Reinforcement Learning Data-Acquiring for Causal Inference of Regulatory Networks

Mohammad Alali and Mahdi Imani

Abstract— **Gene regulatory networks (GRNs) consist of multiple interacting genes whose activities govern various cellular processes. The limitations in genomics data and the complexity of the interactions between components often pose huge uncertainties in the models of these biological systems. Meanwhile, inferring/estimating the interactions between components of the GRNs using data acquired from the normal condition of these biological systems is a challenging or, in some cases, an impossible task. Perturbation is a well-known genomics approach that aims to excite targeted components to gather useful data from these systems. This paper models GRNs using the Boolean network with perturbation, where the network uncertainty appears in terms of unknown interactions between genes. Unlike the existing heuristics and greedy data-acquiring methods, this paper provides an optimal Bayesian formulation of the data-acquiring process in the reinforcement learning context, where the actions are perturbations, and the reward measures step-wise improvement in the inference accuracy. We develop a semi-gradient reinforcement learning method with function approximation for learning near-optimal data-acquiring policy. The obtained policy yields near-exact Bayesian optimality with respect to the entire uncertainty in the regulatory network model, and allows learning the policy offline through planning. We demonstrate the performance of the proposed framework using the well-known p53-Mdm2 negative feedback loop gene regulatory network.**

I. Introduction

Gene regulatory networks (GRNs) consist of multiple interacting genes whose activities characterize mechanisms involved in complex diseases such as cancer [1]–[4]. Knowledge about GRNs could answer fundamental questions in biology and human health, including how complex GRNs regulate the response of tissues and cells to stressors, such as injury and infection [5]. Many efforts have been made to infer and model GRNs [6], yet it remains a challenging task to build accurate models, primarily due to the lack of access to targeted data needed for causal and high-quality inference.

Perturbation is a common procedure for acquiring targeted data in biological systems [7]–[11]. The perturbations are often achieved using drug-induced excitations that overexpress or suppress targeted genes over time. The objective of perturbation is to make targeted changes in the dynamics of GRNs, and acquire data that reveal the most about the underlying mechanism of GRNs.

In practice, most biological perturbations are conducted through a trial-and-error process by biologists, which can be suboptimal or inefficient for the majority of complex biological systems. Some attempts have been previously made toward systematic dynamic perturbation of GRNs. These include perturbation strategies for networks modeled by ordinary differential equations (ODE) [12], [13], static networks [14]–[17], and deterministic dynamical models [18]–[22]. However, these techniques are built on different heuristics or simplified assumption that the network uncertainty vanishes upon taking any single perturbation [23]–[27]. Therefore, these methods become inefficient for data-acquiring in complex regulatory networks with possibly limited data-acquiring resources. Moreover, in [28], we developed an optimal finite-horizon perturbation policy for small regulatory networks with a small number of unknown interactions and short horizons. In this approach, an exhaustive search is performed over all possible network models and state transitions. Evidently, this method cannot be generalized to larger networks with more unknown interactions and longer horizons due to its computational complexity; hence, its applications are limited.

This paper models GRNs through the Boolean network with perturbation (BNp) [29]–[32]. This BNp model is capable of properly capturing the stochasticity in GRNs and benefits from simplicity and interpretability. The state values of the genes in this model represent the activation and inactivation of the genes, and their time-varying behavior is modeled through the Markov decision process (MDP). The primary objective of this paper is to derive a data-acquiring policy that can select time-dependent perturbations that lead to the highest accuracy of the inference process. Since the unknown parts of GRNs are often the interactions between different genes, a good data-acquiring policy should help the causal inference of interacting parameters and, equivalently, the dynamics of GRNs.

The maximum a posteriori (MAP) is used in this paper as a criterion for the inference process. The MAP inference selects the model with the largest posterior probability as the GRN model, where the largest posterior probability measures the confidence about the inferred model (i.e., the probability of true inference). The maximum posterior probability near 1 represents an accurate inference process, where the true model is distinguishable from other models. In contrast, the maximum posterior probability close to 0 represents scenarios where models cannot be confidently distinct from each other using the available data. Accurate and confident inference in GRNs through data acquired from their normal condition is impossible. GRNs without perturbations often spend most of their time in a small subset of states (i.e., attractor states), which limits access to diverse data required

M. Alali and M. Imani are with the Department of Electrical and Computer Engineering at Northeastern University. Emails: alali.m@northeastern.edu, m.imani@northeastern.edu

for accurate inference. Therefore, given the cost and limitation of biological data, it is critical to intelligently perturb these biological systems and acquire data that helps the most in the inference of these biological systems.

In this work, we provide optimal Bayesian formulation for data-acquiring of GRNs. The objective of the data-acquiring is an accurate causal inference of unknown regulatory interactions. The optimal data-acquiring is achieved by defining the belief state, which maps the partially known Boolean network model to a known MDP in the belief space. This belief formulation allows reinforcement learning representation of the data-acquiring process. We develop a semi-gradient reinforcement learning method with function approximation for learning near-optimal data-acquiring policy, which meets the near-exact Bayesian optimality given all available uncertainty in a GRN model. The high performance of the proposed method is demonstrated using the p53-Mdm2 negative feedback loop network.

II. Gene Regulatory Network Model

In this paper, we consider a Boolean network with perturbation (BNp) for modeling GRNs [29], [30]. According to this model, the inactivation and activation of each gene can be represented through 0 and 1, and the interactions between each of the genes govern the GRNs' dynamics. Consider a GRN with d genes. The state values of these genes at time step k can be represented in a single vector $\mathbf{x}_k = [\mathbf{x}_k(1), ..., \mathbf{x}_k(d)]^T$, where $\mathbf{x}_k(i) \in \{0, 1\}$ denotes the state value of the ith gene at time step k. The state transition in BNp is modeled through the Markov process, where the next state is only dependent on the previous state, and the input/perturbation as:

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}) \oplus \mathbf{u}_{k-1} \oplus \mathbf{n}_k\,, \tag{1}$$

for $k = 1, 2, \ldots$, where $\mathbf{n}_k \in \{0, 1\}^d$ is the Boolean transition noise at time k, $\mathbf{u}_{k-1} \in \mathcal{U} = \{\mathbf{u}^1, ..., \mathbf{u}^L\} \subset \{0, 1\}^d$ is the perturbation at time step $k-1$, "$\oplus$" indicates component-wise modulo-2 addition, and $\mathbf{f}$ represents the *network function*. The way that the perturbation impacts the system's state is by flipping the value of specific genes' states. For instance, $\mathbf{u}_{k-1}(i) = 1$ flips the state value of the ith gene, contrary to the case with $\mathbf{u}_{k-1}(i) = 0$.

The network function is often expressed through a Boolean logic or pathway diagram model [22], [33]–[36]. In this paper, we consider the pathway diagram model as:

$$\mathbf{f}(\mathbf{x}_{k-1}) = \overline{\mathbf{R}\,\mathbf{x}_{k-1}}, \tag{2}$$

where $\mathbf{R}$ is the connectivity matrix governing the dynamics of a GRN, and $\overline{\mathbf{v}}$ is a nonlinear operator that maps the positive elements of vector $\mathbf{v}$ to 1 and others to 0. The element in the ith row and jth column of the connectivity matrix, i.e., $(\mathbf{R})_{ij} = r_{ij}$, denotes the type of regulation from gene j to gene i; it takes $+1$ and -1 values for positive and negative regulations, and 0 for no regulation. Each element of the noise is modeled through an independent Bernoulli process with parameter p as: $\mathbf{n}_k(i) \sim \text{Bernoulli}(p)$, where $0 \le p < 0.5$, for $i = 1, ..., d$. A larger p leads to a more stochastic process.

