k, \text{MART1}_k, \text{HADHB}_k, \text{STC2}_k]$ is the state vector at time step k, $\boldsymbol{\beta}_k(\mathbf{a}_{k-1}) \in \{0,1\}^d$ is a Boolean noisy input vector, such that $P(\boldsymbol{\beta}_k(i) = 1) = \mathbf{a}_{k-1}(i)$, for $i = 1, ..., d$, and $\mathbf{n}_k \in \{0,1\}^d$ is a Boolean transition noise vector, such that $P(\mathbf{n}_k(i) = 1) = p$, for $i = 1, ..., d$. The transition noise is assumed to be $p = 0.05$. In practice, the gene states are observed through gene expression technologies such as cDNA microarray or image-based assay [36]. A Gaussian observation model is appropriate for modeling the gene expression data produced by these technologies:

$$\mathbf{y}_k(i) \sim \mathcal{N}(20\,\mathbf{x}_k(i) + 30, 15) \,, \tag{17}$$

for $i = 1, \ldots, 7$; where 20 is called base-line expression, 30 is referred to as the differential expression and standard deviation 15 models the uncertainty in the process. It can be shown that the partially-observed MDP in (16) and (17) can be transformed into an MDP in a continuous belief space [37–39]:

$$\begin{aligned} \mathbf{s}_k &= \mathbf{g}(\mathbf{s}_{k-1}, \mathbf{a}_{k-1}, \boldsymbol{\theta}) \\ &\propto p(\mathbf{y}_k \mid \mathbf{x}_k, \boldsymbol{\theta})\, P(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{a}_k)\, \mathbf{s}_{k-1} \,, \end{aligned} \tag{18}$$

The belief state is a vector of length 128 in a simplex of size 127.

In [35], the expression of WNT5A was found to be highly discriminatory between cells with properties typically associated with high metastatic competence versus those with low metastatic competence. Hence, an intervention that blocked the WNT5A protein from activating its receptor could substantially reduce the ability of WNT5A to induce a metastatic phenotype. These information, which are the biologist knowledge, are reflected in the following unknown immediate reward function:

$$R_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{a}) = \theta_1 \sum_{i=1}^{128} \mathbf{s}(i)\, \delta_{\mathbf{x}^i(1)=0} - \theta_2 ||\mathbf{a}||_1, \tag{19}$$

where the parameter vector in this case is $\boldsymbol{\theta} = [\theta_2\, \theta_2]^T$, with the true values $\boldsymbol{\theta}^* = [5\, 1]^T$ and parameter space $\Theta = [0, 10]^2$. Action/intervention space used by the biologist during the intervention process is assumed to be $\mathbb{A} = 0 \times 0 \times 0 \times [0, 1] \times 0 \times [0, 1] \times 0$, which means that the RET1 and the HADHB genes are used as the control gene to reduce the activation of WNT5A. The decomposable squared exponential, and delta Kronecker kernel functions are used for Gaussian process regression over the belief state and action spaces. Due to the large continuous state and action spaces, the well-known GP-SARSA algorithm [40], which is a well-known technique in the family of Gaussian process temporal difference learning (GPTD), is used as an RL for the approximation of the objective function. The

