Optimal Inference of Hidden Markov Models Through Expert-Acquired Data

Amirhossein Ravari, Seyede Fatemeh Ghoreishi, and Mahdi Imani

***Abstract*—This paper focuses on inferring a general class of hidden Markov models (HMMs) using data acquired from experts. Expert-acquired data contain decisions/actions made by humans/users for various objectives, such as navigation data reflecting drivers' behavior, cybersecurity data carrying defender decisions, and biological data containing the biologist's actions (e.g., interventions and experiments). Conventional inference methods rely on temporal changes in data without accounting for expert knowledge. This paper incorporates expert knowledge into the inference of HMMs by modeling expert behavior as an imperfect reinforcement learning agent. The proposed method optimally quantifies experts' perceptions about the system model, which, alongside the temporal changes in data, contributes to the inference process. The proposed inference method is derived through a combination of dynamic programming and optimal recursive Bayesian estimation. The applicability of this method is demonstrated to a wide range of inference criteria, such as maximum likelihood and maximum a posteriori. The performance of the proposed method is investigated through a comprehensive numerical experiment using a benchmark problem and biological networks.**

***Impact Statement*—Inference is an essential tool for understanding and predicting the behavior of complex systems and processes. Expert knowledge, such as that of scientists, doctors, or engineers, can provide valuable insights into the underlying mechanisms of complex systems. Incorporating expert knowledge and intention into the inference process allows for constructing/learning models that carry valuable expert perception of complex systems. This helps overcome data limitations, deal with the non-identifiability of models and increase the accuracy of the inference process. In particular, the biological application will help to fill the gap between expert knowledge and mathematical/computational approaches, allowing for valuable domain knowledge to be quantified and incorporated into the modeling.**

***Index Terms*—Expert-Enabled Inference, Hidden Markov Models, Inverse Reinforcement Learning, Gene Regulatory Networks.**

I. Introduction

Modeling complex processes requires a deep understanding of their underlying components and their interactions. Hidden Markov Models (HMMs) are an important class of dynamic models with a broad range of applications in fields such as robotics, healthcare, smart grids, and finance [1]–[6]. Inferring the parameters of these models through the available time-series data is a crucial step in modeling complex systems using HMMs. Over the years, several inference techniques have been developed for HMMs [7]–[14]. These include maximum likelihood, expectation-maximization, maximum a posteriori, and Markov chain Monte Carlo techniques [15]–[23]. These techniques are capable of accounting for temporal changes in data and prior probabilities (if available) of models for point-based or distribution inference.

Two common challenges in the inference of HMMs are non-identifiability issues and data limitation [24], [25]. The non-identifiability problem arises when multiple system models have the same probability given the observed data, meaning that the true model is not distinguishable among models given the available data. In addition, data limitation relative to large parameters needed for modeling complex real-world systems negatively impacts inference accuracy and the predictive power of the inferred model.

In many domains, the available data contain subject matter experts' decisions, which is referred to as expert-acquired data. Examples of these domains abound, such as medical data obtained according to a course of treatments made by medical professionals, cybersecurity data containing monitoring or cleaning decisions made by engineers, and navigation data that contain the human drivers' actions. The standard HMM tools fail to account for the experts' intentions explicitly and only incorporate the temporal changes in data into the inference process. Ignoring the expert intention can lead to the unreliability of the inference process, as experts' decisions often carry their rich knowledge about how the system's behavior and operation, and provide valuable insights into the model of complex dynamical systems. For example, the biologist's choice of interventions during the therapy reflects their understanding of the underlying mechanisms of biological systems. By incorporating expert knowledge into the inference process, it is possible to overcome data limitations and the non-identifiability of system models, which are two key challenges in accurately modeling complex systems.

Several methods have been developed to account for expert supervision to enhance the performance of the inference process in HMMs. One class of approaches is supervised inference through labeled data, where manual annotations by experts are used with regard to the system state to enhance the inference process [26], [27]. Data augmentation techniques are another class of techniques that aim to generate additional training data that adhere to expert constraints [28]–[31]. Finally, Bayesian approaches have been developed to serve as regularization mechanisms to influence the parameter estimation towards values that are consistent with the expert's knowledge [32], [33]. While all aforementioned methods rely on expert presence and supervisors to label, impose con-

A. Ravari and M. Imani are with the Department of Electrical and Computer Engineering, and S.F. Ghoreishi is with the Department of Civil and Environmental Engineering and Khoury College of Computer Sciences at Northeastern University. Emails: ravari.a@northeastern.edu, f.ghoreishi@northeastern.edu, m.imani@northeastern.edu

straints, or augment data, this paper develops a method for implicit and non-supervised incorporation of expert knowledge into the inference of HMMs.

It should be noted that inferring the system model differs from recovering the expert reward function or policy, commonly discussed in imitation learning and inverse reinforcement learning [34]–[41]. Assuming the expert reward functions are known, the goal is to implicitly incorporate the information carried by the expert-acquired data into modeling and inference. This paper models the expert as a reinforcement learning (RL) agent, which provides probabilistic modeling of expert perception of complex systems. The non-optimality of the expert is accounted for using well-known Boltzmann and ϵ-greedy policies [42], where the expert deviation from optimal decision-maker, if unknown, can also be quantified during the inference. A combination of dynamic programming and Bayesian filtering is employed for quantifying expert knowledge in partially observable environments. The proposed optimal expert-enabled inference includes an expert-knowledge term in addition to the transition term, measuring the temporal data change in conventional inference techniques. The expert-knowledge term can be incorporated into a wide range of inference criteria, including maximum likelihood, maximum a posteriori, and Bayesian inference. The analytical results demonstrate that incorporating expert perception always helps better distinguish models, as long as the expert decisions are more informative than a random policy.

The primary application of this paper is to infer the model of biological systems, such as gene regulatory networks and microbial communities. These regulatory networks are often modeled through HMMs with binary state variables, where the state values of biological components are observed imperfectly through high-throughput sequencing data. The proposed method's performance is investigated in inferring regulatory interactions in these biological systems using data acquired by biologist during the intervention/perturbation process. Aside from a benchmark problem, the numerical experiments utilize the p53-MDM2 negative feedback loop network and the gut microbial community to compare the results of the proposed method with conventional inference methods.

The article is organized as follows. Section II describes the HMM and the optimal infinite-horizon policy for the underlying state process of HMM. The formulation of optimal expert-enabled inference and its recursive matrix-form implementation are discussed in Sections III and IV, respectively. Discussions and analysis have been mentioned in Section V. Finally, numerical examples and concluding remarks are presented in Section VI and Section VII, respectively.

II. Mathematical Preliminaries

In this section, a brief description of the HMMs is provided, followed by modeling the optimal policy derived for arbitrary data collection objectives.

A. Finite-State Hidden Markov Models

The HMMs characterize the dynamical behavior of systems observed through noisy time-series data. The schematic diagram of HMMs with external inputs is shown in Fig. 1. The time-varying behavior of systems is represented through the state process, where $\mathbf{x}_k$ represents the system state at time step k. This paper considers HMMs with a finite state space, where the system state takes a value in a finite set $\mathcal{X}$. The sequence of states is observed indirectly through the measurement process. The finite-state HMM can be represented by the following two processes [5]:

$$\begin{aligned} \mathbf{x}_k &= \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_{k-1}, \mathbf{a}_{k-1}, \mathbf{n}_k) \quad \text{(state process)} \\ \mathbf{y}_k &= \mathbf{h}_{\boldsymbol{\theta}}(\mathbf{x}_k, \mathbf{v}_k) \quad \text{(measurement process)} \end{aligned} \tag{1}$$

for $k = 1, 2, \ldots$; where $\boldsymbol{\theta}$ is a vector of unknown parameters which takes a value in a set Θ, $\mathbf{a}_{k-1} \in \mathcal{A}$ is the external input to the system at time $k-1$, $\mathbf{n}_k$ is the transition noise at time k, and $\mathbf{f}_{\boldsymbol{\theta}}$ represents the system dynamics. The measurements depend on the sensor/technology in use, where $\mathbf{h}$ is a general function mapping the current state and measurement noise $\mathbf{v}_k$ into the measurement space. The noise processes $\{\mathbf{n}_k, \mathbf{v}_k; k = 1, 2, \ldots\}$ are assumed to be "white" in the sense that the noises at distinct time points are uncorrelated random variables. It is also assumed that the noise processes are uncorrelated with each other and with the initial state $\mathbf{x}_0$.

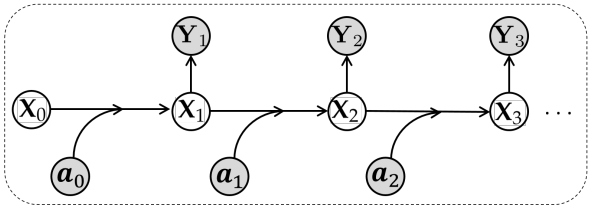


Figure 1: The schematic diagram of HMMs with external inputs.

B. Optimal Policy for Underlying HMM

HMMs with an external set of actions can be modeled through a partially observable Markov decision process. Here, the policy defined over the underlying Markov decision process (MDP) (without accounting for the partial state observability) is described. The underlying state process in HMM is a MDP, which can be represented by a 5-tuple $\langle \mathcal{X}, \mathcal{A}, \mathcal{T}, R, \gamma \rangle$, where $\mathcal{X}$ is the *state space*, $\mathcal{A}$ is the *action space*, $\mathcal{T} : \mathcal{X} \times \mathcal{A} \times \mathcal{X}$ is the *state transition probability function* such that $p(\mathbf{x}' \mid \mathbf{x}, \mathbf{a})$ represents the probability of moving to state $\mathbf{x}'$ after taking action $\mathbf{a}$ in state $\mathbf{x}$, $R : \mathcal{X} \times \mathcal{A} \times \mathcal{X} \to \mathbb{R}$ is a bounded *reward function* such that $R(\mathbf{x}, \mathbf{a}, \mathbf{x}')$ encodes the reward earned when action $\mathbf{a}$ in taken in state $\mathbf{x}$ and the system moves to state $\mathbf{x}'$, and $0 < \gamma < 1$ is a *discount factor*.

More formally, a deterministic stationary policy $\pi : \mathcal{X} \to \mathcal{A}$ for an MDP is a mapping from states to actions. The expected discounted reward function at state $\mathbf{x} \in \mathcal{X}$ after taking action $\mathbf{a} \in \mathcal{A}$ and the following policy π afterward is defined as:

$$Q^{\pi}(\mathbf{x}, \mathbf{a}) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(\mathbf{x}_t, \mathbf{a}_t, \mathbf{x}_{t+1}) \mid \mathbf{x}_0 = \mathbf{x}, \mathbf{a}_0 = \mathbf{a}, \mathbf{a}_{1:\infty} \sim \pi \right]. \tag{2}$$

According to (2), the expected return under the optimal policy π^* can be defined as:

$$Q^{\pi^*}(\mathbf{x}, \mathbf{a}) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(\mathbf{x}_t, \mathbf{a}_t, \mathbf{x}_{t+1}) \mid \mathbf{x}_0 = \mathbf{x}, \mathbf{a}_0 = \mathbf{a}, \mathbf{a}_{1:\infty} \sim \pi^* \right], \tag{3}$$

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

III. Formulation of Optimal Expert-Enabled Inference for HMMs

This section first describes the inference problem in HMMs, including the conventional inference approaches built on maximum likelihood and maximum a posteriori criteria. Then, the proposed approach for inference through expert-acquired data is described. A specific case of expert-enabled inference through data with directly observable states is also analyzed.

A. Conventional Inference in HMMs

The inference in HMMs consists of estimating the unknown parameters of the model in (1) represented by $\boldsymbol{\theta}$, using available data. Let $D_k = \{(\tilde{\mathbf{a}}_0,\tilde{\mathbf{y}}_1),(\tilde{\mathbf{a}}_1,\tilde{\mathbf{y}}_2),...,(\tilde{\mathbf{a}}_{k-1},\tilde{\mathbf{y}}_k)\}$ be the expert-acquired data up to time step k, where $\tilde{\mathbf{a}}_{r-1}$ is the taken action by an expert at time step $r-1$, and $\tilde{\mathbf{y}}_r$ represents the measurement at time step r. The conventional inference techniques aim to infer a model using the maximum likelihood (ML) and maximum a posteriori (MAP) criteria as:

$$\begin{aligned}\hat{\boldsymbol{\theta}}_k^{\mathrm{ML}} &= \operatorname*{argmax}_{\boldsymbol{\theta}\in\Theta} L_k^T(\boldsymbol{\theta}) \text{(Conventional ML Inference)},\\ \hat{\boldsymbol{\theta}}_k^{\mathrm{MAP}} &= \operatorname*{argmax}_{\boldsymbol{\theta}\in\Theta} \Big[\log p(\boldsymbol{\theta}) + L_k^T(\boldsymbol{\theta})\Big] \text{(Conventional MAP Inference)},\end{aligned} \tag{4}$$

where $p(\cdot)$ denotes either a probability density function or a probability mass function, $p(\boldsymbol{\theta})$ is the prior probability, and the log-likelihood value for model $\boldsymbol{\theta}$ can be represented as:

$$L_k^T(\boldsymbol{\theta}) \coloneqq \log p(\tilde{\mathbf{y}}_{1:k} \mid \tilde{\mathbf{a}}_{0:k-1},\boldsymbol{\theta}) = \underbrace{\sum_{r=1}^{k} \log p(\tilde{\mathbf{y}}_r \mid \tilde{\mathbf{y}}_{1:r-1},\tilde{\mathbf{a}}_{0:r-1},\boldsymbol{\theta})}_{\text{Transition Term}}. \tag{5}$$

The transition term in (5) measures the likelihood that the temporal changes in the data are associated with model $\boldsymbol{\theta}$. Clearly, the model with the highest (log)-likelihood or posterior is selected as the inferred model. While these methods are optimal techniques in discriminating models according to their temporal changes, they cannot take into the value expert knowledge carried in expert-acquired data for the inference purpose (i.e., $\tilde{\mathbf{a}}_{0:k-1}$). In the following paragraphs, the proposed expert-enabled inference method, which optimally incorporates expert knowledge, is described.

