Reinforced DAG Learning without Acyclicity Constraints Journal Submissions

Anonymous authors
Paper under double-blind review

Abstract

Recently, reinforcement learning (RL) has proved a promising alternative for conventional local heuristics in score-based approaches to learning directed acyclic causal graphs (DAGs) from observational data. However, the intricate acyclicity constraint still challenges the efficient exploration of the vast space of DAGs in existing methods. In this study, we introduce **ALIAS** (reinforced dAg Learning wIthout Acyclicity conStraints), a novel approach to causal discovery powered by the RL machinery. Our method features an efficient policy for generating DAGs in just a single step with an optimal quadratic complexity, fueled by a novel parametrization of DAGs that directly translates a continuous space to the space of all DAGs, bypassing the need for explicitly enforcing acyclicity constraints. This approach enables us to navigate the search space more effectively by utilizing policy gradient methods and established scoring functions. In addition, we provide compelling empirical evidence for the strong performance of **ALIAS** in comparison with state-of-the-arts in causal discovery over increasingly difficult experiment conditions on both synthetic and real datasets.

1 Introduction

The knowledge of causal relationships required to understand the nature in many scientific sectors (Sachs et al., 2005; Hünermund & Bareinboim, 2023; Cao et al., 2019), especially in intricate situations where randomized experiments are impractical, has motivated the development of causal discovery methods that aim to infer cause-effect relationships from purely passive data over the last decades. Causal discovery is typically formulated as finding the directed acyclic graph (DAG) representing the causal model that most likely generated the observed data. Among the broad literature, score-based methods are one of the most well-recognized approaches, which assigns each possible DAG $\mathcal{G}$ a "score" $\mathcal{S}(\mathcal{D}, \mathcal{G})$ quantifying how much it can explain the observed data $\mathcal{D}$, and then optimize the score over the space of DAGs:

$$\mathcal{G}^\star = \underset{\mathcal{G} \in \text{DAGs}}{\arg\max}\ \mathcal{S}(\mathcal{D}, \mathcal{G}). \tag{1}$$

Solving this optimization problem is generally NP-hard (Chickering, 1996), due to the *huge combinatorial search space* that grows super-exponentially with the number of variables (Robinson, 1977) and the *intricate acyclicity constraint*. Most methods therefore resort to local heuristics, such as GES (Chickering, 2002) which gradually adds edges into a graph one-by-one while laboriously maintaining acyclicity. With the introduction of soft DAG characterizations (Zheng et al., 2018; Yu et al., 2019; Zhang et al., 2022; Bello et al., 2022), the combinatorial optimization problem above is relaxed to a continuous optimization problem, allowing for exploring graphs more effectively, as multiple edges can be added or removed simultaneously in an update.

Recently, reinforcement learning (RL) has emerged into score-based causal discovery (Zhu et al., 2020; Wang et al., 2021; Yang et al., 2023a;b) as the improved search strategy, thanks to its exploration and exploitation abilities. However, existing RL-based methods handle acyclicity either by fusing the soft DAG regularization from Zheng et al. (2018) into the reward Zhu et al. (2020), which wastes time for exploring non-DAGs but still does not prohibit all cycles Wang et al. (2021), or designing *autoregressive policies* (Wang et al., 2021; Deleu et al., 2022; Yang et al., 2023a;b; Deleu et al., 2024) that hinder parallel DAG generation and necessitates learning the transition policies over a multitude of discrete state-action combinations.

Table 1: **Positioning ALIAS among the score-based causal discovery literature.**

Search type[†]	Method (year)	Search space	Generation steps	Generation complexity	Constraint[‡]	Acyclicity assurance[♭]	Nonlinear data	Smooth score not required
Local	GES (2002) (Chickering, 2002)	**DAGs**	-	-	Hard	✓	✓	✓
	NOTEARS (2020) (Zheng et al., 2018; 2020)	Graphs			Soft	✗	✓	✗
	NOCURL (2021) (Yu et al., 2021)	**DAGs**			**None**	✓	✗	✗
	DAGMA (2022) (Bello et al., 2022)	Graphs			Soft	✗	✓	✗
	BaDAG (2023) (Annadani et al., 2023)	**DAGs**			**None**	✓	✓	✗
	COSMO (2024) (Massidda et al., 2024)	Graphs			**None**	✗	✓	✗
Global	RL-BIC (2020) (Zhu et al., 2020)	Graphs	**Single**	**Quadratic**	Soft	✗	✓	✓
	BCD-Nets (2021) (Cundy et al., 2021)	**DAGs**	**Single**	Cubic	**None**	✓	✗	✗
	CORL (2021) (Wang et al., 2021)	Orderings	Multiple (Autoregressive)	Cubic	Hard	✓	✓	✓
	DAG-GFN (2022) (Deleu et al., 2022)	**DAGs**	Multiple (Autoregressive)	Cubic	Hard	✓	✗	✓
	GARL (2023) (Yang et al., 2023b)	Orderings	Multiple (Autoregressive)	Cubic	Hard	✓	✓	✓
	RCL-OG (2023) (Yang et al., 2023a)	Orderings	Multiple (Autoregressive)	Cubic	Hard	✓	✓	✓
	ALIAS (Ours)	**DAGs**	**Single**	**Quadratic**	**None**	✓	✓	✓

[†]Local methods start with an initial graph and update it every iteration, while global methods typically concern with DAG generation parameters.
[‡]Methods with Hard constraints explicitly identify and discard the actions that lead to cycles, while Soft constraints refer to the use of DAG regularizers.
[♭]Methods that guarantee acyclicity only in an annealing limit are considered as do not ensure acyclicity.

In this study, we address the aforementioned limitations of score-based causal discovery methods with a novel RL approach, named **ALIAS** (reinforced d**A**g Learning w**I**thout **A**cyclicity con**S**traints).[1] Our approach employs a generative policy that is capable of generating DAGs in a single-step fashion without any acyclicity regularization or explicit acyclicity maintenance. This enables us to effectively explore and exploit the full DAG space with arbitrary score functions, rather than the restricted ordering space. Specifically, we make the following contributions in this study:

1. At the core of **ALIAS**, we design **Vec2DAG**, a *surjective map* from a continuous domain into the space of all DAGs. We prove that given a fixed number of nodes, this function can translate an unconstrained real-valued vector into a binary matrix that represents a valid DAG, and vice versa–there always exists a vector mapped to every possible DAGs.

2. Thanks to **Vec2DAG**, we are able to devise a policy outputting actions in the continuous domain that are directly associated with high-reward DAGs. The policy is one-step, unconstrained, and costs only a quadratic number of parallel operations w.r.t. the number of nodes, allowing our agent to explore the DAG space very effectively with arbitrary RL method and scoring function. To our knowledge, **ALIAS** is the first score-based causal discovery method based on RL that can explore the exact space of DAGs with an efficient one-step generation, rendering it one of the most efficient realizations of Eq. (1) so far.

3. We demonstrate the effectiveness of the proposed **ALIAS** method in comparison with various state-of-the-arts on a systematic set of numerical evaluations on both synthetic and real-world datasets. Empirical evidence shows that our method consistently surpasses all state-of-the-art baselines under multiple evaluation metrics on varying degrees of nonlinearity, dimensionality, graph density, and model misspecification. For example, our method can achieve an SHD $= 0.2 \pm 0.5$ on very dense graphs with 30 nodes and 8 parents per node on average, and on large graphs with 200 nodes and 400 edges on average, **ALIAS** can still obtain a very low SHD of 2.0 ± 1.9.

We summarize the advantages of **ALIAS** compared with the state-of-the-arts in causal discovery in Table 1 and provide a brief overview of the literature in Appendix A. In addition, Figure 1 shows a snapshot of **ALIAS**'s strong performance in a case of highly complex structures, in which our method can achieve absolute accuracy, while the best baselines in this setting still struggle.

[1]Source code will be made publicly available upon publication.

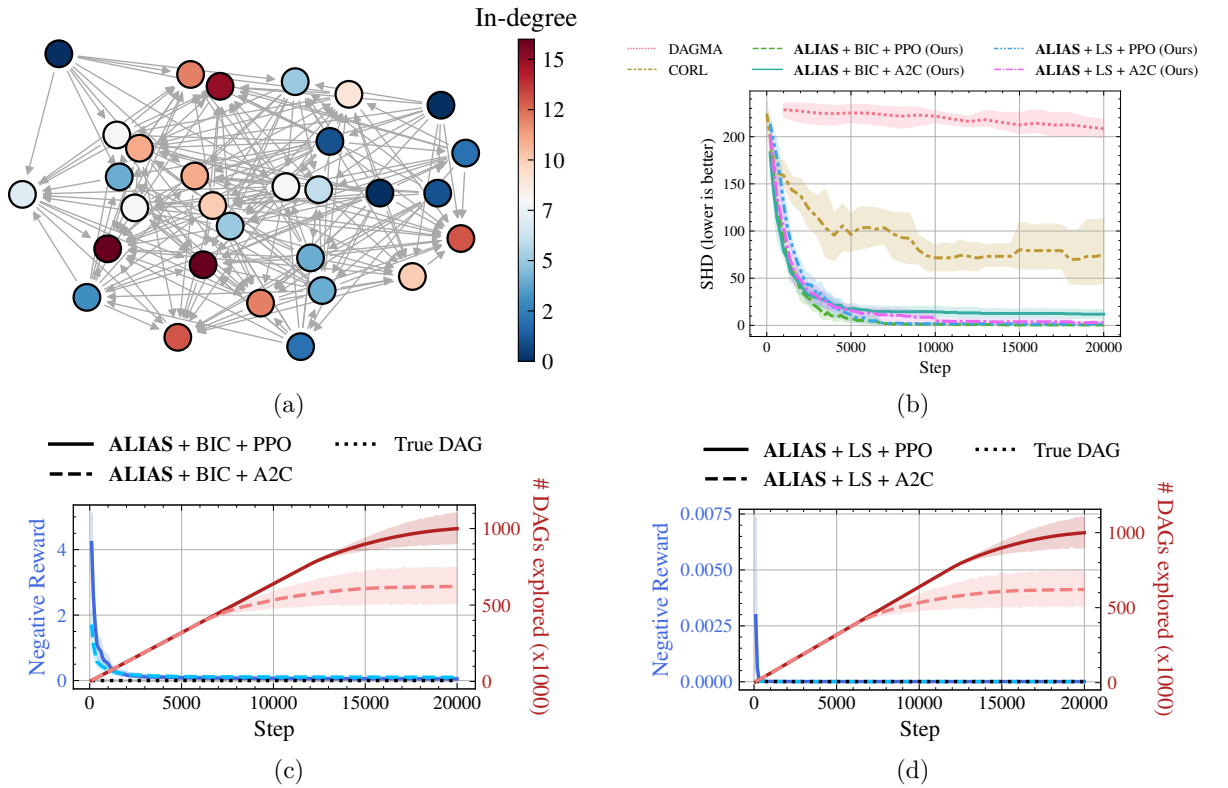


Figure 1: Using merely observational data, the proposed **ALIAS** method **correctly identifies all edges** of (a) a very complex causal dataset with extremely dense connections (linear-Gaussian data with Erdős-Rényi graph of 30 nodes and expected in-degree of 8). (b) We evaluate DAG learning performance in Structural Hamming Distance (SHD, lower is better) on 5 of such datasets with respect to the first 20 000 training steps of four **ALIAS** variants, by combining scoring functions Bayesian Information Criterion (BIC) & Least Squares (LS) with RL methods PPO (Schulman et al., 2017) & A2C (Mnih et al., 2016), in comparison with the best baselines in this setting, namely CORL (Wang et al., 2021) and DAGMA (Bello et al., 2022) (as evaluated in Section 5.2). The best method in this scenario is our **ALIAS** + BIC + PPO variant with zero SHD at the end of the learning process. (c) & (d) For both scores, our method's rewards always approach those of the ground truth DAGs very sharply, which is made possible largely thanks to our efficient DAG parameterization, as well as the continuous exploitation and exploration of the RL algorithms, especially PPO.

2 Background

2.1 Functional Causal Model

Let $\mathbf{X} = (X_1, \ldots, X_d)^\top$ be the d-dimensional random (column) vector representing the variables of interest, $\mathbf{x}^{(k)} = \left(x_1^{(k)}, \ldots, x_d^{(k)}\right)^\top \in \mathbb{R}^d$ denotes the k-th observation of $\mathbf{X}$, and $\mathcal{D} = \left\{\mathbf{x}^{(k)}\right\}_{k=1}^n$ indicates the observational dataset containing n i.i.d. samples of $\mathbf{X}$. Assuming *causal sufficiency*, that is, there are no unobserved endogenous variables, the causal structure among said variables can be described by a DAG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where each vertex $i \in \mathcal{V} = \{1, \ldots, d\}$ corresponds to a random variable X_i, and each edge $(j \to i) \in \mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ implies that X_j is a direct cause of X_i. We also denote the set of all direct causes of a variable as its *parents*, i.e., $\mathrm{pa}_i = \{j \in \mathcal{V} \mid (j \to i) \in \mathcal{E}\}$. The DAG $\mathcal{G}$ is also represented algebraically with a binary *adjacency matrix* $\mathbf{A} \in \{0,1\}^{d \times d}$ where the (i,j)-th entry is 1 iff $(i \to j) \in \mathcal{E}$. Then, the space of all (adjacency matrices of) DAGs of d nodes is denoted by $\mathbb{D}_d \subset \{0,1\}^{d \times d}$. We follow the Functional Causal Model framework (FCM, Pearl, 2009) to assume the data generation process as $X_i := f_i\left(\mathbf{X}_{\mathrm{pa}_i}, E_i\right)$, $\forall i = 1, \ldots, d$, where the noises E_i

are mutually independent. In addition, we also consider *causal minimality* (Peters et al., 2014) to ensure each function f_i is non-constant to any of its arguments. For any joint distribution over $\mathbf{E} = (E_1, \ldots, E_d)$, the functions $\{f_i\}_i^d$ induce a joint distribution over $\mathbf{X}$. The goal of causal discovery is then to recover the acyclic graph $\mathcal{G}$ from empirical samples of $P(\mathbf{X})$.

