Inferring Causal Protein Signaling Networks with Reinforcement Learning via Artificial Bee Colony Neural Architecture Search

Jihao Zhai, Junzhong Ji, Jinduo Liu*
Beijing University of Technology
zhaijihao@emails.bjut.edu.cn, jjz01@bjut.edu.cn, jinduo@bjut.edu.cn

Abstract

Inferring causal protein signaling networks from human immune system cellular data is an important approach to reveal underlying tissue signaling biology and dysfunction in diseased cells. In recent years, reinforcement learning (RL) methods have shown excellent performance in the field of causal protein signaling network inference. However, the complexity of RL models and the need for manual hyperparameter tuning can hinder performance. In this paper, we propose an actor-critic RL model via artificial bee colony (ABC) neural architecture search, called ABCNAS-RL. Specifically, the entire method is divided into two phases: ABC neural architecture search and actor-critic RL search. In phase one, we represent each bee as a set of hyperparameter, utilizing the ABC algorithm searching for optimal hyperparameters of the actor-critic RL model on the training set. In phase two, we use the actor-critic RL model to infer the causal protein signaling network on the test set. The actor network consists of an encoder-decoder architecture, composed of a transformer and a bidirectional gated recurrent unit (BiGRU) with an integrated attention mechanism. The critic network consists of a fully connected neural network that estimates the output state of the actor network. By maximizing cumulative rewards, we ultimately derive the causal protein signaling network. Extensive experimental results on simulated and real datasets verify that ABCNAS-RL outperforms the comparison methods and has superior performance.

1 Introduction

Causal protein signaling networks represent the impact of pathway components on different biomolecules [Sachs *et al.*, 2005a], reflecting the complex stochastic relationships among multiple interacting molecules derived from biological data [Western *et al.*, 2024; Cai *et al.*, 2024]. The creation of intracellular multicolor flow cytometry allows more quantitative simultaneous observations of multiple signaling molecules in many thousands of individual cells and making it easier to infer causal protein signaling network among protein biomolecules [Shah *et al.*, 2024]. The inference of causal protein signaling network from human immune system cell data has gained significant attention in bioinformatics, as it facilitates the proper understanding of normal cellular responses and their potential disruption in disease.

In recent years, numerous methods have been proposed for inferring causal protein signaling network, which can be broadly classified into two categories, one based on traditional machine learning and the other on deep learning methods with complex model structures. For example, the greedy equivalence search (GES) [Chickering, 2002] is a widely utilized score-based method that explores equivalence classes of directed acyclic graphs [Liu *et al.*, 2022]. Zheng et al. [Zheng *et al.*, 2018] first transformed the causal discovery problem from a combinatorial optimization problem to a continuous optimization problem by proposing a continuous optimization structure approach (NoTears) and successfully used it for learning causal protein signaling network. Liu et al. [Liu *et al.*, 2024b] utilized a parallel discrete ABC algorithm for inferring causal protein signaling network from single-cell data. Guo et al. [Guo *et al.*, 2024] proposed FedCSL, a scalable and accurate method for federated causal protein signaling network learning. This group of methods has simple model structures and runs efficiently, but the accuracy of the inferred causal protein signaling network needs to be improved.

For deep learning methods, Zhu et al. [Zhu *et al.*, 2019] developed a causal learning model based on RL that employs a RL framework for the learning of causal protein signaling network. Löwe et al. [Löwe *et al.*, 2022] introduced a new framework called amortized causal discovery, which utilizes shared dynamics to facilitate the learning of causal relationships from data. Tristan et al. [Deleu *et al.*, 2022] proposed a deep generative model that uses a flow network generation method and employs a structural constraint variant to learn causal protein signaling network. Sanchez [Sanchez *et al.*, 2023] proposed DiffAN, a novel identifiable algorithm that utilizes a diffusion probabilistic model to establish a topological ordering, enabling causal discovery under the assumption of an additive noise model. These types of methods can more accurately infer causal protein signaling network, but they typically have higher time complexity and longer run times due to their complex neural network model structures.

*Corresponding Author

In this paper, we propose an actor-critic RL via ABC neural architecture search to infer causal protein signaling network, called ABCNAS-RL. Specifically, we first use the ABC algorithm and cross-entropy loss function to automatically optimize the optimal hyperparameters of the actor-critic RL model, and the optimization process consists of three phases of employed bee search, onlooker bee search, and scout bee search, and each bee corresponds to a set of hyperparameter combinations. Then we use the actor-critic RL model to infer the network structure, the actor adopts a encoder-decoder structure, the encoder adopts a transformer model structure, and the decoder adopts a BiGRU incorporating an attention mechanism. The critic network uses a fully connected layer neural network to estimate the output state of the actor network to output the causal protein signaling network with the highest reward score. Extensive experimental results on simulated and real datasets verify that ABCNAS-RL outperforms the comparison methods and has superior performance.

The key contributions of this paper are summarized below:

- This is the first study to infer causal protein signaling network with RL via ABC neural architecture search, which will provide a significant reference for the causal discovery and bioinformatics fields.
- To minimize the performance impact due to manual hyperparameter selection, we design an ABC neural architecture to adaptively search for optimal hyperparameters of actor-critic RL models.
- To better infer the causal structures and increase feature extraction capability, we add BiGRU with attention mechanism into the RL framework.
- Extensive experiments demonstrate that ABCNAS-RL can infer causal protein signaling networks more accurately, which has significant implications for understanding the underlying causal relationships in biological systems.

2 Related Work

2.1 Causal Protein Signaling Networks

Causal protein signaling network consist of multiple protein biomolecule nodes and causal relationships between different nodes. Inferring causal protein signaling network accurately from Single-cell data is important for understanding the causal relationships of biomolecules in cells and for gaining insight into the pathogenesis of cell-based diseases [Wang *et al.*, 2024]. Recently, Zhang et al. [Zhang *et al.*, 2023] developed a method for learning causal protein signaling network from observed numerical data. Their approach utilizes a regression-based conditional independence test (RCIT) that combines kernel ridge regression and the Hilbert-Schmidt independence criterion with permutation approximation.