III. Problem Formulation

In modeling GRNs, several unknown interactions often need to be inferred according to available data. The causal inference of regulatory parameters is critical for various genomic analyses, including distinguishing healthy and unhealthy GRNs and finding effective therapies for chronic diseases. One of the main challenges in the inference of GRNs is the non-identifiability issue, which refers to the scenario where the true underlying interactions between genes cannot be inferred through data acquired from their normal conditions. Perturbation is a common approach in genomics to over-express or suppress some specific genes in GRNs [7]–[9], [37]. Genetic perturbations are drug-induced excitations that change the gene-expression profile and help acquire data that reveal the most about true underlying regulatory interactions.

Consider m regulatory parameters $\{r^1, ..., r^m\}$ are unknown. These interactions are elements of the connectivity matrix $\mathbf{R}$ in (2). Since each element takes in values from $\{+1, 0, -1\}$, there will be 3^m different possible models (i.e., connectivity matrices) denoted by: $\Theta = \{\theta^1, ..., \theta^{3^m}\}$, where $\theta^j = [\theta^j(1), \cdots, \theta^j(m)]$, and $\theta^j(i)$ denotes the type of the ith unknown interaction under the jth model. Let the prior probability over the ith interaction be denoted by $P(r^i = -1)$, $P(r^i = 0)$, and $P(r^i = +1)$. Assuming the independency of the unknown interactions, the prior probability of all the possible models can be expressed as:

$$P(\theta^j) = \prod_{i=1}^{m} \Big[P(r^i = -1)1_{\theta^j(i)=-1} + P(r^i = 0)1_{\theta^j(i)=0} + P(r^i = +1)1_{\theta^j(i)=+1} \Big], \tag{3}$$

for $j = 1, ..., 3^m$, where $\sum_{j=1}^{3^m} P(\theta^j) = 1$, and $1_{\theta^j(i)=-1}$ is 1 if the ith interacting parameter in the jth model/topology is -1; otherwise, $1_{\theta^j(i)=-1}$ is equal to zero.

Let $\mathbf{u}_{0:k-1} = \{\mathbf{u}_0, ..., \mathbf{u}_{k-1}\}$ be the sequence of perturbations and $\mathbf{x}_{1:k} = \{\mathbf{x}_1, ..., \mathbf{x}_k\}$ be the sequence of the observed states until time step k. The posterior distribution of models can be expressed as $P(\theta \mid \mathbf{x}_{1:k}, \mathbf{u}_{0:k-1})$, for $\theta \in \Theta$. The maximum a posteriori (MAP) inference given the information up to time step k can be expressed as:

$$\theta_k^{\text{MAP}} = \underset{\theta \in \Theta}{\text{argmax}}\, P(\theta \mid \mathbf{x}_{1:k}, \mathbf{u}_{0:k-1}), \tag{4}$$

where θ_k^{MAP} is the model in Θ with the highest posterior probability. To assess the confidence of the MAP inference or equivalently the probability that θ_k^{MAP} is the true underlying model, we define the MAP confidence rate as:

$$C_k = \max_{\theta \in \Theta} P(\theta \mid \mathbf{x}_{1:k}, \mathbf{u}_{0:k-1}), \tag{5}$$

where C_k is the maximum posterior probability of models. The MAP confidence rate takes a value in the range $\frac{1}{3^m} \le C_k \le 1$. The closer C_k to 1, the higher confidence about the true inference by the MAP estimator is. By contrast,

a confidence rate close to $\frac{1}{3^m}$ corresponds to the poorest performance of the MAP inference (i.e., the models are not distinguishable using available data).

The confidence of the MAP inference depends on the selected perturbations (i.e., $\mathbf{u}_{0:k-1}$), which subsequently affect the sequence of states. Therefore, the sequence of perturbations should be selected so that the inferred model becomes more distinguishable from other possible models through the perturbed data. Thus, a better inference under perturbation is achieved when the posterior probability of models becomes more peaked around a single model or, equivalently, when the inferred model has the largest possible posterior probability. More formally, one needs to select the sequence of perturbations $\mathbf{u}^*_{0:k-1}$ to maximize the confidence rate of MAP inference as:

$$\begin{aligned}\mathbf{u}^*_{0:k-1} &= \underset{\mathbf{u}_{0:k-1}\in\mathcal{U}^k}{\operatorname{argmax}}\ \mathbb{E}\left[C_k \mid \mathbf{u}_{0:k-1}\right] \\ &= \underset{\mathbf{u}_{0:k-1}\in\mathcal{U}^k}{\operatorname{argmax}}\ \mathbb{E}\left[\max_{\theta\in\Theta} P(\theta \mid \mathbf{x}_{1:k}, \mathbf{u}_{0:k-1})\right],\end{aligned} \tag{6}$$

where the expectation is with respect to stochasticity in the state process. Since the true network model is unknown, finding the optimal sequence of actions in (6) without full knowledge of the system model or access to large real data is impossible. In the next paragraphs, our proposed Bayesian solution for learning optimal perturbation policy through planning without the need for real data is described.

IV. Proposed Bayesian Reinforcement Learning Data-Acquiring Policy

A. Bayesian Formulation

Given that $(\mathbf{u}_{0:k-1}, \mathbf{x}_{1:k})$ be the sequence of taken perturbations and observed states up to time step k, respectively, we represent the posterior probability of m unknown interactions through:

$$\begin{aligned}\mu_k = &\big[P(r^1 = -1 \mid \mathbf{x}_{1:k}, \mathbf{u}_{0:k-1}),\ P(r^1 = 0 \mid \mathbf{x}_{1:k}, \mathbf{u}_{0:k-1}),\\ &P(r^1 = +1 \mid \mathbf{x}_{1:k}, \mathbf{u}_{0:k-1}), \cdots,\ P(r^m = -1 \mid \mathbf{x}_{1:k}, \mathbf{u}_{0:k-1}),\\ &P(r^m = 0 \mid \mathbf{x}_{1:k}, \mathbf{u}_{0:k-1}),\ P(r^m = +1 \mid \mathbf{x}_{1:k}, \mathbf{u}_{0:k-1})\big].\end{aligned} \tag{7}$$

Note that the μ_k is a vector of size $3m$ and represents the entire uncertainty in the network model. The system uncertainty can also be expressed in terms of the network models as:

$$\vartheta_k = \Big[P(\theta^* = \theta^1 \mid \mathbf{x}_{1:k}, \mathbf{u}_{0:k-1}), \cdots, P(\theta^* = \theta^{3^m} \mid \mathbf{x}_{1:k}, \mathbf{u}_{0:k-1})\Big], \tag{8}$$

where $\vartheta_k(i)$ indicates the posterior probability that model θ^i is the true underlying system model. Note that $\sum_{i=1}^{3^m} \vartheta_k(i) = 1$, and $\vartheta_0 = [\vartheta_0(1), ..., \vartheta_0(3^m)]$ denotes prior probability of the regulatory network models. The posterior probability of models in (8) can be represented in terms of posterior of regulatory interactions μ_k in (7) as:

$$\begin{aligned}\vartheta_k(j) = \prod_{l=1}^{m} \Big[&1_{\theta^j(l)=-1}\,\mu_k(3l-2) + 1_{\theta^j(l)=0}\,\mu_k(3l-1)\\ &+ 1_{\theta^j(l)=+1}\,\mu_k(3l)\Big],\end{aligned} \tag{9}$$

for $j = 1, ..., 3^m$, where $1_{\theta^j(l)=-1}$ is 1 if the lth interacting parameter in the jth model/topology is -1 and otherwise it is zero.

