Weighted Sampling for Online Causal Discovery

Arnab Bhattacharyya
University of Warwick
arnab.bhattacharyya@warwick.ac.uk

Philips George John
Centre for Quantum Technologies
NUS
philips.george.john@u.nus.edu

Sayantan Sen
Centre for Quantum Technologies
NUS
sayantan789@gmail.com

Naganand Yadati
IIIT Bangalore
y.naganand@gmail.com

Abstract

Discovering the underlying causal structure of a system is a fundamental challenge in machine learning, requiring interventional data to distinguish between observationally equivalent models. In this paper, we study the problem of learning causal Bayesian networks in an online setting, where a learner sequentially observes individual samples from a stream of unknown interventional and observational distributions. We formulate this task as an online sequence prediction problem. To overcome the super-exponential size of the DAG search space, we extend dynamic programming algorithms originally developed for uniform DAG counting and sampling within Markov Equivalence Classes (MECs) to support score-decomposable, weighted sampling. We prove that the posterior distribution maintained by our algorithm competes with the optimal causal structure in hindsight, outputting a distribution that is close to the true data-generating mechanism as measured by the Interventional Kullback-Leibler divergence.

1 Introduction

Learning causal relationships from experimental data is a foundational challenge with a myriad of applications across biology, economics, and sociology [King et al., 2004, de Campos et al., 2019, Runge et al., 2019, Hoover, 1990]. In the modern era of data-driven science, massive amounts of observational data are readily available. However, it is a well-established theoretical limitation that observational data alone can only identify a causal graph up to its Markov Equivalence Class (MEC); that is, multiple distinct causal graphs can encode the exact same observational distribution [Verma and Pearl, 1990, Andersson et al., 1997]. To uniquely identify the true causal structure, the data-generating system must be subjected to targeted perturbations, yielding *interventional* data. A large body of literature over the last two decades has focused on combining observational and interventional data to accurately recover the underlying graph [He and Geng, 2008, Hauser and Bühlmann, 2014, Shanmugam et al., 2015, Choo et al., 2022, Squires et al., 2020, Choo and Shiragur, 2023].

Within this space, intervention design strategies are broadly categorized as either non-adaptive or adaptive. In the non-adaptive setting, the sequence of intervention targets is fixed a priori [Shanmugam et al., 2015], whereas adaptive algorithms dynamically select subsequent targets based on the outcomes of prior experiments [He and Geng, 2008, Hauser and Bühlmann, 2014, Squires et al., 2020, Choo and Shiragur, 2023].

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Since performing physical or biological interventions is typically expensive, a standard assumption is that observational data is abundant, and the primary objective is to minimize the number of samples from interventional experiments. To achieve this, many prior works rely on Bayesian inference and impose structural restrictions on the underlying causal model [Heckerman et al., 2006, Annadani et al., 2023, Toth et al., 2022, Greenewald et al., 2019]. However, the number of DAGs within an MEC can grow exponentially with respect to the unoriented edges, and thus, exact posterior updates quickly become intractable [He et al., 2015]. Consequently, researchers often restrict the hypothesis space; e.g., by constraining the graph to a forest [Greenewald et al., 2019] or relying on classical assumptions like *faithfulness* and *causal sufficiency* [Spirtes et al., 2000]. Very recently, progress has been made in exact Bayesian learning under limited interventional budgets [Zhou et al., 2024], as well as in handling mixed noise assumptions within structural equation models (SEMs) [Liu et al., 2024].

The Online Causal Discovery Framework. In this paper, we break away from the traditional static data paradigm and formalize the problem of *online causal discovery* [Cesa-Bianchi and Lugosi, 2006]. Here, interventional samples from an unknown system arrive sequentially in a stream. At each time step, our goal is to construct a predictive estimate of the unknown causal graph. Crucially, in our setup, the sequence of interventions can be completely arbitrary and is not assumed to be chosen by the causal discovery algorithm itself.

This mirrors real-world scientific discovery, where experimental data often arrives in an unstructured stream, perhaps aggregated from historical results, independent laboratories, or existing clinical trials. A compelling example is the reprogramming of T cells for cancer immunotherapy [Miller et al., 2019, Feucht et al., 2019]. Using techniques like single-cell Perturb-Seq [Dixit et al., 2016], researchers sequentially perform CRISPR-mediated single-gene knockouts to observe shifts in T cell states. As data from these varied, sequentially conducted experiments streams in, the goal is to continuously fine-tune a causal model capable of predicting cellular responses to entirely new interventions. Our online setup is perfectly geared to adapt to such arbitrary experimental sequences.

1.1 Main Contributions

We introduce a novel Randomized Weighted Majority (RWM) algorithm for sequential causal prediction and provide rigorous theoretical guarantees for its performance. Our main results can be informally stated as follows:

1. **Efficient Online Learning:** The online algorithm achieves sublinear regret against the best fixed Directed Acyclic Graph (DAG) in hindsight. Furthermore, it samples DAGs in polynomial time with respect to the number of nodes n.
2. **Batch-Optimal Generalization:** Using a stochastic online-to-batch conversion, the algorithm guarantees that the expected Kullback-Leibler (KL) divergence of the interventional distribution converges to the optimal fixed model over the training distribution.
3. **Quantitative Out-of-Distribution (OOD) Transfer:** We provide a quantitative bound for generalization to new interventions. If the training support is rich enough in a quantitative sense, the finite-time regret explicitly bounds the prediction error on unseen interventions, mediated by a covariate shift constant.

1.2 Contrast with Bayesian Approaches

While our sequential update mechanism shares conceptual similarities with Bayesian Causal Discovery (BCD), our theoretical guarantees and setting significantly depart from it. The primary bottleneck of exact BCD is its super-exponential computational complexity, scaling in $O(3^n)$ time [Tian and Pearl, 2001, Koivisto and Sood, 2004]. To bypass this, modern BCD often relies on approximate inference, such as Markov Chain Monte Carlo (MCMC) over graphs and orderings [Friedman and Koller, 2003, Kuipers and Moffa, 2017, Agrawal et al., 2018] or continuous non-convex relaxations [Zheng et al., 2018, Lorch et al., 2021, Cundy et al., 2021]. Because MCMC mixing times on DAG spaces are notoriously difficult to bound and gradient-based relaxations are prone to local optima, these methods inherently sacrifice finite-time theoretical guarantees in exchange for asymptotic posterior consistency.

We note that the recent work by Zhou et al. [2024] has successfully bypassed this super-exponential bottleneck for exact Bayesian inference by leveraging the same polynomial-time counting and sampling algorithms for Markov Equivalence Classes that we utilize. However, their framework is rooted in the static, batch-data paradigm, focusing on minimizing the sample complexity required for exact structure recovery under stochastic interventions. In contrast, our framework operates in a different theoretical regime: *adversarial online learning*. By maintaining dynamic weights rather than a traditional Bayesian posterior, our algorithm achieves non-asymptotic sublinear regret even when the sequence of interventions is sampled from an arbitrary, unknown distribution.

2 Technical Overview

2.1 Online Causal Discovery

The *causal discovery* problem that we consider in this work is as follows. Given *infinite samples* from the observational distribution $P^*(X_1, \ldots, X_n)$ on $[k]^n$, as well as a separate finite sequence of samples from various *interventional distributions* $P^*_{\mathrm{do}(I)} \equiv P^*(X|\mathrm{do}(X_I) = x_I)$ for various interventions (I, x_I), our high-level goal is to output an explicit DAG D on $[n]$ such that, for any intervention I', we can output a distribution $\widehat{P}_{D,I'}$ which is "close to" the true interventional distribution $P^*_{\mathrm{do}(I')}$ (in expectation or with high probability), provided the set of interventions that we saw while learning is rich enough. In this work, we only consider *ideal* interventions where an intervention fixes the value and also removes all the parent-edges in the true causal graph for each intervened variable, without affecting any other causal mechanism (modularity).

Our algorithm for causal discovery works in two phases, namely:

(i) Observational Phase. In the observational phase, given unlimited data, we can exactly identify the Markov Equivalence Class (MEC) of the true causal graph, represented uniquely as a *Completed Partially Directed Acyclic Graph (CPDAG)* C on vertex set $[n]$. Furthermore, we can compute the observed conditional probabilities $\hat{P}(x_i \mid x_R)$ that are arbitrarily close to the true conditional probabilities $P^*(x_i \mid x_R)$ for any node $i \in [n]$ and any subset $R \subseteq \mathsf{Nbr}_C(i)$ of its neighbors. Thus, our search space for the output DAG D is restricted to the *valid Acyclic Moral Orientations (AMOs)* of C, denoted as $\mathsf{EXT}(C)$, representing the true causal structures consistent with the observational data (see Appendix A.1 for the formal definitions). In order for our algorithm to be "efficient", we further restrict ourselves to AMOs with indegree $\leqslant d$ for a parameter $d \in \mathbb{N}$, which we denote by $\mathsf{EXT}_d(C)$, and consider the bounded-parameter regime (constant d) when we want efficient algorithms. Also, even if C is not the true CPDAG (which may be the case when we learn it from finite observational data), the theoretical guarantees of our algorithms for the interventional phase will hold with respect to the class of extensions of the input CPDAG C.