2.2 Reinforcement Learning

Reinforcement learning (RL) utilizes neural networks to model the policy and value functions, using backpropagation to optimize the objective. It also employs RL's decision-making ability to define and optimize goals [Lee *et al.*, 2024]. The Actor-Critic algorithm [Konda and Tsitsiklis, 1999] is a popular reinforcement learning framework that combines policy-based and value-based methods. The Actor selects actions based on a policy, while the Critic evaluates the chosen actions and provides feedback. The Actor updates the policy using this feedback to improve action selection, enabling the agent to converge toward optimal behavior. RL methods have been applied in different real-world applications. Zhou et al. [Zhou *et al.*, 2024] proposed a natural Actor-Critic framework for robust RL with function approximation. Zhang et al. [Zhang *et al.*, 2024] proposed a novel brain effective connectivity discovery method based on meta-RL.

2.3 Neural Architecture Search based Artificial Bee Colony

Despite the impressive progress in neural network architecture design, improving the performance of the existing state-of-the-art models has become increasingly challenging [Fu *et al.*, 2024; Benmeziane *et al.*, 2024]. The emergence and development of neural architecture search (NAS) technology can to some extent solve the problem of difficult manual design of network architecture. The NAS method can automatically search for the optimal architecture for the current task within a pre-defined search space and has been rapidly applied in many fields. Martin et al. [Martin *et al.*, 2024] proposed an ABC optimization of deep convolutional neural networks in the context of biomedical imaging. Asaad et al. [Asaad *et al.*, 2024] employed ABC algorithm to optimize the artificial neural network in heart disease prediction.

3 Preliminary

3.1 Notation and Problem Formulation

Causal protein signaling networks are complex network structures formed by intricate interactions between proteins involved in intracellular signaling. These networks are crucial in cell function, and studying them can enhance our understanding of the signaling mechanisms within cells [Barkhuizen *et al.*, 2022]. In this paper, the vectors (tensors of order one) are denoted by boldface lowercase letters. Matrices (tensors of order two) are denoted by boldface capital letters. a causal protein signaling network is denoted as $\mathbf{G} =< \mathbf{V}, \mathbf{E} >$, where $\mathbf{V}$ is a set of nodes with each node $X_i \in \mathbf{V}$ representing a a protein molecule; and $\mathbf{E}$ is a set of edges with each edge $X_i \to X_j \in \mathbf{E}$ describing an signaling pathway from protein molecule X_i to X_j. Single-cell data is in the form of a two-dimensional matrix $\mathbf{I}$ with m columns and g rows:

$$\mathbf{I} = \begin{bmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{g1} & \cdots & a_{gm} \end{bmatrix}, \tag{1}$$

where a_{ij} $(1 \leq i \leq g, 1 \leq j \leq m)$ is the expression level of the i-th protein biomolecule in the j-th cell. Our task is to reconstruct causal protein signaling network $\mathbf{G}$ based on protein biomolecule expression levels in single-cell data $\mathbf{X}$ and infer signaling pathways between proteins in order to reveal cellular signaling processes.

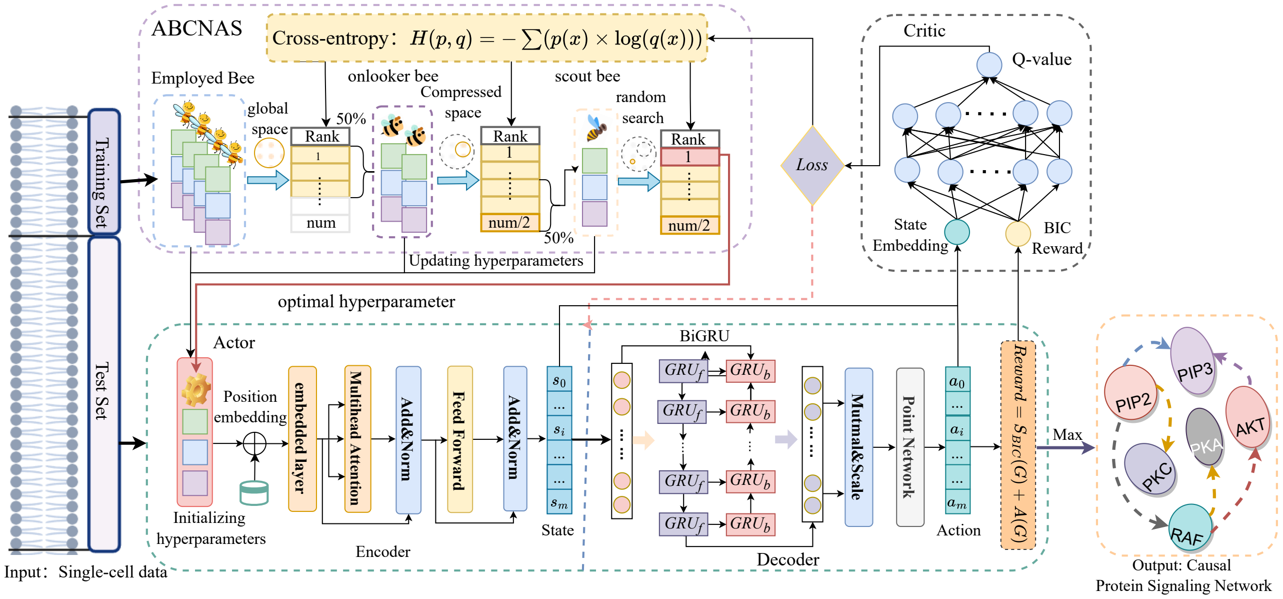

Figure 1: The flowchart of the ABCNAS-RL algorithm, which includes two phases: ABC neural architecture search and actor-critic RL.

3.2 Motivation

Traditional machine learning methods can rapidly infer causal protein signaling network due to their simple structure, but the difficulty in capturing potentially complex signaling pathways leads to poor performance. In recent years, RL methods have shown significant advantages due to the fact that they do not need to rely on various local heuristic information when inferring causal protein signaling network. However, RL models have many hyperparameters and rely on manual settings, leading to poor performance in causal protein signaling network recognition. Luckily, ABC has balanced exploration-exploitation dynamics. ABC employs global search (employed bees), local refinement (onlookers), and random exploration (scouts) to escape local optima efficiently. Therefore, it is a promising attempt to combine the NAS ideology with the tuning of hyperparameter optimization and microstructure of reinforcement learning model based on ABC for causal protein signaling networks.