The MAP inference defined in (4) can be expressed according to (9) as:

$$\theta^{\text{MAP}}_k = \underset{\theta^j : j\in\{1,\ldots,3^m\}}{\operatorname{argmax}}\ \vartheta_k(j), \tag{10}$$

with the confidence of MAP inference being:

$$C_k = \max_{j=1,\ldots,3^m} \vartheta_k(j). \tag{11}$$

B. Belief State and MDP Formulation

Let $(\mathbf{x}^1, \ldots, \mathbf{x}^{2^d})$ be an arbitrary enumeration of all possible Boolean state vectors. We define the *belief state* at time step k as the vector of joint system's state (i.e., $\mathbf{x}_k$) and posterior probability of unknown regulations μ_k:

$$\mathbf{b}_k = [\mathbf{x}_k, \mu_k]^T, \tag{12}$$

where $\mathbf{b}_k$ is a vector of size $d + 3m$, and $\mathbf{b}_0 = [\mathbf{x}_0, \mu_0]^T$ is the initial belief state. $\mathbf{x}_k$ consists of d discrete values of 0s and 1s, and μ_k includes m 3-block elements, where each element takes continuous values between 0 and 1, and sum of elements in each block is 1. Thus, the space of belief state in (12) is $\mathbb{B} = \{0,1\}^d \times (\Delta_3)^m$, which consists of joint Boolean space of size 2^d and m 3-simplexes. Thus, the belief space is infinite-dimensional due to the continuity of the simplex.

Using the definition of belief state in (12), the evolution of the regulatory network's state and posterior distribution of regulatory parameters can be represented as steps of a Markov decision process in the belief space. Given that $\mathbf{b}$ is the current belief state, and $\mathbf{u}$ is the selected perturbation at the current time, the belief state transition can be expressed as:

$$p(\mathbf{b}' \mid \mathbf{b}, \mathbf{u}) = \begin{cases} P(\mathbf{b}'_1 \mid \mathbf{b}, \mathbf{u}) & \text{If } \mathbf{b}' = \mathbf{b}'_1 = [\mathbf{x}^1, \mu'_1]^T \\ \vdots & \\ P(\mathbf{b}'_{2^d} \mid \mathbf{b}, \mathbf{u}) & \text{If } \mathbf{b}' = \mathbf{b}'_{2^d} = [\mathbf{x}^{2^d}, \mu'_{2^d}]^T \\ 0 & \text{Otherwise} \end{cases}, \tag{13}$$

where $\mu'_i(3j-2) = P(r^j = -1 \mid \mathbf{x}' = \mathbf{x}^i, \mathbf{b}, \mathbf{u})$, $\mu'_i(3j-1) = P(r^j = 0 \mid \mathbf{x}' = \mathbf{x}^i, \mathbf{b}, \mathbf{u})$ and $\mu'_i(3j) = P(r^j = +1 \mid \mathbf{x}' = \mathbf{x}^i, \mathbf{b}, \mathbf{u})$, for $j = 1, ..., m$ and $i = 1, ..., 2^d$.

It can be seen from (13) that there are 2^d possible next belief states. The probability of the ith belief state $\mathbf{b}'_i = [\mathbf{x}^i, \mu'_i]^T$ can be expressed as:

$$\begin{aligned}&P(\mathbf{b}_k = \mathbf{b}'_i \mid \mathbf{b}_{k-1} = \mathbf{b}, \mathbf{u}_{k-1} = \mathbf{u})\\ &\quad= P(\mathbf{x}_k = \mathbf{x}^i, \mu_k = \mu'_i \mid \mathbf{b}_{k-1} = \mathbf{b}, \mathbf{u}_{k-1} = \mathbf{u})\\ &\quad= P(\mathbf{x}_k = \mathbf{x}^i \mid \mathbf{x}_{k-1} = \mathbf{x}, \mu_{k-1} = \mu, \mathbf{u}_{k-1} = \mathbf{u})\\ &\qquad\times P(\mu_k = \mu'_i \mid \mathbf{x}_k = \mathbf{x}^i, \mathbf{x}_{k-1} = \mathbf{x}, \mu_{k-1} = \mu, \mathbf{u}_{k-1} = \mathbf{u})\\ &\quad= P(\mathbf{x}_k = \mathbf{x}^i \mid \mathbf{x}_{k-1} = \mathbf{x}, \mu_{k-1} = \mu, \mathbf{u}_{k-1} = \mathbf{u}).\end{aligned} \tag{14}$$

The last line of (14) is obtained given that μ_k can only take a single value μ'_i with probability 1 given $\mathbf{x}_k = \mathbf{x}^i, \mathbf{x}_{k-1} = \mathbf{x}, \mu_{k-1} = \mu, \mathbf{u}_{k-1} = \mathbf{u}$ (described below).

Further simplification of the last expression in (14) leads to the following probability of ith next belief state as:

$$\begin{aligned} &P(\mathbf{b}_k = \mathbf{b}'_i \mid \mathbf{b}_{k-1} = \mathbf{b}, \mathbf{u}_{k-1} = \mathbf{u}) \\ &= \sum_{j=1}^{3^m} P(\mathbf{x}_k = \mathbf{x}^i \mid \mathbf{x}_{k-1} = \mathbf{x}, \mathbf{u}_{k-1} = \mathbf{u}, \theta^j) P(\theta^j \mid \mu_{k-1} = \mu) \\ &= \sum_{j=1}^{3^m} (1-p)^{d-\|\mathbf{x}^i \oplus \mathbf{f}_{\theta^j}(\mathbf{x}) \oplus \mathbf{u}\|_1} p^{\|\mathbf{x}^i \oplus \mathbf{f}_{\theta^j}(\mathbf{x}) \oplus \mathbf{u}\|_1} \vartheta_{k-1}(j), \end{aligned} \tag{15}$$

where according to (9)

$$\vartheta_{k-1}(j) = \prod_{l=1}^{m} \Big[1_{\theta^j(l)=-1} \mu(3l-2) + 1_{\theta^j(l)=0} \mu(3l-1) + 1_{\theta^j(l)=+1} \mu(3l) \Big].$$