B. Expert-Enabled Inference in HMMs

This part describes the inference through expert-acquired data by incorporating expert knowledge. Let $R(.,.,.)$ represent the expert reward function, which denotes the reward associated with the main objectives from which data are acquired. For instance, if the actions are interventions made by biologists, the immediate expert reward function measures the improvement achieved at any given step during the intervention. Let π^* be the optimal policy given the expert reward function. Expert decisions/actions are modeled according to the following well-known ϵ-greedy policy [34]:

$$p(\mathbf{a} \mid \mathbf{x}, R) = \begin{cases} q + \frac{1-q}{|\mathcal{A}|} & \text{If } \mathbf{a} = \pi^*(\mathbf{x}) \\ \frac{1-q}{|\mathcal{A}|} & \text{If } \mathbf{a} \neq \pi^*(\mathbf{x}) \end{cases}, \text{ for } \mathbf{a}\in\mathcal{A}, \mathbf{x}\in\mathcal{X}, \tag{6}$$

where $0 \le q \le 1$ represents the confidence of the expert. Values of q close to 1 represent near-optimal experts, whereas the values close to 0 model random experts. Another popular way to model expert behavior is through the Boltzmann policy, which is also known as softmax policy expressed as [42]:

$$p(\mathbf{a} \mid \mathbf{x}, R) \propto \exp\big(\eta Q^*(\mathbf{x},\mathbf{a})\big) \text{ for } \mathbf{a}\in\mathcal{A}, \mathbf{x}\in\mathcal{X}, \tag{7}$$

where $\eta > 0$ represents the confidence of the expert; large values of η model confident experts, whereas smaller values model experts with more random/non-optimal decisions. It should be noted that both expert models in (6) and (7) are widely used in the inverse reinforcement learning context and can model the imperfect behavior of expert/human [34].

The optimal expert-enabled inference given the expert-acquired data $D_k = \{(\tilde{\mathbf{a}}_0,\tilde{\mathbf{y}}_1),(\tilde{\mathbf{a}}_1,\tilde{\mathbf{y}}_2),...,(\tilde{\mathbf{a}}_{k-1},\tilde{\mathbf{y}}_k)\}$ can be formulated as:

$$\begin{aligned}\hat{\boldsymbol{\theta}}_k^{\mathrm{EE-ML}} &= \operatorname*{argmax}_{\boldsymbol{\theta}\in\Theta} L_k^{\mathrm{EE}}(\boldsymbol{\theta}),\\ \hat{\boldsymbol{\theta}}_k^{\mathrm{EE-MAP}} &= \operatorname*{argmax}_{\boldsymbol{\theta}\in\Theta} \Big[\log p(\boldsymbol{\theta}) + L_k^{\mathrm{EE}}(\boldsymbol{\theta})\Big],\end{aligned} \tag{8}$$

where the expert-enabled log-likelihood $L_k^{\mathrm{EE}}(\boldsymbol{\theta})$ can be represented as:

$$\begin{aligned}L_k^{\mathrm{EE}}(\boldsymbol{\theta}) &\coloneqq \log p(D_k \mid \boldsymbol{\theta}, R)\\ &= \log p(\tilde{\mathbf{y}}_{1:k}, \tilde{\mathbf{a}}_{0:k-1} \mid \boldsymbol{\theta}, R)\\ &= \log \prod_{r=1}^{k} p(\tilde{\mathbf{y}}_r, \tilde{\mathbf{a}}_{r-1} \mid \tilde{\mathbf{a}}_{0:r-2}, \tilde{\mathbf{y}}_{1:r-1}, \boldsymbol{\theta}, R).\end{aligned} \tag{9}$$

Unlike the log-likelihood function in (5), the expert-enabled likelihood function in (9) considers the joint distribution of the measurements and the expert decisions. The last expression in (9) can be further simplified as:

$$\begin{aligned}L_k^{\mathrm{EE}}(\boldsymbol{\theta}) &= \log \prod_{r=1}^{k} \Big[p(\tilde{\mathbf{y}}_r \mid \tilde{\mathbf{a}}_{0:r-1}, \tilde{\mathbf{y}}_{1:r-1}, \boldsymbol{\theta}, R)\\ &\qquad \times p(\tilde{\mathbf{a}}_{r-1} \mid \tilde{\mathbf{a}}_{0:r-2}, \tilde{\mathbf{y}}_{1:r-1}, \boldsymbol{\theta}, R)\Big]\\ &= \underbrace{\sum_{r=1}^{k} \log p(\tilde{\mathbf{y}}_r \mid \tilde{\mathbf{y}}_{1:r-1}, \tilde{\mathbf{a}}_{0:r-1}, \boldsymbol{\theta})}_{L_k^T(\boldsymbol{\theta}) = \text{ Transition Term}}\\ &\qquad + \underbrace{\sum_{r=1}^{k} \log p(\tilde{\mathbf{a}}_{r-1} \mid \tilde{\mathbf{y}}_{1:r-1}, \tilde{\mathbf{a}}_{0:r-2}, \boldsymbol{\theta}, R)}_{L_k^E(\boldsymbol{\theta}) \coloneqq \text{ Expert-Knowledge Term}},\end{aligned} \tag{10}$$

where the expert reward function R is dropped in transition term due to the Markovian properties of the state process. Comparing the expert-enabled and the conventional log-likelihood functions in (5) and (10), one can see that the conventional inference techniques only consider the transition term for the inference process and ignore the expert-knowledge

term. The term expert-knowledge term includes the probability that the expert would take actions reflected in expert-acquired data if $\boldsymbol{\theta}$ was the expert's perception of the system. In general, expert actions would be different if they perceived the system as $\boldsymbol{\theta}^1$ or $\boldsymbol{\theta}^2$ ($\boldsymbol{\theta}^1 \neq \boldsymbol{\theta}^2$). Therefore, given a realization of the actions reflected in expert-acquired data, the goal is to incorporate the expert's perception of the system into the inference process.

A description of how incorporating the expert-knowledge term can help the inference process is described through an illustrative example below (more details are provided in Section V).

C. Directly Observable States: A Special Case

For the system with directly observable states, the transition and expert-enabled terms in (10) can be expressed as:

$$L_k^T(\boldsymbol{\theta}) \coloneqq \sum_{r=1}^{k} \log P(\tilde{\mathbf{x}}_r \mid \tilde{\mathbf{x}}_{r-1}, \tilde{\mathbf{a}}_{r-1}, \boldsymbol{\theta}),$$
$$L_k^E(\boldsymbol{\theta}) \coloneqq \sum_{r=1}^{k} \log p(\tilde{\mathbf{a}}_{r-1} \mid \tilde{\mathbf{x}}_{r-1}, \boldsymbol{\theta}, R),$$

where the measurements $\tilde{\mathbf{y}}_{1:k}$ are replaced by states, $\tilde{\mathbf{x}}_{1:k}$ and probabilities are simplified according to the Markovian properties of the underlying state process.

An Illustrative Example: The following simple system with a single unknown component θ is considered:

$$x_k = \overline{\theta x_{k-1}} \oplus a_{k-1}, \tag{11}$$

where $x \in \{0,1\}$, $\theta \in \{+1,-1\}$ and $a_{k-1} \in \mathcal{A} = \{a^1 = 0, a^2 = 1\}$ and $\overline{v}$ maps the value of an arbitrary scalar v greater than 0 to 1, and otherwise to 0. The possible models for this system are represented by $\theta^1 = +1$ and $\theta^2 = -1$, where $\theta^* = \theta^1$ represents the true system model. The expert objective is to keep the system at $x = 1$. The optimal model-specific policies can be expressed through $\pi^*_{\boldsymbol{\theta}^1}(x=0) = 1, \pi^*_{\boldsymbol{\theta}^1}(x=1) = 0$ and $\pi^*_{\boldsymbol{\theta}^2}(x=0) = 1, \pi^*_{\boldsymbol{\theta}^2}(x=1) = 1$. Assuming $\{\tilde{x}_0 = 0, \tilde{a}_0 = 1, \tilde{x}_1 = 1, \tilde{a}_1 = 0\}$ be the expert-acquired data from the optimal expert (i.e., $q = 1$ in (6) or $\eta = \infty$ in (7)), the expert-enabled log-likelihood value for two models can be expressed as:

$$L_1^{\mathrm{EE}}(\boldsymbol{\theta}^1) = \underbrace{\log P(\tilde{x}_1 = 1 \mid \tilde{x}_0 = 0, \tilde{a}_0 = 1, \boldsymbol{\theta}^1)}_{L_1^T(\boldsymbol{\theta}^1)=\log 1}$$
$$+ \underbrace{\log p(\tilde{a}_0 = 1 \mid \tilde{x}_0 = 0, \boldsymbol{\theta}^1, R) + \log p(\tilde{a}_1 = 0 \mid \tilde{x}_1 = 1, \boldsymbol{\theta}^1, R)}_{L_1^E(\boldsymbol{\theta}^1)=\log 1+\log 1} = 3 \log 1,$$

$$L_1^{\mathrm{EE}}(\boldsymbol{\theta}^2) = \underbrace{\log P(\tilde{x}_1 = 1 \mid \tilde{x}_0 = 0, \tilde{a}_0 = 1, \boldsymbol{\theta}^2)}_{L_1^T(\boldsymbol{\theta}^2)=\log 1}$$
$$+ \underbrace{\log p(\tilde{a}_0 = 1 \mid \tilde{x}_0 = 0, \boldsymbol{\theta}^2, R) + \log p(\tilde{a}_1 = 0 \mid \tilde{x}_1 = 1, \boldsymbol{\theta}^2, R)}_{L_1^E(\boldsymbol{\theta}^2)=\log 1+\log 0} = -\infty.$$

It can be seen that the transition terms for both models are the same; thus, these methods are not differentiable given the temporal changes reflected in the observed data. This is an example of a non-identifiability problem where two models have the same likelihood values according to their transition terms. However, the difference between the expert-knowledge terms decisively differentiates the two models.

IV. Matrix-Based Computations of the Expert-Enabled Inference

The matrix-based computations described below consist of two main steps: 1) updating the state posterior for calculating the transition term, and 2) modeling the expert for computing the expert-knowledge term. These steps aim to compute the two terms in the expert-enabled log-likelihood function in (10). Furthermore, a recursive solution for the computation of the expert-enabled likelihood and inference is provided in this section.

A. Matrix-Based Computation of Expert-Enabled Log-Likelihood Function

Efficient matrix-based computations of the transition and expert-knowledge terms, required for evaluating the expert-enabled log-likelihood function in (10), are described below.

1) Computation of transition term: Let $\{\mathbf{x}^1, \ldots, \mathbf{x}^n\}$ be an arbitrary enumeration of the possible states in $\mathcal{X}$. The following posterior and predictive posterior distributions of the states are defined at time step r:

$$\begin{aligned} \boldsymbol{\Pi}^{\boldsymbol{\theta}}_{r|r-1}(i) &= P\left(\mathbf{x}_r = \mathbf{x}^i \mid \mathbf{y}_{1:r-1}, \mathbf{a}_{0:r-1}, \boldsymbol{\theta}\right), \\ \boldsymbol{\Pi}^{\boldsymbol{\theta}}_{r|r}(i) &= P\left(\mathbf{x}_r = \mathbf{x}^i \mid \mathbf{y}_{1:r}, \mathbf{a}_{0:r-1}, \boldsymbol{\theta}\right), \end{aligned} \tag{12}$$

for $r = 1, \ldots$ and $i = 1, \ldots, n$; where $\boldsymbol{\Pi}^{\boldsymbol{\theta}}_{0|0}(i) = P(\mathbf{x}_0 = \mathbf{x}^i \mid \boldsymbol{\theta})$, for $i = 1, \ldots, n$, is the initial states distribution. If no knowledge about the initial state is available, a uniform prior can be considered, i.e., $\boldsymbol{\Pi}^{\boldsymbol{\theta}}_{0|0}(i) = 1/n$, for $i = 1, \ldots, n$ and $\boldsymbol{\theta} \in \Theta$.