2.2 DAG Scoring

Among multiple DAG scoring functions well-developed in the literature Schwarz, 1978; Heckerman et al., 1995; Rissanen, 1978, here we focus on the popular Bayesian Information Criterion (BIC) (Schwarz, 1978), which is adopted in many works (Chickering, 2002; Zhu et al., 2020; Wang et al., 2021; Yang et al., 2023a) for its flexibility, computationally straightforwardness, and consistency. More particularly, BIC is a log-likelihood score regularized by the number of model parameters, which is the number of graph edges in our problem. As an example, for additive noise models (ANM) $X_i := f_i\left(\mathbf{X}_{\mathrm{pa}_i}\right) + E_i,\ \forall i = 1, \ldots, d$ with Gaussian noises $E_i \sim \mathcal{N}\left(0; \sigma_i^2\right)$, the BIC-based score can be specified as $\mathcal{S}_{\mathrm{BIC}}(\mathcal{D}, \mathcal{G}) = -(n \sum_{i=1}^{d} \ln \frac{\mathrm{SSR}_i}{n} + |\mathcal{G}| \ln n)$, where $\mathrm{SSR}_i = \sum_{k=1}^{n} (\hat{x}_i^{(k)} - x_i^{(k)})^2$ is the sum of squared residuals after regressing X_i on its parents in $\mathcal{G}$, and we adopt the convention that $|\mathcal{G}|$ is the number of edges in $\mathcal{G}$. Additionally assuming equal noise variances gives us with $\mathcal{S}_{\mathrm{BIC}}(\mathcal{D}, \mathcal{G}) = -(nd \ln \frac{\sum_{i=1}^{d} \mathrm{SSR}_i}{nd} + |\mathcal{G}| \ln n)$. The derivations of BIC scores are presented in Appendix B. A simpler yet widely adopted alternative is the least squares (LS) (Zheng et al., 2018; Lachapelle et al., 2020; Yu et al., 2021; Bello et al., 2022; Massidda et al., 2024), defined with an additional l_0 regularization as $\mathcal{S}_{\mathrm{LS}}(\mathcal{D}, \mathcal{G}) = -(\sum_{i=1}^{d} \mathrm{SSR}_i + \lambda_0 |\mathcal{G}|)$, where $\lambda_0 \geq 0$ is a hyper-parameter for penalizing dense graphs. In our empirical studies, following common practices (Zhu et al., 2020; Wang et al., 2021; Yang et al., 2023b;a), linear regression is used for linear data and Gaussian process regression is adopted for nonlinear data. That being said, any valid regression technique can be seamlessly integrated into our method.

2.3 DAG Representations

Typically, to search over the space of DAGs, modern causal discovery methods either optimize over directed graphs with differentiable DAG regularizers Zheng et al. (2018); Yu et al. (2019); Lee et al. (2020); Zheng et al. (2020); Ng et al. (2020); Wei et al. (2020); Zhu et al. (2020); Zhang et al. (2022), or search over causal orderings and then apply variable selection to suppress redundant edges (Cundy et al., 2021; Charpentier et al., 2022; Chen et al., 2019; Wang et al., 2021; Rolland et al., 2022; Sanchez et al., 2023; Yang et al., 2023b). The former approach does not guarantee the acyclicity of the returned graph with absolute certainty, while the latter approach faces challenges in efficiently generating permutations. For instance, in Cundy et al. (2021) the permutation matrix representing the causal ordering is parametrized by the Sinkhorn operator (Sinkhorn, 1964) followed by the Hungarian algorithm (Kuhn, 1955) with a considerable cost of $\mathcal{O}\left(d^3\right)$. Other examples include ordering-based RL methods (Wang et al., 2021; Yang et al., 2023a) that cost at least $\mathcal{O}\left(d^2\right)$ just to generate a single ordering element, thus totaling an $\mathcal{O}\left(d^3\right)$ complexity for generating a DAG.

Our work takes inspiration from Yu et al. (2021), where a novel unconstrained characterization of *weighted* adjacency matrices of DAGs is proposed. Particularly, a "node potential" vector $\mathbf{p} \in \mathbb{R}^d$ is introduced to model an *implicit* causal ordering, where i precedes j if $p_j > p_i$. Hence, the weight matrix

$$\mathbf{A} = \mathbf{W} \odot \mathrm{ReLU}\left(\mathrm{grad}\left(\mathbf{p}\right)\right), \tag{2}$$

where $\mathbf{W} \in \mathbb{R}^{d \times d}$ and $\mathrm{grad}\left(\mathbf{p}\right)_{ij} := p_j - p_i$ is the gradient flow operator (Lim, 2020), can be shown to correspond to a valid DAG (Theorem 2.1 of (Theorem 2.1, Yu et al., 2021)). However, this weight matrix is only applicable for linear models, and the representation is only used as a refinement for the result returned by a constrained optimization problem. An alternative to this characterization is recently introduced by Massidda et al. (2024), where $\mathbf{A} = \mathbf{W} \odot \mathrm{sigmoid}\left(\mathrm{grad}\left(\mathbf{p}\right)/\tau\right)$, yet this approach only ensures acyclicity at the limit of the annealing temperature $\tau \to 0^+$, which is usually not exactly achieved in practice. Additionally, the equivalent DAG formulation of Annadani et al. (2023) uses the node potential $\mathbf{p}$ to represent a smooth *explicit* permutation matrix: $\sigma\left(\mathbf{p}\right) := \lim_{\tau \to 0^+} \mathrm{Sinkhorn}(\mathbf{p} \cdot [1 \ldots d]/\tau)$, again necessitating a temperature scheduler and the expensive Sinkhorn operator, which reportedly requires 300 iterations to converge and an $\mathcal{O}\left(d^3\right)$ complexity of the Hungarian algorithm, for generating a single DAG.

3 Vec2DAG: Unconstrained Parametrization of DAGs

3.1 The Vec2DAG Operator

Extending from the formulation in Eq. (2), we design a deterministic translation from an unconstrained continuous space to the space of general *binary* adjacency matrices of all DAGs, not restricted to linear models. To be more specific, in addition to the node potential vector $\mathbf{p} \in \mathbb{R}^d$, we introduce a strictly upper-triangular "edge potential" matrix $\mathbf{E} \in \mathbb{R}^{d \times d}$, which can be described using $\frac{d \cdot (d-1)}{2}$ parameters. We then combine them with $\mathbf{p}$ to create a unified representation vector $\mathbf{z} \in \mathbb{S}_d = \mathbb{R}^{d \cdot (d+1)/2}$, which is the parameter space of all d-node DAGs in our method. Furthermore, we denote by $\mathbf{p}(\mathbf{z})$ and $\mathbf{E}(\mathbf{z})$ the node and edge potential components constituting $\mathbf{z}$, respectively. Then, our unconstrained DAG parametrization $\mathbf{Vec2DAG}_d$ for d nodes can be defined as follows.

Definition 1. For all $d \in \mathbb{N}^+$ and $\mathbf{z} \in \mathbb{S}_d$:

$$\mathbf{Vec2DAG}_d(\mathbf{z}) := H\left(\mathbf{E}(\mathbf{z}) + \mathbf{E}(\mathbf{z})^\top\right) \odot H\left(\operatorname{grad}(\mathbf{p}(\mathbf{z}))\right), \tag{3}$$

where $H(x) := \begin{cases} 1 & \text{if } x > 0, \\ 0 & \text{otherwise.} \end{cases}$ is known as the Heaviside step function and $\odot$ is the Hadamard (element-wise) product operator.

The procedure to sample a DAG is then denoted as $\mathbf{z} \sim P(\mathbf{z})$, $\mathbf{A} = \mathbf{Vec2DAG}_d(\mathbf{z})$, which can be implemented in a few lines of code, as illustrated in Figure 5 of the Appendix. The validity of our parametrization is justified by the following theorem.

Theorem 1. *For all $d \in \mathbb{N}^+$, let $\mathbf{Vec2DAG}_d : \mathbb{S}_d \to \{0,1\}^{d \times d}$ be defined as in Eq. 3. Then, $\operatorname{Im}(\mathbf{Vec2DAG}_d) = \mathbb{D}_d$, where $\operatorname{Im}(\cdot)$ is the Image operator, and $\mathbb{D}_d$ is the space of all d-node DAGs.*

The proof can be found in Appendix C.1. Our formulation directly represents a DAG by a real-valued vector, which is in stark contrary to existing unconstrained methods that only aim for a DAG sampler (Cundy et al., 2021; Wang et al., 2021; Charpentier et al., 2022; Deleu et al., 2022; Annadani et al., 2023; Yang et al., 2023b;a). More notably, this approach requires **no temperature annealing** like in (Massidda et al., 2024; Annadani et al., 2023), and can generate a valid DAG in a **single step** since sampling $\mathbf{z}$ can be done instantly in an unconstrained manner. In addition, this merely costs $\boldsymbol{\mathcal{O}\left(d^2\right)}$ **parallelizable operations** compared with the $\mathcal{O}\left(d^3\right)$ cost of sequentially generating permutations using the Sinkhorn operator (Cundy et al., 2021; Charpentier et al., 2022; Annadani et al., 2023) and multiple-step RL methods (Wang et al., 2021; Yang et al., 2023b;a). Moreover, our generation technique is one-step, and thus **does not require learning any transition function**, which vastly reduces the computational burden compared with RL methods based on sequential decisions.

3.2 Properties of Vec2DAG

In this section, we show that our parameterization **Vec2DAG** has some important additional properties that set it apart from past formulations.

Lemma 1. *(Scaling and Translation Invariance). For all $d \in \mathbb{N}^+$, let $\mathbf{Vec2DAG}_d : \mathbb{S}_d \to \{0,1\}^{d \times d}$ be defined as in Eq. 3. Then, for all $\mathbf{z} \in \mathbb{S}_d$, $\alpha > 0$, and $\boldsymbol{\beta} \in \mathbb{S}_d$ such that $|\mathbf{p}(\boldsymbol{\beta})_i| < 1/2 \min_j \left|\mathbf{p}(\mathbf{z})_i - \mathbf{p}(\mathbf{z})_j\right|$ and $\left|\mathbf{E}(\boldsymbol{\beta})_{ij}\right| < \left|\mathbf{E}(\mathbf{z})_{ij}\right|$ $\forall i, j$, we have $\mathbf{Vec2DAG}_d(\mathbf{z}) = \mathbf{Vec2DAG}_d(\alpha \cdot (\mathbf{z} + \boldsymbol{\beta}))$.*

This insight is proven in Appendix C.2, indicating that any DAG can be diversely constructed by infinitely many representations, suggesting a dense parameter space where representations of different DAGs are close to each other. This leads us to the next point, which shows an upper bound of the distance between an arbitrary representation with a representation of any DAG.

Lemma 2. *(Proximity between DAGs). Let $\mathbf{z} \in \mathbb{S}_d$. Then, for any DAG $\mathbf{A} \in \mathbb{D}_d$ and $\epsilon > 0$, there exists $\mathbf{z}_\mathbf{A}$ in the unit ball $B(\infty; \|\mathbf{z}\|_\infty + \epsilon)$ around $\mathbf{z}$ such that $\mathbf{Vec2DAG}_d(\mathbf{z}_\mathbf{A}) = \mathbf{A}$.*

Algorithm 1 ALIAS method for causal discovery.

Input: Dataset $\mathcal{D} = \left\{\mathbf{x}^{(k)}\right\}_{k=1}^{n}$, score function $\mathcal{S}\left(\mathcal{D}, \cdot\right)$, batch size B, and learning rate η.
Output: Estimated causal DAG $\hat{\mathcal{G}}$.

1: **while** not terminated **do**
2: Draw a minibatch of B actions from the policy: $\left\{\mathbf{z}^{(k)} \sim \pi_{\boldsymbol{\theta}}\right\}_{k=1}^{B}$.
3: Collect rewards $\left\{r^{(k)} := \frac{1}{n \times d}\mathcal{S}\left(\mathcal{D}, \textbf{Vec2DAG}_d\left(\mathbf{z}^{(k)}\right)\right)\right\}_{k=1}^{B}$. ▷ Sec. 3.1
4: Update policy as: $\boldsymbol{\theta} := \boldsymbol{\theta} + \eta\left(\frac{1}{B}\sum_{k=1}^{B}\nabla_{\boldsymbol{\theta}}\ln\pi_{\boldsymbol{\theta}}\left(\mathbf{z}^{(k)}\right)\cdot r^{(k)}\right)$. ▷ Sec. 4.1
5: **end while**
6: $\mathbf{z} \sim \pi_{\theta}$, $\hat{\mathcal{G}} := \textbf{Vec2DAG}\left(\mathbf{z}\right)$.
7: Post-process $\hat{\mathcal{G}}$ by pruning if needed and return. ▷ Sec. 4.2

We provide the proof in Appendix C.3. This property is not straightforward in existing constrained optimization approaches (Zheng et al., 2018; Lee et al., 2020; Zheng et al., 2020; Lachapelle et al., 2020; Bello et al., 2022), and suggests that the true DAG may be found closer to the initial position if we start from a smaller scale in our framework. We leverage this result in our implementation by restricting $\mathbb{S}_d$ to a hypercube $[-\gamma, \gamma]^{d\cdot(d+1)/2}$ with a relatively small $\gamma = 10$. This has the effect of regularizing the search space but still does not invalidate our Theorem 1, i.e., we can still reach every possible DAGs when searching in this restricted space.

4 ALIAS: Reinforced DAG Learning without Acyclicity Constraints

Given the defined DAG formulation from real-valued vectors, instead of employing continuous optimization over the new space, we leverage reinforcement learning as the search strategy to find the highest-scoring DAG.

4.1 Policy Gradient for DAG Search

Policy and Action. Utilizing RL, we seek for a policy π that outputs a continuous action $\mathbf{z} \in \mathbb{S}_d = \mathbb{R}^{d\cdot(d+1)/2}$, which is the parameter space of DAGs of d nodes. In this work, we consider stochastic policies for better exploration, i.e., we parametrize our policy by an isotropic Gaussian distribution with learnable means and variances: $\pi_{\boldsymbol{\theta}}\left(\mathbf{z}\right) = \mathcal{N}\left(\mathbf{z}; \boldsymbol{\mu}_{\boldsymbol{\theta}}, \text{diag}\left(\boldsymbol{\sigma}^2_{\boldsymbol{\theta}}\right)\right)$. Since our policy generates a DAG representation in just one step, every trajectory starts with the same initial state and terminates after only one transition, so the agent does not need to be aware of the state in our method.

Reward. The reward of an action in our method is set as the graph score of the DAG induced by that action with respect to the observed dataset $\mathcal{D}$ (Section 2.2), and divided by $n \times d$ to maintain numerical stability without modifying the monotonicity of the score: $\mathcal{R}\left(\mathbf{z}\right) := \frac{1}{n\times d}\mathcal{S}\left(\mathcal{D}, \textbf{Vec2DAG}\left(\mathbf{z}\right)\right)$.

Policy Gradient Algorithm. Since our action space is continuous, we employ policy gradient methods, which are well established for handling continuous actions, rather than the value-based approach as in recent RL-based techniques (Wang et al., 2021; Yang et al., 2023b;a). The training objective is to maximize the expected return defined as $\mathcal{J}\left(\boldsymbol{\theta}\right) = \mathbb{E}_{\mathbf{p}\sim\pi_{\boldsymbol{\theta}}}\left[\mathcal{R}\left(\mathbf{p}\right)\right]$. The differential entropy of the policy can also be added as a regularization term to encourage exploration (Mnih et al., 2016), however we find in our experiments that the stochasticity offered by the policy suffices for exploration. That said, we also investigate the effect of entropy regularization in our empirical studies. During training, the parameter $\boldsymbol{\theta}$ is updated in the direction suggested by the policy gradient as given by the policy gradient theorem (Sutton et al., 1999): $\nabla_{\boldsymbol{\theta}}\mathcal{J}\left(\boldsymbol{\theta}\right) = \mathbb{E}_{\mathbf{p}\sim\pi_{\boldsymbol{\theta}}(\mathbf{p})}\left[\nabla_{\boldsymbol{\theta}}\ln\pi_{\boldsymbol{\theta}}\left(\mathbf{p}\right)\cdot\mathcal{R}\left(\mathbf{p}\right)\right]$.