The value of μ'_i in $\mathbf{b}'_i = [\mathbf{x}^i, \mu'_i]$ can also be expressed as:

$$\begin{aligned} \mu'_i(3l-2) &= \sum_{j=1}^{3^m} 1_{\theta^j(l)=-1} \vartheta'_i(j), \\ \mu'_i(3l-1) &= \sum_{j=1}^{3^m} 1_{\theta^j(l)=0} \vartheta'_i(j), \\ \mu'_i(3l) &= \sum_{j=1}^{3^m} 1_{\theta^j(l)=+1} \vartheta'_i(j), \end{aligned} \tag{16}$$

where

$$\begin{aligned} \vartheta'_i(j) &= P(\theta^j \mid \mathbf{x}_k = \mathbf{x}^i, \mathbf{x}_{k-1} = \mathbf{x}, \mu_{k-1} = \mu, \mathbf{u}_{k-1} = \mathbf{u}) \\ &= \frac{P(\mathbf{x}_k = \mathbf{x}^i \mid \mathbf{x}_{k-1} = \mathbf{x}, \mathbf{u}_{k-1} = \mathbf{u}, \theta^j) \vartheta_{k-1}(j)}{\sum_{n=1}^{3^m} P(\mathbf{x}_k = \mathbf{x}^i \mid \mathbf{x}_{k-1} = \mathbf{x}, \mathbf{u}_{k-1} = \mathbf{u}, \theta^n) \vartheta_{k-1}(n)} \\ &= \frac{(1-p)^{d-\|\mathbf{x}^i \oplus \mathbf{f}_{\theta^j}(\mathbf{x}) \oplus \mathbf{u}\|_1} p^{\|\mathbf{x}^i \oplus \mathbf{f}_{\theta^j}(\mathbf{x}) \oplus \mathbf{u}\|_1} \vartheta_{k-1}(j)}{\sum_{n=1}^{3^m} (1-p)^{d-\|\mathbf{x}^i \oplus \mathbf{f}_{\theta^n}(\mathbf{x}) \oplus \mathbf{u}\|_1} p^{\|\mathbf{x}^i \oplus \mathbf{f}_{\theta^n}(\mathbf{x}) \oplus \mathbf{u}\|_1} \vartheta_{k-1}(n)}. \end{aligned} \tag{17}$$

C. Reinforcement Learning Formulation of Data-Acquiring Process

Using the concept of belief state, the data-acquiring process can be seen as steps of an MDP in the belief space. As described in the following paragraphs, the MDP representation enables reinforcement learning formulation of the perturbation process and consequently finding the near-optimal Bayesian perturbation policy. The immediate reward function $R : \mathbb{B} \times \mathcal{U} \times \mathbb{B}$ can be expressed in terms of enhancing the inference accuracy, where $R(\mathbf{b}, \mathbf{u}, \mathbf{b}')$ represents the change in the confidence of the MAP inference when system moves from belief state $\mathbf{b}$ to belief state $\mathbf{b}'$ upon taking the perturbation $\mathbf{u}$. The reward function can be represented as:

$$\begin{aligned} R(\mathbf{b}, \mathbf{u}, \mathbf{b}') = &\prod_{l=1}^{m} \max\{\mathbf{b}'(d+3l-2), \mathbf{b}'(d+3l-1), \mathbf{b}'(d+3l)\} \\ &- \prod_{l=1}^{m} \max\{\mathbf{b}(d+3l-2), \mathbf{b}(d+3l-1), \mathbf{b}(d+3l)\}. \end{aligned} \tag{18}$$

The above reward function quantifies a single-step change in the confidence of the MAP inference, e.g., $C_k - C_{k-1}$. The positive values of the reward correspond to cases with more peaked posterior distribution upon the last perturbation, whereas negative values represent cases with less peaked posterior probability after taking the last perturbation. Note that as an alternative, a more complicated and time-dependent reward function can be incorporated into the proposed policy. For instance, in domains with the varied cost of perturbations, the cost of perturbations can also be incorporated into the reward function. Meanwhile, if the objective is to accurately infer a single or a subset of unknown interactions (as opposed to all unknown interactions), this can also be incorporated into the reward function.

Let $\pi : \mathbb{B} \to \mathcal{U}$ be a deterministic policy, which associates a perturbation to each sample in the belief space. The expected discounted reward function at belief state $\mathbf{b} \in \mathbb{B}$ after taking perturbation $\mathbf{u} \in \mathcal{U}$ and following policy π afterward is defined as:

$$Q^{\pi}(\mathbf{b}, \mathbf{u}) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(\mathbf{b}_t, \mathbf{u}_t, \mathbf{b}_{t+1}) \mid \mathbf{b}_0 = \mathbf{b}, \mathbf{u}_0 = \mathbf{u}, \pi \right], \tag{19}$$

where $0 \le \gamma \le 1$ is the discount factor, and the expectation is taken with respect to the uncertainty in the belief transition. The optimal Q-function, denoted by Q^*, provides the maximum expected return, where $Q^{\pi^*}(\mathbf{b}, \mathbf{u})$ indicates the expected discounted reward after taking perturbation $\mathbf{u}$ in belief state $\mathbf{b}$ and following optimal policy π^* afterward. An optimal stationary policy π^* attains the maximum expected return for all states as: $\pi^*(\mathbf{b}) = \operatorname{argmax}_{\mathbf{u} \in \mathcal{U}} Q^{\pi^*}(\mathbf{b}, \mathbf{u})$. It should be noted that $\pi^*(\mathbf{b})$ makes a decision according to the current belief state, which includes the current state of the system (i.e., $\mathbf{x}_k$) as well as the posterior probability of unknown regulations (i.e., μ_k). Therefore, the policy in the belief state is the optimal Bayesian perturbation policy. This Bayesian policy guarantees optimal perturbation of regulatory networks given all available information and uncertainty reflected in the belief state.

D. Linear Q-Function Approximations in Belief Space

The exact computation of Q^* and consequently π^* is not possible since the belief space has an infinite dimension. In this paper, we propose a function approximation for representing the Q-function in (19), and an efficient approximation of the optimal policy. Let $\boldsymbol{\phi}(\mathbf{b}, \mathbf{u})$ be a set of basis function defined according to the belief state $\mathbf{b} \in \mathbb{B}$ and perturbation $\mathbf{u} \in \mathcal{U}$. The Q-function can be approximated as:

$$Q(\mathbf{b}, \mathbf{u}) \approx \hat{Q}(\mathbf{b}, \mathbf{u}) = \boldsymbol{\phi}^T(\mathbf{b}, \mathbf{u})\, \mathbf{w}, \tag{20}$$

where $\mathbf{w}$ is a column weight vector of the same size as the basis function. The basis function could contain a nonlinear combination of the belief state and perturbation. This representation of the Q-function allows learning the policy in a large and continuous belief space. More information about the choice of basis function is provided in our numerical experiments in Section V.