We define the *controlled transition matrix* of size $n \times n$ associated with a model under action $\mathbf{a}$ parameterized by $\boldsymbol{\theta}$ as:

$$(M_{\boldsymbol{\theta}}(\mathbf{a}))_{ij} = P\left(\mathbf{x}_r = \mathbf{x}^i \mid \mathbf{x}_{r-1} = \mathbf{x}^j, \mathbf{a}_{r-1} = \mathbf{a}, \boldsymbol{\theta}\right), \tag{13}$$

for $i, j = 1, \ldots, n$.

Additionally, given a value of the observation vector $\mathbf{y}_r$ at time r, the *update vector* $T_{\boldsymbol{\theta}}(\mathbf{y}_r)$ of size n associated with model $\boldsymbol{\theta}$ is defined by:

$$(T_{\boldsymbol{\theta}}(\mathbf{y}_r))_i = p\left(\mathbf{y}_r \mid \mathbf{X}_r = \mathbf{x}^i, \boldsymbol{\theta}\right), \tag{14}$$

for $i = 1, \ldots, n$.

Let $\boldsymbol{\Pi}^{\boldsymbol{\theta}}_{r-1|r-1}$ be the posterior distribution of state using the expert-acquired data $(\tilde{\mathbf{a}}_{0:r-2}, \tilde{\mathbf{y}}_{1:r-1})$. Given $\tilde{\mathbf{a}}_{r-1}$ be the expert action at time step $r-1$, the predictive posterior distribution of states at time step r can be expressed as:

$$\boldsymbol{\Pi}^{\boldsymbol{\theta}}_{r|r-1} = M_{\boldsymbol{\theta}}(\tilde{\mathbf{a}}_{r-1})\, \boldsymbol{\Pi}^{\boldsymbol{\theta}}_{r-1|r-1}. \tag{15}$$

Upon observing the measurement at time step r, $\tilde{\mathbf{y}}_t$, the posterior distribution of state $\boldsymbol{\Pi}^{\boldsymbol{\theta}}_{r|r}$, can be achieved using the following Bayesian recursion [43]:

$$\boldsymbol{\Pi}^{\boldsymbol{\theta}}_{r|r} = \frac{T_{\boldsymbol{\theta}}(\tilde{\mathbf{y}}_r) \circ \boldsymbol{\Pi}^{\boldsymbol{\theta}}_{r|r-1}}{||T_{\boldsymbol{\theta}}(\tilde{\mathbf{y}}_r) \circ \boldsymbol{\Pi}^{\boldsymbol{\theta}}_{r|r-1}||_1}. \tag{16}$$

where "$\circ$" denotes Hadamard product (i.e., the component-wise multiplication of two vectors). The recursive procedure in (16) consists of the conditional probability of the last measurement given the system state denoted by $T_{\boldsymbol{\theta}}(\tilde{\mathbf{y}}_r)$, and the predictive posterior of system state $\boldsymbol{\Pi}^{\boldsymbol{\theta}}_{r|r-1}$ given the information available up to time step $r-1$.

The transition term in (10) associated with model $\boldsymbol{\theta}$ can be expressed in terms of the predictive posterior distribution as:

$$\begin{aligned} L^T_k(\boldsymbol{\theta}) &= \sum_{r=1}^{k} \log p(\tilde{\mathbf{y}}_r \mid \tilde{\mathbf{y}}_{1:r-1}, \tilde{\mathbf{a}}_{0:r-1}, \boldsymbol{\theta}) \\ &= \sum_{r=1}^{k} \log \left[\sum_{j=1}^{n} p(\tilde{\mathbf{y}}_r \mid \mathbf{x}_r = \mathbf{x}^j, \boldsymbol{\theta}) P(\mathbf{x}_r = \mathbf{x}^j \mid \tilde{\mathbf{y}}_{1:r-1}, \tilde{\mathbf{a}}_{0:r-1}, \boldsymbol{\theta}) \right] \\ &= \sum_{r=1}^{k} \log \left[\sum_{j=1}^{n} (T_{\boldsymbol{\theta}}(\tilde{\mathbf{y}}_r))_j \, \boldsymbol{\Pi}^{\boldsymbol{\theta}}_{r|r-1}(j) \right] \\ &= \sum_{r=1}^{k} \log \left[\| T_{\boldsymbol{\theta}}(\tilde{\mathbf{y}}_r) \circ \boldsymbol{\Pi}^{\boldsymbol{\theta}}_{r|r-1} \|_1 \right]. \end{aligned} \tag{17}$$

where the predictive posterior can be computed according to the recursion in (15)-(16).

2) Computation of Expert-Knowledge Term: The expert-knowledge term in (10) can be further expanded for a given $\boldsymbol{\theta} \in \Theta$ as:

$$\begin{aligned} L^E_k(\boldsymbol{\theta}) &= \sum_{r=1}^{k} \log p(\tilde{\mathbf{a}}_{r-1} \mid \tilde{\mathbf{a}}_{0:r-2}, \tilde{\mathbf{y}}_{1:r-1}, \boldsymbol{\theta}) \\ &= \sum_{r=1}^{k} \log \left[\sum_{i=1}^{n} p(\tilde{\mathbf{a}}_{r-1} \mid \mathbf{x}_{r-1} = \mathbf{x}^i, \boldsymbol{\theta}) \right. \\ &\qquad \left. \times P(\mathbf{x}_{r-1} = \mathbf{x}^i \mid \tilde{\mathbf{a}}_{0:r-2}, \tilde{\mathbf{y}}_{1:r-1}, \boldsymbol{\theta}) \right] \\ &= \sum_{r=1}^{k} \log \left[\sum_{i=1}^{n} p(\tilde{\mathbf{a}}_{r-1} \mid \mathbf{x}_{r-1} = \mathbf{x}^i, \boldsymbol{\theta}) \, \boldsymbol{\Pi}^{\boldsymbol{\theta}}_{r-1|r-1}(i) \right], \end{aligned} \tag{18}$$

where the computation of the last term can be achieved according to the expert behavior model in (6) and (7).

Let $\pi^*_{\boldsymbol{\theta}}$ be the optimal model-specific policy associated with the model parameterized by $\boldsymbol{\theta}$. The expert-knowledge term in (18) according to the ϵ-greedy policy can be written as:

$$\begin{aligned} L^E_k(\boldsymbol{\theta}) = \sum_{r=1}^{k} \log \left[\sum_{i=1}^{n} \left(\left(q + \frac{1-q}{|\mathcal{A}|} \right) 1_{\tilde{\mathbf{a}}_{r-1} = \pi^*_{\boldsymbol{\theta}}(\mathbf{x}^i)} \right. \right. \\ \left. \left. + \frac{1-q}{|\mathcal{A}|} 1_{\tilde{\mathbf{a}}_{r-1} \neq \pi^*_{\boldsymbol{\theta}}(\mathbf{x}^i)} \right) \boldsymbol{\Pi}^{\boldsymbol{\theta}}_{r-1|r-1}(i) \right]. \end{aligned} \tag{19}$$

Let also $Q^*_{\boldsymbol{\theta}}(.,.)$ be the optimal model-specific Q-function for the model parameterized by $\boldsymbol{\theta}$. The expert knowledge term for the softmax policy can also be expressed as:

$$L^E_k(\boldsymbol{\theta}) = \sum_{r=1}^{k} \log \left[\sum_{i=1}^{n} \frac{\exp\left(\eta \, Q^*_{\boldsymbol{\theta}}(\mathbf{x}^i, \tilde{\mathbf{a}}_{r-1})\right)}{\sum_{\mathbf{a} \in \mathcal{A}} \exp\left(\eta \, Q^*_{\boldsymbol{\theta}}(\mathbf{x}^i, \mathbf{a})\right)} \boldsymbol{\Pi}^{\boldsymbol{\theta}}_{r-1|r-1}(i) \right]. \tag{20}$$

Note that the computation of the optimal model-specific Q-function can be achieved through a dynamic programming method [44] for relatively small and finite state and action spaces, and approximate dynamic programming or reinforcement learning techniques [45] for large and continuous spaces.

This part briefly describes the value iteration method [44] for the computation of the model-specific policies and Q-functions in finite state and action spaces. The matrix-form representation of the expert reward function, $R(.,.,.)$, for a given action, $\mathbf{a} \in \mathcal{A}$ can be expressed as:

$$(\mathbf{R}(\mathbf{a}))_{ij} = R\left(\mathbf{x}^j, \mathbf{a}, \mathbf{x}^i\right), \text{ for } i, j = 1, .., n. \tag{21}$$

The Bellman operator for model $\boldsymbol{\theta}$ in a matrix-form is defined as [44]:

$$\mathcal{T}^*[\mathbf{V}_{\boldsymbol{\theta}}] = \max_{\mathbf{a} \in \mathcal{A}} \left[(\mathbf{R}(\mathbf{a}) \odot M_{\boldsymbol{\theta}}(\mathbf{a}))^T \mathbf{1}_{n \times 1} + \gamma M^T_{\boldsymbol{\theta}}(\mathbf{a}) \, \mathbf{V}_{\boldsymbol{\theta}} \right], \tag{22}$$

where "max" is applied row-wise, $\mathbf{1}_{n \times 1}$ is a vector of size n with all elements 1, and "$\odot$" denotes the component-wise multiplication of two matrices. Note that the Bellman operator in (22) depends on the model $\boldsymbol{\theta}$, and the row-wise maximum is over possible actions in the action space.

The optimal state value function $\mathbf{V}^*_{\boldsymbol{\theta}}$ for any given model $\boldsymbol{\theta}$ can be obtained by starting with an arbitrary state value vector $\mathbf{V}^0_{\boldsymbol{\theta}}$ (e.g., $\mathbf{V}^0_{\boldsymbol{\theta}} = [0, \cdots, 0]^T$) and recursively applying the Bellman operator to all elements of this vector. This can be expressed in a vectored form as $\mathbf{V}^t_{\boldsymbol{\theta}} = \mathcal{T}^*[\mathbf{V}^{t-1}_{\boldsymbol{\theta}}]$, where the process is guaranteed to converge to the optimal state value function, which is a unique fixed-point solution for Bellman operator. The iterative process of performing the Bellman operator can stop when the maximum difference between elements of value vectors in two consecutive iterations falls below a small prespecified threshold, i.e., $\|\mathbf{V}^{t-1}_{\boldsymbol{\theta}} - \mathbf{V}^t_{\boldsymbol{\theta}}\|_\infty < \lambda$.

The optimal model-specific Q-function can be obtained according to $\mathbf{V}^*_{\boldsymbol{\theta}}$ as:

$$\begin{bmatrix} Q^*_{\boldsymbol{\theta}}(\mathbf{x}^1, \mathbf{a}) \\ \vdots \\ Q^*_{\boldsymbol{\theta}}(\mathbf{x}^n, \mathbf{a}) \end{bmatrix} = (\mathbf{R}(\mathbf{a}) \odot M_{\boldsymbol{\theta}}(\mathbf{a}))^T \mathbf{1}_{n \times 1} + \gamma M^T_{\boldsymbol{\theta}}(\mathbf{a}) \mathbf{V}^*_{\boldsymbol{\theta}}, \tag{23}$$

for $\mathbf{a} \in \mathcal{A}$. The optimal model-specific policy can be computed as $\pi^*_{\boldsymbol{\theta}}(\mathbf{x}^i) = \text{argmax}_{\mathbf{a} \in \mathcal{A}} Q^*_{\boldsymbol{\theta}}(\mathbf{x}^i, \mathbf{a})$, for $i = 1, ..., n$. Replacing $\pi^*_{\boldsymbol{\theta}}$ or $Q^*_{\boldsymbol{\theta}}$ into (19) or (20) leads to the exact computation of the expert-knowledge term.

The transition and expert-knowledge terms computed in (17) and (19) or (20) can be used for expert-enabled inference in HMMs with arbitrary parameter space. Each expert-enabled log-likelihood evaluation requires employing a dynamic programming technique and a forward Bayesian recursion. As of the conventional inference methods, inference through batch expert-acquired data demands searching over the parameter space and selecting the model with the largest expert-enabled log-likelihood or log-posterior values.