Note that since our trajectories are one-step and our environment is deterministic, the state-action value function is always equal to the immediate reward, and therefore there is no need for a critic to estimate the value function. Hence, vanilla policy gradient works well out-of-the-box for our framework, yet in practice our

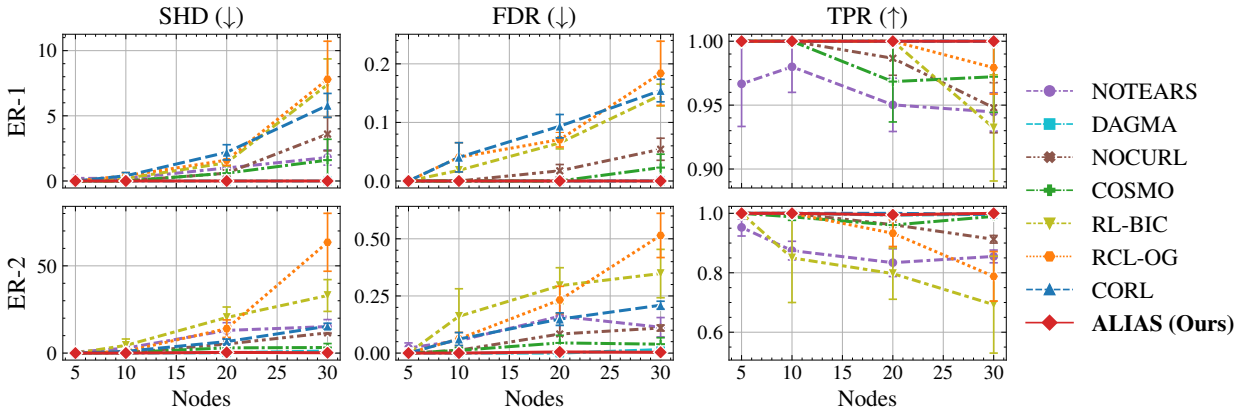


Figure 2: **Causal Discovery Performance on Linear-Gaussian Data.** ER-1 and ER-2 denote Erdős-Rényi graph models with expected in-degrees of 1 and 2, respectively. We compare the proposed **ALIAS** method with NOTEARS (Zheng et al., 2018), DAGMA (Bello et al., 2022), NOCURL (Yu et al., 2021), COSMO (Massidda et al., 2024), RL-BIC (Zhu et al., 2020), CORL (Wang et al., 2021), and RCL-OG (Yang et al., 2023a). The performance metrics are Structural Hamming Distance (SHD, lower is better), False Detection Rate (FDR, lower is better), and True Positive Rate (TPR, higher is better). Shaded areas depict standard errors over 5 independent runs.

method can be implemented with more advanced algorithms for improved training efficiency. Our practical implementation considers the basic policy algorithm Advantage Actor-Critic (A2C, Mnih et al., 2016) and a more advanced method Proximal Policy Optimization (PPO, Schulman et al., 2017). In addition, while policy gradient only ensures local convergence under suitable conditions (Sutton et al., 1999), our empirical evidence remarks that our method can reach the exact ground truth DAG in notably many cases.

4.2 Post Processing

With limited sample sizes, due to overfitting, redundant edges may still be present in the returned DAG whose highest score. One approach towards suppressing the false discovery rate is to greedily remove edges with non-substantial contributions in the score. For linear models, a standard approach is to threshold the absolute values of the estimated weight matrix $\mathbf{W}$ at a certain level δ (Zheng et al., 2018; Ng et al., 2020; Bello et al., 2022), i.e., removing all edges $(i \to j)$ with $|\mathbf{W}_{ij}| < \delta$. For nonlinear models, the popular CAM pruning method (Bühlmann et al., 2014) can be employed for generalized additive models (GAMs), which performs a GAM regression on the parents set and exclude the parents that do not pass a predefined significance level. An alternative pruning method that does not depend on the causal model is based on conditional independence (CI), i.e., by imposing Faithfulness (Spirtes et al., 2000), for each $j \in \mathrm{pa}_i$ in the graph found so far, we remove the edge $(j \to i)$ if $X_i \perp\!\!\!\perp X_j \mid X_{\mathrm{pa}_i \setminus \{j\}}$, which is a direct consequence of the Faithfulness assumption and can be realized with available CI tests like KCIT (Zhang et al., 2011). In addition, for the least squares score, we can increase the regularization strength on the number of edges to encourage sparsity during the learning process. Our numerical experiments investigate the effects of all these approaches. To summarize, Algorithm 1 highlights the key steps of our **ALIAS** method.

5 Numerical Evaluations

5.1 Experiment Setup

We conduct extensive empirical evaluations on both simulated and real datasets, where the ground truth DAGs are available, to compare the efficiency of the proposed **ALIAS** method with up-to-date state-of-the-arts in causal discovery, including the constrained continuous optimization approaches with soft DAG constraints NOTEARS (Zheng et al., 2018; 2020) and DAGMA (Bello et al., 2022), unconstrained continuous optimization

Table 2: **Causal discovery performance on dense graphs (30-node ER-8) and high-dimensional graphs (200-node ER-2) with linear-Gaussian data.** The performance metrics are Structural Hamming Distance (SHD, lower is better), False Detection Rate (FDR, lower is better), and True Positive Rate (TPR, higher is better). The numbers are *mean ± standard deviation* over 5 independent runs. **Bold**: best performance, underline: second-best performance. RL-BIC & CORL fail to run high-dimensional tasks.

Method	Dense graphs (30 nodes, ≈ 240 edges)			High-dimensional graphs (200 nodes, ≈ 400 edges)		
	SHD (↓)	FDR (↓)	TPR (↑)	SHD (↓)	FDR (↓)	TPR (↑)
NOTEARS (Zheng et al., 2018)	141.2 ± 26.5	0.25 ± 0.06	0.55 ± 0.07	53.8 ± 14.6	0.06 ± 0.02	0.93 ± 0.02
DAGMA (Bello et al., 2022)	67.6 ± 17.8	0.14 ± 0.04	0.82 ± 0.04	9.6 ± 6.1	0.02 ± 0.01	0.99 ± 0.01
NOCURL (Yu et al., 2021)	147.6 ± 12.8	0.32 ± 0.03	0.63 ± 0.01	227.6 ± 39.2	0.20 ± 0.07	0.59 ± 0.04
COSMO (Massidda et al., 2024)	97.4 ± 15.3	0.24 ± 0.03	0.80 ± 0.04	158.0 ± 43.6	0.25 ± 0.06	0.87 ± 0.05
RL-BIC (Zhu et al., 2020)	180.6 ± 48.6	0.43 ± 0.13	0.42 ± 0.32	-	-	-
CORL (Wang et al., 2021)	82.4 ± 49.9	0.23 ± 0.11	0.87 ± 0.10	-	-	-
RCL-OG (Yang et al., 2023a)	199.7 ± 15.9	0.47 ± 0.03	0.51 ± 0.08	1076.6 ± 64.3	0.89 ± 0.01	0.32 ± 0.03
ALIAS (Ours)	$\mathbf{0.2 \pm 0.5}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{2.0 \pm 1.9}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$

approaches NOCURL (Yu et al., 2021) and COSMO (Massidda et al., 2024), as well as three RL-based methods RL-BIC (Zhu et al., 2020), CORL (Wang et al., 2021), and RCL-OG (Yang et al., 2023a). A brief description of these methods along with their implementation details and hyper-parameter specifications are provided in Appendix D and the evaluation metrics are described in Appendix D.2.1. For the main experiments, we use the variant with BIC score and PPO algorithm for our method, and examine other variants in the ablation studies. We report supplementary results, including ablation studies and runtime, in Appendix E.

5.2 Linear Data with Gaussian and non-Gaussian Noises

For a given number of nodes d, we first generate a DAG following the Erdős-Rényi graph model (Erdős & Rényi, 1960) with an expected in-degree of $k \in \mathbb{N}^+$, denoted by ER-k. Next, edge weights are randomly sampled from the uniform distribution $P(\mathbf{W})$, and the noises are drawn from the standard Gaussian $E_i \sim \mathcal{N}(0, 1)$. To make this setting more challenging, we use a wider range $P(\mathbf{W}) = \mathcal{U}([-5, -2] \cup [2, 5])$ compared with the common range of $\mathcal{U}([-2, -0.5] \cup [0.5, 2])$ in previous studies (Zheng et al., 2018; Zhu et al., 2020; Wang et al., 2021; Bello et al., 2022). We then sample $n = 1\,000$ observations for each dataset and. This causal model is identifiable due to the equal noise variances (Peters et al., 2014). For fairness, we also apply the same pruning procedure with linear regression coefficients thresholded at 0.3 for all methods and use the equal-variance BIC (Section 2.2) for RL-BIC, CORL, RCL-OG, and **ALIAS**.

Small to moderate graphs. In Figure 2 we report the causal discovery performance for linear-Gaussian data with small to moderate graph sizes and densities, showing that our method consistently achieves near-perfect performance in all metrics, which can be expected thanks to its ability to explore the DAG space competently. Overall, the closest method with comparable performance to our method in this case is DAGMA, followed by COSMO, which are among the most advanced continuous optimization approaches, while other methods, including RL-based ones, still struggle even in this simplest scenario.In addition, apart from ER graphs, the results for Scale-Free (SF) graphs can also be found in Appendix E.

Dense & High-dimensional graphs. We next test the proposed method's ability to adapt to highly complex scenarios, including the cases with very dense graphs (ER-8 graphs) and larger number of nodes (200-node graphs). In this case, we use the common weight range of $\mathcal{U}([-2, -0.5] \cup [0.5, 2])$ to avoid numerical instabilities due to larger graphs. Table 2 depicts that for dense graphs, our method makes almost no mistake while the best baseline in this case, which is DAGMA, still has a significantly large SHD of 67.6 ± 17.8. The performance gap is narrower in the high-dimensional setting with 200 nodes, yet our method remains the leading approach with an SHD of only 2, compared with an SHD of nearly 10 for the second-best method DAGMA.

Table 3: **Causal discovery performance under noise misspecification on linear data with 30-node ER-2 graphs.** The numbers are *mean ± standard deviation* over 5 runs. **Bold**: best performance, underline: second-best performance.

	SHD (lower is better)			
Method\Noise	Exp (1)	Gumbel (0, 1)	Laplace (0, 1)	Uniform (−1, 1)
NOTEARS (Zheng et al., 2018)	6.0 ± 3.6	4.0 ± 4.2	3.0 ± 3.1	6.4 ± 9.6
DAGMA (Bello et al., 2022)	1.0 ± 2.2	**0.2 ± 0.5**	1.0 ± 2.2	4.8 ± 3.4
NOCURL (Yu et al., 2021)	10.8 ± 1.9	6.6 ± 3.7	4.2 ± 2.17	29.0 ± 6.8
COSMO (Massidda et al., 2024)	5.4 ± 4.5	5.0 ± 5.2	7.6 ± 5.5	5.6 ± 4.0
RL-BIC (Zhu et al., 2020)	66.8 ± 29.5	34.4 ± 21.3	31.4 ± 22.6	31.4 ± 25.8
CORL (Wang et al., 2021)	12.0 ± 3.2	15.4 ± 5.3	15.6 ± 4.9	15.0 ± 2.0
RCL-OG (Yang et al., 2023a)	77.3 ± 30.1	79.8 ± 51.0	41.3 ± 36.6	57.0 ± 51.6
ALIAS (Ours)	**0.4 ± 0.6**	0.4 ± 0.9	**0.8 ± 0.8**	**0.4 ± 0.6**

Effect of sample size. We further investigate the behavior of **ALIAS** under data scarcity and redundancy. We again consider the difficult configuration of ER-8 graphs, and vary the sample size from very limited (100) to redundant (5 000) in Figure 3, where it is shown that our method with just 100 samples can surpass DAGMA even with 5 000 samples.

Model misspecification. Next, we consider the model misspecification scenarios when the Gaussian noise assumption is violated. In Table 3, we benchmark all methods on linear data with four types of non-Gaussian noises. The results indicate that our method is still the most robust to noise mis-specification, with an SHD of less than one in all four cases, and is the lead performer in three out of four configurations. In addition, we further study the performance of our method under different model misspecification scenarios, including mismatched causal model, noisy data, and the presence of hidden confounders, in Appendix E.4.

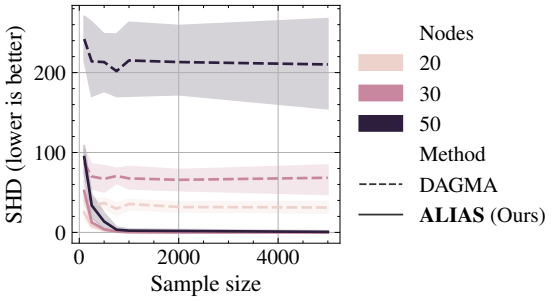


Figure 3: **Causal Discovery performance (linear-Gaussian data on ER-8 graphs) as function of sample size** (100 **to** 5 000). Shaded areas depict standard errors over 5 independent runs.

5.3 Nonlinear Data with Gaussian Processes

In this section, to answer the question of whether our method can operate beyond the standard linear-Gaussian setting, we follow the evaluations in (Zhu et al., 2020; Wang et al., 2021; Yang et al., 2023a) to sample each causal mechanism f_i from a Gaussian process with an RBF kernel of unit bandwidth, and the noises follow normal distributions with different variances sampled uniformly. We also follow Zhu et al. (2020); Wang et al. (2021) to apply Gaussian process regression to calculate the BIC with non-equal variances (Section 2.2) for RL-BIC, CORL, RCL-OG, and our **ALIAS** method. For NOTEARS, DAGMA, and COSMO, we use their nonlinear versions where Multiple-layer Perceptrons (MLP) are used to model nonlinear relationships.

The empirical results reported in Table 4 verify the effectiveness of our method even on nonlinear data. Our method outperforms all other baselines in all metrics, either with or without pruning. Most remarkably, even without pruning, our method correctly identifies nearly every edge with an expected SHD lower than 1. However, by using CAM pruning, there is a slight degrade in performance of most methods, which could be due to CAM's inability to capture complex causal mechanisms drawn from Gaussian processes.

Furthermore, following (Lachapelle et al., 2020), we also study the case of causal model mis-specification with Post-nonlinear models Zhang & Hyvärinen, 2009 in Appendix E.

Table 4: **Causal discovery performance on nonlinear data with Gaussian processes on 10-node ER-4 graphs.** The performance metrics are Structural Hamming Distance (SHD, lower is better), False Detection Rate (FDR, lower is better), and True Positive Rate (TPR, higher is better). The numbers are *mean ± standard deviation* over 5 runs. **Bold**: best performance, underline: second-best performance. Since the graphs are dense and the noise is additive, we also study the effect of pruning the output graphs with CAM pruning (Bühlmann et al., 2014).