(ii) Online Interventional Phase. During the online phase, the learner faces a sequence of time-varying environments. At each time step t, the environment specifies a *hard intervention* which consists of a tuple $I_t \subseteq [n]$ of variables and an assignment $v_{I_t} \in [k]^{I_t}$ of values to the variables I_t. The learner then receives a sample $x^{(t)} \in [k]^n$ drawn from the true interventional distribution $P^*_{\mathrm{do}(I_t)} \equiv P^*(X \mid \mathrm{do}(X_{I_t} = v_{I_t}))$, which ensures that $x^{(t)}_i = v_{I_t}[i]$ for each $i \in I_t$.

For any joint distribution $\widehat{P}$ on $[k]^n$, a DAG D on vertices $[n]$ and intervention target $I \subseteq [n]$, let us define $\widehat{P}_{D,I}$ to be the distribution on $[k]^n$ given by:

$$\widehat{P}_{D,I}(x) = \prod_{i \notin I} \widehat{P}\left(x_i | x_{\mathsf{Pa}_D(i)}\right), \tag{1}$$

To learn a DAG D that performs well across all environments, we apply the *Randomized Weighted Majority (RWM)* algorithm. The RWM algorithm maintains a distribution over the candidate causal DAGs $D \in \mathsf{EXT}_d(C)$. For a hard intervention I_t, the interventional log-loss for any candidate DAG D relies only on the non-intervened nodes, evaluated using our pre-computed observational conditionals:

$$\ell_t(D) \triangleq -\log \hat{P}_{D,I_t}(x^{(t)}) \tag{2}$$

which, when $\widehat{P} = P^*$ and P^* is Markovian with respect to D, is the log loss of $P^*_{\mathrm{do}(I_t)}$ for the sample $x^{(t)}$. The weight of a DAG D at the beginning of round t is defined by its cumulative exponentiated log-loss from all previous rounds:

$$W_t(D) \triangleq \exp\left(-\eta \sum_{k=1}^{t-1} \ell_k(D)\right) = \prod_{i=1}^{n} w_{i,t}(i, \mathsf{Pa}_D(i)) \tag{3}$$

where

$$w_{i,t}(i, \mathsf{Pa}_D(i)) \triangleq \exp\left(\eta \sum_{k<t} \mathbb{I}[i \notin I_k] \log \widehat{P}\left(x_i^{(k)} \,\middle|\, x^{(k)}_{\mathsf{Pa}_D(i)}\right)\right) = \prod_{k<t \,:\, i \notin I_k} \hat{P}\left(x_i^{(k)} \mid x^{(k)}_{\mathsf{Pa}_D(i)}\right)^{\eta}$$

encapsulates the cumulative local score up to time $t-1$ for node i for some hyperparameter $\eta > 0$. This factorization ensures the global weight function remains score-decomposable at every step. The pseudocode of our RWM sampler for causal discovery is as follows:

Algorithm 1: RWM Sampler for Causal Discovery

Input: CPDAG C, maximum indegree d, learning rate η, time horizon T.

1 **Initialize:** Local weights $w_{i,1}(i, R) \leftarrow 1$ for all $i \in V$, $R \subseteq \mathsf{Nbr}_C(i)$ with $|R| \leqslant d$.
2 **for** $t = 1, \ldots, T$ **do**
3 | **Sample** a DAG $D_t \sim \nu_t(D) \propto W_t(D)$ *// via Algorithm 3*
4 | **Observe** intervention target $I_t \subseteq [n]$ and sample $x^{(t)} \sim P^*_{\mathrm{do}(I_t)}$
5 | **for** *all* $i \notin I_t$ *and all candidate parent sets* $R \subseteq \mathsf{Nbr}_C(i)$ *with* $|R| \leqslant d$ **do**
6 | | $w_{i,t+1}(i, R) \leftarrow w_{i,t}(i, R) \cdot \hat{P}\left(x_i^{(t)} \mid x_R^{(t)}\right)^{\eta}$
7 | **for** *all* $i \in I_t$ *and all candidate parent sets* $R \subseteq \mathsf{Nbr}_C(i)$ *with* $|R| \leqslant d$ **do**
8 | | $w_{i,t+1}(i, R) \leftarrow w_{i,t}(i, R)$

2.2 Bypassing the Computational Bottleneck.

The sampler (Algorithm 1) relies on sampling exactly from the distribution weighted by $W_t(D)$ (Step 3 in Algorithm 1). However, the number of valid extensions $|\mathsf{EXT}_d(C)|$ can be super-exponential in the number of nodes n even for constant indegree d. Explicitly maintaining weights for every DAG and sampling from this space is computationally intractable.

Our core algorithmic contribution is demonstrating that since the interventional weight $W_t(D)$ factorizes over the nodes of the graph, we can reduce the RWM sampling step to an *exact weighted sampling* problem over the undirected chordal chain components of the CPDAG C. For this, we rely on the techniques developed in Wienöbst et al. [2023] to first compute a partition function efficiently and then to use the partition function computation to efficiently sample the DAG D_t in Line 3.

Thus, we show that Algorithm 1 can be implemented in polynomial (in n, k, $1/\varepsilon$) time (for bounded in-degree d), and it moreover satisfies the following online guarantee:

Theorem 2.1. *Suppose $\mathcal{I}$ is a finite set of intervention targets which are seen in the online interventional phase, and $\mu(I) \triangleq \frac{1}{T}\sum_{t=1}^{T} \mathbb{I}[I_t = I]$ denotes the relative frequency of intervention I among $I_1, \ldots, I_T$. Then, for any class of DAGs $\mathcal{C}$, if $T \geqslant \widetilde{\Omega}\left(\frac{n^3 d}{\varepsilon^2}\right)$:*

$$\mathop{\mathbb{E}}_{x^{(1:T)}} \frac{1}{T}\sum_{t=1}^{T} d_{\mathrm{KL}}\left(P^*_{\mathrm{do}(I_t)} \,\middle\|\, \widehat{P}_{D_t, I_t}\right) \leqslant \min_{D \in \mathcal{C}} \sum_{I \in \mathcal{I}} \mu(I)\, d_{\mathrm{KL}}\left(P^*_{\mathrm{do}(I)} \,\middle\|\, \widehat{P}_{D,I}\right) + \varepsilon. \tag{4}$$

In the above, observe that the interventions $I_1, \ldots, I_T$ can be completely arbitrary (indeed, adversarial) and that D_t is predicted *before* the intervention I_t is revealed. Thus, if $\mathcal{C}$ contains the ground-truth DAG, the *time-averaged KL-loss* of the predicted interventional distributions is at most ε.

To prove the above theorem, we first bound the expected time-average of *step-wise KL losses*: $\frac{1}{T}\sum_{t=1}^{T} d_{\mathrm{KL}}\left(P^*_{\mathrm{do}(I_t)} \,\middle\|\, \widehat{P}_{D_t, I_t}\right) - \min_{D \in \mathcal{C}} \frac{1}{T}\sum_{t=1}^{T} d_{\mathrm{KL}}\left(P^*_{\mathrm{do}(I_t)} \,\middle\|\, \widehat{P}_{D, I_t}\right)$ in terms of the *regret* of the online prediction algorithm used to choose the DAGs $D_1, \ldots, D_T$. Since Algorithm 1 uses the randomized weighted majority (RWM) algorithm with the logarithmic loss for the predictions $D_1, \ldots, D_T$, we can combine our bound along with a standard RWM regret bound (see Lemma A.22) and the appropriate construction of $\widehat{P}$ from P^* (see Lemma B.6) to prove Theorem 2.1.

2.3 Online-to-Batch Conversion for Causal Discovery with Seen Interventions

In this section, we prove Theorem 2.2 which assumes that the sequence of intervention targets seen by the online algorithm is *stochastic* rather than adversarial, and sampled independently from an unknown and arbitrary distribution μ. This allows us to give a sample complexity guarantee for outputting a *single prediction* D_τ such that $\mathbb{E}_{I\sim\mu}\left[d_{\mathrm{KL}}\left(P^*_{\mathrm{do}(I)}\,\middle\|\,\widehat{P}_{D_\tau,I}\right)\right]\leqslant\varepsilon$ in expectation (over the randomness of the samples and the algorithm). Here D_τ is chosen by sampling τ uniformly at random from $\{1,\ldots,T\}$.