B. Expert-Enabled Inference under Unknown Expert Confidence

The expert-knowledge terms in (19) and (20) require knowledge about the expert's confidence, denoted by the parameters q or η. In practice, the expert's confidence may be unknown, and thus one needs to take this into account during the expert-enabled inference process. In such scenarios, the expert's confidence can be treated as an additional parameter and be inferred alongside other model parameters during the inference

process. Given the ϵ-greedy model of the expert behavior with unknown confidence, the joint inference of the model and the expert's confidence can be achieved as:

$$
\begin{aligned}
(q_k^{\text{EE-ML}}, \boldsymbol{\theta}_k^{\text{EE-ML}}) &= \underset{\boldsymbol{\theta}\in\Theta \& q\in[0\,1]}{\operatorname{argmax}} \left[L_k^{\mathrm{T}}(\boldsymbol{\theta}) + L_k^{\mathrm{E}}(\boldsymbol{\theta})\right] \\
&= \underset{\boldsymbol{\theta}\in\Theta \& q\in[0\,1]}{\operatorname{argmax}} \Bigg(\sum_{r=1}^{k} \log\left[\|T_{\boldsymbol{\theta}}(\tilde{\mathbf{y}}_r) \circ \boldsymbol{\Pi}_{r|r-1}^{\boldsymbol{\theta}}\|_1\right] \\
&\qquad + \sum_{r=1}^{k} \log\Bigg[\sum_{i=1}^{n} \Bigg(\left(q + \frac{1-q}{|\mathcal{A}|}\right) 1_{\tilde{\mathbf{a}}_{r-1}=\pi_{\boldsymbol{\theta}}^*(\mathbf{x}^i)} \\
&\qquad + \frac{1-q}{|\mathcal{A}|} 1_{\tilde{\mathbf{a}}_{r-1}\neq\pi_{\boldsymbol{\theta}}^*(\mathbf{x}^i)} \Bigg) \boldsymbol{\Pi}_{r-1|r-1}^{\boldsymbol{\theta}}(i) \Bigg] \Bigg),
\end{aligned}
\tag{24}
$$

where the first term, $L_k^T(\theta)$, includes only the model parameter, while the expert knowledge term, $L_k^E(\theta)$, includes both the model parameters and expert confidence. Note that the main difference in expert-enabled inference with unknown and known expert confidence is the addition of a scalar parameter representing the expert confidence, which should be optimized alongside the model parameter $\boldsymbol{\theta}$. The optimization can be achieved by quantization of the q value or other optimization algorithms suited for continuous variables. The inclusion of this single parameter in the optimization process is expected to be less impactful in the accuracy of inference in domains with large parameters.

C. Optimal Recursive Expert-Enabled Inference for HMMs with Finite Parameter Space

In this section, for HMMs with finite parameter space, the exact recursive computation of the expert-enabled inference is derived. The finite parameter space is common in many domains modeled by HMMs, such as network systems where the unknown parameters are often the possible topologies for the network. Meanwhile, quantization of the parameter space is a standard approach to achieving a finite parameter space in practice.

A recursive formulation of the expert-enabled log-likelihood function in (9) can be expressed as:

$$
\begin{aligned}
L_k^{\text{EE}}(\boldsymbol{\theta}) &= \sum_{r=1}^{k} \log p(\tilde{\mathbf{y}}_r \mid \tilde{\mathbf{a}}_{0:r-1}, \tilde{\mathbf{y}}_{1:r-1}, \boldsymbol{\theta}) \\
&\qquad + \sum_{r=1}^{k} \log p(\tilde{\mathbf{a}}_{r-1} \mid \tilde{\mathbf{a}}_{0:r-2}, \tilde{\mathbf{y}}_{1:r-1}, \boldsymbol{\theta}) \\
&= L_{k-1}^{\text{EE}}(\boldsymbol{\theta}) + \underbrace{\log p(\tilde{\mathbf{y}}_k \mid \tilde{\mathbf{y}}_{1:k-1}, \tilde{\mathbf{a}}_{0:k-1}, \boldsymbol{\theta})}_{l_k^T(\boldsymbol{\theta}):=\text{Transition Increment}} \\
&\qquad + \underbrace{\log p(\tilde{\mathbf{a}}_{k-1} \mid \tilde{\mathbf{y}}_{1:k-1}, \tilde{\mathbf{a}}_{0:k-2}, \boldsymbol{\theta})}_{l_k^E(\boldsymbol{\theta}):=\text{Expert-Knowledge Increment}},
\end{aligned}
\tag{25}
$$

where $L_{k-1}^{\text{EE}}(\boldsymbol{\theta})$ is the previous expert-enabled log-likelihood function, and the transition and the expert knowledge increments represent the addition to the log-likelihood value after observing the last expert-acquired data, i.e., $(\tilde{\mathbf{a}}_{k-1}, \tilde{\mathbf{y}}_k)$.

The transition and the expert-knowledge increments can be expressed as:

$$
\begin{aligned}
l_k^T(\boldsymbol{\theta}) &= \log \|T_{\boldsymbol{\theta}}(\tilde{\mathbf{y}}_k) \circ \boldsymbol{\Pi}_{k|k-1}^{\boldsymbol{\theta}}\|_1, \\
l_k^E(\boldsymbol{\theta}) &= \log\left[\sum_{i=1}^{n} \frac{\exp\left(\eta\, Q_{\boldsymbol{\theta}}^*(\mathbf{x}^i, \tilde{\mathbf{a}}_{k-1})\right)}{\sum_{\mathbf{a}\in\mathcal{A}} \exp\left(\eta\, Q_{\boldsymbol{\theta}}^*(\mathbf{x}^i, \mathbf{a})\right)} \boldsymbol{\Pi}_{k-1|k-1}^{\boldsymbol{\theta}}(i)\right] \quad \text{softmax} \\
&= \log\Bigg[\sum_{i=1}^{n} \Bigg(\left(q + \frac{1-q}{|\mathcal{A}|}\right) 1_{\tilde{\mathbf{a}}_{k-1}=\pi_{\boldsymbol{\theta}}^*(\mathbf{x}^i)} + \\
&\qquad \frac{1-q}{|\mathcal{A}|} 1_{\tilde{\mathbf{a}}_{k-1}\neq\pi_{\boldsymbol{\theta}}^*(\mathbf{x}^i)} \Bigg) \boldsymbol{\Pi}_{k-1|k-1}^{\boldsymbol{\theta}}(i) \Bigg] \quad \epsilon\text{-greedy}.
\end{aligned}
\tag{26}
$$

Given a finite parameter space Θ, the proposed method divides the expert-enabled inference process into offline and online steps. In the offline step, M dynamic programming techniques are run in parallel, each tuned to a specific model/parameter, $\{\boldsymbol{\theta}^1, ..., \boldsymbol{\theta}^M\} = \Theta$. The outcomes of the offline step are the optimal model-specific Q-values, i.e., $Q_{\boldsymbol{\theta}}^*(.,.)$, for $\boldsymbol{\theta} \in \Theta$. Note that the offline step contains the major computational costs of the proposed method. In the online step, upon observing any new expert-acquired data, the optimal recursive expert-enabled MAP inference is updated. The expert-enabled log-likelihood values get updated according to the transition and expert-knowledge increments for all models. The process is fully recursive; as new expert-acquired data at time $k+1$ arrives, the expert-enabled inference can be updated by only computing the transition and expert-knowledge increments in (26). The procedure is summarized in Algorithm 1.

Algorithm 1 The proposed Optimal Expert-Enabled Inference for Hidden Markov Models with Finite Parameter Space

Parameter space $\Theta = \{\boldsymbol{\theta}^1, ..., \boldsymbol{\theta}^M\}$; known expert reward $(\mathbf{R}(\mathbf{a}))_{ij} = R(\mathbf{x}^j, \mathbf{a}, \mathbf{x}^i)$; transition matrices $M^{\boldsymbol{\theta}}(\mathbf{a})$; known expert confidence q; Value Iteration threshold $\lambda > 0$.

Offline Step: Parallel Dynamic Programming Methods

for $\boldsymbol{\theta} \in \Theta$ **do**
  Run dynamic programming using (22) and (23) to obtain $\pi_{\boldsymbol{\theta}}^*$
end for

Online Step: Recursive Expert-Enabled MAP Inference

$L_0(\boldsymbol{\theta}) = 0$, set $\boldsymbol{\Pi}_{0|0}^{\boldsymbol{\theta}}$, for $\boldsymbol{\theta} \in \Theta$.

for $k = 1, 2, ...$ **do** /* Upon receiving new data $(\tilde{\mathbf{a}}_{k-1}, \tilde{\mathbf{y}}_k)$ */

  for $\boldsymbol{\theta} \in \Theta$ **do**

    Prediction: $\boldsymbol{\Pi}_{k|k-1}^{\boldsymbol{\theta}} = M_{\boldsymbol{\theta}}(\tilde{\mathbf{a}}_{k-1})\, \boldsymbol{\Pi}_{k-1|k-1}^{\boldsymbol{\theta}}$

    Unnormalized Posterior update: $\boldsymbol{\beta}_k^{\boldsymbol{\theta}} = T_{\boldsymbol{\theta}}(\tilde{\mathbf{y}}_k) \circ \boldsymbol{\Pi}_{k|k-1}^{\boldsymbol{\theta}}$

    Posterior Update: $\boldsymbol{\Pi}_{k|k}^{\boldsymbol{\theta}} = \boldsymbol{\beta}_k^{\boldsymbol{\theta}} / \|\boldsymbol{\beta}_k^{\boldsymbol{\theta}}\|_1$.

    Compute $l_k^T(\boldsymbol{\theta})$ and $l_k^E(\boldsymbol{\theta})$ using (26):

    Expert-Enabled Likelihood:

$$L_k^{\text{EE}}(\boldsymbol{\theta}) = L_{k-1}^{\text{EE}}(\boldsymbol{\theta}) + l_k^T(\boldsymbol{\theta}) + l_k^E(\boldsymbol{\theta}).$$

  end for

  Optimal Expert-Enabled ML Inference:

$$\hat{\boldsymbol{\theta}}_k^{\text{EE-ML}} = \underset{\boldsymbol{\theta}\in\Theta}{\operatorname{argmax}} \log L_k^{\text{EE}}(\boldsymbol{\theta}).$$

end for

V. Discussions and Analysis

In this section, the impact of the incorporation of expert knowledge in the inference process is examined. The conventional ML inference and the expert-enabled ML inference for HMMs can be expressed according to (4) and (8) as:

$$\begin{aligned}\hat{\boldsymbol{\theta}}_k^{\mathrm{ML}} &= \underset{\boldsymbol{\theta}\in\Theta}{\operatorname{argmax}}\, L_k^T(\boldsymbol{\theta}),\\ \hat{\boldsymbol{\theta}}_k^{\mathrm{EE-ML}} &= \underset{\boldsymbol{\theta}\in\Theta}{\operatorname{argmax}} \Big[L_k^T(\boldsymbol{\theta}) + L_k^E(\boldsymbol{\theta})\Big].\end{aligned} \tag{27}$$

In logarithmic form, the expert-knowledge term, $L_k^E(\boldsymbol{\theta})$, can be seen as an additional term that appears in expert-enabled ML inference. The goal is to analyze the impact of the expert knowledge term on the deviation of optimization solutions in (27). Consider the expert knowledge term $L_k^E(\boldsymbol{\theta})$, derived in (19):

$$\begin{aligned}L_k^E(\boldsymbol{\theta}) = \sum_{r=1}^{k} \log\Bigg[\sum_{i=1}^{n}\Bigg(&\underbrace{(q+\frac{1-q}{|\mathcal{A}|})1_{\tilde{\mathbf{a}}_{r-1}=\pi^*_{\boldsymbol{\theta}}(\mathbf{x}^i)}}_{\text{optimal model-specific term}}\\ &+\underbrace{\frac{1-q}{|\mathcal{A}|}1_{\tilde{\mathbf{a}}_{r-1}\neq\pi^*_{\boldsymbol{\theta}}(\mathbf{x}^i)}}_{\text{non-optimal model-specific term}}\Bigg)\boldsymbol{\Pi}^{\boldsymbol{\theta}}_{r-1|r-1}(i)\Bigg].\end{aligned} \tag{28}$$

The optimal model-specific term takes $(q+\frac{1-q}{|\mathcal{A}|})$ if the expert action is matched with the actions prescribed by the optimal model-specific policy for model $\boldsymbol{\theta}$, otherwise, it takes $(1-q)/|\mathcal{A}|$, as indicated by the non-optimal model-specific term in (19). For extremely confident expert modeled by $q \approx 1$, the expert-knowledge term can be simplified as:

$$L_k^E(\boldsymbol{\theta}) = \sum_{r=1}^{k} \log\Bigg(\sum_{i=1}^{n} 1_{\tilde{\mathbf{a}}_{r-1}=\pi^*_{\boldsymbol{\theta}}(\mathbf{x}^i)}\,\boldsymbol{\Pi}^{\boldsymbol{\theta}}_{r-1|r-1}(i)\Bigg), \tag{29}$$

which only includes the optimal model-specific term. Thus, for a confidence rate close to 1, the optimal-model-specific term plays a larger role than non-optimal model-specific terms, as experts are expected to make more optimal decisions.

For experts with behavior close to random decision makers, the expert behavior can be modeled using q close to 0. The expert-knowledge term in (19) can be expressed for these cases as:

$$L_k^E(\boldsymbol{\theta}) = \sum_{r=1}^{k} \log \sum_{i=1}^{n}\Bigg[\frac{1}{|\mathcal{A}|}\,\boldsymbol{\Pi}^{\boldsymbol{\theta}}_{r-1|r-1}(i)\Bigg] = k\log\frac{1}{|\mathcal{A}|}. \tag{30}$$

The expert knowledge, in this case, is independent of $\boldsymbol{\theta}$, as it takes a fixed value for all models $\boldsymbol{\theta} \in \Theta$, and can be perceived as a constant addition to the transition terms of all models. Thus, the expert-enabled inference and regular inference in (27) become the same under random experts.