Learning the near-optimal data-acquiring policy, in this case, consists of finding the weight vector $\mathbf{w}^*$ that can approximate $Q^{\pi^*}(.,.)$. We employ the semi-gradient SARSA algorithm for learning the weights [38]. Given that the perturbation space is $\mathcal{U} = \{\mathbf{u}^1, ..., \mathbf{u}^L\}$, the epsilon-greedy

policy at belief state $\mathbf{b}_t$ can be defined as:

$$\pi_{\mathbf{w}}^{\epsilon\text{-greedy}}(\mathbf{b}_t) = \begin{cases} \underset{\mathbf{u}\in\mathcal{U}}{\operatorname{argmax}} \boldsymbol{\phi}(\mathbf{b}_t, \mathbf{u})^T \mathbf{w} & \text{w.p. } 1-\epsilon \\ \text{random}\{\mathbf{u}^1, ..., \mathbf{u}^L\} & \text{w.p. } \epsilon \end{cases}, \quad (21)$$

where $0 \leq \epsilon \leq 1$ controls the level of exploration during the learning process.

At any given episode, we start from an initial belief state $\mathbf{b}_0 = [\mathbf{x}_0, \mu_0]^T$. If the initial belief is not known, the initial belief can be selected randomly from the belief space, i.e., $\mathbf{b}_0 \in \mathbb{B}$. Let $\mathbf{w}^-$ be the current weights, and $\mathbf{b}_t$ and $\mathbf{u}_t$ be the belief and perturbation at step t. A realization of the next belief state $\mathbf{b}_{t+1}$ can be obtained according to the belief transition probabilities in (13). The next perturbation, $\mathbf{u}_{t+1}$, at belief state $\mathbf{b}_{t+1}$ can be calculated according to the epsilon-greedy policy with $\mathbf{w}^-$ as:

$$\mathbf{u}_{t+1} \sim \pi_{\mathbf{w}^-}^{\epsilon\text{-greedy}}(\mathbf{b}_{t+1}). \quad (22)$$

The reward value for this transition can be obtained according to (18) as $r_{t+1} = R(\mathbf{b}_t, \mathbf{u}_t, \mathbf{b}_{t+1})$. The created $(\mathbf{b}_t, \mathbf{u}_t, \mathbf{b}_{t+1}, \mathbf{u}_{t+1}, r_{t+1})$ at each step of episode can be used for updating the weight vector. This can be achieved by minimizing the temporal difference error using the following stochastic gradient descent procedure [38]:

$$\mathbf{w}^+ = \mathbf{w}^- + \alpha \Big[r_{t+1} + \gamma \boldsymbol{\phi}^T(\mathbf{b}_{t+1}, \mathbf{u}_{t+1}) \mathbf{w}^- - \boldsymbol{\phi}^T(\mathbf{b}_t, \mathbf{u}_t) \mathbf{w}^- \Big] \boldsymbol{\phi}(\mathbf{b}_t, \mathbf{u}_t), \quad (23)$$

where α is the learning rate. The process starts with the weights being zero, then the weights are iteratively updated according to the belief transitions and perturbation obtained according to the epsilon-greedy policy. The process consists of multiple episodes of fixed length, where the learning stops when the average accumulated reward in the last episodes meets the desired inference performance, or when changes in the weights over consecutive episodes become insignificant.

Upon termination of the learning process and finding final weights $\mathbf{w}^*$, the near-optimal Bayesian perturbation policy at any given belief state $\mathbf{b}$ can be computed using the greedy form of the ϵ-greedy policy in (22) as:

$$\pi^*(\mathbf{b}) = \underset{\mathbf{u}\in\mathcal{U}}{\operatorname{argmax}} \hat{Q}_{\mathbf{w}^*}(\mathbf{b}, \mathbf{u}) = \underset{\mathbf{u}\in\mathcal{U}}{\operatorname{argmax}} \boldsymbol{\phi}(\mathbf{b}, \mathbf{u})^T \mathbf{w}^*. \quad (24)$$

The belief transitions can be created through planning without the need for any real data, and the weight updates (i.e., learning process) in (23) can be achieved efficiently and sequentially, without requiring any gradient computations. Therefore, the proposed method can be employed for learning data-acquiring policy of GRNs with a possibly large number of unknowns (i.e., large belief or perturbation spaces).

V. Numerical Experiments

The numerical experiments in this section evaluate the performance of the proposed framework using an example of GRNs with unknown regulations. At first, we will investigate the performance of our approach on a GRN with four genes and two unknown regulations. We will further show the effectiveness of our obtained perturbation policy on the same network, but in a case where we have four unknown interactions. All the results provided in the numerical experiments are averaged over 1000 trials.

The well-known *p53-Mdm2* negative-feedback gene regulatory network [39] is considered for assessing the performance of the proposed framework. This regulatory network is responsible for coding the tumor suppressor protein p53 in human bodies. The p53 gene activation has a vital role in cellular responses to different stress signals that might lead to genome instability. The pathway diagram of the p53-Mdm2 network is shown in Figure 1. The blunt and normal arrows define suppressive and activating regulations, respectively. This gene regulatory network consists of four genes, represented in a state vector $\mathbf{x}_k = [\textit{ATM}, \textit{p53}, \textit{Wip1}, \textit{MDM2}]^T$. Since each of these four genes can take values of 0 or 1, there will be a total of $2^4 = 16$ possible states. The connectivity matrix for this GRN can be extracted from the pathway diagram in Figure 1 as:

$$\mathbf{R} = \begin{bmatrix} 0 & 0 & -1 & 0 \\ +1 & 0 & -1 & -1 \\ 0 & +1 & 0 & 0 \\ -1 & +1 & +1 & 0 \end{bmatrix}. \quad (25)$$

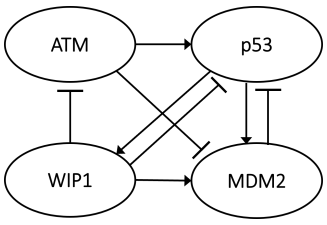

Fig. 1: Activation/repression pathway diagram for the p53-Mdm2 negative feedback loop Boolean network model.

We assume the unknown parts of regulatory networks are regulatory interactions in the connectivity matrix $\mathbf{R}$ in (25). A uniform prior probability is considered for each interaction, where the probability of taking -1, 0, or $+1$ is $\frac{1}{3}$. The process noise $p = 0.001$ is used for our numerical experiments. The perturbation space includes $\mathcal{U} = \{\mathbf{u}^1 = [1,0,0,0]^T, \mathbf{u}^2 = [0,1,0,0]^T, \mathbf{u}^3 = [0,0,1,0]^T, \mathbf{u}^4 = [0,0,0,1]^T\}$, which each flips the value of a single gene at a time. This type of perturbation takes place in practice using drugs often designed to alter the activity of a single gene. For our proposed method, a maximum of 50 perturbations is assumed for the horizon length while performing the testing. Therefore, a larger episode length of 100 is used for our experiments during training/planning to account for discounted rewards in the last perturbation steps. The other hyperparameters used in our experiments are as follows: $\alpha = 10^{-3}$, $\gamma = 0.9$, and $\epsilon = 0.1$. The performance of the proposed policy is compared with the random perturbation policy and no-perturbation case. To the best of the authors' knowledge, most practical perturbations are performed using a trial-and-error process according to biologists' knowledge. Therefore, currently, there is no existing approach