Theorem 2.2 (RWM Online-to-batch Learning w.r.t Seen Interventions)**.** *Suppose Algorithm 1 is run for T rounds, and suppose that at each time step $t\in[T]$, the intervention target I_t is sampled independently from a fixed but unknown distribution μ on a finite set of interventions $\mathcal{I}$, and subsequently the sample $x^{(t)}$ is sampled from $P^*_{\mathrm{do}(I_t)}$. Let D_τ denote the DAG output by sampling τ uniformly at random from $[T]$ and choosing the round τ prediction of Algorithm 1.*

Then, if $T\geqslant\widetilde{\Omega}\left(\frac{n^3d}{\varepsilon^2}\right)$, then we have

$$\mathbb{E}\left[\sum_{I\in\mathcal{I}}\mu(I)\,d_{\mathrm{KL}}\left(P^*_{\mathrm{do}(I)}\,\middle\|\,\widehat{P}_{D_\tau,I}\right)\right]\leqslant\min_{D\in\mathsf{EXT}_d(C)}\sum_{I\in\mathcal{I}}\mu(I)\,d_{\mathrm{KL}}\left(P^*_{\mathrm{do}(I)}\,\middle\|\,\widehat{P}_{D,I}\right)+\varepsilon,\qquad(5)$$

where the outer expectations are taken over the random draws of the interventions $I_{1:T}$, the samples $x^{(1:T)}$, and the algorithm's internal randomization for the predictions $D_{1:T}$ as well as the choice τ.

The proof of this result is exactly analogous to the proof of Theorem 2.1 in the previous section, except that we use the assumption that the intervention sequence is independently sampled from a distribution μ and interchange the expectations in the proof so that we can bound $\mathbb{E}\left[\sum_{I\in\mathcal{I}}\mu(I)\,d_{\mathrm{KL}}\left(P^*_{\mathrm{do}(I)}\,\middle\|\,\widehat{P}_{D_\tau,I}\right)\right]-\min_{D\in\mathsf{EXT}_d(C)}\sum_{I\in\mathcal{I}}\mu(I)\,d_{\mathrm{KL}}\left(P^*_{\mathrm{do}(I)}\,\middle\|\,\widehat{P}_{D,I}\right)$ in terms of the regret.

2.4 Generalizing to Unseen Interventions

We establish the following result showing that if the distribution μ from Theorem 2.2 puts sufficient mass on a verifying set, then the Bayes net on the predicted DAG D_τ (as in Theorem 2.2) has bounded prediction error on any intervention I' that is not "too different" from the interventions in the training set.

In the theorem statement below, a verifying set $\mathcal{V}_{\mathrm{struct}}$ for a true causal DAG G^* is a set of interventions that *separates* each undirected edge in the CPDAG of $[G^*]$; in the sense that $\mathcal{V}_{\mathrm{struct}}$ has an intervention target I containing exactly one of the endpoints of the edge. Hyttinen et al. [2013], Shanmugam et al. [2015] showed that interventions on such a set is necessary and sufficient to learn G^* (if no other information about G^* is available).

We define the *minimum structural margin* $\Delta>0$ to be the smallest possible KL error $d_{\mathrm{KL}}\left(P^*_{\mathrm{do}(I)}\,\middle\|\,\widehat{P}_{D,I}\right)$ that any incorrect DAG $D\in\mathcal{C}\backslash\{G^*\}$ makes when considering all the interventions I in the verifying set.

For an unseen intervention I', we define the covariate shift constant $C_{\mathrm{shift}}(I')$ to be the *worst case (probability) density ratio* $\frac{P^*_{\mathrm{do}(I')}(\mathsf{Pa}^*(v)=pa)}{P^*(\mathsf{Pa}^*(v)=pa)}$, over all unintervened variables $v\notin I'$ and over all *supported* parent-assignments pa for each such variable v.

Theorem 2.3. *Assume the true causal graph $G^*\in\mathcal{C}$, and that the training distribution μ has support over both the empty set $\varnothing$ and a verifying set $\mathcal{V}_{\mathrm{struct}}$ for G^*. Let $\mu_{\min}\triangleq\min_{I\in\mathcal{V}_{\mathrm{struct}}\cup\{\varnothing\}}\mu(I)$. Run Algorithm 1 and output D_τ by sampling $\tau\sim[T]$ uniformly, in the same setup as Theorem 2.2. Then, for any unseen target intervention I' with a finite covariate shift constant $C_{\mathrm{shift}}(I')$, we have*

$$\mathbb{E}\left[d_{\mathrm{KL}}\left(P^*_{\mathrm{do}(I')}\,\middle\|\,\widehat{P}_{D_\tau,I'}\right)\right]\leqslant\varepsilon,\qquad(6)$$

provided the number of rounds $T\geqslant\mathsf{poly}\left(n,\log k,\frac{1}{\varepsilon},\frac{1}{\Delta},C_{\mathrm{shift}}(I'),\frac{1}{\mu_{\min}}\right)$.

The proof of Theorem 2.3 involves bounding (see Theorem C.3) the expected KL loss of D_τ by $\left(C_{\text{shift}}(I') + \frac{M}{\Delta}\right) \frac{B_\tau}{\mu_{\min}}$, where M is the largest possible KL divergence $d_{\text{KL}}\left(P^*_{\text{do}(I)} \middle\| \widehat{P}_{D,I}\right)$ for any $\widehat{P}_{D,I}$ and B_τ is the expected average regret bound for the the online prediction algorithm.

Since each $\widehat{P}_{D,I}$ will satisfy $-\log \widehat{P}_{D,I}(x) \leqslant n \log \frac{4nk^2}{\varepsilon}$ for all x due to the construction of $\widehat{P}$ (see Lemma B.6), we will get the theorem by using the RWM regret bounds for bounding B_τ and taking $M \triangleq n \log \frac{4nk^2}{\varepsilon}$.

At a high level, Theorem C.3 is proved as follows: We split the expectation into two sub-events, i.e. *structural failure* where we predict the wrong DAG ($D_\tau \neq G^*$) and the correct structure case where $D_\tau = G^*$. In the event of structural failure, we can lower-bound the expected μ-weighted KL errors across the verifying set by $\mu_{\min} \cdot \Delta \cdot \Pr(D_\tau \neq G^*)$, and upper bound the expectation by B_τ (since $G^* \in \mathcal{C}$ so the minimizer term is 0) to obtain $\Pr(D_\tau \neq G^*) \leqslant \frac{B_\tau}{\mu_{\min}\Delta}$, and hence bound the expected KL divergence in this case by $\frac{M\, B_\tau}{\mu_{\min}\Delta}$.

When $D_\tau = G^*$, all the parent structures in the D_τ match the true parent structures, and we can use the chain rule of KL divergence for Bayesian networks (tensorization) to express the expected KL $\mathbb{E}\left[d_{\text{KL}}\left(P^*_{\text{do}(I')} \middle\| \widehat{P}_{D_\tau,I'}\right)\right]$ as a sum of KLs betweem the node-conditonal distributions for each unintervened node, which we can the upper bound in terms of $C_{\text{shift}}(I')$ and the sum of these expected node-conditional KLs in the observational regime (w.r.t P^* rather than $P^*_{\text{do}(I')}$). In this case as well, we bound the observational KL error by the average regret to get a bound of $C_{\text{shift}}(I')\frac{B_\tau}{\mu(\varnothing)}$ ($I = \varnothing$ for observational regime) $\leqslant C_{\text{shift}}(I')\frac{B_\tau}{\mu_{\min}}$. Combining the two cases gives the theorem.

Remark 2.4 (Interventional KL Divergence (IKL))**.** Our generalization bound can be contextualized within the Interventional Kullback-Leibler (IKL) divergence framework [Wildberger et al., 2023]. The IKL divergence was introduced to quantify the structural and distributional distance between a true causal model and an estimate across a finite set of interventional environments. In our setup, the expected average regret B_τ serves as an upper bound on the expected IKL divergence over the training support $\text{supp}(\mu)$.

While Wildberger et al. [2023] establish that the Interventional Kullback-Leibler $D_{\text{IKL}} = 0$ is sufficient to guarantee exact model recovery under specific environmental conditions (analogous to our verifying set), our finite-sample analysis extends this principle to the out-of-distribution regime. Specifically, Theorem C.3 demonstrates how bounding the IKL divergence on a training set sequentially guarantees bounded standard KL-divergence on an unseen intervention I', parameterized by the margin of identifiability Δ and the strength of the interventional covariate shift $C_{\text{shift}}(I')$.