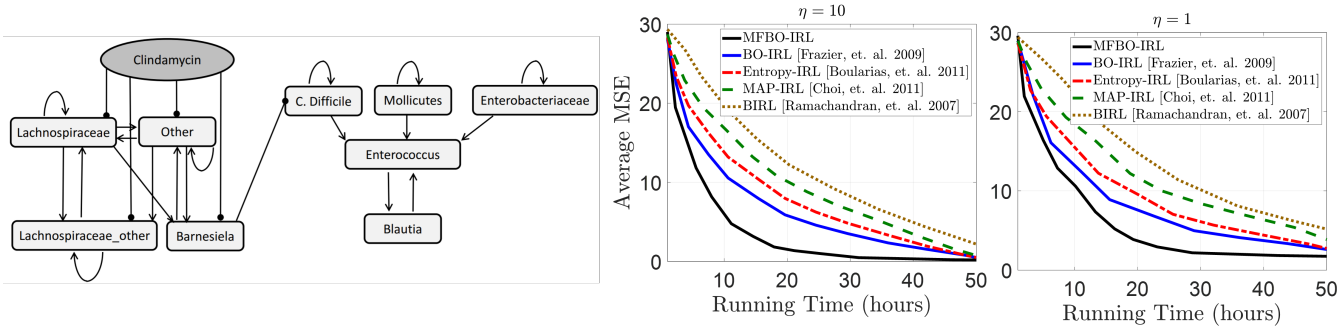


Fig. 5: Boolean regulatory network of gut microbiome.

TABLE I: The Average results obtained for random MDPs.

	$\eta = 10$			$\eta = 1$		
	$T = 100$	$T = 1{,}000$	$T = 10{,}000$	$T = 500$	$T = 1{,}000$	$T = 10{,}000$
Prop. MFBO-IRL	6.07	1.72	0.88	9.66	2.92	2.02
BO-IRL [Frazier, et. al. 2009]	12.04	7.64	6.64	16.93	9.25	8.02
Entropy-IRL [Boularias, et. al. 2011]	15.31	10.09	8.90	19.83	12.72	10.84
BIRL [Ramachandran, et. al. 2007]	16.32	10.84	9.10	19.99	12.90	11.39

results of the proposed framework are compared with four well-known techniques: Bayesian optimization-based inverse reinforcement learning (BO-IRL) [32], relative entropy inverse reinforcement learning (Entropy-IRL) [41], maximum aposteriori inverse reinforcement learning (MAP-IRL) [5], and Bayesian inverse reinforcement learning (BIRL) [9]. For the proposed MFBO-IRL, the set of episode numbers for training the GP-SARSA algorithm is set to be $\mathbb{N} = \{100, 500, 1{,}000\}$, whereas for the rest of the techniques, the episode number is set to be 500.

The following stopping criteria are used for various methods: MFBO-IRL, BO-IRL, and MAP-IRL algorithms all stop when changes in the estimated values of all parameters over a window of length 20 fall below 5% of their range; BIRL with Gaussian proposal distribution stops upon completion of $6{,}000$ iterations, where this number of iterations is chosen through trial and error to guarantee to achieve the best results by the IBRL algorithm.

Two sets of demonstration data with the length of 500 are generated according to the Boltzmann soft policy in (6). The sequence of actions in the first demonstration data mostly follow the optimal policy since a larger confidence rate (i.e., $\eta = 10$) is used for the action selection process; whereas the second demonstration data has more randomness due to the smaller confidence rate (i.e., $\eta = 1$) used for the action selection process. Fig. 4 displays the average MSE of estimation of the parameters overrunning time in minutes for both cases. One can observe that accurate estimation is achieved much faster by the proposed MFBO-IRL in comparison to other techniques. A lower average MSE can also be seen for demonstration data with a larger confidence rate (i.e., $\eta = 10$). The reliability aspect of the proposed framework can be understood by looking at both plots in Fig. 4, where much lower MSE in the estimation of the reward function parameters is achieved by the proposed framework in comparison to other competing techniques.

Gut Microbial Community: In this part of numerical experiment, the performance of the proposed framework is examined over data acquired during the intervention process in gut microbial community. The regulatory relationship representing the relationship among microbes in the gut microbial community is presented in Fig. 5. The network contains information of 9 microbes, in which their value represented in a single state vector as: $\mathbf{x}_k = [\text{Lachnospiraceae}_k, \text{Other}_k, \text{Lachnospiraceae\_Other}_k, \text{Barnesiela}_k, \text{C. Difficile}_k, \text{Mollicutes}_k, \text{Entero}_k, \text{Enterobacteriaceae}_k, \text{Enterococcus}_k, \text{Blautia}_k]^T$. The external input to this network is "Clindamycin", which is an antibiotics that fights bacteria in the body and can treat various types of infections. The same model as (16) is used for capturing the dynamical behavior of this system. The network function in this case can be represented as:

$$f(\mathbf{x}_k) = \overline{\mathbf{R}\,\mathbf{x}_k + \mathbf{b}} \tag{20}$$

where

$$\mathbf{R} = \begin{bmatrix} +1 & +1 & +1 & 0 & 0 & 0 & 0 & 0 & 0 \\ +1 & +1 & 0 & +1 & 0 & 0 & 0 & 0 & 0 \\ +1 & +1 & +1 & 0 & 0 & 0 & 0 & 0 & 0 \\ +1 & +1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & +1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & +1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & +1 & 0 & 0 \\ 0 & 0 & 0 & 0 & +1 & +1 & +1 & 0 & +1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & 0 \end{bmatrix},$$

$$\mathbf{b} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & +1 & 0 & 0 & 0 \end{bmatrix}^T.$$

The intervention is taken over Barnesiela and C. Difficile, which can be denoted by the following action space:

$$\mathbb{A} = 0 \times 0 \times 0 \times [0,\ 1] \times [0,\ 1] \times 0 \times 0 \times 0 \times 0. \tag{21}$$

The goal is preventing C. Difficile to be upregulated. For more information about the biological rationale for this, the reader is referred to [42]. Thus, we assume the following reference immediate reward function:

$$R_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{a}) = \theta_1 \sum_{i=1}^{512} \mathbf{s}(i)\, \delta_{\mathbf{x}^i(5)=0} - \theta_2 ||\mathbf{a}||_1 \qquad (22)$$

where the unknown parameters of the reward function are encoded in a single parameter vector $\boldsymbol{\theta} = [\theta_1\,, \theta_2]$, with the true value $\boldsymbol{\theta}^* = [7\,, 2]$. The space of parameters are assumed to be $\Theta = [0,\, 10]^2$. The large size of state space, which is in a simplex of size 512, necessitates performing scalable reinforcement learning for approximation of the objective function. We have used deep Q-learning introduced in [43] with the set of episode/epoch numbers 200, 1000 and 5000. The same stopping criteria used in the previous part are used for this part of experiment. The average MSE of the estimated parameters versus running time per hour for two sets of demonstrations with the length of 1000 and confidence parameter $\eta = 10$ and $\eta = 1$ is presented in Fig. 5. One can see that the proposed framework has the highest rate of reduction in MSE, one average, in running time.

Random MDPs: For the last part of numerical experiments, a simple 6-states and 2-actions MDPs are generated according to the symmetric Dirichlet distribution with parameter $\phi = 10$. The reward is simply assumed to be as $R_{\boldsymbol{\theta}}(\mathbf{s}) = \theta_1\mathbf{s}(1) + \theta_2\mathbf{s}(2) + \theta_3\mathbf{s}(3) + \theta_4\mathbf{s}(4) + \theta_5\mathbf{s}(5) + \theta_6\mathbf{s}(6)$, where $\boldsymbol{\theta} = [\theta_1\ \theta_2\ \theta_4\ \theta_5\ \theta_6]^T$ with the true value $\boldsymbol{\theta}^* = [1\,2\,1.5\ -\ 1\ -\ 2\ -\ 0.5]^T$. This means that the system is desired to spend the most amount of time in the first three states and avoid the last three ones. The parameter space is assumed to be $\Theta = [-3,\, 3]^6$. Since the state and action spaces are small, the well-known Q-learning algorithm is used to approximate the objective function. The set of episode number 100 and 1000 as low and medium fidelity models and the exact dynamic programming (value iteration) technique as the highest fidelity model are used by the proposed framework, while other techniques used 1000 episode number. All methods are stopped after 10 minutes of performing for any given MDP. The average MSE obtained by different methods is shown in Table I. It can be seen that for the same running time, a much lower average MSE is obtained by the proposed MFBO-IRL framework in comparison to other techniques for all data lengths and confidence rates.