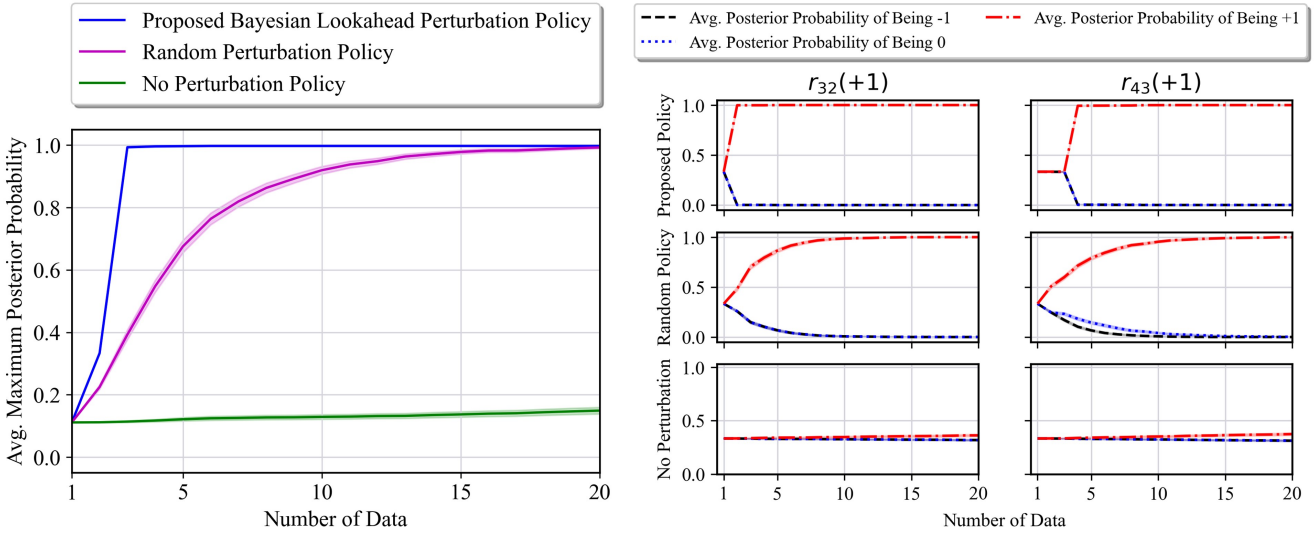

Fig. 2: Average results for the p53-Mdm2 network with two +1 unknown interactions: (a) performance comparison of different policies ; (b) progress of posterior probability of unknown regulations (μ_k).

for systematically perturbing regulatory networks to enhance inference performance. We also consider the no-perturbation case to demonstrate the non-identifiability of the regulatory networks under no perturbation and, moreover, to show the gain achieved through systematic perturbation.

In the first experiment, we consider two +1 unknown interactions from the true model as: $r_{32} = +1$ and $r_{43} = +1$. Two unknown interactions lead to $3^2 = 9$ possible network models. The belief state for this scenario is $\mathbf{b}_k = [\mathbf{x}_k, \mu_k]^T$, which is a vector of size $4+3\times 2 = 10$, and with belief space $\mathbb{B} = \{0,1\}^4 \times (\Delta_3)^2$. This means that at each time step, each gene's state can be either 0 or 1, and the first and second 3-block elements of μ_k can take any continuous value between 0 and 1, summing up to 1. Thus, one can understand the large space of the belief state in this scenario.

We consider the following basis function for our first experiment:

$$\boldsymbol{\phi}(\mathbf{b},\mathbf{u}) = \begin{bmatrix} \tilde{\boldsymbol{\phi}}(\mathbf{b},\mathbf{u}) \\ \lfloor \tilde{\boldsymbol{\phi}}(\mathbf{b},\mathbf{u}) \otimes \tilde{\boldsymbol{\phi}}(\mathbf{b},\mathbf{u}) \rfloor \\ \lfloor \tilde{\boldsymbol{\phi}}(\mathbf{b},\mathbf{u}) \otimes \tilde{\boldsymbol{\phi}}(\mathbf{b},\mathbf{u}) \otimes \tilde{\boldsymbol{\phi}}(\mathbf{b},\mathbf{u}) \rfloor \end{bmatrix} \tag{26}$$

where $\tilde{\boldsymbol{\phi}}(\mathbf{b},\mathbf{u}) = [\mathbf{b}, \sum_{l=1}^{4} l 1_{\mathbf{u}=\mathbf{u}^l}]^T$, $\otimes$ is the outer product, and $\lfloor . \rfloor$, denotes the vectorized representation of a matrix. Note that this basis function contains all the combinations of linear, quadratic, and cubic terms of belief state and perturbation. This nonlinear basis function, along with the weights are used to approximate the Q-function over the belief and perturbation space. The weights are then sequentially adjusted using semi-gradient reinforcement learning to represent near-optimal data-acquiring policy.

In the first scenario, the proposed perturbation policy is trained over 3500 episodes. Figure 2 (a) represents the average results and their 95% confidence bounds for all methods during the perturbation of the true model over 20 time steps. The y-axis shows the average maximum posterior probability obtained over all the possible models. We can see that the regulatory network models under no perturbation are not distinguishable from each other, as the maximum posterior probability stays very small independent of the number of data. For the system under the random perturbation policy, one can see the increase in the maximum posterior probability with respect to the number of data. The reason is that random perturbation (which is not systematically/optimally assigned) helps the system to come out of attractor states and helps the causal inference of the unknown regulatory parameters. Finally, the best results are achieved by the proposed policy, where the maximum posterior has significantly increased even with a small number of data. This clearly illustrates the superiority of the proposed framework in selecting the perturbation sequence, especially for small data sizes.

The average posterior probability of unknown interactions with respect to the number of data (i.e., number of perturbations) is also visualized in Figure 2 (b). The two columns in the subplots correspond to two unknown interactions. Further, the subplots in the first, second, and third rows are associated with our proposed perturbation policy, random perturbation policy, and no-perturbation case, respectively. Moreover, the posterior probability that unknown interactions are +1, 0, and -1 are indicated by red, blue and black curves, respectively. For instance, the subplot in the first row and second column represents the posterior probability of interaction $r_{43}(+1)$ under the proposed perturbation policy. In subplots of the first row, one can see that the average posterior probability of the true interaction has quickly approached 1. However, a slower increase in the curves can be seen in the middle row subplots under the random perturbation policy. In this specific case, one can see that the average posterior probability of r_{43} has a slower increase relative