Organization of the paper In Section 3, we present our efficient algorithms for weighted counting and sampling. We conclude and discuss open questions in Section 5. Due to shortage of space, the formal proofs of the lemmas and theorems are presented in the appendix.

3 Computationally efficient algorithms for weighted counting and sampling

As mentioned before, the naive DAG sampling approach of $D_t \sim \nu_t(D) \propto W_t(D)$ in Algorithm 1 is computationally intractable. However, since the interventional weight $W_t(D)$ factorizes over the nodes of the graph, we can design novel computationally efficient *exact weighted counting and sampling* algorithms of AMOs in a CPDAG.

In order to describe the algorithms, we need to first set up some notation and definitions (formal definitions are given in Appendix A.1).

Graph Terminology and Orientations. We begin by establishing the basic graph-theoretic notation used in our framework. A *directed acyclic graph* (DAG) is a graph containing exclusively directed edges and no directed cycles. A *partially directed* (or mixed) graph $G = (V, E)$ generalizes this concept by containing both directed edges ($u \to v$) and undirected edges ($u - v$). When algorithmically orienting edges, we use the operator $G \oplus E^*$. This denotes the new partially directed graph obtained by taking G and directing a subset of its undirected edges according to the specific orientations provided in E^*, while leaving the rest of G intact.

Markov Equivalence and CPDAGs. In causal discovery, multiple DAGs can imply the exact same set of conditional independencies. Such DAGs are said to be *Markov equivalent*, a relation that partitions the space of all possible DAGs into distinct *Markov Equivalence Classes* (MECs). An MEC is uniquely visually represented by a *Completed Partially Directed Acyclic Graph* (CPDAG), which is formed by taking the union of all DAGs within that equivalence class. The undirected components of a CPDAG possess a highly specific structure: they are *undirected and connected chordal graphs* (UCCGs). An undirected graph is *chordal* if every cycle of length four or greater contains a chord, an edge connecting two non-adjacent vertices in the cycle.

Cliques and Separators. To efficiently process the chordal components of a CPDAG, we rely on two fundamental sub-structures. For any undirected graph G, we denote the set of all its *maximal cliques* by $\Pi(G)$. Furthermore, a subset of vertices S is defined as a *minimal separator* if it disconnects at least one pair of vertices in G, and no strict subset of S can achieve that same separation. We denote the set of all minimal separators of G by $\Delta(G)$.

Score-Decomposable Weights. To evaluate and sample DAGs efficiently, we assign a score-decomposable weight to each valid extension. Specifically, the total weight of a DAG D factorizes into the product of local weights:

$$W(D) = \prod_{i=1}^{n} \phi_i(i, \mathsf{Pa}_D(i)) \tag{7}$$

where $\phi_i(i, R)$ is an arbitrary, non-negative local weight for node i given its parent set R. To abstract away the specific scoring mechanics from the counting algorithm, we assume access to a *weight oracle* $\mathsf{WO}(v, R)$. Given a node v and a candidate parent subset $R \subseteq \mathsf{Nbr}_C(v)$, the oracle returns the local weight $\phi_v(v, R)$ in $O(1)$ time. Since we only consider DAGs D whose indegree is bounded, the number of possible parent set configurations is polynomial in n and k and thus this weight oracle can be efficiently implemented.

We first give an algorithm to efficiently evaluate the *partition function*: the sum of weights of all consistent DAG extensions of a partially directed graph C, subject to a maximum indegree bound d:

$$Z(C) \triangleq \sum_{D \in \mathsf{EXT}_d(C)} W(D) \tag{8}$$

Our exact weighted counting result is stated as follows.

Theorem 3.1 (Computational Complexity of $Z(G)$)**.** *Let C be a CPDAG with n vertices, m edges, and undirected chordal chain components having a maximum treewidth* tw*. Assuming an $O(1)$ oracle* WO *for evaluating local weights, and utilizing dynamic programming memoization, Algorithm 2 computes the total partition function $Z(C)$ in time $\tilde{O}\left(n^2 m + n^2 \cdot 2^{\mathrm{tw}}\right)$.*

At a high level, the WEIGHTED-AMO-COUNTING algorithm computes the partition function using a recursive, divide-and-conquer dynamic programming strategy over the structural components of chordal graphs. The algorithm exploits a fundamental property of chordal graphs: any valid acyclic maximal orientation (AMO) can be systematically generated by selecting a "root" set S, which must be either a maximal clique or a minimal separator, and orienting all incident edges strictly outward from S. This outward orientation acts as a separator, breaking the remaining unoriented edges into smaller, mutually independent UCCG components $G_1, \ldots, G_k$. By memoizing the results, the total partition function is recursively computed as the aggregate weight of all valid internal orderings of the root S, multiplied by the product of the partition functions of the independent child components.

The critical departure from prior exact counting algorithms, such as that of Wienöbst et al. [2023], lies in how the internal orderings of the root set S are evaluated. In the uniform (unweighted) setting, the number of valid topological orderings within a clique is purely combinatorial and can be calculated trivially (e.g., as $|S|!$). However, in our general score-decomposable setting, every distinct topological ordering of S assigns different parent sets to the nodes within S, yielding entirely different local weights. Brute-forcing all $|S|!$ permutations to sum their respective weights would introduce an expensive factorial computation time.

To bypass this bottleneck, our algorithm embeds a subset-level DP optimization inspired by the Held-Karp algorithm for the Traveling Salesperson Problem [Held and Karp, 1962]. Rather than

Algorithm 2: WEIGHTED-AMO-COUNTING(C, G, d)

Input: Ambient CPDAG C, UCCG G of C, maximum indegree d
Output: $Z(G \mid C)$

if $Z(G \mid C)$ *is in* MemoTable **then**
  return MemoTable$[G, C]$;
Compute $\Pi(G)$ and $\Delta(G)$;
$Z_{total} \leftarrow 0$;
if $|V(G)| \leqslant d$ **then**
  for *every valid AMO* O *of* G **do**
    $Z_{total} \leftarrow Z_{total} + W(G \oplus O \mid C)$; `// via Oracle WO`
  MemoTable$[G, C] \leftarrow Z_{total}$;
  return Z_{total};
for *every* $S \in \Delta(G) \cup \Pi(G)$ **do**
  Compute $C_G(S)$ using the MLS algorithm. Let $G_1, \ldots, G_k$ be its undirected components;
  Let $\Delta(S, G) \leftarrow \{S' \in \Delta(G) \mid S' \subset S\}$;
  `// Held-Karp Subset DP for` Z_{root}
  Initialize $Z_{HK}(\varnothing) \leftarrow 1$;
  for $i = 1$ **to** $|S|$ **do**
    for *every subset* $U \subseteq S$ *with* $|U| = i$ **do**
      if $U \in \Delta(S, G)$ **then**
        $Z_{HK}(U) \leftarrow 0$; `// Enforce prefix-free constraint`
      else
        $Z_{HK}(U) \leftarrow 0$;
        for *every* $v \in U$ **do**
          $w_v \leftarrow \mathsf{WO}(v, (U \backslash \{v\}) \cup \mathsf{Pa}_C(v))$;
          $Z_{HK}(U) \leftarrow Z_{HK}(U) + Z_{HK}(U \backslash \{v\}) \times w_v$;
  $Z_{sub} \leftarrow 1$;
  for *every* $j \in [k]$ **do**
    $Z_{sub} \leftarrow Z_{sub} \times$ WEIGHTED-AMO-COUNTING$(G_j, C \oplus C_G(S), d)$;
  $Z_{total} \leftarrow Z_{total} + (Z_{HK}(S) \times Z_{sub})$;
MemoTable$[G, C] \leftarrow Z_{total}$;
return Z_{total};

enumerating specific permutations, we compute an aggregate weight $Z_{HK}(U)$ representing the sum of weights for all valid orderings of a subset $U \subseteq S$. The DP builds iteratively from subset sizes 1 up to $|S|$. When evaluating a node v as the "last" element added to U, its parent set within S is exactly the remainder of the subset, $U \backslash \{v\}$. This allows the algorithm to invoke the weight oracle in $O(1)$ time and multiply the local weight by the previously computed aggregate $Z_{HK}(U \backslash \{v\})$.

Now we will move to the weighted sampling algorithm.