4 Method

4.1 Main Idea

ABCNAS-RL consists of two phases. In phase one, we represent each bee as a unique set of hyperparameter combinations, utilizing the ABC algorithm and cross-entropy loss function to automatically search for optimal hyperparameters for the RL model on the training set. The optimization process consists of three phases of employed bees' search, onlooker bees' search, and scout bees' search. In phase two, we use the RL model with hyperparameters optimized by the ABC algorithm to infer the causal protein signaling network on the test set. The model follows an actor-critic architecture. The actor network consists of an encoder-decoder architecture, composed of a transformer and a BiGRU with an integrated attention mechanism. The critic network consists of a fully connected neural network that estimates the output state of the actor network. By maximizing cumulative rewards, we ultimately derive the causal protein signaling network. Figure 1 illustrates the flowchart of the ABCNAS-RL algorithm.

4.2 ABC Neural Architecture Search

We divide the input dataset into a training set and a test set, and first search for the optimal hyperparameters of the RL model on the training set using the ABC algorithm and the cross-entropy loss function. The hyperparameters of the RL model are chosen as $epoch$, lr and n_{layer}, where $epoch$ is the number of times the entire training dataset is completely traversed by the neural network, lr denotes the learning rate, and n_{layer} is the number of layers of the fully-connected network in the Critic network. We define the search space and step size of these three hyperparameters respectively. Each bee corresponds to a set of hyperparameter combinations, and the cross-entropy loss function is defined as follows:

$$H(p,q) = -\sum(p(x) \times \log(q(x))), \tag{2}$$

where $p(x)$ represents the causal distribution of the true causal protein signaling network and $q(x)$ represents the causal distribution predicted by the model. This formula measures the difference between the true distribution and the model-predicted distribution; the smaller the difference $H(p,q)$, the closer the two distributions are, i.e., the better the model's prediction matches the true situation, and thus the better the model performs. The optimization search process consists of three phases: employed bee search, onlooker bee search and scout bee search. This method combines global search, local search, and random search to effectively avoid local optima. The specific steps are as follows:

Employed Bee Phase

First, initialize N employed bees in the training dataset, each corresponding to a hyperparameter combination, with the search space being the global space $\mathbf{R}$. Assuming the search space is d-dimensional, the hyperparameter combination $\mathbf{P}_i$ can be expressed as:

$$\mathbf{P}_i = [p_{i1}, p_{i2}, \ldots, p_{id}], \tag{3}$$

where p_{ij} is the j-th hyperparameter of the i-th employed bee and d is 3.

Onlooker Bee Phase

The top 50% of the employed bees proceed to the onlooker bee phase. In the onlooker bee phase, the search space is compressed based on information from the previous phase, resulting in the local search space $\mathbf{L}$. The local search space is defined as:

$$\mathbf{L} = [l_1, l_2, \ldots, l_d], \tag{4}$$

where l_i is the local search range of the i-th hyperparameter, and d is the number of hyperparameters.

Each onlooker bee performs local optimization within the neighborhood of the hyperparameters. The new hyperparameter combination $\mathbf{P}_{new}$ can be expressed as:

$$\mathbf{P}_{new} = \mathbf{P}_{best} + \phi \cdot (\mathbf{P}_{best} - \mathbf{P}_i), \tag{5}$$

where ϕ is a random number, $-1 \leq \phi \leq 1$, $\mathbf{P}_i$ is the hyperparameter combination of the current onlooker bee, and $\mathbf{P}_{best}$ is the best solution in the current iteration.

After the onlooker bee search phase, the new hyperparameter combinations are sorted based on the cross-entropy loss function, and the bottom 50% are converted to scout bees.

Scout Bee Phase

In the scout bee phase, scout bees perform random searches outside the current search space to reduce the probability of falling into local optima. The global search space is defined as $\mathbf{R}$, and scout bees randomly generate new hyperparameter combinations $\mathbf{P}_{random}$ outside the current search space $\mathbf{L}$. The formula is:

$$\mathbf{P}_{random} = \mathbf{P}_{min} + rand \cdot (\mathbf{P}_{max} - \mathbf{P}_{min}), \tag{6}$$

where $\mathbf{P}_{min}$ and $\mathbf{P}_{max}$ are the minimum and maximum values of the global search space, respectively, and $rand$ is a random number, $0 \leq rand \leq 1$.

After multiple iterations and updating the rank table, the optimal hyperparameter combination is selected as the neural network structure and hyperparameters for the actor-critic RL model.

4.3 Actor-Critic Reinforcement Learning

After searching for the optimal actor-critic RL model hyperparameters on the training set, we use the actor-critic RL model to perform causal protein signaling network inference on the test set. The modules of Actor-Critic Reinforcement Learning are described in detail below.

Actor

Transformer-based Encoder: a common way is to use $\mathbf{I}$ as input to the network directly. However, the high noise characteristic of single-cell data presents a great challenge for general feed-forward neural networks to capture the underlying causal relationships directly using $\mathbf{I}$ as states. Consequently, incorporating an encoder module to preprocess the single-cell data proves beneficial in extracting useful information and finding better causal protein signaling network.

For the model design of the encoder, we utilize the Transformer, which involves first embedding the inputs via a linear layer, followed by processing them through multiple identical encoder blocks comprising a multi-head self-attention layer and a feed-forward layer; we posit that multi-head self-attention is well-suited for extracting temporal features from single-cell data, as it reduces reliance on external information and better captures internal correlations within the single-cell data. the encoder block operations are as follows:

$$\begin{aligned} X' &= Linear(X^m) \in \mathbb{R}^{m \times n \times h}, \\ Q_l &= W^Q X'^l + \epsilon^Q, \\ K_l &= W^K X'^l + \epsilon^K, \\ V_l &= W^V X'l^l + \epsilon^V, \end{aligned} \tag{7}$$

where X^m denotes the embedded input, $Linear$ is a fully connected linear layer that provides a linear transformation of the input X', h denotes the number of hidden layer nodes of the fully connected linear layer, X'^l expresses the l-th input after dividing the embedding X' into L head self-attention layer. W^Q, W^K and W^V denote the network parameters for the self-attention layer respectively, ϵ^Q, ϵ^K and ϵ^V are the bias vector. Then, we can get Q_l, K_l and V_l which denote the query, key, and value vector of the self-attention layer respectively, So the self-attention can be calculated as follows:

$$\begin{aligned} \text{Attn}_l &= \text{softmax}\left(\frac{Q_l K_l^T}{\sqrt{d_{K_l}}}\right) V_l, \\ S_{enc} &= \text{Concat}\left(Attn_1, Attn_2, ..., Attn_L\right), \end{aligned} \tag{8}$$

where d denotes the number of elements in the last dimension of the query, key, and value vector Q_l, K_l, and V_l, Attn_l describes the l-th head attention vector. Then, we can collect all L heads of the self-attention vectors. and get the embedded state S_{enc}.