	No Pruning			**CAM Pruning**		
Method	**SHD (↓)**	**FDR (↓)**	**TPR (↑)**	**SHD (↓)**	**FDR (↓)**	**TPR (↑)**
NOTEARS (Zheng et al., 2020)	28.4 ± 3.2	0.33 ± 0.14	0.33 ± 0.09	28.6 ± 2.3	0.32 ± 0.13	0.32 ± 0.08
DAGMA (Bello et al., 2022)	25.8 ± 3.7	0.32 ± 0.09	0.40 ± 0.12	26.0 ± 4.0	0.31 ± 0.09	0.39 ± 0.13
NOCURL (Yu et al., 2021)	35.2 ± 2.1	0.47 ± 0.20	0.15 ± 0.08	35.0 ± 1.9	0.46 ± 0.19	0.15 ± 0.08
COSMO (Massidda et al., 2024)	26.4 ± 4.5	0.30 ± 0.09	0.39 ± 0.10	27.0 ± 5.5	0.28 ± 0.09	0.35 ± 0.12
RL-BIC (Zhu et al., 2020)	39.0 ± 4.5	0.06 ± 0.14	0.05 ± 0.10	39.2 ± 4.1	0.05 ± 0.12	0.04 ± 0.09
CORL (Wang et al., 2021)	8.4 ± 4.1	0.19 ± 0.09	0.90 ± 0.06	9.6 ± 3.9	0.10 ± 0.08	0.82 ± 0.08
RCL-OG (Yang et al., 2023a)	7.0 ± 3.1	0.16 ± 0.07	0.94 ± 0.04	9.2 ± 3.0	0.12 ± 0.09	0.84 ± 0.05
ALIAS (Ours)	$\mathbf{0.8 \pm 0.8}$	$\mathbf{0.01 \pm 0.02}$	$\mathbf{0.99 \pm 0.01}$	$\mathbf{4.6 \pm 2.1}$	$\mathbf{0.01 \pm 0.03}$	$\mathbf{0.89 \pm 0.04}$

Table 5: **Causal discovery performance on real-world flow cytometry data (Sachs et al., 2005) with 11 nodes, 17 edges, and 853 samples.** Running time is compared among RL-based methods. The figures for RL-BIC are as originally reported. **Bold**: best performance, underline: second-best performance. Since the causal model is potentially non-additive, we also consider CIT-based pruning with KCIT (Zhang et al., 2011).

	CAM Pruning			**CIT Pruning**		
Method	**Total edges**	**Correct (↑) edges**	**SHD (↓)**	**Total edges**	**Correct (↑) edges**	**SHD (↓)**
NOTEARS (Zheng et al., 2020)	8	5	13	7	5	13
DAGMA (Bello et al., 2022)	6	2	15	6	2	15
NOCURL (Yu et al., 2021)	4	2	15	4	2	15
COSMO (Massidda et al., 2024)	5	2	16	5	2	16
RL-BIC (Zhu et al., 2020)	10	7	11	-	-	-
CORL (Wang et al., 2021)	9	3	14	10	3	15
RCL-OG (Yang et al., 2023a)	9	5	13	9	5	13
ALIAS (Ours)	10	**8**	**10**	9	**8**	**9**

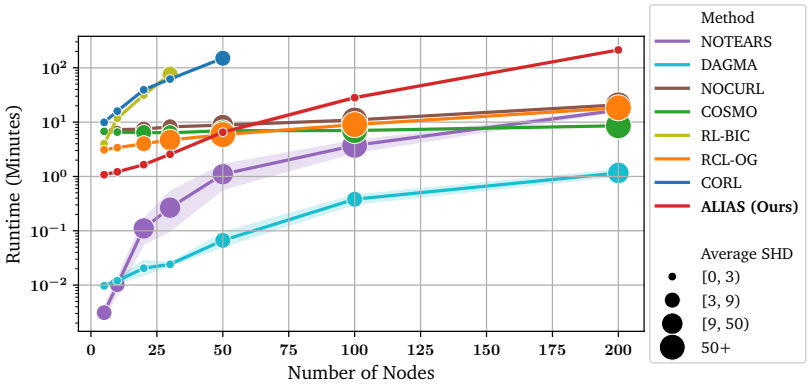
5.4 Real Data

Next, to confirm the validity of our method past synthetic data, we evaluate it on the popular benchmark flow cytometry dataset (Sachs et al., 2005), which involves a protein signaling network based on expression levels of proteins and phospholipids. We employ the observational partition of the dataset with 853 samples, 11 nodes, and 17 edges.

The empirical results provided in Table 5 show that our method **ALIAS** both achieves the best SHD and number of correct edges among all approaches. Specifically, using CAM pruning under the assumption of generalized additive noise models, we achieve the lowest SHD of 10 compared with the second-best of 11 by RL-BIC. Meanwhile, when using CIT-based pruning, we can even further reduce the SHD to 9, with 8 out of 9 identified edges are correct.

Nodes	ALIAS	CORL	RCL-OG
50	0.3	45.7 (135.7×)	46.4 (137.7×)
100	0.2	146.8 (900.0×)	188.6 (1,155.9×)
150	0.3	359.6 (1,261.5×)	471.3 (1,653.2×)
200	0.4	640.6 (1,596.8×)	915.5 (2,282.0×)

(a) Average sampling time for each DAG in milliseconds.



(b) Causal Discovery Runtime.

Figure 4: **Runtime analysis of ALIAS.** (a) We demonstrate the efficiency of our $\mathcal{O}\left(d^2\right)$ sampling technique compared with the $\mathcal{O}\left(d^3\right)$ approaches in CORL and RCL-OG. (b) We study the runtime of all methods with respect to graph size. The numbers are *mean ± standard deviation* over 5 random datasets (ER-2 graphs).

5.5 Runtime Analysis

Here, we analyze the efficiency of **ALIAS** in details. First, to show the significance of our optimal quadratic complexity for sampling DAGs, our Figure 4a compares the runtime of our policy with the autoregressive sampling approaches in CORL and RCL-OG with cubic complexity. It can be seen that, our DAG sampling policy is much faster than other approaches and does not significantly slow down with increasing graph sizes. Meanwhile, CORL and RCL-OG are only nearly 140 times slower than **ALIAS** at only 50 nodes, and the speedup ratio drastically increases with the growth of the graph. Second, in Figure 4b, we detail the runtime and performance of all methods with varying numbers of nodes. For small- to moderate-sized graphs of up to 50 nodes, our method is even faster than gradient-based methods NOCURL and COSMO. For larger graphs, RL-BIC and CORL become computationally expensive very rapidly, and while other methods become faster than **ALIAS**, their performance quickly degrade with significantly larger SHDs than our method.

5.6 Ablation Studies

In Figure 1, we study the effect of the choice of graph scorer, RL method, and number of training steps onto the performance of **ALIAS** compared with DAGMA and CORL as the representatives for continuous optimization and RL-based approaches. It can be seen that all variants of our method surpass the baselines, using as few as 1 000 training steps. While all variants perform equivalently well, PPO proves to be a better choice than A2C, with both variants PPO + BIC and PPO + LS can reach very close to zero SHD, while those of A2C are not as performant (Figure 1b). The influence of other hyper-parameters can be found in Appendix E.

Furthermore, to show that the effectiveness of **ALIAS** is not only thanks to the **Vec2DAG** alone, but also the application of RL, we replace RL in our method with a continuous optimizer, which is popular among modern causal discovery methods. However, since Vec2DAG is discrete, which renders the objective non-differentiable as is, we make slight modifications to make it amenable for continuous optimization, and the adapted version for linear data is given as:

$$\text{Vec2DAG}^{\text{cont.}} := (\mathbf{E}(\mathbf{z}) + \mathbf{E}(\mathbf{z})^\top) \odot H(\text{grad}(\mathbf{p}(\mathbf{z}))),$$

which still represents the weighted adjacency matrix of a DAG. Still, $H(\cdot)$ is not differentiable, so we further use the Straight-Through estimator (Bengio et al., 2013) to estimate its gradients. Specifically, we use $H(\text{grad}(\mathbf{p}(\mathbf{z})))$ for the forward pass whereas the gradients of the inner part $\frac{\partial \text{grad}(\mathbf{p}(\mathbf{z}))}{\partial \mathbf{z}}$ is used for the

Table 6: **Role of RL in ALIAS.** We replace RL with continuous optimization using Adam (Kingma, 2014) and compare with the RL version. The numbers are *mean ± standard deviation* over 5 random datasets on 10-node ER-2 graphs.

Method	SHD (↓)	FDR (↓)	TPR (↑)
Vec2DAG + continuous optimization (lr $= 10^{-2}, \lambda_1 = 10^{-3}$)	14.0 ± 1.4	0.52 ± 0.12	0.42 ± 0.13
Vec2DAG + continuous optimization. (lr $= 10^{-2}, \lambda_1 = 10^{-5}$)	9.6 ± 5.8	0.36 ± 0.13	0.65 ± 0.19
Vec2DAG + continuous optimization (lr $= 10^{-2}, \lambda_1 = 10^{-7}$)	11.8 ± 3.5	0.45 ± 0.17	0.48 ± 0.14
Vec2DAG + continuous optimization (lr $= 10^{-3}, \lambda_1 = 10^{-3}$)	10.0 ± 3.8	0.37 ± 0.11	0.56 ± 0.14
Vec2DAG + continuous optimization (lr $= 10^{-3}, \lambda_1 = 10^{-5}$)	11.8 ± 5.3	0.44 ± 0.13	0.55 ± 0.19
Vec2DAG + continuous optimization (lr $= 10^{-3}, \lambda_1 = 10^{-7}$)	8.6 ± 0.9	0.32 ± 0.06	0.62 ± 0.03
Vec2DAG + continuous optimization (lr $= 10^{-4}, \lambda_1 = 10^{-3}$)	14.2 ± 4.2	0.53 ± 0.13	0.49 ± 0.11
Vec2DAG + continuous optimization (lr $= 10^{-4}, \lambda_1 = 10^{-5}$)	12.0 ± 4.7	0.45 ± 0.12	0.59 ± 0.08
Vec2DAG + continuous optimization (lr $= 10^{-4}, \lambda_1 = 10^{-7}$)	12.8 ± 6.2	0.45 ± 0.17	0.55 ± 0.10
Vec2DAG + continuous optimization (lr $= 10^{-5}, \lambda_1 = 10^{-3}$)	12.6 ± 8.2	0.49 ± 0.23	0.55 ± 0.26
Vec2DAG + continuous optimization (lr $= 10^{-5}, \lambda_1 = 10^{-5}$)	13.0 ± 7.1	0.51 ± 0.21	0.51 ± 0.22
Vec2DAG + continuous optimization (lr $= 10^{-5}, \lambda_1 = 10^{-7}$)	12.2 ± 5.5	0.49 ± 0.17	0.51 ± 0.18
Vec2DAG + RL	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$

backward pass. This is not done similarly for the first term in **Vec2DAG** because that would require an additional weight matrix to represent linear coefficients, which is redundant compared with the above. Then, since the BIC score used in our RL approach is also non-differentiable, we adopt a likelihood-based loss similar to BIC as follows (which is also used in, e.g., GOLEM, Ng et al., 2020):

$$\mathcal{L}(\mathbf{z}) = \ln\left(\frac{1}{n \times d}\|\mathbf{X} - \mathbf{X} \cdot \text{Vec2DAG}^{\text{cont.}}(\mathbf{z})\|_2^2\right) + \lambda_1|\mathbf{z}|,$$

where λ_1 is the sparsity regularization coefficient. We minimize this loss until convergence using the Adam optimizer (Kingma, 2014) (same as NOCURL, COSMO, and the RL algorithm PPO in our method).

We provide the results in Table 6 with a wide range of hyperparameter choices for the above approach. It can be seen that even in this simple case, the continuous optimization approach performs poorly and cannot compete with our RL approach, confirming that **ALIAS**'s effectiveness is attributed greatly by RL, not just **Vec2DAG** alone.

6 Conclusions

In this study, a novel causal discovery method based on RL is proposed. With the introduction of a new DAG characterization that bridges an unconstrained continuous space to the constrained DAG space, we devise an RL policy that can generate DAGs efficiently without any enforcement of the acyclicity constraint, which helps improve the search for the optimal score drastically. Experiments on a wide array of both synthetic and real datasets confirm the effectiveness of our method compared with state-of-the-art baselines.

Regarding limitations, the RL approaches in our study, which are online RL methods, may be limited in sample efficiency, as exploration data is not effectively ultilized to prioritize visiting promising DAGs, thus potentially requiring more explorations than needed to reach the optimal DAG. Towards this end, more sample-efficient RL approaches, such as Optimistic PPO (Cai et al., 2020), or reward redesign can be considered to enhance exploration in our method, and thus further improve its efficiency.

Future work may involve deepening the understanding on the convergence properties of our method and extending it to more intriguing settings like causal discovery with interventional data and hidden variables.

References

Yashas Annadani, Nick Pawlowski, Joel Jennings, Stefan Bauer, Cheng Zhang, and Wenbo Gong. BayesDAG: Gradient-based posterior sampling for causal discovery. In *Advances in Neural Information Processing Systems*, 2023.

Kevin Bello, Bryon Aragam, and Pradeep Ravikumar. DAGMA: learning DAGs via M-matrices and a log-determinant acyclicity characterization. *Advances in Neural Information Processing Systems*, pp. 8226–8239, 2022.

Emmanuel Bengio, Moksh Jain, Maksym Korablyov, Doina Precup, and Yoshua Bengio. Flow network based generative models for non-iterative diverse candidate generation. In *Advances in Neural Information Processing Systems*, pp. 27381–27394, 2021.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Yoshua Bengio, Salem Lahlou, Tristan Deleu, Edward J Hu, Mo Tiwari, and Emmanuel Bengio. Gflownet foundations. *Journal of Machine Learning Research*, pp. 1–55, 2023.

Peter Bühlmann, Jonas Peters, and Jan Ernest. CAM: Causal additive models, high-dimensional order search and penalized regression. *The Annals of Statistics*, 42:2526–2556, 2014.

Qi Cai, Zhuoran Yang, Chi Jin, and Zhaoran Wang. Provably efficient exploration in policy optimization. In *Proceedings of the International Conference on Machine Learning*, pp. 1283–1294, 2020.

Yinan Cao, Christopher Summerfield, Hame Park, Bruno Lucio Giordano, and Christoph Kayser. Causal inference in the multisensory brain. *Neuron*, 102:1076–1087, 2019.

Bertrand Charpentier, Simon Kibler, and Stephan Günnemann. Differentiable DAG sampling. In *Proceedings of the International Conference on Learning Representations*. 2022.

Wenyu Chen, Mathias Drton, and Y Samuel Wang. On causal discovery with an equal-variance assumption. *Biometrika*, 106:973–980, 2019.

David Maxwell Chickering. Learning Bayesian networks is NP-complete. *Learning from data: Artificial intelligence and statistics V*, pp. 121–130, 1996.

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

Diego Colombo, Marloes H Maathuis, Markus Kalisch, and Thomas S Richardson. Learning high-dimensional directed acyclic graphs with latent and selection variables. *The Annals of Statistics*, pp. 294–321, 2012.