Theorem 3.2 (Computational Complexity of Exact Sampling)**.** *Given a CPDAG C with n nodes, m edges, and bounded indegree d, and assuming the partition functions $Z(G \mid C)$ have been computed and memoized via Algorithm 2, there exists an algorithm (Algorithm 3) to sample a random AMO $D \in \mathsf{EXT}_d(C)$ with exact probability $P(D) = \frac{W(D)}{Z(C)}$ in time bounded by $\tilde{O}\left(n^2(m + 2^d\right)$.*

Once the counting algorithm has populated the memoization table with the exact partition functions, we can sample a valid orientation with probability exactly proportional to its weight. The sampling procedure operates top-down. For a given undirected chordal component G, it first determines the macro-structure of the orientation by sampling a root set S^* (a maximal clique or minimal separator). The probability of selecting a specific root is exactly the fraction of the total partition weight $Z(G)$ that flows through S. This probability is computed by taking the Held-Karp aggregate weight $Z_{HK}(S)$ for the root, multiplying it by the precomputed partition functions of its resulting subcomponents, and dividing by the total weight $Z(G)$.

Once the root set S^* is chosen, all edges between S^* and the rest of the graph are forced outward. The remaining problem is to sample a valid internal topological ordering for the vertices within S^*. To do this without enumerating permutations, the algorithm employs a backtracking strategy that

builds the topological ordering *backwards* from sink to source. Iteratively, it selects which vertex v should be the "last" (sink) node in the ordering of the current unoriented subset $U \subseteq S^*$. By using the intermediate subset weights $Z_{HK}(U\backslash\{v\})$ generated during the Held-Karp DP phase, it samples the sink node v^* with proportional probability, directs all other remaining edges in U toward v^*, and removes v^* from the active set. This reconstructs the internal orientations while exactly preserving the target probability distribution.

Since the outward orientation of the chosen root set S^* acts as a separator, the remaining unoriented edges fragment into smaller, mutually independent chordal subcomponents. The algorithm recursively applies the same sampling procedure to each of these disjoint subcomponents.

Algorithm 3: EXACT-WEIGHTED-SAMPLING(C, G, d, MemoTable)

Input: Ambient CPDAG C, UCCG G, maximum indegree d, MemoTable
Output: Sampled AMO O drawn $\propto W(G \oplus O \mid C)$

if $|V(G)| \leqslant d$ **then**
  Sample valid AMO O of G with probability $\propto W(G \oplus O \mid C)$;
  return O;
Initialize distribution P_{trans} over candidate root blocks S;
for *every* $S \in \Delta(G) \cup \Pi(G)$ **do**
  Compute $C_G(S)$ and its components $G_1, \ldots, G_k$;
  Compute $Z_{HK}(S)$ using the Held-Karp subset DP (as in Algorithm 2);
  $P_{trans}(S) \leftarrow Z_{HK}(S) \prod_{i=1}^{k} \mathsf{MemoTable}[G_i, C \oplus C_G(S)]$;
Normalize P_{trans} by dividing by $\mathsf{MemoTable}[G, C]$;
Sample $S^* \sim P_{trans}$;
$C^* \leftarrow C \oplus C_G(S^*)$;
$O_{out} \leftarrow E_{\mathrm{MLS}}(S^*)$;
$U \leftarrow S^*$;
while $U \neq \varnothing$ **do**
  for *every* $v \in U$ **do**
    $w_v \leftarrow Z_{HK}(U\backslash\{v\}) \times \mathsf{WO}(v, (U\backslash\{v\}) \cup \mathsf{Pa}_{C^*}(v))$;
    $P_{node}(v) \leftarrow w_v / Z_{HK}(U)$;
  Sample $v^* \sim P_{node}$;
  for *every* $u \in U\backslash\{v^*\}$ **do**
    $O_{out} \leftarrow O_{out} \cup \{u \to v^*\}$; `// Direct remaining nodes to sink`
  $U \leftarrow U\backslash\{v^*\}$;
`// Recursively sample subcomponents`
for *each* $G_i \in C_G(S^*)$ **do**
  $O_{sub} \leftarrow$ EXACT-WEIGHTED-SAMPLING($G_i, C^*, d, \mathsf{MemoTable}$);
  $O_{out} \leftarrow O_{out} \cup O_{sub}$;
return O_{out};

4 Experiments

As a practical validation of our theoretical results, we have implemented our algorithm in Python and compared our results to classic algorithms such as GIES and hill-climbing.

Synthetic The first experiment uses synthetic data, but does not assume access to the true CPDAG. For each of 32 random Erdos-Renyi graph instances with linear gaussian models at the nodes, we first estimate the CPDAG from $100,000$ observational samples using GES, then run RWM on this estimated CPDAG for 600 interventional rounds, and finally evaluate the held-out interventional negative log-likelihood (NLL) on rounds $600 - -1000$. This experiment tests a realistic two-stage setting in which the CPDAG is only estimated from (a large amount of) observational data and is not perfect: the mean skeleton SHD to the true graph was 1.31. Our algorithm remains close to both the true-DAG oracle and GIES, and improves over the uniform MEC mixture. The results are shown in Table 1.

Table 1: Synthetic experiment results

Method	Held-out NLL per node	Difference from true-DAG oracle
True DAG oracle	2.0918 ± 0.0055	0.0000
GIES	2.0936 ± 0.0054	$+0.0018 \pm 0.0009$
RWM posterior mixture	2.0972 ± 0.0058	$+0.0054 \pm 0.0015$
Uniform MEC mixture	2.0985 ± 0.0057	$+0.0064 \pm 0.0015$

Sachs protein-signalling As an additional validation on real-world data, we run a Sachs protein-signalling experiment Sachs et al. [2005] in the setting targeted by our method: an observational phase supplies a CPDAG, and limited interventional samples are then used to reweight DAG extensions within that CPDAG. More details, as well as the results, are given in the Appendix F.

Runtime benchmarks Where our approach truly shines is in *scalability*. We consider a runtime benchmark isolating the stage targeted by our method: given a supplied CPDAG, update the weighted MEC distribution, compute the exact normalizer, and sample DAGs. We used bounded-width CPDAGs given by the k-th power of a path, for $k \in \{2, 3, 4\}$, and compare against `bnlearn hc` restricted to the same CPDAG skeleton. RWM scales to $n = 5000$ variables while computing $\log Z$ and drawing 1000 DAG samples in under 30 seconds for all tested widths. In contrast, the restricted `bnlearn hc` baseline takes several minutes at $n = 5000$, with timeouts at 600 seconds for $k = 4$. Thus, conditioned on a given bounded-width CPDAG, RWM provides substantially faster weighted inference over DAG extensions compared to rerunning a standard single-DAG learner. The results are in Appendix F

5 Conclusion and Open Questions

In this work, we introduced the framework of online causal discovery, motivated by real-world settings such as sequential clinical trials and single-cell perturbation experiments where interventional data arrives in an arbitrary, streaming fashion. To address this, we developed a Randomized Weighted Majority algorithm for sequential causal prediction. By leveraging recent algorithmic advancements in Markov Equivalence Class counting, our approach achieves non-asymptotic sublinear regret in polynomial time, bypassing the super-exponential computational bottlenecks and approximate inference issues inherent in standard Bayesian Causal Discovery.

Future Work and Open Questions. The limitations of our work lead to several interesting questions for future research:

- **Latent Confounding:** A natural open question is whether our polynomial-time online updates can be extended to settings with unobserved confounders, potentially by maintaining weights over Ancestral Graphs (MAGs) or Partial Ancestral Graphs (PAGs) rather than DAGs.
- **Finite Observational Data:** Our computational efficiency relies on the assumption that the true observational CPDAG is known exactly. When observational samples are bounded, the true MEC is uncertain. Extending our exact DP framework to jointly weight graphs across multiple uncertain or empirically estimated CPDAGs remains an open problem [Koivisto and Sood, 2004, Eaton and Murphy, 2007, Brouillard et al., 2020].
- **Active Online Exploration:** Our framework handles arbitrary and adversarial sequences of interventions. However, if the algorithm is allowed to *choose* the next experiment, can we design an active bandit-style policy that dynamically targets interventions to optimally minimize the covariate shift constant $C_{\mathrm{shift}}(I')$ for a desired downstream task?

Acknowledgments Arnab Bhattacharyya's research is supported by a start-up grant at the University of Warwick. Philips George John's research is supported by the CQT++ Core Research Funding Grant from Prof. Divesh Aggarwal at the Centre for Quantum Technologies, NUS. Sayantan Sen's research is supported by the NRF Investigatorship award (NRF-NRFI10-2024-0006) and CQT Young Researcher Career Development Grant (25-YRCDG-SS). Naganand Yadati's research is supported by a seed grant at the International Institute of Information Technology, Bangalore.