BiGRU-based Decoder: the decoder is responsible for generating the output sequence from the encoded state S_{enc}. In our model, we use a bidirectional GRU to enhance the decoding process by capturing dependencies in both forward and backward directions, which provides a more comprehensive context for generating accurate outputs.

A GRU cell is defined by the following equations:

$$\mathbf{z}_t = \sigma(\mathbf{W}_z \mathbf{x}_t + \mathbf{U}_z \mathbf{h}_{t-1} + \mathbf{b}_z), \tag{9}$$

$$\mathbf{r}_t = \sigma(\mathbf{W}_r \mathbf{x}_t + \mathbf{U}_r \mathbf{h}_{t-1} + \mathbf{b}_r), \tag{10}$$

$$\tilde{\mathbf{h}}_t = \tanh(\mathbf{W}_h \mathbf{x}_t + \mathbf{U}_h (\mathbf{r}_t \odot \mathbf{h}_{t-1}) + \mathbf{b}_h), \tag{11}$$

$$\mathbf{h}_t = (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \tilde{\mathbf{h}}_t, \tag{12}$$

where $\mathbf{z}_t$ is the update gate vector, $\mathbf{r}_t$ is the reset gate vector, $\tilde{\mathbf{h}}_t$ is the candidate hidden state vector, $\mathbf{h}_t$ is the final hidden state vector, σ is the sigmoid activation function, $\tanh$ is the hyperbolic tangent activation function, $\mathbf{W}_z$, $\mathbf{W}_r$, and $\mathbf{W}_h$ are the weight matrices for the input $\mathbf{x}_t$, $\mathbf{U}_z$, $\mathbf{U}_r$, and $\mathbf{U}_h$ are the weight matrices for the hidden state $\mathbf{h}_{t-1}$, and $\mathbf{b}_z$, $\mathbf{b}_r$, and $\mathbf{b}_h$ are the bias vectors.

In the BiGRU, we have two GRUs: one processes the input sequence in the forward direction and the other in the backward direction. The hidden states from both directions are concatenated to form the final hidden state:

$$\mathbf{h}_t = [\overrightarrow{\mathbf{h}}_t; \overleftarrow{\mathbf{h}}_t], \tag{13}$$

where $\overrightarrow{\mathbf{h}}_t$ is the hidden state of the forward GRU and $\overleftarrow{\mathbf{h}}_t$ is the hidden state of the backward GRU.

To further enhance the decoder, we incorporate an attention mechanism that allows the model to focus on different parts of the input sequence when generating each output token. The attention score for each encoder hidden state is computed as:

$$e_{ij} = \mathbf{v}^\top \tanh(\mathbf{W}_1\mathbf{h}_i + \mathbf{W}_2\mathbf{s}_{j-1}), \tag{14}$$

where e_{ij} is the attention score, $\mathbf{h}_i$ is the encoder hidden state at time step i, $\mathbf{s}_{j-1}$ is the decoder hidden state from the previous time step $j-1$, $\mathbf{v}$ is the attention vector, $\mathbf{W}_1$ and $\mathbf{W}_2$ are the weight matrices for the encoder hidden state and decoder hidden state respectively.

The attention weights are obtained by normalizing the scores using a softmax function:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_k \exp(e_{ik})}, \tag{15}$$

where α_{ij} is the attention weight for the i-th encoder hidden state at time step j.

The context vector $\mathbf{c}_j$ is computed as a weighted sum of the encoder hidden states:

$$\mathbf{c}_j = \sum_i \alpha_{ij}\mathbf{h}_i, \tag{16}$$

where $\mathbf{c}_j$ is the context vector at time step j.

The decoder hidden state is then updated using the context vector and the previous hidden state:

$$\mathbf{s}_j = \text{GRU}(\mathbf{y}_{j-1}, [\mathbf{c}_j; \mathbf{s}_{j-1}]), \tag{17}$$

where $\mathbf{s}_j$ is the decoder hidden state at time step j, $\mathbf{y}_{j-1}$ is the previous output, and GRU is the gated recurrent unit function that updates the hidden state based on the concatenated context vector and previous hidden state.

The updated decoder hidden state $\mathbf{s}_j$ is then used to generate the final output of the sequence. By incorporating the bidirectional GRU and attention mechanism, the decoder can effectively leverage the encoded information and generate more accurate and context-aware outputs.

Critic

For critic we use a n_{layer} feed-forward neural network with a ReLU activation function. The input of the critic network is the decoder output (actions) and the rewards. To better assess the value of the learning causal protein signaling network, we use the Bayesian information criterion score (BIC) as the reward function as follows:

$$\mathrm{S_{BIC}}(\mathbf{G}) = \sum_{j=1}^{n}\left[\sum_{i=1}^{t}\log p\left(x_{ij} \mid \mathrm{Pa}\left(x_{ij}\right);\theta_j\right) - \frac{|\theta_j|}{2}\log t\right], \tag{18}$$

Where θ_j is the parameter associated with each likelihood, and $|\theta_j|$ denotes the dimension of the parameter. BIC score enables us to identify the optimal causal relationships that best aligns with the single-cell data, which leads to improved performance of the causal protein signaling network. So the reward can be described as:

$$\text{reward}(\mathbf{G}) = -[\mathrm{S_{BIC}}(\mathbf{G}) + \lambda A(\mathbf{G})], \tag{19}$$

where $\lambda \geq 0$ is a parameter that controls the sparsity of causal protein signaling network and $A(\mathbf{G})$ is the sparse penalty function as $A(\mathbf{G}) = \|\mathbf{G}\|_1$. $\mathrm{S_{BIC}}$ denotes the score for the action (causal protein signaling network). By utilizing fully connected layers, the critic network can effectively capture the intricate relationship between actions and rewards. At the same time, the output of the critic network provides a loss L^{critic} for the actor that trains actor network to produce more highly rewarded actions (causal protein signaling networks).

The ABC phase has a complexity of $O(N)$(bees) and $O(T)$ (iterations). The RL training phase requires $O(R)$ (epochs). Complexity stems from ABC search and RL training. Total complexity is $O(N \times T \times R)$.