V. Conclusion

In this paper, we introduced a multi-fidelity Bayesian optimization framework for scalable, fast, and accurate inverse reinforcement learning. For large systems with any arbitrary reward function containing a relatively small number of parameters, the proposed framework enables incorporating multiple reinforcement learning algorithms with different episode numbers as multiple fidelity approximators for the learning process. The correlation, uncertainty, and computational cost of these approximators are efficiently considered in the proposed framework using the Gaussian process surrogate model and efficiently selecting a sample of parameters and episode number of reinforcement learning for an efficient learning process. In numerical experiments, the proposed algorithm performed significantly faster than competing algorithms. Future work will focus on developing IRL techniques capable of handling reward functions with relatively large parameters (e.g., deep neural networks).

Acknowledgment

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

References

[1] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the twenty-first international conference on Machine learning*, p. 1, ACM, 2004.

[2] A. Y. Ng, S. J. Russell, *et al.*, "Algorithms for inverse reinforcement learning.," in *Proceedings of the 17th international conference on Machine learning*, vol. 1, p. 2, 2000.

[3] J. Choi and K.-E. Kim, "Nonparametric Bayesian inverse reinforcement learning for multiple reward functions," in *Advances in Neural Information Processing Systems*, pp. 305–313, 2012.

[4] S. Zhifei and E. Meng Joo, "A survey of inverse reinforcement learning techniques," *International Journal of Intelligent Computing and Cybernetics*, vol. 5, no. 3, pp. 293–311, 2012.

[5] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning," in *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, 2008.

[6] N. D. Ratliff, J. A. Bagnell, and M. A. Zinkevich, "Maximum margin planning," in *Proceedings of the 23rd international conference on Machine learning*, pp. 729–736, ACM, 2006.

[7] D. Silver, J. A. Bagnell, and A. Stentz, "Perceptual interpretation for autonomous navigation through dynamic imitation learning," in *Robotics Research*, pp. 433–449, Springer, 2011.

[8] U. Syed and R. E. Schapire, "A game-theoretic approach to apprenticeship learning," in *Advances in neural information processing systems*, pp. 1449–1456, 2008.

[9] D. Ramachandran and E. Amir, "Bayesian inverse reinforcement learning.," in *International Joint Conferences on Artificial Intelligence (IJCAI)*, vol. 7, pp. 2586–2591, 2007.

[10] B. Michini and J. P. How, "Bayesian nonparametric inverse reinforcement learning," in *Joint European conference on machine learning and knowledge discovery in databases*, pp. 148–163, Springer, 2012.

[11] J. Zheng, S. Liu, and L. M. Ni, "Robust Bayesian inverse reinforcement learning with sparse behavior noise," in *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[12] C. A. Rothkopf and C. Dimitrakakis, "Preference elicitation and inverse reinforcement learning," in *Joint European conference on machine learning and knowledge discovery in databases*, pp. 34–48, Springer, 2011.

[13] G. Neu and C. Szepesvári, "Apprenticeship learning using inverse reinforcement learning and gradient methods," pp. 295–302, 2007.

[14] S. K. S. Ghasemipour, S. S. Gu, and R. Zemel, "SMILe: Scalable meta inverse reinforcement learning through context-conditional policies," in *Advances in Neural Information Processing Systems*, pp. 7881–7891, 2019.

[15] Z. Wu, L. Sun, W. Zhan, C. Yang, and M. Tomizuka, "Efficient sampling-based maximum entropy inverse reinforcement learning with application to autonomous driving," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5355–5362, 2020.

[16] S. Levine, Z. Popovic, and V. Koltun, "Nonlinear inverse reinforcement learning with Gaussian processes," in *Advances in Neural Information Processing Systems*, pp. 19–27, 2011.