Chris Cundy, Aditya Grover, and Stefano Ermon. BCD nets: Scalable variational approaches for Bayesian causal discovery. In *Advances in Neural Information Processing Systems*, pp. 7095–7110, 2021.

Tristan Deleu, António Góis, Chris Emezue, Mansi Rankawat, Simon Lacoste-Julien, Stefan Bauer, and Yoshua Bengio. Bayesian structure learning with generative flow networks. In *Proceedings of the Uncertainty in Artificial Intelligence*, pp. 518–528, 2022.

Tristan Deleu, Mizu Nishikawa-Toomey, Jithendaraa Subramanian, Nikolay Malkin, Laurent Charlin, and Yoshua Bengio. Joint Bayesian inference of graphical structure and parameters with a single generative flow network. *Advances in Neural Information Processing Systems*, 2024.

Paul Erdős and Alfréd Rényi. On the evolution of random graphs. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, 1960.

Dominique Marie-Annick Haughton. On the choice of a model to fit data from an exponential family. *The Annals of Statistics*, pp. 342–355, 1988.

David Heckerman, Dan Geiger, and David M Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, pp. 197–243, 1995.

Paul Hünermund and Elias Bareinboim. Causal inference and data fusion in econometrics. *The Econometrics Journal*, pp. utad008, 2023.

Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Harold W Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, pp. 83–97, 1955.

Sébastien Lachapelle, Philippe Brouillard, Tristan Deleu, and Simon Lacoste-Julien. Gradient-based neural DAG learning. In *Proceedings of the International Conference on Learning Representations*, 2020.

Hao-Chih Lee, Matteo Danieletto, Riccardo Miotto, Sarah T Cherng, and Joel T Dudley. Scaling structural learning with NO-BEARS to infer causal transcriptome networks. In *Proceedings of the Pacific Symposium on Biocomputing*, pp. 391–402, 2020.

Lek-Heng Lim. Hodge Laplacians on graphs. *Siam Review*, 62:685–715, 2020.

Riccardo Massidda, Francesco Landolfi, Martina Cinquini, and Davide Bacciu. Constraint-free structure learning with smooth acyclic orientations. In *Proceedings of the International Conference on Learning Representations*, 2024.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, pp. 1928–1937, 2016.

Ignavier Ng, AmirEmad Ghassami, and Kun Zhang. On the role of sparsity and DAG constraints for learning linear DAGs. In *Advances in Neural Information Processing Systems*, pp. 17943–17954, 2020.

Judea Pearl. *Causality*. Cambridge University Press, 2009.

Jonas Peters, Joris M Mooij, Dominik Janzing, and Bernhard Schölkopf. Causal discovery with continuous additive noise models. *Journal of Machine Learning Research*, 2014.

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, pp. 1–8, 2021.

Aaditya Ramdas, Sashank Jakkam Reddi, Barnabás Póczos, Aarti Singh, and Larry Wasserman. On the decreasing power of kernel and distance based nonparametric hypothesis tests in high dimensions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2015.

Joseph Ramsey, Madelyn Glymour, Ruben Sanchez-Romero, and Clark Glymour. A million variables and more: the Fast Greedy Equivalence Search algorithm for learning high-dimensional graphical causal models, with an application to functional magnetic resonance images. *International Journal of Data Science and Analytics*, 3:121–129, 2017.

Jorma Rissanen. Modeling by shortest data description. *Automatica*, pp. 465–471, 1978.

Robert W Robinson. Counting unlabeled acyclic digraphs. In *Combinatorial Mathematics V*, pp. 28–43. Springer, 1977.

Paul Rolland, Volkan Cevher, Matthäus Kleindessner, Chris Russell, Dominik Janzing, Bernhard Schölkopf, and Francesco Locatello. Score matching enables causal discovery of nonlinear additive noise models. In *Proceedings of the International Conference on Machine Learning*, pp. 18741–18753, 2022.

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*, 308:523–529, 2005.

Pedro Sanchez, Xiao Liu, Alison Q O'Neil, and Sotirios A. Tsaftaris. Diffusion models for causal discovery via topological ordering. In *Proceedings of the International Conference on Learning Representations*, 2023.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Gideon Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, pp. 461–464, 1978.

Richard Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The Annals of Mathematical Statistics*, pp. 876–879, 1964.

Peter Spirtes and Clark Glymour. An algorithm for fast recovery of sparse causal graphs. *Social Science Computer Review*, 9:62–72, 1991.

Peter Spirtes, Clark N Glymour, Richard Scheines, and David Heckerman. *Causation, prediction, and search*. MIT Press, 2000.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, 1999.

Mark Towers, Jordan K. Terry, Ariel Kwiatkowski, John U. Balis, Gianluca de Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Arjun KG, Markus Krimmel, Rodrigo Perez-Vicente, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Andrew Tan Jin Shen, and Omar G. Younis. Gymnasium, 2023.

Xiaoqiang Wang, Yali Du, Shengyu Zhu, Liangjun Ke, Zhitang Chen, Jianye Hao, and Jun Wang. Ordering-based causal discovery with reinforcement learning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 3566–3573, 2021.

Dennis Wei, Tian Gao, and Yue Yu. DAGs with No Fears: A closer look at continuous optimization for learning Bayesian networks. In *Advances in Neural Information Processing Systems*, pp. 3895–3906, 2020.

Dezhi Yang, Guoxian Yu, Jun Wang, Zhengtian Wu, and Maozu Guo. Reinforcement causal structure learning on order graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 10737–10744, 2023a.

Dezhi Yang, Guoxian Yu, Jun Wang, Zhongmin Yan, and Maozu Guo. Causal discovery by graph attention reinforcement learning. In *Proceedings of the SIAM International Conference on Data Mining*, pp. 28–36, 2023b.

Yue Yu, Jie Chen, Tian Gao, and Mo Yu. DAG-GNN: DAG structure learning with graph neural networks. In *Proceedings of the International Conference on Machine Learning*, pp. 7154–7163, 2019.

Yue Yu, Tian Gao, Naiyu Yin, and Qiang Ji. DAGs with No Curl: An efficient DAG structure learning approach. In *Proceedings of the International Conference on Machine Learning*, pp. 12156–12166, 2021.

Keli Zhang, Shengyu Zhu, Marcus Kalander, Ignavier Ng, Junjian Ye, Zhitang Chen, and Lujia Pan. gCastle: A python toolbox for causal discovery. *arXiv preprint arXiv:2111.15155*, 2021.

Kun Zhang and Aapo Hyvärinen. On the identifiability of the post-nonlinear causal model. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pp. 647–655, 2009.

Kun Zhang, Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. Kernel-based conditional independence test and application in causal discovery. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pp. 804–813, 2011.

Zhen Zhang, Ignavier Ng, Dong Gong, Yuhang Liu, Ehsan Abbasnejad, Mingming Gong, Kun Zhang, and Javen Qinfeng Shi. Truncated matrix power iteration for differentiable DAG learning. In *Advances in Neural Information Processing Systems*, pp. 18390–18402, 2022.

Xun Zheng, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing. DAGs with NO TEARS: Continuous optimization for structure learning. In *Advances in Neural Information Processing Systems*, pp. 9472–9483, 2018.

Xun Zheng, Chen Dan, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing. Learning sparse nonparametric DAGs. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2020.

Shengyu Zhu, Ignavier Ng, and Zhitang Chen. Causal discovery with reinforcement learning. In *Proceedings of the International Conference on Learning Representations*, 2020.

```python
def Vec2DAG(z) -> np.ndarray:
    p = z[:d]                    # R^d
    E = np.zeros((d, d))
    E[np.triu_indices(d, -1)] = z[d:]   # R^(d(d-1)/2)

    A = (E + E.T > 0) * (p[:, None] < p[None, :])
    return A
```

Figure 5: Unconstrained DAG parameterization. This function takes as input a real-valued vector $\mathbf{z} \in \mathbb{R}^{d\cdot(d+1)/2}$ and deterministically transforms it into an adjacency matrix of a d-node DAG.

Appendix

A Related Work

Constraint-based[2] methods like PC, FCI (Spirtes & Glymour, 1991; Spirtes et al., 2000) and RFCI (Colombo et al., 2012) form a prominent class of causal discovery approaches. They first exploit conditional independence relationships statistically exhibited in data via a series of hypothesis tests to recover the skeleton, which is the undirected version of the DAG, and then orient the remaining edges using probabilistic graphical rules. These methods can only identify the causal graph up to its Markov equivalent class (MEC), which contains potentially many DAGs inducing the same data. In addition, their performance heavily relies on the quality of the conditional independence tests, which can deteriorate rapidly with the number of conditioning variables (Ramdas et al., 2015), rendering them unsuitable for large or dense graphs.

Score-based methods is another major class of DAG learners, where each DAG is assigned a properly defined score based on its compatibility with observed data, then the DAG learning problem becomes the optimization problem for the DAG yielding the best score. Score-based methods can be further categorized based on the search approach as follows.

Combinatorial greedy search methods such as GES (Chickering, 2002) and FGES (Ramsey et al., 2017) resort to greedy heuristics to reduce the search space and enforce acyclicity by adding one edge at a time after explicitly checking that it would not introduce any cycle, yet this comes at the cost of the sub-optimality of the result.

Continuous optimization methods improve upon combinatorial optimization methods in scalability by the ingenious smooth acyclicity constraint, introduced and made popular with NOTEARS (Zheng et al., 2018), which turns said combinatorial optimization into a continuous optimization problem. This enables bypassing the adversary between combinatorial enumeration and scalability to allow for exploring the DAG space much more effectively, where multiple edges can be added or removed in an update. Following developments, e.g., Yu et al. (2019); Lee et al. (2020); Zheng et al. (2020); Ng et al. (2020); Yu et al. (2021); Zhang et al. (2022); Wei et al. (2020) contribute to extending and improving the soft DAG characterization in scalability and convergence. Notably, unconstrained DAG parameterizations are also proposed by Yu et al. (2021) and Massidda et al. (2024), which simplify the optimization problem from a constrained to an unconstrained problem. However, continuous optimization methods restrict the choices of the score to be differentiable functions, which exclude many well-studied scores such as BIC, BDe, MDL, or independence-based scores (Bühlmann et al., 2014)..

Reinforcement learning methods have emerged in recent years as the promising replacement for the greedy search heuristics discussed so far, thanks to its search ability via exploration and exploitation. As the pioneer in this line of work, Zhu et al. (2020) introduced the first RL agent that is trained to generate high-reward graphs. To handle acyclicity, they incorporate the soft DAG constraint from Zheng et al. (2018) into the reward function to penalize cyclic graphs. Unfortunately, this may not discard all cycles in the

[2]Note that the term "constraint" here largely refers to statistical constraints, such as conditional independence, while "constraint" in our method refers to the acyclicity enforcement.

solution, but also increase computational cost drastically due to the unnecessary reward calculations for non-DAGs. To mitigate this issue, subsequent studies (Wang et al., 2021; Yang et al., 2023b;a) turn to finding the best-scoring causal ordering instead and subsequently apply variable selection onto the result to obtain a DAG, which naturally relieves our concerns with cycles. More particularly, CORL (Wang et al., 2021) is the first RL method operating on the ordering space, which defines states as incomplete permutations and actions as the element to be added next. GARL (Yang et al., 2023b) is proposed to enhance ordering generation by exploiting prior structural knowledge with the help of graph attention networks. Meanwhile, RCL-OG (Yang et al., 2023a) introduces a notion of order graph that drastically reduces the state space size from $\mathcal{O}(d!)$ to only $\mathcal{O}\left(2^d\right)$. It is also worth noting that the emerging generative flow networks (GFlowNet, Bengio et al., 2021; 2023) offer another technique for learning (distributions of) DAGs (Deleu et al., 2022), in which the generation of DAGs is also viewed as a sequential generation problem, where edges are added one-by-one with explicit exclusions of edges introducing cycles, and the transition probabilities are learned via flow matching objectives. However, the generation of these orderings and DAGs are usually formulated as a Markov decision process, in which elements are iteratively added to the structure in a multiple-step fashion, which prevents efficient concurrent DAG generations and requires learning the transition functions, which is computationally involved given the multitude of discrete state-action combinations.

B Details about BIC scores

BIC is a parametric score that assumes a model for the causal mechanisms, e.g., linear-Gaussian, which comes with a set of parameters ψ. This score is used to approximate the likelihood of data given the model after marginalizing out the model parameters:

$$\ln p(\mathcal{D} \mid \mathcal{G}) = \ln \int p(\mathcal{D} \mid \psi, \mathcal{G})\, p(\psi \mid \mathcal{G})\, \mathrm{d}\psi \approx -\frac{\mathrm{BIC}}{2}. \tag{4}$$

Given a DAG $\mathcal{G}$, the BIC score is defined as follows:

$$\mathrm{BIC}(\mathcal{D}, \mathcal{G}) = -2 \ln p\left(\mathcal{D} \mid \hat{\psi}, \mathcal{G}\right) + |\mathcal{G}| \ln n, \tag{5}$$

where $\hat{\psi}$ is the maximum-likelihood estimator of $p(\mathcal{D} \mid \psi, \mathcal{G})$, $|\mathcal{G}|$ is the number of edges in $\mathcal{G}$, and n is the number of samples in $\mathcal{D}$.

The BIC is consistent in the sense that if a causal model is identifiable, asymptotically, the true DAG has the highest score among all other DAGs Haughton, 1988. Meanwhile, for limited samples, it prevents overfitting by penalizing edges that do not improve the log-likelihood significantly.

B.1 Non-equal variances case

Recall that the additive noise model under Gaussian noise is given by $X_i := f_i\left(\mathbf{X}_{\mathrm{pa}_i}\right) + E_i$, where $E_i \sim \mathcal{N}\left(0, \sigma_i^2\right)$. This implies $X_i \sim \mathcal{N}\left(f_i\left(\mathbf{X}_{\mathrm{pa}_i}\right), \sigma_i^2\right)$ and the log-likelihood of an empirical dataset $\mathcal{D} = \left\{\mathbf{x}^{(k)}\right\}_{k=1}^{n}$ is given by

$$\mathcal{L} = \ln p(\mathcal{D} \mid f, \boldsymbol{\sigma}, \mathcal{G}) \tag{6}$$

$$= -\frac{1}{2} \sum_{k=1}^{n} \sum_{i=1}^{d} \frac{\left(x_i^{(k)} - f_i\left(x_{\mathrm{pa}_i}^{(k)}\right)\right)^2}{\sigma_i^2} \tag{7}$$

$$- \frac{n}{2} \sum_{i=1}^{d} \ln \sigma_i^2 + \mathrm{const}, \tag{8}$$

where the constant does not depend on any variable.