5 Experimental Setting

In this section, we introduce the environment configuration, datasets, evaluation metrics, baseline methods, and parameter settings. The configuration comprises a powerful NVIDIA GeForce RTX 3090, coupled with a high-performance NVIDIA GeForce RTX 3080Ti GPU, alongside the computational prowess of an AMD Ryzen 9 5950X 16-Core Processor CPU. The code is available at https://github.com/ZJH66/ABCNAS-RL.

5.1 Data Description

In this paper, we used simulated dataset and the real dataset. We generated 4 simulated datasets Sim1 to Sim4 with reference to the [Zhu *et al.*, 2019], and each dataset contains a different number of nodes (v = 5, 10, 20, 50). The real multi-parameter fluorescence-activated cell sortera data set [Sachs *et al.*, 2005b] to learn causal protein signaling network based on expression levels of proteins and phospholipids, and the data sets are available at https://www.science.org/doi/10.1126/science.1105809#supplementary.

5.2 Evaluation Metrics

We compared the result learned to ground-truth network on common graph metrics [Xiong *et al.*, 2025; Liu *et al.*, 2024a]: (1) Precision; (2) Recall; (3) F1; (4) Accuracy; (5) Structural Hamming Distance (SHD). A high-performing algorithm is characterized by higher values of Precision, Recall, F1 and Accuracy, as well as lower values of SHD.

5.3 Baseline Methods and Parameter Setting

The parameters of the comparison algorithms are chosen based on the corresponding literature. For a more fair comparison, we conduct experiments on generated simulation datasets with 5 to 50 node numbers and make appropriate fine-tuning to the original parameters so that the comparison algorithms can show the best performance.

Data (Nodes)	Metrics	Methods								
		NoTears	RL	GES	FedCSL	RCIT	DiffAN	ACD	PDABC	ABCNAS-RL
Sim1 (5)	Precision↑	0.58 ± 0.00	0.56 ± 0.00	0.54 ± 0.00	0.55 ± 0.10	0.35 ± 0.13	0.57 ± 0.06	0.51 ± 0.05	0.64 ± 0.03	0.68 ± 0.03
	Recall↑	0.64 ± 0.00	0.52 ± 0.00	0.54 ± 0.00	0.35 ± 0.07	0.35 ± 0.10	0.62 ± 0.07	0.35 ± 0.08	0.64 ± 0.02	0.64 ± 0.03
	F1↑	0.60 ± 0.00	0.54 ± 0.00	0.54 ± 0.00	0.45 ± 0.12	0.35 ± 0.08	0.59 ± 0.04	0.43 ± 0.07	0.64 ± 0.03	0.66 ± 0.02
	Accuracy↑	0.83 ± 0.00	0.82 ± 0.00	0.81 ± 0.00	0.81 ± 0.07	0.78 ± 0.08	0.82 ± 0.04	0.79 ± 0.07	0.84 ± 0.03	0.85 ± 0.03
	SHD↓	3.00 ± 0.00	4.00 ± 0.00	4.00 ± 0.00	5.15 ± 1.10	7.12 ± 0.95	3.14 ± 1.22	5.46 ± 0.32	3.22 ± 0.74	2.13 ± 0.62
Sim2 (10)	Precision↑	0.43 ± 0.00	0.45 ± 0.00	0.24 ± 0.00	0.46 ± 0.01	0.58 ± 0.04	0.54 ± 0.04	0.50 ± 0.02	0.76 ± 0.01	0.79 ± 0.01
	Recall↑	0.54 ± 0.00	0.66 ± 0.00	0.38 ± 0.00	0.46 ± 0.01	0.65 ± 0.05	0.29 ± 0.02	0.59 ± 0.01	0.72 ± 0.01	0.70 ± 0.01
	F1↑	0.47 ± 0.00	0.54 ± 0.00	0.27 ± 0.00	0.46 ± 0.01	0.62 ± 0.04	0.37 ± 0.03	0.54 ± 0.02	0.74 ± 0.01	0.75 ± 0.01
	Accuracy↑	0.78 ± 0.00	0.81 ± 0.00	0.76 ± 0.00	0.63 ± 0.08	0.81 ± 0.04	0.74 ± 0.05	0.81 ± 0.02	0.86 ± 0.01	0.87 ± 0.01
	SHD↓	9.00 ± 0.00	8.00 ± 0.00	13.00 ± 0.00	20.50 ± 4.20	9.15 ± 0.91	11.65 ± 0.85	9.21 ± 2.45	6.15 ± 1.85	5.05 ± 1.13
Sim3 (20)	Precision↑	0.53 ± 0.00	0.57 ± 0.00	0.62 ± 0.00	0.60 ± 0.04	0.44 ± 0.03	0.41 ± 0.03	0.62 ± 0.06	0.65 ± 0.06	0.71 ± 0.09
	Recall↑	0.53 ± 0.00	0.57 ± 0.00	0.53 ± 0.00	0.14 ± 0.03	0.41 ± 0.02	0.81 ± 0.06	0.61 ± 0.06	0.62 ± 0.06	0.68 ± 0.08
	F1↑	0.53 ± 0.00	0.57 ± 0.00	0.57 ± 0.00	0.23 ± 0.03	0.42 ± 0.03	0.54 ± 0.04	0.62 ± 0.06	0.63 ± 0.05	0.69 ± 0.08
	Accuracy↑	0.82 ± 0.00	0.85 ± 0.00	0.83 ± 0.00	0.78 ± 0.04	0.71 ± 0.09	0.82 ± 0.01	0.84 ± 0.01	0.85 ± 0.02	0.87 ± 0.01
	SHD ↓	12.00 ± 0.00	10.00 ± 0.00	11.00 ± 0.00	19.15 ± 1.12	15.12 ± 2.15	12.40 ± 1.07	13.50 ± 3.21	13.51 ± 2.58	11.40 ± 3.06
Sim4 (50)	Precision↑	0.46 ± 0.00	0.55 ± 0.00	0.59 ± 0.00	0.48 ± 0.02	0.59 ± 0.03	0.49 ± 0.02	0.57 ± 0.02	0.63 ± 0.01	0.73 ± 0.01
	Recall↑	0.48 ± 0.00	0.55 ± 0.00	0.54 ± 0.00	0.67 ± 0.01	0.57 ± 0.03	0.51 ± 0.02	0.59 ± 0.01	0.63 ± 0.01	0.71 ± 0.01
	F1↑	0.47 ± 0.00	0.55 ± 0.00	0.56 ± 0.00	0.56 ± 0.01	0.58 ± 0.02	0.50 ± 0.02	0.58 ± 0.02	0.63 ± 0.01	0.72 ± 0.01
	Accuracy↑	0.78 ± 0.00	0.81 ± 0.00	0.83 ± 0.00	0.80 ± 0.02	0.75 ± 0.02	0.74 ± 0.03	0.83 ± 0.02	0.87 ± 0.01	0.89 ± 0.01
	SHD↓	49.00 ± 0.00	45.00 ± 0.00	45.00 ± 0.00	42.11 ± 2.14	42.62 ± 2.11	47.13 ± 1.34	37.15 ± 1.85	33.25 ± 1.56	25.37 ± 1.26