[17] M. Wulfmeier, P. Ondruska, and I. Posner, "Maximum entropy deep inverse reinforcement learning," *arXiv preprint arXiv:1507.04888*, 2015.

[18] M. Guo, Z. Zhong, W. Wu, D. Lin, and J. Yan, "Irlas: Inverse reinforcement learning for architecture search," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9021–9029, 2019.

[19] D. S. Brown and S. Niekum, "Machine teaching for inverse reinforcement learning: Algorithms and applications," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 7749–7758, 2019.

[20] F. S. Melo and M. Lopes, "Learning from demonstration using MDP induced metrics," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 385–401, Springer, 2010.

[21] S. Zhifei and E. M. Joo, "A review of inverse reinforcement learning theory and recent advances," in *2012 IEEE Congress on Evolutionary Computation*, pp. 1–8, IEEE, 2012.

[22] D. P. Bertsekas, *Dynamic programming and optimal control*, vol. 1. Athena scientific Belmont, MA, 1995.

[23] W. B. Powell, *Approximate Dynamic Programming: Solving the curses of dimensionality*, vol. 703. John Wiley & Sons, 2007.

[24] R. S. Sutton, A. G. Barto, *et al.*, *Introduction to reinforcement learning*, vol. 2. MIT press Cambridge, 1998.

[25] M. Imani, S. F. Ghoreishi, and U. M. Braga-Neto, "Bayesian control of large MDPs with unknown dynamics in data-poor environments," in *Advances in neural information processing systems*, pp. 8156–8166, 2018.

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

[27] J. Choi and K.-E. Kim, "Map inference for Bayesian inverse reinforcement learning," in *Advances in Neural Information Processing Systems*, pp. 1989–1997, 2011.

[28] 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.

[29] 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.

[30] P. Auer, "Using confidence bounds for exploitation-exploration trade-offs," *Journal of Machine Learning Research*, vol. 3, no. Nov, pp. 397–422, 2002.

[31] M. G. Azar, I. Osband, and R. Munos, "Minimax regret bounds for reinforcement learning," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 263–272, JMLR. org, 2017.

[32] 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.

[33] 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.

[34] 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.

[35] M. Bittner, P. Meltzer, Y. Chen, Y. Jiang, E. Seftor, M. Hendrix, M. Radmacher, R. Simon, Z. Yakhini, A. Ben-Dor, *et al.*, "Molecular classification of cutaneous malignant melanoma by gene expression profiling," *Nature*, vol. 406, no. 6795, p. 536, 2000.

[36] J. Hua, C. Sima, M. Cypert, G. C. Gooden, S. Shack, L. Alla, E. A. Smith, J. M. Trent, E. R. Dougherty, and M. L. Bittner, "Dynamical analysis of drug efficacy and mechanism of action using GFP reporters," *Journal of Biological Systems*, vol. 20, no. 04, pp. 403–422, 2012.

[37] M. Imani and U. M. Braga-Neto, "Point-based methodology to monitor and control gene regulatory networks via noisy measurements," *IEEE Transactions on Control Systems Technology*, 2018.

[38] M. Imani and U. M. Braga-Neto, "Finite-horizon LQR controller for partially-observed Boolean dynamical systems," *Automatica*, vol. 95, pp. 172–179, 2018.

[39] M. Imani and U. M. Braga-Neto, "Control of gene regulatory networks with noisy measurements and uncertain inputs," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 2, pp. 760–769, 2018.

[40] Y. Engel, S. Mannor, and R. Meir, "Reinforcement learning with Gaussian processes," in *Proceedings of the 22nd international conference on Machine learning*, pp. 201–208, ACM, 2005.

[41] A. Boularias, J. Kober, and J. Peters, "Relative entropy inverse reinforcement learning," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 182–189, 2011.

[42] 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.

[43] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, 2015.