Statement of AI usage The authors prepared the initial draft of this paper without using any large language model (LLM), extending the ideas from Bhattacharyya et al. [2025] and Wienöbst et al. [2023]. ChatGPT 6 Astra was used subsequently to design and write code for the experiments, and to edit the draft. The authors take full responsibility for all the results in the paper.

References

Raj Agrawal, Caroline Uhler, and Tamara Broderick. Minimal I-MAP MCMC for scalable structure discovery in causal DAG models. In *International Conference on Machine Learning (ICML)*, pages 111–120. PMLR, 2018.

Steen A Andersson, David Madigan, and Michael D Perlman. A characterization of markov equivalence classes for acyclic digraphs. *The Annals of Statistics*, 25(2):505–541, 1997.

Yashas Annadani, Nick Pawlowski, Joel Jennings, Stefan Bauer, Cheng Zhang, and Wenbo Gong. Bayesdag: Gradient-based posterior inference for causal discovery. *Advances in Neural Information Processing Systems*, 36:1738–1763, 2023.

Anne Berry, Richard Krueger, and Genevieve Simonet. Maximal label search algorithms to compute perfect and minimal elimination orderings. *SIAM Journal on Discrete Mathematics*, 23(1):428–446, 2009.

Arnab Bhattacharyya, Sutanu Gayen, Eric Price, Vincent YF Tan, and NV Vinodchandran. Near-optimal learning of tree-structured distributions by chow and liu. *SIAM Journal on Computing*, 52 (3):761–793, 2023.

Arnab Bhattacharyya, Sutanu Gayen, Philips George John, Sayantan Sen, and N. V. Vinodchandran. Distribution learning meets graph structure sampling. In *NeurIPS*, 2025.

Philippe Brouillard, Sébastien Lachapelle, Alexandre Lacoste, Simon Lacoste-Julien, and Alexandre Drouin. Differentiable causal discovery from interventional data. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 21865–21877, 2020.

Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

Davin Choo and Kirankumar Shiragur. Adaptivity complexity for causal graph discovery. In *Uncertainty in Artificial Intelligence*, pages 391–402. PMLR, 2023.

Davin Choo, Kirankumar Shiragur, and Arnab Bhattacharyya. Verification and search algorithms for causal dags. *Advances in Neural Information Processing Systems*, 35:12787–12799, 2022.

Chris Cundy, Aditya Grover, and Stefano Ermon. BCD Nets: Scalable variational approaches for bayesian causal discovery. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, page 7095–7110, 2021.

Luis M de Campos, Andrés Cano, Javier G Castellano, and Serafín Moral. Combining gene expression data and prior knowledge for inferring gene regulatory networks via bayesian networks using structural restrictions. *Statistical Applications in Genetics and Molecular Biology*, 18(3):20180042, 2019.

Gabriel Andrew Dirac. On rigid circuit graphs. In *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, volume 25, pages 71–76. Springer, 1961.

Atray Dixit, Oren Parnas, Biyu Li, Jenny Chen, Charles P Fulco, Livnat Jerby-Arnon, Nemanja D Marjanovic, Danielle Ptashkin, Roberta Chong, Autumn Herbst, et al. Perturb-seq: Dissecting molecular circuits with scalable single-cell RNA profiling of pooled genetic screens. *Cell*, 167(7): 1853–1866, 2016.

Daniel Eaton and Kevin Murphy. Exact bayesian structure learning from uncertain interventions. In *Artificial Intelligence and Statistics (AISTATS)*, pages 107–114. PMLR, 2007.

Judith Feucht, Justin Sun, Anton Dobrin, Ariyan Surun, Alina Michels, Cora Tie, Sakunthala Ramanayake, Mohsen Karimi, Beatriz Aranda-Orgilles, Alice E Zucchetti, et al. Calibration of CAR activation potential directs alternative T cell fates and therapeutic potency. *Nature Medicine*, 25(1):82–88, 2019.

Nir Friedman and Daphne Koller. Being bayesian about network structure. a bayesian approach to structure discovery in bayesian networks. *Machine Learning*, 50(1):95–125, 2003.

Kristjan Greenewald, Dmitriy Katz, Karthikeyan Shanmugam, Sara Magliacane, Murat Kocaoglu, Enric Boix Adsera, and Guy Bresler. Sample efficient active learning of causal trees. *Advances in Neural Information Processing Systems*, 32, 2019.

Alain Hauser and Peter Bühlmann. Two optimal strategies for active learning of causal models from interventional data. *International Journal of Approximate Reasoning*, 55(4):926–939, 2014.

Yang-Bo He and Zhi Geng. Active learning of causal networks with intervention experiments and optimal designs. *Journal of Machine Learning Research*, 9(11), 2008.

Yangbo He, Jinzhu Jia, and Bin Yu. Counting and exploring sizes of markov equivalence classes of directed acyclic graphs. *The Journal of Machine Learning Research*, 16(1):2589–2609, 2015.

David Heckerman, Christopher Meek, and Gregory Cooper. A bayesian approach to causal discovery. In *Innovations in Machine Learning: Theory and Applications*, pages 1–28. Springer, 2006.

Michael Held and Richard M Karp. A dynamic programming approach to sequencing problems. *Journal of the Society for Industrial and Applied mathematics*, 10(1):196–210, 1962.

Kevin D Hoover. The logic of causal inference: Econometrics and the conditional analysis of causation. *Economics & Philosophy*, 6(2):207–234, 1990.

Antti Hyttinen, Frederick Eberhardt, and Patrik O Hoyer. Experiment selection for causal discovery. *Journal of Machine Learning Research*, 14(93):3041–3071, 2013.

Ross D King, Kenneth E Whelan, Ffion M Jones, Philip GK Reiser, Christopher H Bryant, Stephen H Muggleton, Douglas B Kell, and Stephen G Oliver. Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature*, 427(6971):247–252, 2004.

Mikko Koivisto and Kismat Sood. Exact bayesian structure discovery in bayesian networks. *Journal of Machine Learning Research*, 5(May):549–573, 2004.

Jack Kuipers and Giusi Moffa. Partition MCMC for inference on acyclic digraphs. *Journal of the American Statistical Association*, 112(517):282–299, 2017.

Wenqin Liu, Biwei Huang, Erdun Gao, Qiuhong Ke, Howard D. Bondell, and Mingming Gong. Causal discovery with mixed linear and nonlinear additive noise models: A scalable approach. In *Proceedings of the Third Conference on Causal Learning and Reasoning (CLeaR)*, volume 236, pages 1237–1263. PMLR, 2024.

Lars Lorch, Jonas Rothfuss, Bernhard Schölkopf, and Andreas Krause. DiBS: Differentiable bayesian structure learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pages 24111–24123, 2021.

Brian C Miller, Debattama R Sen, Riad Al Abosy, Kevin Bi, Yashaswi V Virkud, Martin W LaFleur, Kathleen B Yates, Ana Lako, Katie Felt, Guray S Naik, et al. Subsets of exhausted CD8+ T cells differentially mediate tumor control and respond to checkpoint blockade. *Nature Immunology*, 20 (3):326–336, 2019.

Jakob Runge, Sebastian Bathiany, Erik Bollt, Gustau Camps-Valls, Dim Coumou, Ethan Deyle, Clark Glymour, Marlene Kretschmer, Miguel D Mahecha, Jordi Muñoz-Marí, et al. Inferring causation from time series in earth system sciences. *Nature communications*, 10(1):2553, 2019.

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

Marco Scutari. Learning bayesian networks with the bnlearn r package. *Journal of statistical software*, 35(1):1–22, 2010.

Karthikeyan Shanmugam, Murat Kocaoglu, Alexandros G Dimakis, and Sriram Vishwanath. Learning causal graphs with small interventions. *Advances in Neural Information Processing Systems*, 28, 2015.

Tomi Silander and Petri Myllymäki. A simple approach for finding the globally optimal bayesian network structure. In *Conference on Uncertainty in Artificial Intelligence*, pages 445–452, 2006.

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

Chandler Squires, Sara Magliacane, Kristjan Greenewald, Dmitriy Katz, Murat Kocaoglu, and Karthikeyan Shanmugam. Active structure learning of causal dags via directed clique trees. *Advances in Neural Information Processing Systems*, 33:21500–21511, 2020.

Jin Tian and Judea Pearl. Causal discovery from changes. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 512–521, 2001.

Christian Toth, Lars Lorch, Christian Knoll, Andreas Krause, Franz Pernkopf, Robert Peharz, and Julius Von Kügelgen. Active bayesian causal inference. *Advances in Neural Information Processing Systems*, 35:16261–16275, 2022.