Table 1: The results of 9 methods on simulated dataset. The gray values indicate that the method achieved the best results.

To demonstrate the competitiveness of our ABCNAS-RL in an intuitive way, we compare with eight other state-of-the-art or classic algorithms. These algorithms include: continuous optimization for structure learning (NoTears) [Zheng *et al.*, 2018], reinforcement learning (RL) [Zhu *et al.*, 2019] ,greedy equivalence search (GES) [Chickering, 2002], federal causal structure learning (FedCSL) [Guo *et al.*, 2024], regression-based conditional independence test (RCIT) [Zhang *et al.*, 2023], diffusion models for causal discovery via topological ordering (DiffAN) [Sanchez *et al.*, 2023], amortized causal discovery (ACD)[Löwe *et al.*, 2022] and parallel discrete ABC algorithm (PDABC) [Liu *et al.*, 2024b]. We refer to the parameters of all comparative algorithm source literature and optimize on the experimental dataset to ensure fairness.

For ABCNAS-RL algorithm, We conducted a parameter sensitivity analysis experiment on the first simulated dataset. The ABC algorithm reached optimal performance when the number of bees N and the number of iterations T were set to 20 and 10, respectively, resulting in the minimum value of the cross-entropy loss, as shown in Figure 2. Increasing these values further led to higher computational time costs, but the performance remained stable. Subsequently, the parameters for the reinforcement learning model were automatically determined using the ABC algorithm.

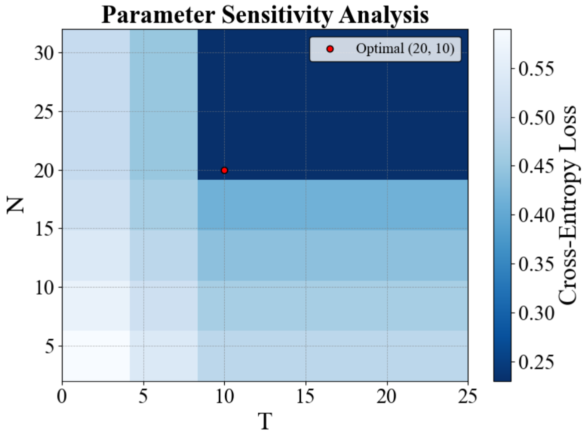

Figure 2: Parameter Sensitivity Analysis.

6 Experimental Results and Discussions

6.1 Results on Simulated Dataset

We comprehensively test and compare the above 9 algorithms on 4 simulated datasets. The detailed results on 4 simulated datasets are shown in Table 1. For ABCNAS-RL, we take the first 30% of all data sets as the training set, and the last 70% as the test set. From Table 1, we can find that ABCNAS-RL outperforms the other 8 algorithms in all metrics on the 4 simulated datasets.

ABCNAS-RL consistently outperforms the other algorithms in multiple metrics, particularly in Precision and F1 Score across all datasets. For instance, in Sim1, ABCNAS-RL achieved a Precision of 0.68 and an F1 Score of 0.66, indicating its effectiveness in correctly identifying relevant connections while minimizing false positives. Similarly, in Sim4, it maintained superior performance with a Precision of 0.73, showcasing its robustness as the complexity of the dataset increases. Moreover, ABCNAS-RL demonstrates a notable advantage in SHD, achieving the lowest error rates compared to other methods. This signifies that ABCNAS-RL not only excels in precision and recall but also minimizes the discrepancies in the learned structure, making it a reliable choice for tasks requiring accurate causal inference.

6.2 Results on Real-world Dataset

We merged the 14 sub-datasets into a total dataset with a sample size of 11672 and treat the first 30% as the training set and the last 70% as the test set. The results are shown in Table 2.

By Recall we can easily find that highlight the strengths and weaknesses of each approach from Table 2. FedCSL and RCIT, both of which identified 6 edges, displayed notably lower performance metrics. This shortfall is likely due to their assumption of acyclic graphs, which do not align well with the inherently cyclic nature of biological protein signaling networks, leading to reduced accuracy and effectiveness. RL, DiffAN, and ACD each identified 7 edges cor-

Methods	Precision ↑	Recall↑	F1↑	Accuracy↑	SHD↓
NoTears	0.47 ± 0.00	0.47 ± 0.00	0.47 ± 0.00	0.85 ± 0.00	14.00 ± 0.00
RL	0.50 ± 0.00	0.41 ± 0.00	0.45 ± 0.00	0.86 ± 0.00	15.00 ± 0.00
GES	0.23 ± 0.00	0.47 ± 0.00	0.31 ± 0.00	0.70 ± 0.00	36.00 ± 0.00
FedCSL	0.50 ± 0.12	0.35 ± 0.15	0.41 ± 0.13	0.86 ± 0.12	15.15 ± 1.25
RCIT	0.46 ± 0.07	0.35 ± 0.12	0.39 ± 0.03	0.82 ± 0.04	16.13 ± 1.36
DiffAN	0.44 ± 0.13	0.41 ± 0.15	0.42 ± 0.14	0.84 ± 0.10	16.72 ± 0.79
ACD	0.41 ± 0.04	0.41 ± 0.05	0.41 ± 0.04	0.83 ± 0.03	17.30 ± 0.72
PDABC	0.53 ± 0.02	0.47 ± 0.02	0.50 ± 0.02	0.87 ± 0.03	14.35 ± 0.62
ABCNAS-RL	0.62 ± 0.03	0.47 ± 0.02	0.53 ± 0.02	0.88 ± 0.02	12.75 ± 0.55