Using the expert knowledge term in (19), given the large available data, the largest value for $L_k^E(\boldsymbol{\theta})$ should correspond to $\boldsymbol{\theta} = \boldsymbol{\theta}^*$ and when the posterior is peaked over the true state as:

$$\boldsymbol{\Pi}^{\boldsymbol{\theta}}_{r-1|r-1}(i) = \begin{cases}1 & \mathbf{x}^i = \mathbf{x}_{r-1}\\ 0 & \text{otherwise}\end{cases}, i = 1,\ldots,n. \tag{31}$$

In practice, the posterior distribution of the true model is expected to be peaked around the true state, especially in domains with small measurement noise and a long history of observations. For models other than the true model, the posterior distribution is generally not expected to be peaked around the true state, as observed data from $\boldsymbol{\theta}^*$ might not match the expected trajectory for models $\boldsymbol{\theta} \neq \boldsymbol{\theta}^*$.

For those models that are not distinguishable according to their transition terms, the incorporation of expert knowledge helps them to be differentiable in general. If the goal is to infer the true underlying system model, the incorporation of knowledge from a biased expert could lead to a biased inference. However, if the goal is to infer the system model that represents the expert's perception of systems, then the proposed method can be directly used to infer such models. For instance, inferring (inaccurate or biased) biologists' perception of systems helps them to test new hypotheses or design new experiments to confirm or reject their perception.

VI. Numerical Experiments

This section analyzes the performance of the proposed expert-enabled inference method using three problems, a benchmark, and two biological systems. ϵ-greedy policy with parameter q is used through numerical experiments for modeling the expert behavior. The ML and MAP are used as inference criteria, and the results of the proposed expert-enabled approach are compared with the optimal conventional ML and MAP inference techniques. The following three metrics are used for comparison purposes: 1) true inference rate $1_{\hat{\boldsymbol{\theta}}_k=\boldsymbol{\theta}^*}$, which takes on a value of 1 if the inferred model $\hat{\boldsymbol{\theta}}_k$ at time step k matches the true model $\boldsymbol{\theta}^*$ and 0 otherwise; 2) inference error $\|\hat{\boldsymbol{\theta}}_k - \boldsymbol{\theta}^*\|_1$, which measures the absolute sum of errors between the inferred model and the true model; 3) maximum posterior probability $\max_{\boldsymbol{\theta}\in\Theta} p(\boldsymbol{\theta} \mid D_k)$, which denotes the largest posterior probability of models. All average results presented through the numerical experiments include the standard error of the mean to better analyze the consistency and robustness of the proposed method.

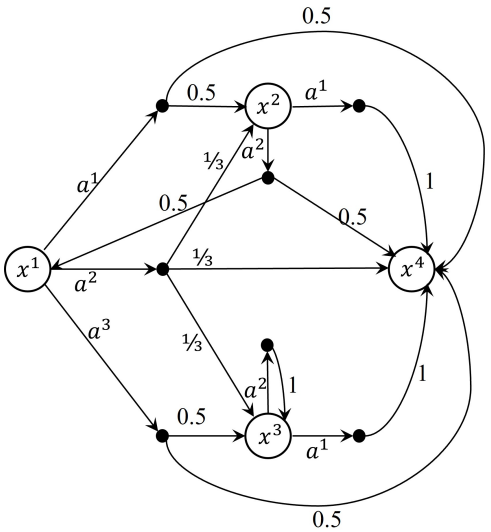


Figure 2: Benchmark finite state MDP problem

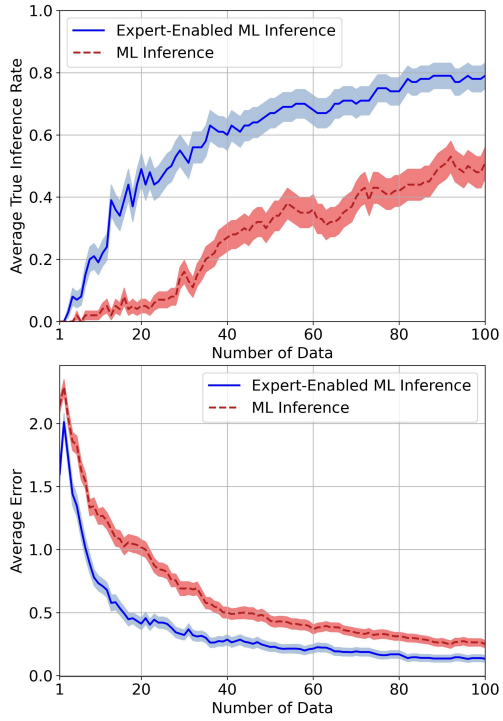


Figure 3: Average true inference and error obtained by proposed expert-enabled ML and regular ML methods for the benchmark problem.

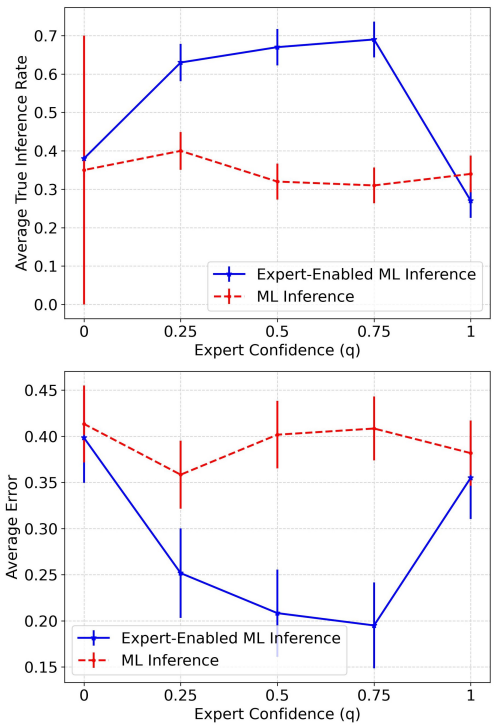


Figure 4: Average inference rate and error with respect to the confidence rate utilized by the proposed method for the benchmark.

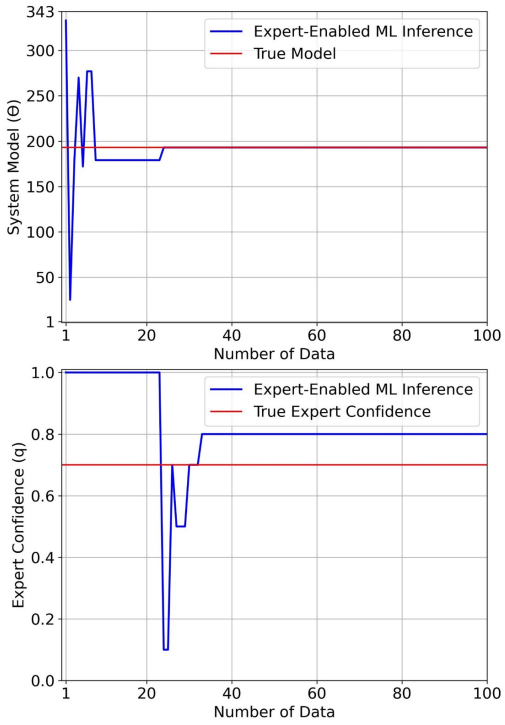


Figure 5: Inferred model and expert's confidence obtained by the proposed method for the benchmark problem.

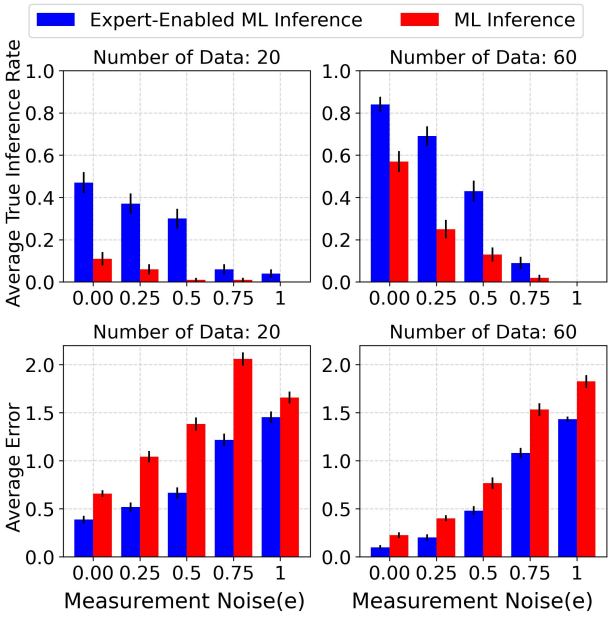


Figure 6: Average inference rate and error with respect to Measurement noise(e) in the benchmark problem.

A. A Benchmark HMM Problem

The first experiment is conducted using a benchmark finite-state HMM [46]. The system contains four states denoted by $\mathcal{X} = \{x^1, x^2, x^3, x^4\}$. The state transition probabilities are shown by arrows in Fig. 2. The transition probabilities from state x^1 to other states is assumed to be unknown, which can

be expressed using the following parameter vector:

$$\boldsymbol{\theta} = [(r^1_{12}, r^1_{13}, r^1_{14}), (r^2_{12}, r^2_{13}, r^2_{14}), (r^3_{12}, r^3_{13}, r^3_{14})] \quad (32)$$

where $r^i_{1j} = p(x_k = x^j \mid x_{k-1} = x^1, a_{k-1} = a^i)$.

Each element of unknown parameter vector r^i_{1j} is assumed to take its true value from $\{0, \frac{1}{2}, \frac{1}{3}, 1\}$, subject to $r^i_{12} + r^i_{13} + r^i_{14} = 1$, for $i = 1, 2, 3$. This leads to 343 possible models for the system represented by $\Theta = \{\boldsymbol{\theta}^1, ..., \boldsymbol{\theta}^{343}\}$. The true system model shown in Fig. 2 can be expressed using $\boldsymbol{\theta}^* = [(\frac{1}{2}, 0, \frac{1}{2}), (\frac{1}{3}, \frac{1}{3}, \frac{1}{3}), (0, \frac{1}{2}, \frac{1}{2})]$. The system state is assumed to be partially observable through the following measurement process:

$$y_k = \begin{cases} x_k & \text{with probability } 1 - e \\ x' \in \{\mathcal{X} - x_k\} & \text{with probability } e \end{cases}, \quad (33)$$

where x_k is the underlying true state at time step k, which is observed correctly with probability $1 - e$ and observed wrongly with probability e. Therefore, $0 \leq e \leq 1$ represents the measurement noise, where $e = 0$ is a special case with direct state observability.

We assume the expert objective is to guide the system to move to state $\mathbf{x}^4$ as early as possible and stay in that state. This can be presented using the following expert reward function:

$$R(x, a, x') = -1 + 5\,1_{x'=x^4}, \text{ for } x, x' \in \{x^1, x^2, x^3, x^4\}, a \in \mathcal{A}, \quad (34)$$

where the indicator $1_{x'=x^4}$ assigns a reward of 5 for observing state x^4 and -1 for taking any action. Using $\gamma = 0.95$ and $\lambda = 0.01$ in the offline step of Algorithm 1, the optimal expert policy used for generating expert-acquired data is as follows: $\pi^*_{\boldsymbol{\theta}^*}(x^1) = a^1$ or $a^3, \pi^*_{\boldsymbol{\theta}^*}(x^2) = a^1, \pi^*_{\boldsymbol{\theta}^*}(x^3) = a^1$. Note that the confidence rate q represents the probability that the expert follows the optimal policy compared to non-optimal actions.

The ML is considered as a criterion for inference in this case. The expert confidence is assumed to be $q = 0.75$ and measurement noise to be $e = 0.25$ for generating the expert-acquired data. Fig. 3 represents the average accuracy of inference obtained by the proposed expert-enabled ML inference, compared to the conventional ML inference method. It is evident that the proposed method achieves a much higher inference rate. As more data are observed, the average true inference rates for both methods increase. The superior performance of the proposed method can also be seen in terms of a smaller average error, indicating the importance of incorporating expert knowledge.

The impact of utilizing the wrong expert's confidence on the performance of the proposed method is investigated by generating 100 datasets with a length of 100 using experts with a confidence rate of $q = 0.75$. The average results of the proposed method with different expert confidence levels (true and untrue) are shown in Fig. 4. As expected, the highest inference rate and the minimum average error are obtained for $q = 0.75$, which matches the confidence in the data. The expert-enabled ML inference significantly outperforms the ML inference when the true expert confidence rate is utilized. The performance of the proposed method decreases as the confidence rate used in the method becomes further from the true confidence rate. However, in almost all cases, the expert-enabled ML inference outperforms the regular ML inference, demonstrating the proposed method's non-sensitivity to the exact expert confidence rate. Note that for the extreme case of an optimal expert with $q = 1$, the results are significantly impacted. This is because of the deterministic representation of the expert knowledge in this case, which rules out models with a single mismatch from what the optimal expert is expected to take under those models. Therefore, a rough approximation of the expert confidence can often provide good inference performance.

To analyze how to deal with domains where there is no knowledge about expert confidence, we use optimization in equation (24) for joint inference of parameters and expert confidence. Fig. 5 shows the inferred model and expert confidence over time given a single expert-acquired data. The solid red lines represent the true confidence rate and model. One can see that the inferred model approaches the true model in less than 30 steps with the proposed method. The inferred confidence rate becomes closer to the true confidence rate as more data becomes available. Note that the small difference in the inferred confidence rate is to the availability of a single trajectory for expert-acquired data, which does not carry enough information for an accurate assessment of the expert confidence rate.