Thomas Verma and Judea Pearl. Equivalence and synthesis of causal models. In *Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence*, pages 255–270, 1990.

Marcel Wienöbst, Max Bannach, and Maciej Liśkiewicz. Polynomial-time algorithms for counting and sampling markov equivalent dags with applications. *Journal of Machine Learning Research*, 24(213):1–45, 2023.

Jonas Bernhard Wildberger, Siyuan Guo, Arnab Bhattacharyya, and Bernhard Schölkopf. On the interventional kullback-leibler divergence. In *Conference on Causal Learning and Reasoning*, pages 328–349. PMLR, 2023.

Xun Zheng, Bryon Aragam, Pradeep K Ravikumar, and Eric P Xing. DAGs with NO TEARS: Continuous optimization for structure learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, 2018.

Zihan Zhou, Muhammad Qasim Elahi, and Murat Kocaoglu. Sample efficient bayesian learning of causal graphs from interventions. *Advances in Neural Information Processing Systems*, 37: 9178–9204, 2024.

Appendix

A Additional Preliminaries

In this section, we give the additional preliminary definitions and facts which are used in the subsequent technical sections, and which were not given in the technical overview.

A.1 Preliminaries

A.2 Notation

$[n]$ denotes the set $\{1, \ldots, n\}$. Given a set S, $\mathsf{Sym}(S)$ denotes the *symmetry group* of S; that is, all permutations (bijective functions) mapping $S \to S$. The power set of S is denoted by 2^S. For concise expressions and readability, we use the asymptotic complexity notion of $\widetilde{O}(\cdot)$, where we hide poly-logarithmic dependencies of the parameters. By stating i.i.d samples from a distribution P, we mean independently and identically distributed samples from P. For a positive integer ℓ, $\mathsf{Unif}([\ell])$ denotes the uniform distribution on the set $[\ell]$, where each $i \in [\ell]$ is chosen with equal probability of $1/\ell$. We let $\Delta(\mathcal{D})$ denote the set of probability distributions over the elements of a set $\mathcal{D}$. For two such distributions P, Q over $\mathcal{D}$, the KL-*divergence* from Q to P is defined as: $d_{\mathrm{KL}}(P \,\|\, Q) = \sum_{x \in \mathcal{D}} P(x) \log \frac{P(x)}{Q(x)}$

A directed acyclic graph (DAG) G is a directed graph that has no *directed cycles*; that is, it has no paths of the form $u \to v \to \cdots \to u$, where $u, v \in V(G)$. If G is a directed graph, a topological ordering of $V(G)$ is a total-order relation $\preceq$ such that $u \to v \in E(G) \implies u \preceq v$. G is a DAG if and only if $V(G)$ can be topologically ordered.

A *partially directed* or *mixed* graph $G = (V, E)$ is a graph where the edge set E has two kinds of edges: for a pair of vertices $u, v \in V$, there can be (i) an *undirected edge* $\{u, v\}$ (unordered pair) which is denoted $u - v$, or (ii) a *directed edges* (u, v) (denoted $u \to v$) or (v, u) (denoted $v \to u$), but not both. Every edge is either undirected or directed, and an undirected edge $u - v$ is always used instead of a pair of directed edges $u \to v$, $v \to u$. If V is a vertex set, we denote by $\mathsf{Edges}(V)$ the set of all possible directed and undirected edges on V. Note that the definition of partially directed graphs includes both undirected graphs (every edge is undirected) and directed graphs (every edge is directed) as special cases. A *path* from vertex u to vertex v in a partially directed graph is of the form $u \rightleftharpoons w_1 \rightleftharpoons \cdots \rightleftharpoons w_r \rightleftharpoons v$, where $r \geqslant 0$ and each edge $x \rightleftharpoons y$ in the path is either undirected or directed along the path; as $u \to w_1$, $w_i \to w_{i+1}$, $w_r \to v$.

If $G = (V, E)$ is an undirected graph and $S, T \subseteq V$ are non-empty, $E(S, T)$ denotes the subset of edges of V with one endpoint in S and the other in T. $G[S] = (S, E(S, S))$ denotes the *induced subgraph* of G on S. If $G = (V, E)$ is a *partially directed graph*, $E(S, T)$ instead denotes the union of two sets: (i) the set of directed edges where the *initial vertex* is in S and the *terminal vertex* is in T, and (ii) the set of undirected edges where one endpoint is in S and the other is in T. $G[S]$ is still $(S, E(S, S))$.

Orienting a PDAG ($G \oplus G^*$ and $G \oplus E^*$) If $G = (V, E)$ is a partially directed graph, another partially directed graph $G^* = (V^*, E^*)$ is said to be consistent with G if: (i) $V^* \subseteq V$, (ii) E^* is such that $u - v \in E^*$ implies $u - v \in E(V^*, V^*)$, and $u \to v \in E^*$ implies either $u \to v \in E(V^*, V^*)$ or $u - v \in E(V^*, V^*)$. That is, E^* is obtained by directing some of the undirected edges in $E(V^*, V^*)$ and keeping some of the undirected/directed edges in $E(V^*, V^*)$ as they are. If G, G^* are partially directed graphs and G^* is consistent with G, $G \oplus G^*$ denotes the partially-directed graph on vertex set $V = V(G)$ where (i) any undirected edge of G which is directed in G^* is directed in $G \oplus G^*$ as per its orientation in G^*, and (ii) the remaining edges of G (whether directed or undirected) are included in $G \oplus G^*$ as they are (in G). We also use the same notation $G \oplus E^*$ if E^* is a subset of (mixed) edges on $V(G)$ which are consistent with $E(G)$ (as described above), so that $G \oplus E^* \equiv G \oplus G^*$ for $G^* = (V, E^*)$.

Definition A.1 (Chordal graph)**.** An undirected graph G is said to be *chordal* if every cycle of length at least 4 in G has a *chord*; that is, an edge that connects two vertices which are non-adjacent in the cycle.

Definition A.2 (Bayesian networks)**.** A probability distribution P over n variables $X_1, \ldots, X_n$ is said to be a *Bayesian network (Bayes net in short) on a directed acyclic graph* G with n nodes if[1] for every $i \in [n]$, X_i is conditionally independent of $X_{\mathsf{nd}(i)}$ given $X_{\mathsf{pa}(i)}$. Equivalently, P admits the factorization:

$$P(x) = \Pr_{X \sim P}[X = x] = \prod_{i=1}^{n} \Pr_{X \sim P}[X_i = x_i \mid \forall j \in \mathsf{pa}(i), X_j = x_j] \qquad \text{for all } x. \tag{9}$$

Definition A.3 (Markov equivalent DAGs)**.** Let $D = ([n], E)$ be a DAG. A distribution P on n random variables $(X_1, \ldots, X_n)$ is said to be *Markov* with respect to D, or *D-structured*, if it factorizes as

$$P(X_1 = x_1, \ldots, X_n = x_n) = \prod_{i \in [n]} P(X_i = x_i | X_j = x_j \ \forall\, j \in \mathsf{Pa}_D(i)),$$

where $\mathsf{Pa}_D(i)$ denotes the set of parents of the node i in D. Two DAGs D_1 and D_2 on $[n]$ are said to be Markov equivalent if, for any distribution $P(X_1, \ldots, X_n)$, P is Markov to D_1 if and only if P is Markov to D_2. We denote this equivalence relation by $D_1 \sim_{\mathrm{M}} D_2$.

Definition A.4 (MEC, CPDAG and UCCG)**.** The Markov equivalence relation $\sim_{\mathrm{M}}$ partitions the set of all DAGs on vertices $[n]$ into *Markov Equivalence Classes* (MECs). The MEC for a DAG D is denoted by $[D]$. An MEC can be uniquely represented by a *CPDAG (completed partially directed acyclic graph)*. The CPDAG for an MEC M is the *union graph* of all the DAGs in the equivalence class M. That is, $\mathsf{CPDAG}(M) \triangleq \left(V, \bigcup_{H=(V,E_H) \in M} E_H\right)$. The undirected components of a CPDAG are said to be *undirected and connected chordal graphs (UCCGs)*.

Max-cliques and minimal separators Given an undirected graph G, the set of all maximal cliques of G is denoted by $\Pi(G)$. For a connected graph G with vertex set V, and for any pair of vertices $a, b \in V$, a subset of vertices $S \subseteq V \backslash \{a, b\}$ is said to be an (a, b)-separator if a and b are in different connected components in $G[V \backslash S]$. If no proper subset of S separates the vertices a and b, we define S to be a minimal (a, b)-separator. A set $S \subseteq V$ is said to be a minimal separator of G if S is a minimal separator for at least one pair of vertices in $V \backslash S$. Given an undirected graph G, the set of all minimal separators of G is denoted by $\Delta(G)$.