Table 2: The result of 9 methods on Precision, Recall, F1, Accuracy and SHD. The gray values indicate the best results.

rectly. While these algorithms performed better than FedCSL and RCIT, their overall metrics were only moderate, indicating that while they provide a reasonable number of correct edges, they lack exceptional performance across the board. NoTears, GES, PDABC, and ABCNAS-NAS each successfully identified 8 correct edges. However, GES faced significant issues with low Precision and Accuracy, primarily due to generating a high number of redundant edges, which detracted from its overall utility. NoTears, PDABC, and ABCNAS-NAS demonstrated a more balanced performance, though they did not surpass ABCNAS-RL in terms of metric excellence. ABCNAS-RL emerged as the most proficient algorithm, correctly identifying 8 edges while also achieving a high Precision of 0.62. In summary, ABCNAS-RL can infer causal protein signaling network stably and accurately.

The causal protein signaling network inferred by ABCNAS-RL reveals a complex network of interconnected signaling pathways, including key pathways such as (Pkc, Jnk, P38, Pka), (PKA,P38, Akt) and calcium signaling through Plc. Pkc, Pka, and Raf act as central regulators, activating downstream effectors like Jnk, P38, and Akt through both direct and indirect mechanisms. These signaling networks mediate crucial cellular processes such as proliferation, survival, differentiation, and stress responses, highlighting the intricate crosstalk between pathways. This causal network provides valuable insights into how cells integrate and respond to various signals, offering potential implications for understanding disease mechanisms and developing targeted therapeutic strategies.

To clearly show the significant differences between these algorithms, we use the Friedman test and T test to attest to the significant difference between these algorithms. If the p-value obtained from the test is less than 0.05, we consider that a significant difference exists in the corresponding experimental results.The Friedman test indicates a significant difference between the nine algorithms (p-value <0.05). Furthermore, we perform the T test on the results, which reinforce the conclusion that ABCNAS-RL provides a robust advantage in inferring the causal protein signaling network.

6.3 Ablation Analysis

In this ablation study of encoder and decoder configurations, we compare the performance of several encoder and decoder types on Sim1 data, including Transformer, Graph Attention Network (GAT), BiLSTM and BiGRU, across multiple evaluation metrics. Figure 3 visualizes the results of the ablation experiments. It is evident that the Transformer + BiGRU configuration achieves the highest values in Precision (0.92), Recall (0.89), F1 (0.90), Accuracy (0.90) and SHD (2). In contrast, other encoder and decoder configurations, such as Transformer + BiLSTM and GAT + BiGRU, show noticeably lower performance on these metrics, highlighting the Transformer + BiGRU configuration's strong ability to model sequential dependencies and capture important features from the data. Moreover, Transformer + BiGRU achieves a SHD of 2, significantly lower than other models like GAT + BiLSTM and Transformer + BiLSTM (both having SHD = 5). This result indicates that Transformer + BiGRU provides more accurate structural predictions in the context of inferring causal protein signaling networks.

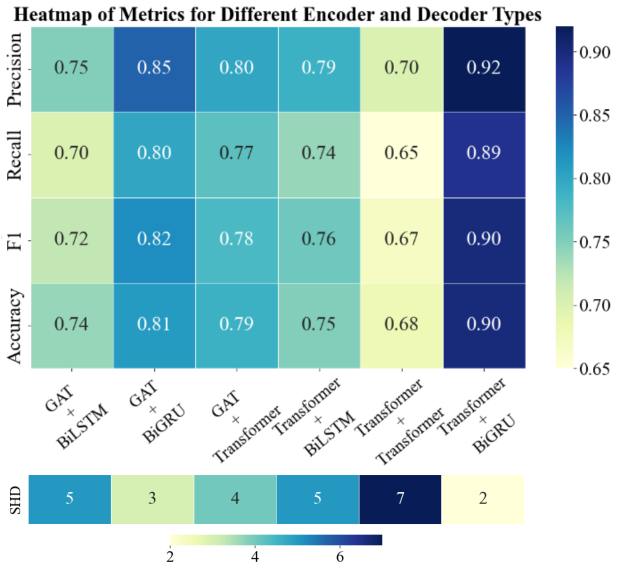

Figure 3: Performance heatmap of different encoder-decoders.

In summary, Transformer + BiGRU excels in both classification accuracy and structural prediction tasks. Its combination of Transformer's powerful feature extraction capabilities and BiGRU's bidirectional sequence modeling ability enables it to capture both forward and backward dependencies, making it the most effective configuration in this experiment.

7 Conclusion and Limitation

In this paper, we propose a novel method combining ABC neural architecture search and reinforcement learning (ABCNAS-RL) to infer causal protein signaling networks from single-cell data. The algorithm optimizes hyperparameters using ABC and applies the model to learn causal network structures. Experiments show the method effectively identifies causal protein relationships, which has significant implications for understanding the biological systems.

The main limitation of the current work is efficiency, with relatively long search times. In future work, we plan to further optimize the computational efficiency of the algorithm to reduce training and inference time, especially when handling large-scale datasets. To enhance ABCNAS-RL's performance, we will develop efficient parallel computing methods for faster training and inference, enabling better tools in bioinformatics and systems biology.

Acknowledgements

Thanks for the professional and constructive suggestions from the 5 reviewers. This work was Sponsored by Beijing Nova Program (20240484635), partly supported by National Natural Science Foundation of China Research Program (62106009, 62276010, 62171297), in part by R&D Program of Beijing Municipal Education Commission (KM202210005030, KZ202210005009).

References

[Asaad *et al.*, 2024] Manal Mohammed Othman Farea Asaad, Juliana Wahid, and Abdul Razak Rahmat. Employing artificial bee colony algorithm to optimize the artificial neural network in heart disease prediction. In *AIP Conference Proceedings*, volume 2895. AIP Publishing, 2024.

[Barkhuizen *et al.*, 2022] Melinda Barkhuizen, Kasper Renggli, Sylvain Gubian, Manuel C Peitsch, Carole Mathis, and Marja Talikka. Causal biological network models for reactive astrogliosis: a systems approach to neuroinflammation. *Scientific Reports*, 12(1):4205, 2022.