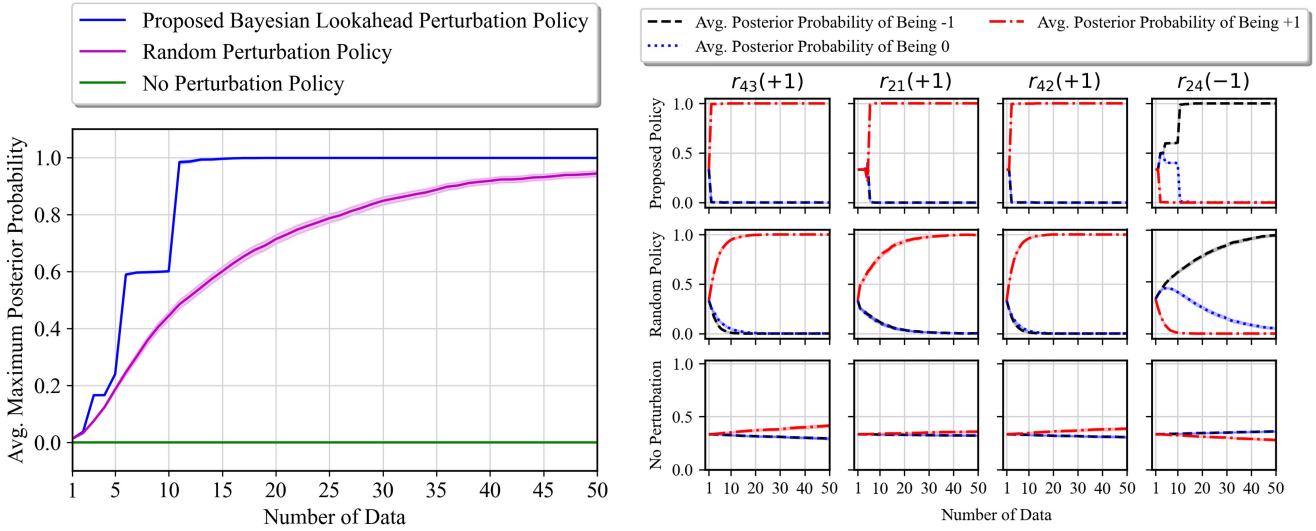

Fig. 3: Average results for the p53-Mdm2 network with three +1 and one -1 unknown interactions: (a) performance comparison of different policies ; (b) progress of posterior probability of unknown regulations (μ_k).

to the other unknown interaction r_{32} under the proposed policy and the random perturbation policy. Finally, as it can be seen in the last row of the subplots, the average posterior probability for all interactions stays around $\frac{1}{3}$, similar to their prior probabilities. This indicates the poor performance of inference, and difficulty of distinguishing the true topology using the data from no-perturbation case.

In the second part of our experiments, we consider a more complex scenario with three +1 and one -1 unknown interactions as: $r_{43} = +1$, $r_{21} = +1$, $r_{42} = +1$, and $r_{24} = -1$. This scenario is considered to be more challenging due to the following two reasons: 1) On one hand, this scenario has four unknown interactions, which leads to $3^4 = 81$ possible network models; 2) On the other hand, including -1 interactions makes the causal inference of regulatory networks more challenging, primarily due to the attractor structure of GRNs, and the fact that most genes spend their time at rest (inactivated state) under no-perturbation.

In this case, we used the following basis function for Q-function approximation:

$$\boldsymbol{\phi}(\mathbf{b},\mathbf{u}) = \begin{bmatrix} \begin{bmatrix} 1 \\ \tilde{\boldsymbol{\phi}}(\mathbf{b},\mathbf{u}) \\ \lfloor \tilde{\boldsymbol{\phi}}(\mathbf{b},\mathbf{u}) \otimes \tilde{\boldsymbol{\phi}}(\mathbf{b},\mathbf{u}) \rfloor \\ \lfloor \tilde{\boldsymbol{\phi}}(\mathbf{b},\mathbf{u}) \otimes \tilde{\boldsymbol{\phi}}(\mathbf{b},\mathbf{u}) \otimes \tilde{\boldsymbol{\phi}}(\mathbf{b},\mathbf{u}) \rfloor \end{bmatrix} \otimes \begin{bmatrix} 1_{\mathbf{u}=\mathbf{u}^1} \\ 1_{\mathbf{u}=\mathbf{u}^2} \\ 1_{\mathbf{u}=\mathbf{u}^3} \\ 1_{\mathbf{u}=\mathbf{u}^4} \end{bmatrix} \end{bmatrix}, \tag{27}$$

where $\tilde{\boldsymbol{\phi}}(\mathbf{b} = [\mathbf{x},\mu],\mathbf{u}) = [2\mathbf{x} - 1, \mu, 1.1 - \max\vartheta]^T$, and ϑ is the posterior probability of models corresponding to μ computable using (9). This basis function is much more complex than the previous one, and has been shown to be an effective choice for this complex scenario.

The proposed perturbation policy in this case is trained over 17,500 episodes. Figure 3 (a) represents the average results of all methods over 50 time steps. Similar to the previous case, the maximum posterior probability of models under no perturbation stays very small even with larger data. For the random perturbation policy, the average maximum posterior probability stays smaller than the previous experiment. From Figure 3 (a), one can see that the average maximum posterior probability does not get to 1 even after 50 randomly perturbed data. In contrast, much larger performance is achieved under the proposed perturbation policy. The average maximum posterior probability under the proposed policy is increased to 0.6 after about 6 perturbations, and it has increased to almost 1 after only 11 perturbations. This demonstrates the capability of the proposed framework in systematically choosing the perturbation sequence and increasing the performance of the inference process.

Finally, Figure 3 (b) shows the average posterior probability of the four unknown interactions. We can observe that in the subplots of the first row, which correspond to the proposed policy, the average posterior probabilities of the true interactions have quickly reached 1. By comparing these results with the ones in the second and the third row, one can see how effective the proposed approach is, especially with fewer data. In particular, the subplot in the first row and fourth column, which shows the posterior probability of interaction $r_{24}(-1)$ under the proposed perturbation policy, gets to 1 in only 11 time steps. However, as can be seen, it takes about 50 data for the random policy to get close to 1.

VI. Conclusion

This paper developed a reinforcement learning data-acquiring policy for causal inference of gene regulatory networks (GRNs) under uncertainty. A Boolean network with perturbation (BNp) model is considered for representing the GRNs. The unknown interactions between genes represent partial knowledge about the model of GRNs. We use maximum a posteriori (MAP) as a criterion for inference of unknown regulatory interactions in GRN models. The

data-acquiring is used for selecting the best perturbations for flipping the state value of targeted genes at any given time to maximize the confidence of the MAP inference. We introduced a semi-gradient reinforcement learning method with function approximation for learning near-optimal data-acquiring policy through planning without the need for real data. Eventually, we demonstrated the high performance of the proposed policy through a set of numerical experiments.

ACKNOWLEDGMENT

The authors acknowledge the support of the National Institute of Health award 1R21EB032480-01, National Science Foundation award IIS-2202395, ARMY Research Office award W911NF2110299, and Oracle Cloud credits and related resources provided by the Oracle for Research program.

REFERENCES

[1] E. H. Davidson, *The regulatory genome: gene regulatory networks in development and evolution*. Elsevier, 2010.

[2] I. Shmulevich, E. R. Dougherty, and W. Zhang, "From Boolean to probabilistic Boolean networks as models of genetic regulatory networks," *Proceedings of the IEEE*, vol. 90, no. 11, pp. 1778–1792, 2002.

[3] Z. Zou, H. Chen, P. Poduval, Y. Kim, M. Imani, E. Sadredini, R. Cammarota, and M. Imani, "BioHD: an efficient genome sequence search platform using HyperDimensional memorization," in *Proceedings of the 49th Annual International Symposium on Computer Architecture*, pp. 656–669, 2022.

[4] M. Imani, M. Imani, and S. F. Ghoreishi, "Optimal Bayesian biomarker selection for gene regulatory networks under regulatory model uncertainty," in *American Control Conference (ACC)*, IEEE, 2022.