The maximum likelihood estimator for f_i can be found via least square methods, and that of σ_i^2 can be found by solving $\frac{\partial \mathcal{L}}{\partial \sigma_i^2} = 0$, which yields

$$\hat{\sigma}_i^2 = \frac{1}{n} \underbrace{\sum_{k=1}^{n} \left(x_i^{(k)} - \hat{f}_i \left(x_{\mathrm{pa}_i}^{(k)} \right) \right)^2}_{\mathrm{SSR}_i}. \tag{9}$$

Plugging this back to Eqn. (6) gives

$$\hat{\mathcal{L}} = -\frac{n}{2} \sum_{i=1}^{d} \ln \frac{\mathrm{SSR}_i}{n} + \mathrm{const}. \tag{10}$$

Finally, we obtain the BIC score for the non-equal variances case by incorporating this into Eqn. (5):

$$\mathrm{BIC}\left(\mathcal{D}, \mathcal{G}\right) = n \sum_{i=1}^{d} \ln \frac{\mathrm{SSR}_i}{n} + |\mathcal{G}| \ln n + \mathrm{const}, \tag{11}$$

B.2 Equal variances case

Similarly to the unequal variances case, by assuming $\sigma_1 = \ldots = \sigma_d = \sigma$, we solve for $\frac{\partial \mathcal{L}}{\partial \sigma^2} = 0$ and obtain

$$\hat{\sigma}^2 = \frac{1}{nd} \sum_{i=1}^{d} \underbrace{\sum_{k=1}^{n} \left(x_i^{(k)} - \hat{f}_i \left(x_{\mathrm{pa}_i}^{(k)} \right) \right)^2}_{\mathrm{SSR}_i}. \tag{12}$$

Substituting this estimate into Eqn. (6) yields us with

$$\hat{\mathcal{L}} = -\frac{nd}{2} \ln \frac{\sum_{i=1}^{d} \mathrm{SSR}_i}{nd}. \tag{13}$$

For the last step, the BIC score for the equal variance case is given by substitution as

$$\mathrm{BIC}\left(\mathcal{D}, \mathcal{G}\right) = nd \ln \frac{\sum_{i=1}^{d} \mathrm{SSR}_i}{nd} + |\mathcal{G}| \ln n + \mathrm{const}. \tag{14}$$

C Proofs

C.1 Proof of Theorem 1

Theorem. *For all $d \in \mathbb{N}^+$, let $\mathbf{Vec2DAG}_d : \mathbb{S}_d \to \{0,1\}^{d \times d}$ be defined as in Eq. 3. Then, $\mathrm{Im}\left(\mathbf{Vec2DAG}_d\right) = \mathbb{D}_d$.*

To show that **Vec2DAG** is surjective, we first show that every point in the parameter space maps to a directed graph without any cycle, and vice-versa, for any DAG, there always exists a vector that maps to it.

Lemma 3. *For all $d \in \mathbb{N}^+$, let $\mathbf{Vec2DAG}_d : \mathbb{S}_d \to \{0,1\}^{d \times d}$ be defined as in Eq. 3. Then, $\mathbf{Vec2DAG}_d\left(\mathbf{z}\right) \in \mathbb{D}_d \ \forall \mathbf{z} \in \mathbb{S}_d$.*

Proof. The proof follows the same argument with Theorem 2.1 of Yu et al. (2021). Specifically, let $\mathbf{A} = \mathbf{Vec2DAG}_d\left(\mathbf{z}\right) = H\left(\mathbf{E}\left(\mathbf{z}\right) + \mathbf{E}\left(\mathbf{z}\right)^\top\right) \odot H\left(\mathrm{grad}\left(\mathbf{p}\left(\mathbf{z}\right)\right)\right)$. Then, the necessary condition for an edge from i to j to exist is $p_i < p_j$. By contradiction, assuming there exists a cycle $(i_1 \to \ldots \to i_k \to i_1)$ in $\mathbf{A}$, then it follows that $p_{i_1} < \cdots < p_{i_k} < p_{i_1}$. This contradicts with the total-ordering property of real values, thus concluding our argument. □

Lemma 4. *For all $d \in \mathbb{N}^+$, let $\mathbf{Vec2DAG}_d : \mathbb{S}_d \to \{0,1\}^{d\times d}$ be defined as in Eq. 3. Then, for any DAG $\mathbf{A} \in \mathbb{D}_d$ and $\epsilon > 0$, there exists $\mathbf{z} \in (-\epsilon, \epsilon)^{d\cdot(d+1)/2} \subset \mathbb{S}_d$ such that $\mathbf{Vec2DAG}_d(\mathbf{z}) = \mathbf{A}$.*

Proof. We prove by construction. Let $\mathrm{an}_i = \{j \neq i \mid \text{there is a path from } j \text{ to } i\}$ be the set of ancestors of i. It follows that if $i \to j$ is in $\mathbf{A}$ then every ancestor of i is an ancestor of j, so $\mathrm{an}_i \subset \mathrm{an}_j$ and $\mathrm{an}_j = \mathrm{an}_i \cup \{i\} \cup (\mathrm{an}_j \setminus \mathrm{an}_i \setminus \{i\})$, which means $|\mathrm{an}_i| < |\mathrm{an}_j|$. Therefore, we can construct the unnormalized node potentials as $\tilde{p}_i = |\mathrm{an}_i|$ and normalize it to fit into the $(-\epsilon, \epsilon)$ range as

$$p_i := \frac{\tilde{p}_i - \min \tilde{p}_i}{\max \tilde{p}_i - \min \tilde{p}_i} \times \epsilon - \frac{1}{2}\epsilon. \tag{15}$$

We then construct the upper-triangular edge potentials matrix as

$$E_{ij} = \begin{cases} 1/2\epsilon & \text{if } i < j \text{ and } A_{ij} + A_{ji} = 1, \\ -1/2\epsilon & \text{otherwise.} \end{cases} \tag{16}$$

□

Lemmas 3 and 4 then completes our proof of Theorem 1.

C.2 Proof of Lemma 1

Lemma 5. *(Scaling and Translation Invariance). For all $d \in \mathbb{N}^+$, let $\mathbf{Vec2DAG}_d : \mathbb{S}_d \to \{0,1\}^{d\times d}$ be defined as in Eq. 3. Then, for all $\mathbf{z} \in \mathbb{S}_d$, $\alpha > 0$, and $\boldsymbol{\beta} \in \mathbb{S}_d$ such that $|\mathbf{p}(\boldsymbol{\beta})_i| < 1/2 \min_j \left|\mathbf{p}(\mathbf{z})_i - \mathbf{p}(\mathbf{z})_j\right|$ and $\left|\mathbf{E}(\boldsymbol{\beta})_{ij}\right| < \left|\mathbf{E}(\mathbf{z})_{ij}\right|$ $\forall i, j$, we have $\mathbf{Vec2DAG}_d(\mathbf{z}) = \mathbf{Vec2DAG}_d(\alpha \cdot (\mathbf{z} + \boldsymbol{\beta}))$.*

Proof. We first show that $\mathbf{Vec2DAG}_d(\mathbf{z}) = \mathbf{Vec2DAG}_d(\alpha \cdot \mathbf{z})$ $\forall \alpha > 0$. Let $\mathbf{A} = \mathbf{Vec2DAG}_d(\mathbf{z})$. This follows from the fact that

$$\mathbf{Vec2DAG}_d(\mathbf{z})_{ij} = 1 \Leftrightarrow \begin{cases} \mathbf{p}(\mathbf{z})_i < \mathbf{p}(\mathbf{z})_j \\ \mathbf{E}(\mathbf{z})_{ij} + \mathbf{E}(\mathbf{z})_{ji} > 0 \end{cases} \tag{17}$$

$$\Leftrightarrow \begin{cases} \alpha\mathbf{p}(\mathbf{z})_i < \alpha\mathbf{p}(\mathbf{z})_j \\ \alpha\mathbf{E}(\mathbf{z})_{ij} + \alpha\mathbf{E}(\mathbf{z})_{ji} > 0 \end{cases} \tag{18}$$

$$\Leftrightarrow \mathbf{Vec2DAG}_d(\alpha \cdot \mathbf{z})_{ij} = 1, \tag{19}$$

or equivalently, $\mathbf{Vec2DAG}_d(\mathbf{z}) = \mathbf{Vec2DAG}_d(\alpha \cdot \mathbf{z})$.

Next, we prove $\mathbf{Vec2DAG}_d(\mathbf{z}) = \mathbf{Vec2DAG}_d(\mathbf{z} + \boldsymbol{\beta})$ if $|\mathbf{p}(\boldsymbol{\beta})_i| < 1/2 \min_j \left|\mathbf{p}(\mathbf{z})_i - \mathbf{p}(\mathbf{z})_j\right|$ and $\left|\mathbf{E}(\boldsymbol{\beta})_{ij}\right| < \left|\mathbf{E}(\mathbf{z})_{ij}\right|$ $\forall i, j$. First, it can be seen that if $p_i < p_j$ and $p'_i, p'_j \in (-1/2(p_j - p_i), 1/2(p_j - p_i))$, then $p_i + p'_i < 1/2(p_i + p_j) < p_j + p'_j$. Therefore, $\mathbf{p}(\mathbf{z})_i < \mathbf{p}(\mathbf{z})_j \Leftrightarrow \mathbf{p}(\mathbf{z})_i + \mathbf{p}(\boldsymbol{\beta})_i < \mathbf{p}(\mathbf{z})_j + \mathbf{p}(\boldsymbol{\beta})_j$ if $|\mathbf{p}(\boldsymbol{\beta})_i| < 1/2 \left|\mathbf{p}(\mathbf{z})_i - \mathbf{p}(\mathbf{z})_j\right|$, which also applies when $|\mathbf{p}(\boldsymbol{\beta})_i| < 1/2 \min_j \left|\mathbf{p}(\mathbf{z})_i - \mathbf{p}(\mathbf{z})_j\right|$.

Similarly, $\mathbf{E}(\mathbf{z})_{ij} > 0 \Leftrightarrow \mathbf{E}(\mathbf{z})_{ij} + \mathbf{E}(\boldsymbol{\beta})_{ij} > 0$ if $\left|\mathbf{E}(\boldsymbol{\beta})_{ij}\right| < \left|\mathbf{E}(\mathbf{z})_{ij}\right|$. Combining these two points entails that if $\left|\mathbf{E}(\boldsymbol{\beta})_{ij}\right| < \left|\mathbf{E}(\mathbf{z})_{ij}\right|$ $\forall i, j$ then we obtain

$$\mathbf{Vec2DAG}_d(\mathbf{z})_{ij} = 1 \Leftrightarrow \begin{cases} \mathbf{p}(\mathbf{z})_i < \mathbf{p}(\mathbf{z})_j \\ \mathbf{E}(\mathbf{z})_{ij} + \mathbf{E}(\mathbf{z})_{ji} > 0 \end{cases} \tag{20}$$

$$\Leftrightarrow \begin{cases} \mathbf{p}(\mathbf{z})_i + \mathbf{p}(\boldsymbol{\beta})_i < \mathbf{p}(\mathbf{z})_j + \mathbf{p}(\boldsymbol{\beta})_j \\ \left(\mathbf{E}(\mathbf{z})_{ij} + \mathbf{E}(\boldsymbol{\beta})_{ij}\right) + \left(\mathbf{E}(\mathbf{z})_{ji} + \mathbf{E}(\boldsymbol{\beta})_{ji}\right) > 0 \end{cases} \tag{21}$$

$$\Leftrightarrow \mathbf{Vec2DAG}_d(\mathbf{z} + \boldsymbol{\beta})_{ij} = 1. \tag{22}$$

Finally, combining Eqs. 19 and 22 concludes our proof for Lemma 1:

$$\mathbf{Vec2DAG}_d\left(\alpha \cdot (\mathbf{z} + \boldsymbol{\beta})\right) = \mathbf{Vec2DAG}_d\left(\mathbf{z} + \boldsymbol{\beta}\right) = \mathbf{Vec2DAG}_d\left(\mathbf{z}\right).$$

□

C.3 Proof of Lemma 2

Lemma 6. *(Proximity between DAGs). Let $\mathbf{z} \in \mathbb{S}_d$. Then, for any DAG $\mathbf{A} \in \mathbb{D}_d$ and $\epsilon > 0$, there exists $\mathbf{z}_{\mathbf{A}}$ in the unit ball $B\left(\infty; \|\mathbf{z}\|_\infty + \epsilon\right)$ around $\mathbf{z}$ such that $\mathbf{Vec2DAG}_d\left(\mathbf{z}_{\mathbf{A}}\right) = \mathbf{A}$.*

Proof. Lemma 4 established that there exists a representation $\mathbf{z}_{\mathbf{A}} \in (-\epsilon, \epsilon)^{d\cdot(d+1)/2}$ for any DAG $\mathbf{A}$ and $\epsilon > 0$. Therefore, the distance between $\mathbf{z}$ and $\mathbf{z}_{\mathbf{A}}$ is bounded in l_∞ norm by the triangle inequality as $|\mathbf{z} - \mathbf{z}_{\mathbf{A}}|_\infty \leq |\mathbf{z}|_\infty + |\mathbf{z}_{\mathbf{A}}| < |\mathbf{z}|_\infty + \epsilon$. □

D Experiment Details

D.1 Datasets

D.1.1 Synthetic Linear-Gaussian Data

To simulate data for a given number of nodes d, we first generate a DAG following the Erdős-Rényi graph model (Erdős & Rényi, 1960) with a random ordering and an expected in-degree of $k \in \mathbb{N}^+$, denoted by ER-k. Next, edge weights are randomly sampled from the uniform distribution $\mathcal{U}\left([-5, -2] \cup [2, 5]\right)$, giving a weighted matrix $\mathbf{W} \in \mathbb{R}^{d\times d}$ where zero entries indicate no connections, then the noises are sampled from the standard Gaussian distribution $E_i \sim \mathcal{N}(0, 1)$.

Finally, we sample $n = 1,000$ observations for each dataset following the linear assignment $X_i := \sum_{j\in \mathrm{pa}_i} W_{ji} X_j + E_i$. While linear-Gaussian models are non-identifiable in general (Spirtes et al., 2000), the instances with equal noise variances adopted in this experiment are known to be fully identifiable (Peters et al., 2014). This data generation process is similar to multiple other works such as (Zheng et al., 2018; Zhu et al., 2020; Wang et al., 2021) and is conducted using the well-established gCastle[3] utility for causality research (Zhang et al., 2021).

D.1.2 Nonlinear Data with Gaussian Processes

We evaluate the performance of the proposed method **ALIAS** with competitors on the exact 5 datasets used by Zhu et al. (2020) in their experiment, which are produced by Lachapelle et al. (2020) (`https://github.com/kurowasan/GraN-DAG`, MIT license). In addition, we also consider the first 5 datasets of the PNL-GP portion of their datasets for the model misspecification experiment.

D.2 Experiment Details

D.2.1 Evaluation Metrics

The estimated graphs are assessed against ground truth DAGs on multiple evaluation metrics, including the commonly employed Structural Hamming Distance (SHD, lower is better), False Detection Rate (FDR, lower is better), and True Positive Rate (TPR, higher is better). SHD counts the minimal number of edge additions, removals, and reversals in order to turn one graph into another. FDR is the ratio of incorrectly estimated edges over all estimated edges, while TPR is the proportion of correctly identified edges over all true edges. We also utilize gCastle for the calculations of the aforementioned metrics.