The impact of the measurement noise on the performance of the proposed method is examined in this part. The top plots of Fig. 6 represent the average inference rates with respect to the measurement noise intensity for two different numbers of data. It can be seen that expert-enabled ML inference outperforms regular ML inference in all cases. The performance of both methods decreases as the level of measurement noise increases. However, the reduction in average inference rates obtained by the proposed expert-enabled inference method is less than that of regular ML inference. A similar trend can be observed for the average errors in the second row of Fig. 6. It can be seen that the incorporation of expert knowledge helps enhance the inference rate under various measurement noise intensities.

B. Biological Networks Application

The application of interest in this paper is biological systems such as gene regulatory networks and microbial communities. These regulatory networks are composed of a number of interacting components, such as bacteria, microbes, genes, and small molecules, which are often observed in high-throughput sequencing data [47]. These systems are often modeled by hidden Markov models with binary state variables [48], [49]. Most available data from these biological systems contain the decisions made by biologists or microbiologists, such as data acquired during interventions for disease treatment or experimental perturbation for hypothesis testing.

Consider a regulatory network with d components. The state process can be expressed as $\{\mathbf{x}_k; k = 0, 1, \ldots\}$, where $\mathbf{x}_k \in \{0, 1\}^d$ represents the activation/inactivation state of the components at time k. The components' state is updated at each discrete time using the following Boolean signal

model [50], [51]:

$$\mathbf{x}_k = \overline{\mathbf{C}\mathbf{x}_{k-1} + \mathbf{b}} \oplus \mathbf{a}_{k-1} \oplus \mathbf{n}_k\,, \tag{35}$$

for $k = 1, 2, \ldots$, where $\mathbf{C}$ is a matrix of size $d \times d$, $\overline{\mathbf{v}}$ maps the elements of an arbitrary vector $\mathbf{v}$ greater than 0 to 1 and smaller than or equal to 0 to 0, $\mathbf{a}_{k-1} \in \mathcal{A} \subset \{0,1\}^d$ is an external input (e.g., intervention, perturbation, etc.) at time step $k-1$, $\mathbf{n}_k \in \{0,1\}^d$ is noise process at time k, "$\oplus$" denotes component-wise binary addition (the exclusive-or logic operation). The elements in the ith row and jth column of the connective matrix, $(\mathbf{C})_{ij}$ model the type of regulatory interaction from component j to component i. $(\mathbf{C})_{ij}$ takes a value in $\{-1, 0, +1\}$, where 0 represents no interaction, and $+1$ and -1 represent positive and negative regulatory interactions, respectively. The stress vector $\mathbf{b} \in \{0,1\}^d$ specifies the external input to each gene, where $\mathbf{b}(i) = 1$ represents the ith system component is under stress. The noise process is modeled through d independent Bernoulli process with parameter p (i.e., $0 \le p \le 0.5$) as: $\mathbf{n}_k(i) \sim$ Bernoulli(p), for $i = 1, ..., d$. The small values of p represent less noisy Boolean network models, whereas larger values correspond to stochastic models.

The measurement process in HMM modeling of biological systems depends on the type of biological data (e.g., type of gene expression or omics data). Without loss of generality, the measurements are considered to be from cDNA microarrays [52] or live-cell imaging-based assays [53], and the following Gaussian model is employed for the observation model:

$$\mathbf{y}_k(j) = m + \delta \mathbf{x}_k(j) + \mathbf{v}_k(j)\,,\ k = 1, 2, \ldots, \tag{36}$$

for $j = 1, ..., d$; where $\mathbf{v}_k(j) \sim \mathcal{N}(0, \sigma^2)$ is an uncorrelated zero-mean Gaussian noise vector, m is a baseline expression corresponding to the "zero" state of components, δ is a differential expression value that indicates by how much the "one" state of each component is over-expressed over the "zero" state.

The unknown parameters of regulatory networks are often missing interactions between different components or bias units. Letting n regulatory interactions be unknown, there will be $M = 3^n$ possible network models for the regulatory networks. These possible models can be represented by $\Theta = \{\boldsymbol{\theta}^1, ..., \boldsymbol{\theta}^M\}$, where $\mathbf{C}_{\boldsymbol{\theta}}$ is the connectivity matrix associated with model $\boldsymbol{\theta} \in \Theta$.

The controlled transition matrix in (13) can be expressed for the regulatory network with model $\boldsymbol{\theta}$ as:

$$(M_{\boldsymbol{\theta}}(\mathbf{a}))_{ij} = p^{\|\overline{\mathbf{C}_{\boldsymbol{\theta}}\mathbf{x}^j + \mathbf{b}} \oplus \mathbf{a} \oplus \mathbf{x}^i\|_1} (1-p)^{d - \|\overline{\mathbf{C}_{\boldsymbol{\theta}}\mathbf{x}^j + \mathbf{b}} \oplus \mathbf{a} \oplus \mathbf{x}^i\|_1}\,,$$

for $i, j = 1, \ldots, 2^d$. The update vector defined in (14) can be written according to the observation model in (36) as:

$$\big(T(\mathbf{y}_k)\big)_i = \frac{1}{(2\pi\sigma^2)^{\frac{d}{2}}} \exp\left(-\frac{\sum_{j=1}^{d}\big(\mathbf{y}_k(j) - \delta \mathbf{x}^i(j) - m\big)^2}{2\sigma^2}\right),$$

for $i = 1, \ldots, 2^d$.

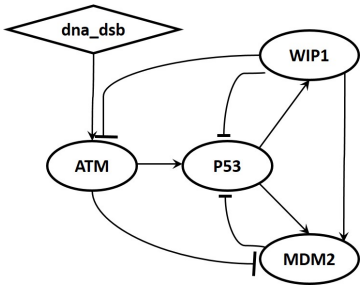


Figure 7: The pathways diagram for the P53 network.

1) p53-MDM2 Negative Feedback Loop Regulatory Network: The p53-MDM2 negative feedback loop gene regulatory network plays a critical role in various types of cancers, including the ovarian, esophageal, larynx, and lung [54], [55]. This network includes four genes, ATM, p53, Wip1, and MDM2, and the stress input "dna_dsb", which shows the presence of DNA double-strand breaks. The pathway diagram for the network is shown in Fig. 7; the solid arrows demonstrate the activating rules (+1 regulatory interactions), and the blunt arrows demonstrate the suppressive rules (-1 regulatory interactions). This Boolean model in (35) in stress response can be represented for this network using the following connectivity matrix and bias units:

$$\mathbf{C} = \begin{bmatrix} c_{11}=0 & c_{12}=0 & c_{13}=-1 & c_{14}=0 \\ c_{21}=+1 & c_{22}=0 & c_{23}=-1 & c_{24}=-1 \\ c_{31}=0 & c_{32}=+1 & c_{33}=0 & c_{34}=0 \\ c_{41}=-1 & c_{42}=+1 & c_{43}=+1 & c_{44}=0 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{37}$$

The network in healthy conditions stays at state "0000", meaning all 4 genes are often in inactivated states. This is, however, not the case for the network in stressed conditions. The stress often leads to unnecessary activation of various genes and uncontrolled cell proliferation. Therefore, in genomics intervention, the experts aim to keep the network at state "0000" through drug-induced controls. The intervention is achieved through the following control inputs: $\mathcal{A} = \{\mathbf{a}^1 = [0,0,0,0]^T, \mathbf{a}^2 = [1,0,0,0]^T, \mathbf{a}^3 = [0,1,0,0]^T, \mathbf{a}^4 = [0,0,1,0]^T, \mathbf{a}^5 = [0,0,0,1]^T\}$, where $\mathbf{a}^2$ to $\mathbf{a}^5$ can alter the value of a single gene at a time. The expert reward function can be defined as:

$$R(\mathbf{x}, \mathbf{a}, \mathbf{x}') = -\|\mathbf{x}'\|_1 - \frac{1}{2}\|\mathbf{a}\|_1,$$

where the cost of any gene inactivation is -1, and the control input altering the gene state has the reward of $-\frac{1}{2}$. The negative reward value for the control represents the expert's desire to take minimum controls due to their potential side effects. The expert aims to make the system far from the state "1111", which has the lowest reward value and makes the system close to the state "0000", which yields the largest reward value.

The expert-acquired data is assumed to be collected using the true network model in (37). The following five interacting parameters are considered to be unknown: c_{13}, c_{21}, c_{32}, c_{42} and c_{43}. Since each interaction can take in $\{-1, 0, +1\}$, there will be $M = 3^5 = 243$ possible network models with $\boldsymbol{\theta}^* = (-1, +1, +1, +1, +1)$. The following parameters are used for

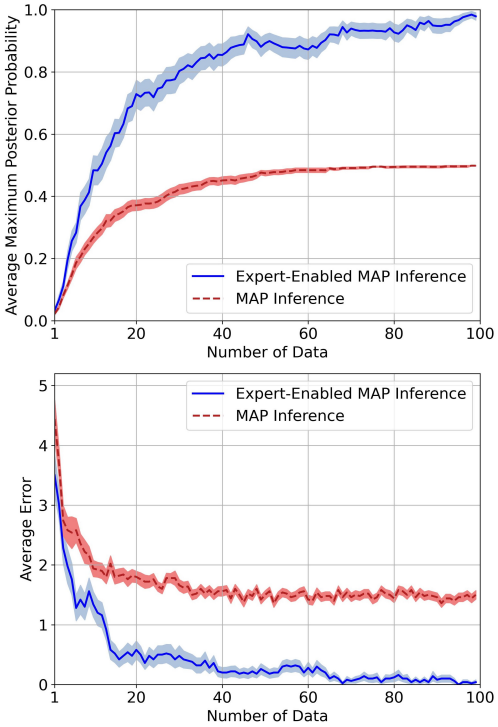


Figure 8: Average true inference and error obtained by proposed expert-enabled MAP and regular MAP methods for the p53-MDM2 network.

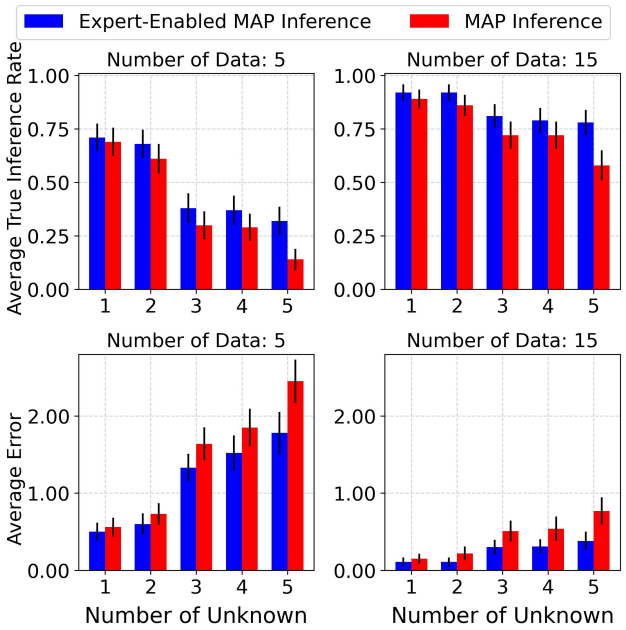


Figure 9: Average inference rate and error with respect to the number of unknowns in the p53-MDM2 network.

the experiments: $p = 0.05$, $q = 0.75$, $\sigma = 10$, $m = 20$, $\delta = 30$, $\gamma = 0.95, \lambda = 0.01$. These are typical parameter values for modeling genomics and microbial communities [50], [51]. The prior probability of all models is assumed to be the same, i.e., $p(\boldsymbol{\theta}) = 1/M$, for $\boldsymbol{\theta} \in \Theta$.

The results of expert-enabled MAP inference and regular MAP inference are demonstrated in Fig. 8. The top plot demonstrates the average maximum posterior probability. It can be seen that the proposed method significantly outperforms the conventional MAP inference method. In fact, under the proposed method, the true inference rate is around 90% after 40 data points, whereas the rate for regular MAP inference does not exceed 50% even after 100 data points. The average error results in the right plot of Fig. 8 better illustrate the reason behind the high performance of the proposed method. One can see that the average error under the proposed expert-enabled inference becomes close to 0 after 40 data points, whereas for regular MAP inference, it does not approach zero and remains around 1.5. This comes from the non-identifiability of some parameters of the model when using only the temporal changes in data.

This part of the experiment examines the impact of measurement noise on the performance of the proposed method. The results in terms of average error and maximum posterior probability obtained under the proposed expert-enabled MAP inference and the regular MAP inference for three different measurement noise levels are presented in Table I. It can be seen that as the measurement noise increases, the maximum posterior probability decreases and the error increases. This reduction is much more significant for conventional inference techniques. In fact, the proposed method significantly outperforms conventional inference under large noise levels. As more data becomes available, the inference rate for both methods increases. For the large noise case of $\sigma = 45$, the maximum posterior probability for conventional MAP inference is still 0.13, whereas the proposed method's rate is 0.83. This indicates how the incorporation of expert knowledge in domains with noisy data can help the probability of distinguishing models from each other and improve the maximum posterior probability.