[5] F. He, X. Ru, and T. Wen, "NRF2, a transcription factor for stress response and beyond," *International Journal of Molecular Sciences*, vol. 21, no. 13, p. 4777, 2020.

[6] M. Banf and S. Y. Rhee, "Computational inference of gene regulatory networks: approaches, limitations and opportunities," *Biochimica et Biophysica Acta (BBA)-Gene Regulatory Mechanisms*, vol. 1860, no. 1, pp. 41–52, 2017.

[7] K. Y. Yip, R. P. Alexander, K.-K. Yan, and M. Gerstein, "Improved reconstruction of in silico gene regulatory networks by integrating knockout and perturbation data," *PloS one*, vol. 5, no. 1, p. e8121, 2010.

[8] G. Krouk, J. Lingeman, A. M. Colon, G. Coruzzi, and D. Shasha, "Gene regulatory networks in plants: learning causality from time and perturbation," *Genome biology*, vol. 14, no. 6, pp. 1–7, 2013.

[9] J. W. Freimer, O. Shaked, S. Naqvi, N. Sinnott-Armstrong, A. Kathiria, C. M. Garrido, A. F. Chen, J. T. Cortez, W. J. Greenleaf, J. K. Pritchard, *et al.*, "Systematic discovery and perturbation of regulatory genes in human T cells reveals the architecture of immune networks," *Nature Genetics*, pp. 1–12, 2022.

[10] O. Wolkenhauer, P. Wellstead, K.-H. Cho, J. R. Banga, and E. Balsa-Canto, "Parameter estimation and optimal experimental design," *Essays in biochemistry*, vol. 45, pp. 195–210, 2008.

[11] M. Andrec, B. N. Kholodenko, R. M. Levy, and E. Sontag, "Inference of signaling and gene regulatory networks by steady-state perturbation experiments: structure and accuracy," *Journal of theoretical biology*, vol. 232, no. 3, pp. 427–441, 2005.

[12] F. Steinke, M. Seeger, and K. Tsuda, "Experimental design for efficient identification of gene regulatory networks using sparse bayesian models," *BMC systems biology*, vol. 1, no. 1, pp. 1–15, 2007.

[13] A. Fiedler, S. Raeth, F. J. Theis, A. Hausser, and J. Hasenauer, "Tailored parameter optimization methods for ordinary differential equation models with steady-state constraints," *BMC systems biology*, vol. 10, no. 1, pp. 1–19, 2016.

[14] S. M. Ud-Dean and R. Gunawan, "Optimal design of gene knockout experiments for gene regulatory network inference," *Bioinformatics*, vol. 32, no. 6, pp. 875–883, 2015.

[15] Z. Dong, T. Song, and C. Yuan, "Inference of gene regulatory networks from genetic perturbations with linear regression model," *PloS one*, vol. 8, no. 12, p. e83263, 2013.

[16] X. Cai, J. A. Bazerque, and G. B. Giannakis, "Inference of gene regulatory networks with sparse structural equation models exploiting genetic perturbations," *PLoS computational biology*, vol. 9, no. 5, p. e1003068, 2013.

[17] A. Feiglin, A. Hacohen, A. Sarusi, J. Fisher, R. Unger, and Y. Ofran, "Static network structure can be used to model the phenotypic effects of perturbations in regulatory networks," *Bioinformatics*, vol. 28, no. 21, pp. 2811–2818, 2012.

[18] T. E. Ideker, V. Thorsson, and R. M. Karp, "Discovery of regulatory interactions through perturbation: inference and experimental design," in *Pacific symposium on biocomputing*, vol. 5, pp. 302–313, 2000.

[19] T. Akutsu, S. Kuhara, O. Maruyama, and S. Miyano, "Identification of genetic networks by strategic gene disruptions and gene overexpressions under a Boolean model," *Theoretical Computer Science*, vol. 298, no. 1, pp. 235–251, 2003.

[20] S. M. Ud-Dean and R. Gunawan, "Optimal design of gene knockout experiments for gene regulatory network inference," *Bioinformatics*, vol. 32, no. 6, pp. 875–883, 2016.

[21] J. Zhong, Y. Liu, J. Lu, and W. Gui, "Pinning control for stabilization of Boolean networks under knock-out perturbation," *IEEE Transactions on Automatic Control*, vol. 67, no. 3, pp. 1550–1557, 2021.

[22] I. Shmulevich, E. R. Dougherty, and W. Zhang, "Gene perturbation and intervention in probabilistic Boolean networks," *Bioinformatics*, vol. 18, no. 10, pp. 1319–1331, 2002.

[23] S. Tong and D. Koller, "Active learning for structure in Bayesian networks," in *International joint conference on artificial intelligence*, vol. 17, pp. 863–869, Citeseer, 2001.

[24] D. Heckerman, "A Bayesian approach to learning causal networks," *arXiv preprint arXiv:1302.4958*, 2013.

[25] M. Imani and S. F. Ghoreishi, "Graph-based Bayesian optimization for large-scale objective-based experimental design," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[26] K. Shanmugam, M. Kocaoglu, A. G. Dimakis, and S. Vishwanath, "Learning causal graphs with small interventions," *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[27] A. E. Allahverdyan and A. Galstyan, "Active inference for binary symmetric hidden Markov models," *Journal of Statistical Physics*, vol. 161, no. 2, pp. 452–466, 2015.

[28] M. Imani and S. F. Ghoreishi, "Optimal finite-horizon perturbation policy for inference of gene regulatory networks," *IEEE Intelligent Systems*, 2020.

[29] I. Shmulevich and E. R. Dougherty, *Probabilistic Boolean networks: the modeling and control of gene regulatory networks*. SIAM, 2010.

[30] M. Imani, R. Dehghannasiri, U. M. Braga-Neto, and E. R. Dougherty, "Sequential experimental design for optimal structural intervention in gene regulatory networks based on the mean objective cost of uncertainty," *Cancer informatics*, vol. 17, 2018.

[31] A. Ravari, S. F. Ghoreishi, and M. Imani, "Optimal recursive expert-enabled inference in regulatory networks," *IEEE Control Systems Letters*, vol. 7, pp. 1027–1032, 2023.

[32] M. Alali and M. Imani, "Inference of regulatory networks through temporally sparse data," *Frontiers in control engineering*, vol. 3, 2022.

[33] M. Imani and S. F. Ghoreishi, "Bayesian optimization objective-based experimental design," in *2020 American Control Conference (ACC)*, pp. 3405–3411, IEEE, 2020.

[34] S. F. Ghoreishi and M. Imani, "Offline fault detection in gene regulatory networks using next-generation sequencing data," in *53rd Asilomar Conference on Signals, Systems and Computers*, IEEE, 2019.

[35] M. Imani and S. F. Ghoreishi, "Adaptive real-time filter for partially-observed Boolean dynamical systems," in *46th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2021.

[36] 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.

[37] A. Garg, I. Xenarios, L. Mendoza, and G. DeMicheli, "An efficient method for dynamic analysis of gene regulatory networks and in silico gene perturbation experiments," in *Annual International Conference on Research in Computational Molecular Biology*, pp. 62–76, Springer, 2007.

[38] R. S. Sutton and A. G. Barto, "Introduction to reinforcement learning," 1998.

[39] 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, pp. 371–377, 2009.