[3] `https://github.com/huawei-noah/trustworthyAI/tree/master/gcastle`, version 1.0.3, Apache-2.0 license.

D.2.2 Implementations of Methods

Our set of baseline methods contains a wide range of both well-established and recent approaches, including the constrained continuous optimization approaches with soft DAG constraints NOTEARS (Zheng et al., 2018; 2020) and DAGMA (Bello et al., 2022), unconstrained continuous optimization approaches NOCURL (Yu et al., 2021) and COSMO (Massidda et al., 2024), as well as three RL-based methods RL-BIC (Zhu et al., 2020), CORL (Wang et al., 2021), and RCL-OG (Yang et al., 2023a), where the implementations are publicly available. More specifically,

- NOTEARS (Zheng et al., 2018; 2020) is a continuous optimization method that optimizes over the continuous space of weighted adjacency $\mathbf{W} \in \mathbb{R}^{d \times d}$ for linear models, with a soft constraint $\mathbf{W}$ is DAG $\Leftrightarrow h(\mathbf{W}) = e^{\mathbf{W} \circ \mathbf{W}} - d = 0$, which is solved using augmented Lagrangian methods. For nonlinear data, $\mathbf{W}$ is used to mask the input of the MLPs that model the nonlinear causal mechanisms.
- DAGMA (Bello et al., 2022) is an alternative proposal to NOTEARS, with the acyclicity constraint $\mathbf{W}$ is DAG $\Leftrightarrow h(\mathbf{W}) = \log\det(s\mathbf{I} - \mathbf{W} \circ \mathbf{W}) - d \log s = 0$ for all $s > 0$.
- NOCURL (Yu et al., 2021) is proposed as an unconstrained continuous optimization method that represents the weight matrix of a linear model by $\mathbf{A} = \mathbf{W} \odot \mathrm{ReLU}(\mathrm{grad}(\mathbf{p}))$, where $\mathbf{W} \in \mathbb{R}^{d \times d}$ and $\mathrm{grad}(\mathbf{p})_{ij} := p_j - p_i$. Their optimization strategy is to first find a solution from an unconstrained method like NOTEARS, then refine it with this new unconstrained representation.
- COSMO (Massidda et al., 2024) improves upon NOCURL by parametrizing the adjacency matrix with $\mathbf{A} = \mathbf{W} \odot \mathrm{sigmoid}(\mathrm{grad}(\mathbf{p})/\tau)$, which can be shown to converge to a DAG when $\tau \to 0^+$. They optimize directly with this formulation instead of employing the two-stage approach like NOCURL.
- RL-BIC (Zhu et al., 2020) is the first RL-based method, which involves an actor-critic agent that learns to output high-reward graphs. The acyclicity is incorporated into the reward to penalize cyclic graphs.
- CORL (Wang et al., 2021) is the first RL method for ordering-based causal discovery, which revolves around an agent that learns to produce causal orderings with high rewards. The policy sequentially generates each element of the ordering, then the causal order is pruned to obtain a DAG.
- RCL-OG (Yang et al., 2023a) is an alternative RL approach for ordering-based causal discovery, which improves the state space design of CORL. It reduces the state space size from permutations with a factorial size $\mathcal{O}(d!)$ to only $\mathcal{O}(2^d)$.

For NOTEARS, RL-BIC, and CORL, we also adopt the implementations from the gCastle package. For RCL-OG, we employ the implementation provided by the authors at `https://www.sdu-idea.cn/codes.php?name=RCL-OG` (no license provided). Since RCL-OG is essentially a Bayesian method, we use the best-scoring DAG from its 1 000 posterior samples as the output for a fair comparison with other methods.

Default hyper-parameters for each implementation are used unless specifically indicated. The detailed configurations of all methods are provided in Appendix D.2.3.

Our proposed **ALIAS** method is implemented using the Stable-Baselines3[4] toolset (Raffin et al., 2021) with the Advantage Actor-Critic (A2C, Mnih et al., 2016) and Proximal Policy Optimization (PPO, (Schulman et al., 2017)) methods, and a custom DAG environment built on top of Gymnasium[5] (Towers et al., 2023). See Appendix D.2.3 the hyper-parameters for our methods.

Experiments are executed on a mix of several machines running Ubuntu 20.04/22.04 with the matching Python environments, including the following configurations:

- AMD EPYC 7742 CPU, 1TB of RAM, and 8 Nvidia A100 40GB GPUs.

[4] `https://github.com/DLR-RM/stable-baselines3`, MIT license.
[5] `https://github.com/Farama-Foundation/Gymnasium`, MIT license.

Table 7: Default hyper-parameters for our method **ALIAS** throughout the experiments. Unmentioned hyper-parameters are left unchanged.

Experiment	Linear data (Figures 1, 2 and 3, Tables 2, 3, 14, 15, 16, 17, and 18)	Nonlinear GP data (Table 4 and Table 19)	Real data (Table 5)
Batch size (No. parallel environments)	64		
Training steps	20,000		
No. steps to run for each update	1		
Data standardization (dimension-wise)	No (variances should remain equal)	No (data std. is near unit)	Yes (data std. is large)
RL method	PPO		
Advantage normalization	Yes		
Learning rate	Stable-Baseline3' default (0.0003 for PPO and 0.0007 for A2C)		
Regression method	Linear Regression	Gaussian Process Regression	
Scoring method	BIC equal variances, assuming Gaussian noises	BIC non-equal variances, assuming Gaussian noises	
l_0 regularization for LS score	0.000001	-	-

Table 8: Default hyper-parameters for CORL Wang et al. (2021) throughout the experiments. Unmentioned hyper-parameters are left unchanged.

Experiment	Linear data (Figures 1, 2 and 3, Tables 2, 3, 14, 15, 16, 17, and 18)	Nonlinear GP data (Table 4 and Table 19)	Real data (Table 5)
Batch size	64		
Training steps	10,000		
Data standardization (dimension-wise)	No (variances should remain equal)	No (data std. is near unit)	Yes (data std. is large)
Regression method	Linear Regression	Gaussian Process Regression	
Scoring method	BIC equal variances, assuming Gaussian noises	BIC non-equal variances, assuming Gaussian noises	

- Intel Xeon Platinum 8452Y CPU, 1TB of RAM, 4 Nvidia H100 80GB GPUs.
- Intel Core i9 13900KF CPU, 128GB of RAM, 1 Nvidia 4070Ti Super 16GB GPU.

The first configuration is used for batched executions of all methods, while the second is for resource-intensive experiments like high-dimensional ones with RL-based baselines, and the last configuration is for prototyping experiments. We find that RL-based baselines heavily rely on GPU with much slower (hours of) runtime on CPU, even on 30-node graphs, while our method is less dependent on GPU and can even handle datasets of up to 100 nodes in 45 minutes purely on CPU (single process).

D.2.3 Hyper-parameters

We provide the hyper-parameters for each method and experiment scenario as follows:

- **ALIAS** (Ours): Table 7.
- NOTEARS and DAGMA: Table 13.
- NOCURL: Table 11.
- COSMO: Table 12.
- RL-BIC: Table 9.
- CORL: Table 8.
- RCL-OG: Table 10.

Table 9: Default hyper-parameters for RL-BIC Zhu et al. (2020) throughout the experiments. Unmentioned hyper-parameters are left unchanged.

Experiment	Linear data (Figures 1, 2 and 3, Tables 2, 3, 14, 15, 16, 17, and 18)	Nonlinear GP data (Table 4 and Table 19)	Real data (Table 5)
Batch size	64		
Training steps	20,000		
Data standardization (dimension-wise)	No (variances should remain equal)	No (data std. is near unit)	Yes (data std. is large)
Regression method	Linear Regression	Gaussian Process Regression	
Scoring method	BIC equal variances, assuming Gaussian noises	BIC non-equal variances, assuming Gaussian noises	

Table 10: Default hyper-parameters for RCL-OG Yang et al. (2023a) throughout the experiments. Unmentioned hyper-parameters are left unchanged.

Experiment	Linear data (Figures 1, 2 and 3, Tables 2, 3, 14, 15, 16, 17, and 18)	Nonlinear GP data (Table 4 and Table 19)	Real data (Table 5)
Batch size	32		
Training steps	40,000		
Data standardization (dimension-wise)	No (variances should remain equal)	No (data std. is near unit)	Yes (data std. is large)
Regression method	Linear Regression	Gaussian Process Regression	
Scoring method	BIC equal variances, assuming Gaussian noises	BIC non-equal variances, assuming Gaussian noises	

Table 11: Default hyper-parameters for NOCURL Yang et al. (2023a) throughout the experiments. The values are the best parameters yielding the lowest SHD over linear-Gaussian datasets with 30-node ER-4 graphs, found via the tuning process provided by COSMO's implementation. Unmentioned hyper-parameters are left unchanged.

Experiment	Linear data (Figures 1, 2 and 3, Tables 2, 3, 14, 15, 16, 17, and 18)	Nonlinear GP data (Table 4 and Table 19)	Real data (Table 5)
Batch size	64		
Inner iterations	5000		
Data standardization (dimension-wise)	No (variances should remain equal)	No (data std. is near unit)	Yes (data std. is large)
Learning rate	0.0009747753554628831		
Regularization strength	0.007478648909986116		

Table 12: Default hyper-parameters for COSMO Massidda et al. (2024) throughout the experiments. The values are the best parameters yielding the lowest SHD over linear-Gaussian datasets with 30-node ER-4 graphs for linear data, and MLP datasets with 40-node ER-4 graphs for nonlinear data. They are found via the tuning process provided by COSMO's implementation. Unmentioned hyper-parameters are left unchanged.

Experiment	Linear data (Figures 1, 2 and 3, Tables 2, 3, 14, 15, 16, 17, and 18)	Nonlinear GP data (Table 4 and Table 19)	Real data (Table 5)
Batch size	64		
Max epochs	5000	2000	
Data standardization (dimension-wise)	No (variances should remain equal)	No (data std. is near unit)	Yes (data std. is large)
Learning rate	0.004424703475697184	0.0011606444486776536	
l_1 regularization strength	0.0007589315865487066	0.000999704401738756	
l_2 regularization strength	0.029171977709975934	0.0011406751425196925	
Priority regularization strength	0.0007604972601205271	0.0014549388704106592	
Temperature	0.0008407426566089702	0.0007651953117707655	
Shift	0.005860546462049756	0.009324030532123762	
MLP hidden units	0	10	

Table 13: Default hyper-parameters for NOTEARS Zheng et al. (2018; 2020) and DAGMA Bello et al. (2022) throughout the experiments. Unmentioned hyper-parameters are left unchanged.

Experiment	Linear data (Figures 1, 2 and 3, Tables 2, 3, 14, 15, 16, 17, and 18)	Nonlinear GP data (Table 4 and Table 19)	Real data (Table 5)
Data standardization (dimension-wise)	No (variances should remain equal)	No (data std. is near unit)	Yes (data std. is large)
SEM	Linear	MLP	
Loss type	l_2		

E Additional Results

E.1 Results on ER graphs

In Table 14, we provide detailed numerical results for Figure 3, i.e., effect of dimensionality and sample size on linear-Gaussian data and ER-8 graphs.

Table 14: Causal discovery performance as function of dimensionalities and sample size on ER-8 graphs.

Nodes	Samples	Method	SHD (↓)	FDR (↓)	TPR (↑)
20	100	DAGMA	40.6±7.9	0.09±0.03	0.78±0.05
		ALIAS	**24.8±10.2**	**0.04±0.02**	**0.86±0.05**
	500	DAGMA	36.8±8.4	0.08±0.03	0.8±0.05
		ALIAS	**3.0±2.4**	**0.01±0.01**	**0.98±0.01**
	1000	DAGMA	35.6±8.1	0.07±0.03	0.8±0.05
		ALIAS	**1.0±1.4**	**0.0±0.01**	**0.99±0.01**
	2000	DAGMA	31.8±8.0	0.07±0.02	0.82±0.06
		ALIAS	**0.2±0.4**	**0.0±0.0**	**1.0±0.0**
	5000	DAGMA	31.2±9.4	0.07±0.02	0.83±0.06
		ALIAS	**0.2±0.4**	**0.0±0.0**	**1.0±0.0**
30	100	DAGMA	87.0±23.6	0.19±0.06	0.78±0.05
		ALIAS	**51.8±17.7**	**0.1±0.04**	**0.86±0.05**
	500	DAGMA	67.0±21.2	0.13±0.05	0.81±0.05
		ALIAS	**3.8±1.3**	**0.01±0.0**	**0.99±0.0**
	1000	DAGMA	67.6±17.8	0.14±0.04	0.82±0.04
		ALIAS	**0.2±0.4**	**0.0±0.0**	**1.0±0.0**
	2000	DAGMA	65.8±17.6	0.13±0.04	0.82±0.04
		ALIAS	**0.0±0.0**	**0.0±0.0**	**1.0±0.0**
	5000	DAGMA	68.4±22.8	0.14±0.05	0.81±0.06
		ALIAS	**0.0±0.0**	**0.0±0.0**	**1.0±0.0**
50	100	DAGMA	242.0±34.8	0.34±0.05	0.76±0.03
		ALIAS	**94.2±22.6**	**0.14±0.04**	**0.9±0.02**
	500	DAGMA	213.2±46.5	0.31±0.07	0.78±0.03
		ALIAS	**13.6±12.4**	**0.02±0.02**	**0.99±0.01**
	1000	DAGMA	215.2±59.5	0.3±0.08	0.77±0.04
		ALIAS	**2.2±2.2**	**0.0±0.0**	**1.0±0.0**
	2000	DAGMA	213.2±57.8	0.31±0.07	0.78±0.05
		ALIAS	**1.8±1.9**	**0.0±0.0**	**1.0±0.0**
	5000	DAGMA	210.2±72.4	0.3±0.1	0.78±0.05
		ALIAS	**0.7±1.2**	**0.0±0.0**	**1.0±0.0**

E.2 Results on SF graphs

Table 15 investigates the effect of dimensionality and sample size on linear-Gaussian data and Scale-Free (SF) graphs with 8 parents per node.

Table 15: Causal discovery performance as function of dimensionalities and sample size on SF-8 graphs.