Table I: Average performance of proposed expert-enabled MAP and regular MAP inference methods with respect to the measurement noise intensity for the p53-MDM2 network.

		Expert-Enabled MAP Inference		**MAP Inference**	
k	σ	Avg. Error	Max. Post.	Avg. Error	Max. Post.
20	15	0.87 ± 0.18	0.56 ± 0.04	1.4 ± 0.23	0.33 ± 0.07
	30	1.36 ± 0.23	0.36 ± 0.04	2.35 ± 0.24	0.06 ± 0.01
	45	1.47 ± 0.23	0.28 ± 0.03	2.87 ± 0.27	0.03 ± 0.00
40	15	0.15 ± 0.08	0.88 ± 0.03	0.59 ± 0.14	0.60 ± 0.04
	30	$0.65 \pm 0.0.14$	0.66 ± 0.04	1.74 ± 0.25	0.14 ± 0.02
	45	0.76 ± 0.18	0.58 ± 0.05	2.48 ± 0.26	0.05 ± 0.01
100	15	0 ± 0	0.99 ± 0.00	0.14 ± 0.06	0.90 ± 0.02
	30	0.1 ± 0.05	0.91 ± 0.02	0.82 ± 0.17	0.43 ± 0.03
	45	0.29 ± 0.10	0.83 ± 0.03	1.87 ± 0.19	0.13 ± 0.02

The impact of the number of unknown regulations on the performance of the proposed expert-enabled method is demonstrated in Fig. 9. The top plots represent the average true inference rate, while the bottom plots show the average error with respect to the number of unknown regulations. It can be observed that the results of expert-enabled and conventional MAP inference are closer for smaller unknowns, owing to the decrease in complexity of inference under smaller unknowns. The most significant difference between these two approaches corresponds to the case with five unknowns, where the proposed method achieves a much smaller average error

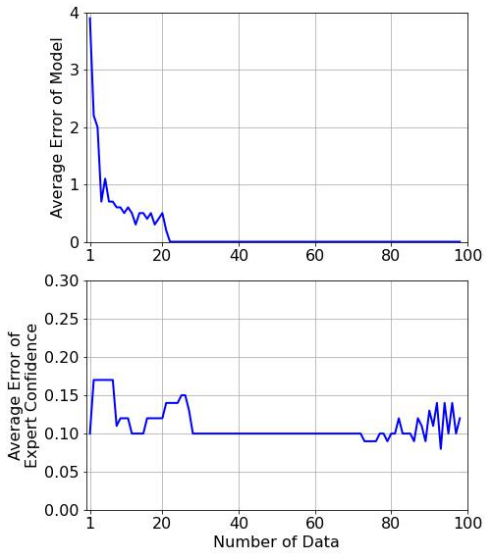


Figure 10: Average inference error of model and expert's confidence obtained by the proposed method for the p53-MDM2 network.

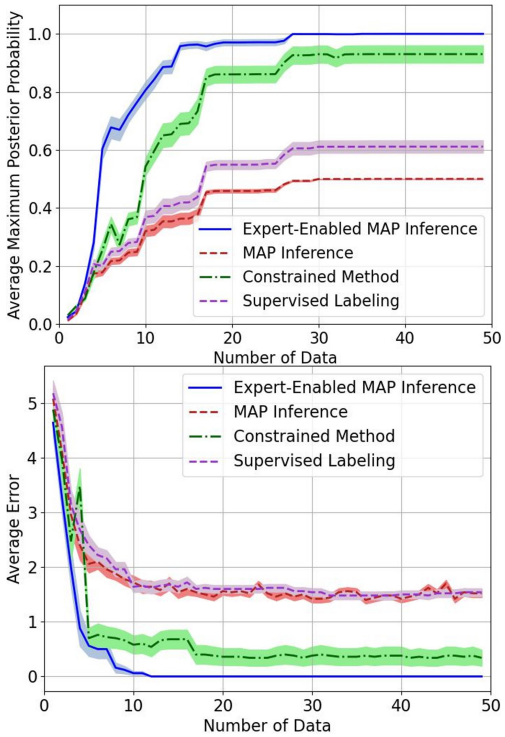


Figure 11: Comparison of average true inference and error obtained by proposed expert-enabled MAP method with constrained, supervised labeling, and regular MAP methods for the p53-MDM2 network.

and higher inference rate. Meanwhile, for a larger number of data shown in the second column plots in Fig. 9, the average error has decreased and the inference rate has increased for both methods, while the expert-enabled inference approach still outperforms the regular MAP inference in all cases.

The joint inference of the expert confidence and system model is illustrated in Fig. 10. The top figure represents the average error for the inferred expert confidence and system model using 10 datasets generated from an expert with the confidence rate $q = 0.85$. One can see that the error model approaches 0 after 20 data, and the expert confidence is also inferred with about a 0.1 distance from the true confidence rate throughout the process. This indicates a higher applicability of the proposed expert-enabled inference in domains with unknown expert confidence.

In this part of the numerical experiments, the performance of the proposed method is compared with two popular inference methods that incorporate expert knowledge. The same set of parameters used for experiments in Fig. 8 is considered here. In the first comparing approach, the known expert reward function is used for imposing constraints on the space of models [30], [31]. In fact, expert knowledge is used to remove the models that expert actions cannot be strongly justified. This can be expressed through the following constrained inference problem:

$$\hat{\boldsymbol{\theta}}_k = \underset{\boldsymbol{\theta}\in\Theta^{\text{Cons}}}{\operatorname{argmax}} \sum_{r=1}^{k} \log p(\tilde{\mathbf{y}}_r \mid \tilde{\mathbf{a}}_{0:r-1}, \tilde{\mathbf{y}}_{1:r-1}, \boldsymbol{\theta}), \tag{38}$$

where $\Theta^{\text{Cons}} = \{\boldsymbol{\theta} \in \Theta \mid p(\tilde{\mathbf{a}}_{0:k-1} \mid \tilde{\mathbf{y}}_{1:k}, \boldsymbol{\theta}) > \delta\}$. The restricted models are those in which the actions taken by the experts are the least likely.

The second method for comparison is supervised labeling [26], [27], which explicitly incorporates expert knowledge about the underlying system state to enhance the inference process. This information is derived from a proficient expert who is familiar with the true underlying system state. Therefore, this approach necessitates explicit expert knowledge, while the proposed method is capable of implicitly incorporating expert knowledge without the need for expert supervision. Implicit knowledge incorporation is particularly applicable in the absence of an expert or when the data collected by an expert in the past is no longer available. Specifically, the expert provides perfect labeling at 20% of randomly selected time steps. Fig. 11 represents the comparison results in terms of the average maximum posterior probability and the average error. It can be observed that the proposed method outperforms all other methods, especially when dealing with small data sizes. The results of supervised labeling are similar to those of regular MAP inference methods, even though labeling has been performed. This demonstrates the impact of the expert-knowledge term in the proposed method, which is not accounted for in supervised labeling. The constrained inference method with the threshold $\delta = 0.45$ performs better than the other two competing methods. However, the reliance on a threshold and the non-systematic incorporation of expert knowledge have led to lower performance of this approach compared to the proposed method.

2) Gut Microbial Community: In this part of the numerical experiment, the performance of the proposed framework is examined using the gut microbial community [56]. This microbial community plays a critical role in the normal intestinal functions. Mutation or damage to this community can lead to different serious illnesses such as obesity, diabetes, and even neurological disorders. The pathway diagram of the gut microbial community network is shown in Fig. 12. The state vector consists of $\mathbf{x}_k =$

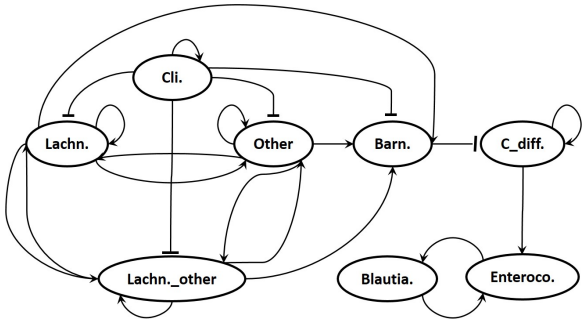


Figure 12: The pathways diagram for the gut microbial community.

$[Cli., Lachn., Lachn.\_other, Other, Barn., C\_diff, Enteroco., Blautia]^T$. The 8 components of this network lead to the state space of size $2^8 = 256$. The connectivity matrix and bias vector for the gut microbial network can be expressed using the pathway diagram in Fig. 12 as:

$$\mathbf{C} = \begin{bmatrix} +1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & +1 & +1 & +1 & 0 & 0 & 0 & 0 \\ -1 & +1 & +1 & +1 & 0 & 0 & 0 & 0 \\ -1 & +1 & +1 & +1 & 0 & 0 & 0 & 0 \\ -1 & +1 & +1 & +1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & +1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & +1 & 0 & +1 \\ 0 & 0 & 0 & 0 & 0 & 0 & +1 & 0 \end{bmatrix}, \quad (39)$$

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

The intervention in the gut microbial community in patients with antibiotic resistance issues takes place in the presence of Clindamycin. In this condition, it is critical to prevent the activation of *C_diff* and the inactivation of *Barn.*, which are both found to be associated with these undesirable conditions. Therefore, the expert reward function during the intervention can be expressed as:

$$R(\mathbf{x}, \mathbf{a}, \mathbf{x}') = 1_{\mathbf{x}'(5)=1} - 1_{\mathbf{x}'(6)=1} - 0.5\|\mathbf{a}\|_1,$$

for $\mathbf{x}, \mathbf{x}' \in \mathcal{X}, \mathbf{a} \in \mathcal{A}$. The action space includes $\mathcal{A} = \{\mathbf{a}^1, \mathbf{a}^2, \mathbf{a}^3, \mathbf{a}^4\}$, where $\mathbf{a}^1$ represents no alteration, $\mathbf{a}^2$ corresponds to altering $\mathbf{x}(2)$ and $\mathbf{x}(3)$, $\mathbf{a}^3$ correspond to altering $\mathbf{x}(3)$ and $\mathbf{x}(4)$, finally $\mathbf{a}^4$ correspond to altering $\mathbf{x}(2)$ and $\mathbf{x}(4)$.

For the experiments, the following four unknown interactions are considered: $c_{24} = +1$, $c_{33} = +1$, $c_{53} = +1$, $c_{66} = +1$. These unknowns lead to $3^4 = 81$ possible network models for the regulatory networks. We assume a uniform prior probability for these models prior to observing any data. The following parameters are also used for this experiment: $p = 0.01$, $q = 0.75$, $\sigma = 10$, $m = 20$, $\delta = 30$, $\gamma = 0.95, \lambda = 0.01$.

The average results in terms of the maximum posterior probability, the true inference rate, and error are shown in Fig. 13. The proposed expert-enabled MAP inference resulted in a higher maximum posterior probability compared to regular MAP inference, demonstrating the impact of the expert knowledge term in distinguishing models. Additionally, the proposed method has shown a significantly higher average rate of true inference. The average rate obtained by the expert-enabled method exceeded 0.8 upon observing 20 data, whereas the average rate under regular MAP inference did not exceed 0.6 even with 100 data. Similarly, the proposed method yields a much smaller average error compared to regular MAP. The average error became 0 under the proposed method after 80 data, whereas the average error did not fall below 0.5 under the regular MAP inference approach. Finally, a comparison between the left and middle plots of Fig. 13 shows that the model with the maximum posterior probability selected by the proposed method mostly corresponds to the true system model, as evidenced by the similar trends of the blue solid lines in these two plots. On the other hand, for regular MAP inference, the true inference rate is lower than the maximum posterior probability, indicating that the non-true models also hold the maximum posterior probability.

VII. Conclusion And Future Work

This paper presents an optimal inference technique for complex systems modeled by hidden Markov models (HMMs) and observed through expert-acquired data. Unlike conventional methods that consider the temporal data changes for the inference process, the proposed approach accounts for expert knowledge reflected in available data in terms of actions or decisions. Expert behavior is modeled as a non-optimal reinforcement learning agent, where a stochastic policy is used for modeling expert policy. Using this model, the expert's perception of the system is quantified and incorporated to derive optimal expert-enabled inference. The proposed method is built on the combination of dynamic programming and Bayesian filtering techniques, where a recursive and exact inference solution is provided. The analytical results demonstrate that the additional term in the expert-enabled likelihood function makes the true system model more distinguishable. This superiority is shown to be significant in the presence of confident experts, compared to less-confident experts with more randomness in their policies. For domains with a priori unknown confidence level, the joint inference of system model and expert confidence is formulated. Numerical experiments using benchmark and biological systems demonstrate the significance of incorporating expert knowledge in terms of better inference rate and robustness compared to conventional techniques. Also, the importance of expert-enabled inference in noisier data is empirically demonstrated, and a comparison between the proposed method and other inference methods in terms of average posterior probability and average error is presented.

Future steps include extensions of the proposed expert-enabled inference to large and complex systems, including those with continuous/large state and action spaces, as well as domains consisting of multiple cooperative and competitive experts. Additionally, studying the impacts of incorporating knowledge from biased experts on the inference solution is among the next steps of the paper.