[Benmeziane *et al.*, 2024] Hadjer Benmeziane, Imane Hamzaoui, Zayneb Cherif, and Kaoutar El Maghraoui. Medical neural architecture search: Survey and taxonomy. *International Joint Conference on Artificial Intelligence*, pages 7932–7940, 2024.

[Cai *et al.*, 2024] Hengrui Cai, Yixin Wang, Michael Jordan, and Rui Song. On learning necessary and sufficient causal graphs. *Advances in Neural Information Processing Systems*, 36, 2024.

[Chickering, 2002] David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3(Nov):507–554, 2002.

[Deleu *et al.*, 2022] Tristan Deleu, António Góis, Chris Emezue, Mansi Rankawat, Simon Lacoste-Julien, Stefan Bauer, and Yoshua Bengio. Bayesian structure learning with generative flow networks. *Uncertainty in Artificial Intelligence*, pages 518–528, 2022.

[Fu *et al.*, 2024] Pinhan Fu, Xinyan Liang, Tingjin Luo, Qian Guo, Yayu Zhang, and Yuhua Qian. Core-structures-guided multi-modal classification neural architecture search. *International Joint Conference on Artificial Intelligence*, pages 3980–3988, 2024.

[Guo *et al.*, 2024] Xianjie Guo, Kui Yu, Lin Liu, and Jiuyong Li. Fedcsl: A scalable and accurate approach to federated causal structure learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 12235–12243, 2024.

[Konda and Tsitsiklis, 1999] Vijay Konda and John Tsitsiklis. Actor-critic algorithms. *Advances in neural information processing systems*, 12, 1999.

[Lee *et al.*, 2024] Jonathan Lee, Annie Xie, Aldo Pacchiano, Yash Chandak, Chelsea Finn, Ofir Nachum, and Emma Brunskill. Supervised pretraining can learn in-context reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024.

[Liu *et al.*, 2022] Jinduo Liu, Junzhong Ji, Guangxu Xun, and Aidong Zhang. Inferring effective connectivity networks from fmri time series with a temporal entropy-score. *IEEE Transactions on Neural Networks and Learning Systems*, 33(10):5993–6006, 2022.

[Liu *et al.*, 2024a] Jinduo Liu, Lu Han, and Junzhong Ji. Mcan: Multimodal causal adversarial networks for dynamic effective connectivity learning from fmri and eeg data. *IEEE Transactions on Medical Imaging*, 43(8):2913–2923, 2024.

[Liu *et al.*, 2024b] Jinduo Liu, Jihao Zhai, and Junzhong Ji. Inferring causal protein signalling networks from single-cell data based on parallel discrete artificial bee colony algorithm. *CAAI Transactions on Intelligence Technology*, 9(6):1587–1604, 2024.

[Löwe *et al.*, 2022] Sindy Löwe, David Madras, Richard Zemel, and Max Welling. Amortized causal discovery: Learning to infer causal graphs from time-series data. *Conference on Causal Learning and Reasoning*, pages 509–525, 2022.

[Martin *et al.*, 2024] Adri Gomez Martin, Carlos Fernandez del Cerro, Monica Abella Garcia, and Manuel Desco Menendez. Artificial bee colony optimization of deep convolutional neural networks in the context of biomedical imaging. *arXiv preprint arXiv:2402.15246*, 2024.

[Sachs *et al.*, 2005a] Karen Sachs, Omar Perez, Dana Pe'er, Douglas A Lauffenburger, and Garry P Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529, 2005.

[Sachs *et al.*, 2005b] Karen Sachs, Omar Perez, Dana Pe'er, Douglas A Lauffenburger, and Garry P Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529, 2005.

[Sanchez *et al.*, 2023] Pedro Sanchez, Xiao Liu, Alison Q O'Neil, and Sotirios A Tsaftaris. Diffusion models for causal discovery via topological ordering. *The Eleventh International Conference on Learning Representations*, 2023.

[Shah *et al.*, 2024] Amil M Shah, Peder L Myhre, Victoria Arthur, Pranav Dorbala, Humaira Rasheed, Leo F Buckley, Brian Claggett, Guning Liu, Jianzhong Ma, Ngoc Quynh Nguyen, et al. Large scale plasma proteomics identifies novel proteins and protein networks associated with heart failure development. *Nature communications*, 15(1):528, 2024.

[Wang *et al.*, 2024] Yifei Wang, Pengju Ding, Congjing Wang, Shiyue He, Xin Gao, and Bin Yu. Rpi-ggcn: Prediction of rna–protein interaction based on interpretability gated graph convolution neural network and co-regularized variational autoencoders. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[Western *et al.*, 2024] Daniel Western, Jigyasha Timsina, Lihua Wang, Ciyang Wang, Chengran Yang, Bridget Phillips, Yueyao Wang, Menghan Liu, Muhammad Ali, Aleksandra Beric, et al. Proteogenomic analysis of human cerebrospinal fluid identifies neurologically relevant regulation and implicates causal proteins for alzheimer's disease. *Nature Genetics*, pages 1–13, 2024.

[Xiong *et al.*, 2025] Wen Xiong, Jinduo Liu, Junzhong Ji, and Fenglong Ma. Brain effective connectivity estimation via fourier spatiotemporal attention. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pages 1657–1668, 2025.

[Zhang *et al.*, 2023] Hao Zhang, Yewei Xia, Yixin Ren, Jihong Guan, and Shuigeng Zhou. Differentially private nonlinear causal discovery from numerical data. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(10):12321–12328, 2023.

[Zhang *et al.*, 2024] Zuozhen Zhang, Junzhong Ji, and Jinduo Liu. Metarlec: Meta-reinforcement learning for discovery of brain effective connectivity. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(9):10261–10269, 2024.

[Zheng *et al.*, 2018] Xun Zheng, Bryon Aragam, Pradeep K Ravikumar, and Eric P Xing. Dags with no tears: Continuous optimization for structure learning. *Advances in Neural Information Processing Systems*, 31:518–528, 2018.

[Zhou *et al.*, 2024] Ruida Zhou, Tao Liu, Min Cheng, Dileep Kalathil, PR Kumar, and Chao Tian. Natural actor-critic for robust reinforcement learning with function approximation. *Advances in neural information processing systems*, 36, 2024.

[Zhu *et al.*, 2019] Shengyu Zhu, Ignavier Ng, and Zhitang Chen. Causal discovery with reinforcement learning. *arXiv preprint arXiv:1906.04477*, 2019.