Nodes	Samples	Method	SHD (↓)	FDR (↓)	TPR (↑)
20	100	DAGMA	**14.6±7.9**	**0.07±0.06**	0.87±0.05
		ALIAS	16.0±6.3	0.09±0.04	**0.89±0.06**
	500	DAGMA	6.8±3.1	0.02±0.02	0.93±0.02
		ALIAS	**0.6±0.6**	**0.01±0.01**	**1.0±0.01**
	1000	DAGMA	7.0±3.9	0.02±0.02	0.93±0.04
		ALIAS	**0.2±0.4**	**0.0±0.01**	**1.0±0.01**
	2000	DAGMA	9.0±6.8	0.04±0.04	0.91±0.06
		ALIAS	**0.2±0.4**	**0.0±0.01**	**1.0±0.01**
	5000	DAGMA	6.6±5.6	0.03±0.04	0.94±0.05
		ALIAS	**0.2±0.4**	**0.0±0.01**	**1.0±0.01**
30	100	DAGMA	47.2±15.9	0.15±0.08	0.84±0.03
		ALIAS	**30.2±9.9**	**0.1±0.03**	**0.92±0.03**
	500	DAGMA	45.8±17.7	0.16±0.07	0.85±0.05
		ALIAS	**0.2±0.4**	**0.0±0.0**	**1.0±0.0**
	1000	DAGMA	43.6±22.3	0.14±0.09	0.86±0.06
		ALIAS	**0.2±0.4**	**0.0±0.0**	**1.0±0.0**
	2000	DAGMA	36.6±21.6	0.12±0.09	0.89±0.05
		ALIAS	**0.0±0.0**	**0.0±0.0**	**1.0±0.0**
	5000	DAGMA	27.4±10.5	0.09±0.06	0.9±0.02
		ALIAS	**0.0±0.0**	**0.0±0.0**	**1.0±0.0**
50	100	DAGMA	101.2±40.9	0.18±0.09	0.84±0.05
		ALIAS	**54.2±17.6**	**0.11±0.03**	**0.92±0.02**
	500	DAGMA	68.4±23.7	0.13±0.05	0.89±0.03
		ALIAS	**9.8±11.3**	**0.03±0.03**	**0.99±0.01**
	1000	DAGMA	71.8±28.2	0.13±0.06	0.89±0.03
		ALIAS	**5.2±7.4**	**0.01±0.02**	**1.0±0.0**
	2000	DAGMA	65.2±22.0	0.12±0.04	0.9±0.04
		ALIAS	**5.4±9.9**	**0.02±0.03**	**1.0±0.0**
	5000	DAGMA	75.0±56.1	0.14±0.11	0.89±0.08
		ALIAS	**8.8±11.6**	**0.02±0.03**	**0.99±0.01**

E.3 Ablation Studies

E.3.1 Effect of RL method and learning rate

In Table 16, we conduct ablation studies on the effect of the choices of RL method and learning rate on the performance of our **ALIAS** method with the BIC score.

Table 16: Performance sensitivity of **ALIAS** with BIC score subjected to the variations of learning rate. We employ linear-Gaussian datasets with 30 nodes on ER-8 graphs and use CORL Wang et al., 2021 and DAGMA Bello et al., 2022 as reference. For each row, we compute the means and standard deviations over 5 independent datasets. **Bold**: better performance than the baselines. Unless otherwise indicated, the remaining hyper-parameters are used according to Table 7 in the Appendix.

Method	RL method	Learning rate	SHD (↓)	FDR (↓)	TPR (↑)
CORL	-	-	82.4 ± 49.9	0.23 ± 0.11	0.87 ± 0.10
DAGMA	-	-	67.6 ± 17.8	0.14 ± 0.04	0.82 ± 0.04
ALIAS (Ours)	A2C	0.00001	79.4 ± 15.0	0.22 ± 0.04	0.86 ± 0.03
		0.00005	**12.6** ± **7.0**	**0.04** ± **0.02**	**0.98** ± **0.01**
		0.0001	**1.2** ± **0.8**	**0.00** ± **0.00**	**1.00** ± **0.00**
		0.0005	**2.2** ± **3.0**	**0.01** ± **0.01**	**1.00** ± **0.00**
		0.001	**22.8** ± **8.8**	**0.07** ± **0.03**	**0.96** ± **0.02**
		0.005	123.8 ± 111.1	0.34 ± 0.35	0.54 ± 0.48
		0.01	223.4 ± 12.5	0.48 ± 0.07	0.13 ± 0.09
		0.05	232.6 ± 6.7	0.59 ± 0.07	0.11 ± 0.02
	PPO	0.00001	77.8 ± 16.7	0.21 ± 0.05	0.86 ± 0.03
		0.00005	**9.0** ± **6.5**	**0.03** ± **0.02**	**0.99** ± **0.01**
		0.0001	**1.2** ± **0.8**	**0.00** ± **0.00**	**1.00** ± **0.00**
		0.0005	**0.4** ± **0.6**	**0.00** ± **0.00**	**1.00** ± **0.00**
		0.001	**4.0** ± **4.5**	**0.01** ± **0.02**	**0.99** ± **0.01**
		0.005	204.8 ± 86.7	0.65 ± 0.29	0.21 ± 0.40
		0.01	225.4 ± 15.2	0.53 ± 0.09	0.11 ± 0.10
		0.05	231.4 ± 11.3	0.54 ± 0.04	0.13 ± 0.02

E.3.2 Effect of Entropy regularization

In Table 17, we conduct ablation studies on the effect of the choices of entropy regularization on the performance of our **ALIAS** method with the BIC score.

Table 17: Performance sensitivity of **ALIAS** with BIC score subjected to the variations of Entropy regularization weight. We employ linear-Gaussian datasets with 30 nodes on ER-8 graphs and use CORL Wang et al., 2021 and DAGMA Bello et al., 2022 as reference. Our scoring function is BIC. For each row, we compute the means and standard deviations over 5 independent datasets. **Bold**: better performance than the baselines. Unless otherwise indicated, the remaining hyper-parameters are used according to Table 7.

Method	RL method	Learning rate	Entropy Coef.	SHD (↓)	FDR (↓)	TPR (↑)
CORL		-	-	82.4 ± 49.9	0.23 ± 0.11	0.87 ± 0.10
DAGMA		-	-	67.6 ± 17.8	0.14 ± 0.04	0.82 ± 0.04
ALIAS (Ours)	A2C	0.0001	0	$\mathbf{1.20 \pm 0.8}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$
			0.001	$\mathbf{1.2 \pm 0.8}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$
			0.01	$\mathbf{1.6 \pm 1.1}$	$\mathbf{0.01 \pm 0.01}$	$\mathbf{1.00 \pm 0.00}$
			0.1	$\mathbf{28.8 \pm 17.2}$	$\mathbf{0.09 \pm 0.06}$	$\mathbf{0.97 \pm 0.02}$
			1	127.0 ± 29.2	0.31 ± 0.05	0.78 ± 0.06
		0.0005	0	$\mathbf{2.2 \pm 3.0}$	$\mathbf{0.01 \pm 0.01}$	$\mathbf{1.00 \pm 0.00}$
			0.001	$\mathbf{3.4 \pm 4.3}$	$\mathbf{0.01 \pm 0.01}$	$\mathbf{0.99 \pm 0.01}$
			0.01	$\mathbf{0.2 \pm 0.4}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$
			0.1	$\mathbf{48.2 \pm 26.2}$	0.14 ± 0.08	$\mathbf{0.93 \pm 0.03}$
			1	189.2 ± 18.9	0.40 ± 0.03	0.51 ± 0.04
	PPO	0.0001	0	$\mathbf{1.2 \pm 0.8}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$
			0.001	$\mathbf{1.0 \pm 0.7}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$
			0.01	$\mathbf{1.6 \pm 1.1}$	$\mathbf{0.01 \pm 0.01}$	$\mathbf{1.00 \pm 0.00}$
			0.1	$\mathbf{28.6 \pm 15.2}$	$\mathbf{0.09 \pm 0.05}$	$\mathbf{0.97 \pm 0.01}$
			1	130.2 ± 27.0	0.32 ± 0.04	0.77 ± 0.06
		0.0005	0	$\mathbf{0.4 \pm 0.6}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$
			0.001	$\mathbf{0.2 \pm 0.4}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$
			0.01	$\mathbf{1.6 \pm 3.0}$	$\mathbf{0.00 \pm 0.01}$	$\mathbf{1.00 \pm 0.01}$
			0.1	$\mathbf{51.8 \pm 26.3}$	0.15 ± 0.08	$\mathbf{0.92 \pm 0.03}$
			1	178.4 ± 21.2	0.38 ± 0.04	0.58 ± 0.05

E.3.3 Effect of sparsity regularization

In Table 18, we conduct ablation studies on the effect of the choices of sparsity regularization strength of our **ALIAS** method with the LS score.

Table 18: Performance sensitivity of **ALIAS** with LS score subjected to the variations of Sparsity regularization weight for the LS score. We employ linear-Gaussian datasets with 30 nodes on ER-8 graphs and use CORL Wang et al., 2021 and DAGMA Bello et al., 2022 as reference. For each row, we compute the means and standard deviations over 5 independent datasets. **Bold**: better performance than the baselines. Unless otherwise indicated, the remaining hyper-parameters are used according to Table 7.

Method	RL method	Sparsity regularizer λ_0	SHD (↓)	FDR (↓)	TPR (↑)
CORL	-	-	82.4 ± 49.9	0.23 ± 0.11	0.87 ± 0.10
DAGMA	-	-	67.6 ± 17.8	0.14 ± 0.04	0.82 ± 0.04
ALIAS (Ours)	A2C	0	$\mathbf{8.4 \pm 9.5}$	$\mathbf{0.02 \pm 0.03}$	$\mathbf{0.98 \pm 0.02}$
		0.000001	$\mathbf{2.8 \pm 2.2}$	$\mathbf{0.01 \pm 0.01}$	$\mathbf{0.99 \pm 0.00}$
		0.0001	$\mathbf{50.0 \pm 62.0}$	$\mathbf{0.07 \pm 0.08}$	0.83 ± 0.22
		0.01	189.0 ± 31.8	0.34 ± 0.10	0.28 ± 0.16
		1	226.4 ± 10.4	0.50 ± 0.06	0.08 ± 0.04
	PPO	0	$\mathbf{2.0 \pm 1.2}$	$\mathbf{0.01 \pm 0.01}$	$\mathbf{1.00 \pm 0.00}$
		0.000001	$\mathbf{1.2 \pm 1.3}$	$\mathbf{0.00 \pm 0.01}$	$\mathbf{1.00 \pm 0.00}$
		0.0001	$\mathbf{42.0 \pm 59.1}$	$\mathbf{0.04 \pm 0.05}$	0.84 ± 0.23
		0.01	178.8 ± 43.4	0.29 ± 0.11	0.31 ± 0.19
		1	221.6 ± 14.0	0.42 ± 0.09	0.09 ± 0.04

E.4 Model Misspecification Results

In Table 19, we study causal model misspecification on nonlinear data. We assume data is generated via additive noise models with Gaussian process similarly to (Zhu et al., 2020; Wang et al., 2021; Yang et al., 2023a), but test with datasets generated by the Post-nonlinear Gaussian Process model $X_i := \sigma\left(f_i\left(\mathbf{X}_{\mathrm{pa}_i}\right) + \mathrm{Laplace}\,(0,1)\right)$, which is identifiable (Zhang & Hyvärinen, 2009) and produced by Lachapelle et al. (2020).

Table 19: Causal discovery performance on nonlinear data with PNL-GP model. The data is generated with 10-node ER-4 graphs and post nonlinear Gaussian processes as causal mechanisms. The performance metrics are Structural Hamming Distance (SHD), False Detection Rate (FDR), and True Positive Rate (TPR). Lower SHD and FDR values are preferable, while higher values are better for TPR. The numbers are *mean ± standard deviation* over 5 independent runs. Since the graphs are dense, we also study the effect of pruning the output graphs.

	No Pruning			CAM Pruning		
Method	SHD (↓)	FDR (↓)	TPR (↑)	SHD (↓)	FDR (↓)	TPR (↑)
NOTEARS (Zheng et al., 2020)	29.4 ± 2.4	0.47 ± 0.05	0.34 ± 0.07	29.4 ± 2.4	0.46 ± 0.04	0.34 ± 0.07
DAGMA (Bello et al., 2022)	27.6 ± 6.6	0.45 ± 0.14	0.38 ± 0.14	27.4 ± 7.0	0.51 ± 0.17	0.38 ± 0.14
NOCURL (Yu et al., 2021)	34.8 ± 2.5	0.51 ± 0.17	0.15 ± 0.06	34.8 ± 2.5	0.45 ± 0.15	0.15 ± 0.06
COSMO (Massidda et al., 2024)	29.4 ± 1.8	0.41 ± 0.09	0.33 ± 0.04	29.4 ± 1.8	0.41 ± 0.09	0.33 ± 0.04
CORL (Wang et al., 2021)	$\mathbf{6.6 \pm 2.0}$	0.15 ± 0.04	$\mathbf{0.95 \pm 0.03}$	$\mathbf{4.2 \pm 1.5}$	0.08 ± 0.04	$\mathbf{0.94 \pm 0.04}$
RCL-OG (Yang et al., 2023a)	8.6 ± 3.1	0.19 ± 0.07	$\underline{0.9 \pm 0.05}$	7.2 ± 3.0	0.13 ± 0.06	$\underline{0.89 \pm 0.04}$
ALIAS (Ours)	$\underline{6.8 \pm 3.0}$	$\mathbf{0.04 \pm 0.06}$	0.85 ± 0.06	$\underline{6.8 \pm 3.0}$	$\mathbf{0.03 \pm 0.06}$	0.84 ± 0.06

In addition, we also study the robustness of our method under noisy data. Specifically, we corrupt the data by adding Gaussian noises ($\sigma^2 = 0.1$) to $p\%$ random entries of the design matrix $\mathcal{D} \in \mathbb{R}^{n \times d}$, and report the results for the challenging 30-node ER-8 graphs in Table 20, showing that our method still consistently surpasses the baselines, even with increasing noise levels.

Table 20: Causal Discovery Performance under Noisy Data.

Method \ p	1%	3%	5%	10%
NOTEARS	181.4 ± 9.0	189.0 ± 5.6	190.8 ± 6.2	194.4 ± 8.7
DAGMA	67.0 ± 19.1	75.0 ± 21.7	79.0 ± 14.6	94.0 ± 13.7
NOCURL	142.2 ± 9.3	147.0 ± 10.7	146.0 ± 11.4	153.0 ± 11.8
COSMO	96.6 ± 13.7	112.8 ± 17.1	111.0 ± 19.0	136.2 ± 31.9
ALIAS (Ours)	$\mathbf{8.4 \pm 1.7}$	$\mathbf{27.2 \pm 6.8}$	$\mathbf{41.4 \pm 11.1}$	$\mathbf{66.0 \pm 8.9}$

Moreover, we investigate our method's performance under the dependence of noises, which is equivalent to the existence of hidden confounders. Specifically, we create datasets with hidden variables by generating datasets with k additional variables then remove them. We present the results (for 30-node ER-8 graphs) on such datasets in Table 21, where our method also outperforms all baselines, even with more hidden confounders.

Table 21: Causal Discovery Performance under Hidden Confounders.

Method \ k	1	2	3	4
COSMO	93.0 ± 36.2	133.6 ± 25.0	132.8 ± 15.9	136.4 ± 19.1
DAGMA	53.0 ± 26.6	77.6 ± 20.7	96.0 ± 17.0	124.0 ± 11.8
NOCURL	149.2 ± 17.2	159.8 ± 18.8	153.2 ± 12.0	158.4 ± 11.9
NOTEARS	185.2 ± 23.2	172.6 ± 12.7	173.2 ± 11.1	188.8 ± 9.9
ALIAS (Ours)	29.4 ± 13.8	58.2 ± 10.9	82.2 ± 13.6	106.6 ± 5.5