Acknowledgment

The authors acknowledge the support of the National Institute of Health award 1R21EB032480-01, National Science

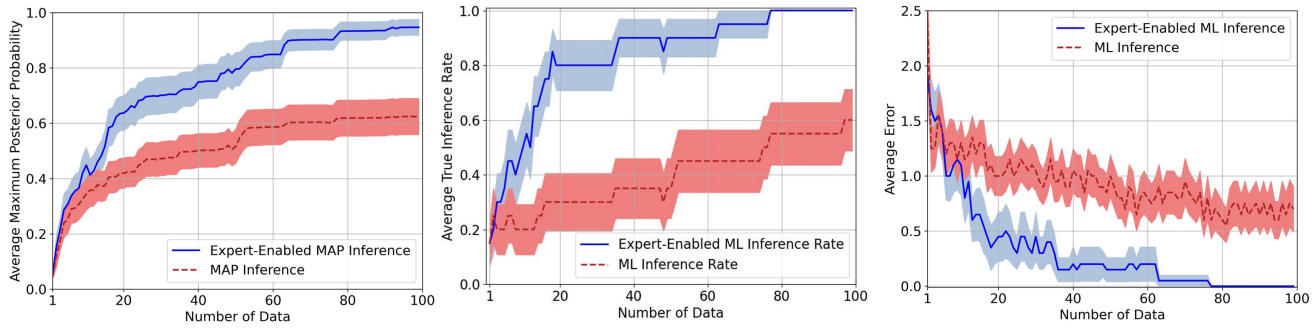


Figure 13: Average maximum posterior probability, true inference rate, and error obtained by proposed expert-enabled MAP and regular MAP methods for the gut microbial community.

Foundation awards IIS-2311969 and IIS-2202395, ARMY Research Laboratory award W911NF2320179, ARMY Research Office award W911NF2110299, and Office of Naval Research award N00014-23-1-2850.

REFERENCES

[1] 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.

[2] S. Särkkä, *Bayesian filtering and smoothing*, vol. 17. Cambridge university press, 2013.

[3] A. Doucet, N. De Freitas, N. J. Gordon, *et al.*, *Sequential Monte Carlo methods in practice*, vol. 1. Springer, 2001.

[4] O. Netzer, P. Ebbes, and T. H. Bijmolt, "Hidden Markov models in marketing," *Advanced methods for modeling markets*, pp. 405–449, 2017.

[5] B. Mor, S. Garhwal, and A. Kumar, "A systematic review of hidden Markov models and their applications," *Archives of computational methods in engineering*, vol. 28, pp. 1429–1448, 2021.

[6] R. J. Elliott, L. Aggoun, and J. B. Moore, *Hidden Markov models: estimation and control*, vol. 29. Springer Science & Business Media, 2008.

[7] H. N. Nagaraja, "Inference in hidden Markov models," *Technometrics*, vol. 48, no. 4, p. 574, 2006.

[8] P. Wang and P. Blunsom, "Collapsed variational Bayesian inference for hidden Markov models," in *Artificial Intelligence and Statistics*, pp. 599–607, PMLR, 2013.

[9] S. S. Hassan, S. Särkkä, and Á. F. García-Fernández, "Temporal parallelization of inference in hidden Markov models," *IEEE Transactions on Signal Processing*, vol. 69, pp. 4875–4887, 2021.

[10] M. Dewar, C. Wiggins, and F. Wood, "Inference in hidden Markov models with explicit state duration distributions," *IEEE Signal Processing Letters*, vol. 19, no. 4, pp. 235–238, 2012.

[11] Y.-A. Ma, N. J. Foti, and E. B. Fox, "Stochastic gradient MCMC methods for hidden Markov models," in *International Conference on Machine Learning*, pp. 2265–2274, PMLR, 2017.

[12] N. Foti, J. Xu, D. Laird, and E. Fox, "Stochastic variational inference for hidden Markov models," *Advances in neural information processing systems*, vol. 27, 2014.

[13] M. Imani and S. F. Ghoreishi, "Two-stage Bayesian optimization for scalable inference in state-space models," *IEEE transactions on neural networks and learning systems*, vol. 33, no. 10, pp. 5138–5149, 2021.

[14] M. Imani, S. F. Ghoreishi, D. Allaire, and U. M. 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.

[15] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[16] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state Markov chains," *The annals of mathematical statistics*, vol. 37, no. 6, pp. 1554–1563, 1966.

[17] B. G. Leroux, "Maximum-likelihood estimation for hidden Markov models," *Stochastic processes and their applications*, vol. 40, no. 1, pp. 127–143, 1992.

[18] C. P. Robert, T. Ryden, and D. M. Titterington, "Bayesian inference in hidden Markov models through the reversible jump Markov chain Monte Carlo method," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 62, no. 1, pp. 57–75, 2000.

[19] O. Cappé, E. Moulines, and T. Rydén, "Inference in hidden Markov models," in *Proceedings of EUSFLAT conference*, pp. 14–16, 2009.

[20] M. Delattre and M. Lavielle, "Maximum likelihood estimation in discrete mixed hidden Markov models using the SAEM algorithm," *Computational Statistics & Data Analysis*, vol. 56, no. 6, pp. 2073–2085, 2012.

[21] V. B. Tadić, "Analyticity, convergence, and convergence rate of recursive maximum-likelihood estimation in hidden Markov models," *IEEE Transactions on Information Theory*, vol. 56, no. 12, pp. 6406–6432, 2010.

[22] P. Fearnhead and P. Clifford, "On-line inference for hidden Markov models via particle filters," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 65, no. 4, pp. 887–899, 2003.

[23] G. Alexandrovich, H. Holzmann, and A. Leister, "Nonparametric identification and maximum likelihood estimation for hidden Markov models," *Biometrika*, vol. 103, no. 2, pp. 423–434, 2016.

[24] D. J. Cole, "Parameter redundancy and identifiability in hidden Markov models," *Metron*, vol. 77, no. 2, pp. 105–118, 2019.

[25] M. Alali and M. Imani, "Reinforcement learning data-acquiring for causal inference of regulatory networks," in *American Control Conference (ACC), IEEE*, 2023.

[26] M. Qiao, W. Bian, R. Y. Da Xu, and D. Tao, "Diversified hidden Markov models for sequential labeling," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 11, pp. 2947–2960, 2015.

[27] C. Zhang and K. Chaudhuri, "Active learning from weak and strong labelers," *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[28] A. Laptev, R. Korostik, A. Svischev, A. Andrusenko, I. Medennikov, and S. Rybin, "You do not need more data: Improving end-to-end speech recognition by text-to-speech data augmentation," in *2020 13th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pp. 439–444, IEEE, 2020.

[29] T. Xiahou, Z. Zeng, and Y. Liu, "Remaining useful life prediction by fusing expert knowledge and condition monitoring information," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 4, pp. 2653–2663, 2020.

[30] V. Asvatourian, P. Leray, S. Michiels, and E. Lanoy, "Integrating expert's knowledge constraint of time dependent exposures in structure learning for Bayesian networks," *Artificial Intelligence in Medicine*, vol. 107, p. 101874, 2020.

[31] D. R. Scobee and S. S. Sastry, "Maximum likelihood constraint inference for inverse reinforcement learning," in *International Conference on Learning Representations*, 2019.

[32] A. Zhang, J. Zhu, and B. Zhang, "Max-margin infinite hidden Markov models," in *International Conference on Machine Learning*, pp. 315–323, PMLR, 2014.

[33] E. Di Lello, "Bayesian time-series models: Expert knowledge-driven inference and learning for engineering applications," 2015.

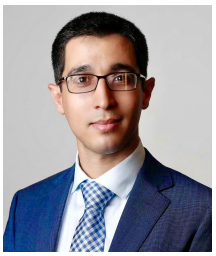
[34] S. Arora and P. Doshi, "A survey of inverse reinforcement learning: Challenges, methods and progress," *Artificial Intelligence*, vol. 297, p. 103500, 2021.

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

[36] N. Jaques, A. Ghandeharioun, J. H. Shen, C. Ferguson, A. Lapedriza, N. Jones, S. Gu, and R. Picard, "Way off-policy batch deep reinforcement learning of implicit human preferences in dialog," *arXiv preprint arXiv:1907.00456*, 2019.
[37] M. Imani and U. Braga-Neto, "Optimal control of gene regulatory networks with unknown cost function," in *Proceedings of the 2018 American Control Conference (ACC 2018)*, IEEE, 2018.
[38] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, "Imitation learning: A survey of learning methods," *ACM Computing Surveys (CSUR)*, vol. 50, no. 2, pp. 1–35, 2017.
[39] J. Ho and S. Ermon, "Generative adversarial imitation learning," *Advances in neural information processing systems*, vol. 29, 2016.
[40] M. Imani and S. F. Ghoreishi, "Scalable inverse reinforcement learning through multifidelity Bayesian optimization," *IEEE transactions on neural networks and learning systems*, vol. 33, no. 8, pp. 4125–4132, 2021.
[41] A. Ravari, S. Ghoreishi, and M. Imani, "Structure-based inverse reinforcement learning for quantification of biological knowledge," in *IEEE Conference on Artificial Intelligence*, 2023.
[42] R. S. Sutton, A. G. Barto, *et al.*, *Introduction to reinforcement learning*, vol. 2. MIT press Cambridge, 1998.
[43] S. Särkkä and L. Svensson, *Bayesian filtering and smoothing*, vol. 17. Cambridge university press, 2023.
[44] P. R. Kumar and P. Varaiya, *Stochastic systems: Estimation, identification, and adaptive control*, vol. 75. SIAM, 2015.
[45] W. B. Powell, *Approximate Dynamic Programming: Solving the curses of dimensionality*, vol. 703. John Wiley & Sons, 2007.
[46] S. Junges, N. Jansen, R. Wimmer, T. Quatmann, L. Winterer, J.-P. Katoen, and B. Becker, "Finite-state controllers of POMDPs via parameter synthesis," in *Uncertainty in Artificial Intelligence: Thirty-Fourth Conference (2018)*, pp. 519–529, 2018.
[47] L. E. Chai, S. K. Loh, S. T. Low, M. S. Mohamad, S. Deris, and Z. Zakaria, "A review on the computational approaches for gene regulatory network construction," *Computers in biology and medicine*, vol. 48, pp. 55–65, 2014.
[48] M. Imani and U. M. Braga-Neto, "Maximum-likelihood adaptive filter for partially observed Boolean dynamical systems," *IEEE Transactions on Signal Processing*, vol. 65, no. 2, pp. 359–371, 2016.
[49] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang, "Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks," *Bioinformatics*, vol. 18, no. 2, pp. 261–274, 2002.
[50] M. Imani and U. M. Braga-Neto, "Maximum-likelihood adaptive filter for partially observed Boolean dynamical systems," *IEEE Transactions on Signal Processing*, vol. 65, no. 2, pp. 359–371, 2017.
[51] M. Imani, E. R. Dougherty, and U. Braga-Neto, "Boolean kalman filter and smoother under model uncertainty," *Automatica*, vol. 111, p. 108609, 2020.
[52] 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.
[53] 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.
[54] E. Batchelor, A. Loewer, and G. Lahav, "The ups and downs of p53: understanding protein dynamics in single cells," *Nature Reviews Cancer*, vol. 9, no. 5, p. 371, 2009.
[55] M. Fischer, "Conservation and divergence of the p53 gene regulatory network between mice and humans," *Oncogene*, vol. 38, no. 21, pp. 4095–4109, 2019.
[56] 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.

Amirhossein Ravari is a Ph.D. candidate in Electrical Engineering at Northeastern University, Boston, USA. He received his B.S. and M.S. degrees in Electrical Engineering from Sharif University of Technology, Iran, in 2018 and 2021, respectively. His research interests include inverse reinforcement learning and decision/control theory.

Seyede Fatemeh Ghoreishi received her Ph.D. and M.Sc. degrees both in Mechanical Engineering from Texas A&M University in 2019 and 2016, respectively. She is an assistant professor in the College of Engineering and Khoury College of Computer Sciences at Northeastern University. Her research includes machine learning and Bayesian statistics for design and decision-making under uncertainty. Prior to Northeastern, she was a postdoctoral research fellow at the Institute for Systems Research at the University of Maryland. She holds a minor in Applied Statistics from the Department of Statistics at Texas A&M University. 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.



Mahdi Imani received his Ph.D. degree in Electrical and Computer Engineering from Texas A&M University, College Station, TX in 2019. He is an Assistant Professor in the Department of Electrical and Computer Engineering at Northeastern University. His research interests include reinforcement learning, reasoning under uncertainty, and multi-agent systems, with a wide range of applications from computational biology to cyber-physical systems. He is the recipient of the NIH NIBIB Trailblazer award in 2022, NSF CISE Career Research Initiation Initiative award in 2020, the Oracle Research Award in 2022, the Association of Former Students Distinguished Graduate Student Award for Excellence in Research-Doctoral in 2019, and the best paper finalist award from the American Control Conference in 2023 and the 49th Asilomar Conference on Signals, Systems, and Computers in 2015.