Definition A.5 (Prefix-free permutations)**.** If $S \subseteq [n]$ and $\mathcal{R} \subseteq 2^{[n]}$, then $\mathsf{PFP}(S, \mathcal{R})$ denotes the set of all permutations of S that do not have a proper prefix in $\mathcal{R}$. If $S = \{i_1, \ldots, i_k\}$ with $i_1 < \cdots < i_k$, then $S' = \{j_1, \ldots, j_\ell\} \subseteq [n]$ (in ascending order) with $\ell < k$ is a proper prefix of a permutation $\sigma \in \mathsf{Sym}(S)$ if there exists a permutation τ of S' such that $\sigma(i_t) = \tau(j_t)$ for all $t \in [\ell]$.

That is, if a permutation σ of $S = \{i_1, \ldots, i_k\}$ (in ascending order) is represented as a vector $[\sigma(i_1), \cdots, \sigma(i_k)]$, a set S' (with $|S'| = \ell$) is a proper prefix of σ if the first ℓ elements ($\ell < k$) of the vector match some permutation of S'. E.g. if $S = \{1, 2, 3\}$ and $S' = \{1, 2\}$, then S' is a proper prefix of $\sigma_1 = [1, 2, 3]$ but not of $\sigma_2 = [1, 3, 2]$. Note that, by definition, only a proper subset of S can be a proper prefix of a permutation of S, and so $\mathsf{PFP}(S, \mathcal{R}) = \mathsf{PFP}(S, \mathcal{R} \cap (2^S \backslash S))$.

Definition A.6 (Simplicial vertex)**.** Let $G = (V, E)$ be a graph. A vertex $v \in V$ is said to be *simplicial* if the neighbors of v form a clique.

Definition A.7 (Perfect Elimination Ordering)**.** Let $G = (V, E)$ be a chordal graph and $V = \{v_1, \ldots, v_n\}$ be the set of vertices in G. A linear ordering $\rho = (v_1, \ldots, v_n)$ of the vertices in V is said to be a *Perfect Elimination Ordering* (PEO) if for every $i \in [n]$, the vertex v_i is simplical in the induced subgraph $G[\{v_1, \ldots, v_i\}]$.

Definition A.8. Let C be a PDAG on vertex set $[n]$, and let $E_{\mathrm{u}}(C)$ denote the set of undirected edges of C. Then an *orientation* O of C is a function $O : E_{\mathrm{u}}(C) \to \{\pm 1\}$ such that $O(\{u, v\}) = 1$ for $u < v$ indicates that the edge should be directed $u \to v$.

[1] We use the notation X_S to denote $\{X_i : i \in S\}$ for a set $S \subseteq [n]$.

In this case, $C \oplus O$ denotes the directed graph D with vertex set $[n]$ and with $E(D)$ defined such that

$$u \to v \in E(D) \iff \begin{cases} u \to v \in E(C) \\ \{u, v\} \in E(C) \wedge u < v \wedge O(\{u, v\}) = 1 \\ \{u, v\} \in E(C) \wedge u > v \wedge O(\{u, v\}) = -1 \end{cases}$$

A.3 Bayesian Networks

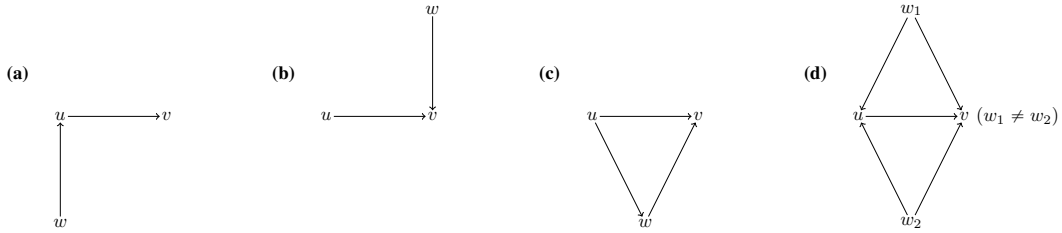
Definition A.9 (Skeleton of a PDAG)**.** Let G be a partially directed graph. The *skeleton* of G, denoted $\mathrm{skel}(G)$, is the underlying undirected graph obtained by transforming all directed edges in G to undirected edges. That is, if $G = (V, E)$, $\mathrm{skel}(G) = (V, E')$ where $E' = \{u - v : u - v \in E \vee u \to v \in E \vee v \to u \in E\}$.

Definition A.10 (v-structure)**.** Let G be a partially directed graph. A *v-structure* is an ordered triplet of vertices (a, b, c) such that the induced subgraph $G[\{a, b, c\}]$ is $a \to b \leftarrow c$. That is, there are directed edges from a and c to the middle vertex b, but these parents are not connected by any (directed or undirected) edge.

Lemma A.11 (Markov equivalence criterion Verma and Pearl [1990])**.** *Two DAGs D_1 and D_2 are Markov equivalent if and only if D_1 and D_2 have the same skeleton and the same v-structures.*

Definition A.12 (Chain graph)**.** A graph G is said to be a *chain graph* if there does not exist any directed cycle in G.

Definition A.13 (Strongly Protected edge)**.** Let G be a graph and $e = u \to v$ be an edge in G. The edge e is said to be *strongly protected* if $u \to v$ occurs in at least one of the following four configurations as an induced subgraph in G:



The following result of Andersson et al. [1997] characterizes when a graph G is an MEC.

Lemma A.14 (Andersson's Rules for CPDAGs [Andersson et al., 1997, Theorem 4.1])**.** *A graph G is an MEC if and only if it satisfies the following four conditions:*

(i) G is a chain graph.

(ii) For every chain component τ of G, G_τ is chordal.

(iii) The configuration $a \to b - c$ does not occur as an induced subgraph of G.

(iv) Every directed edge $u \to v \in G$ is strongly protected in G.

Now we define the notion of a consistent extension of a partially directed graph.

Definition A.15 (Consistent extension)**.** If G is a (partially directed) graph, a *consistent extension* of G is an orientation O of *all* of the undirected edges in $E(G)$ such that (i) $G \oplus O$ is acyclic (i.e., a DAG), and (ii) $G \oplus O$ does not have any new v-structures (aka immoralities) that are not present in G. The set of all consistent extensions of a graph G is denoted by $\mathsf{EXT}(G)$.

The following observation follows directly from the definition of CPDAGs and consistent extensions.

Observation A.16. If G is a CPDAG representing a MEC M, then each distinct $O \in \mathsf{EXT}(G)$ gives a DAG $G \oplus O \in M$, and vice versa. Thus, $\mathsf{EXT}(G)$ is isomorphic to the MEC M.

Definition A.17 (Acyclic Moral Orientation)**.** If G is an undirected chordal graph (thus it is also a CPDAG by definition), each consistent extension of G is an orientation O of *all* the edges of G such that (i) $G \oplus O$ is acyclic and (ii) $G \oplus O$ has *no* v-structures. Such an orientation is called an *Acyclic Moral Orientation* (AMO) of G, and the set of AMOs is denoted $\mathsf{AMO}(G)$.

Claim A.18 (AMO substructure via UCCGs Wienöbst et al. [2023] Theorem 17 and Corollary 18)**.** *If C is a CPDAG with UCCGs $H_1, \ldots, H_k$, there is a bijection between the set of consistent extensions of C and the Cartesian product of the sets of AMOs of $H_1, \ldots, H_k$ (taken together with the oriented edges of C). The AMOs of $H_1, \ldots, H_k$ are also independent of each other given the oriented edges of C.*

Lemma A.19 ([Wienöbst et al., 2023, Lemma 13])**.** *Let $G = (V, E)$ be a UCCG. A topological ordering τ of the vertices of G represents an AMO if and only if it is the reverse of a PEO.*

Observation A.20 (Wienöbst et al. [2023])**.** If $G = (V, E)$ is an undirected chordal graph, every AMO of G can be uniquely represented as the reverse of a perfect elimination ordering (PEO) of G that starts with a set $S \in \Delta(G) \cup \Pi(G)$ ordered using a permutation $\sigma \in \mathsf{PFP}(S, \Delta(G))$.

Consistent extension Counting Given a CPDAG G on vertices $[n]$, $\mathsf{EXT}(G)$ denotes the set of consistent extensions of G (see Definition A.15), which is isomorphic to the MEC $[G]$. The extension counting problem in Wienöbst et al. [2023] is to compute the *cardinality* of this set; that is, to compute

$$\#\mathsf{EXT}(G) \triangleq \sum_{O \in \mathsf{EXT}(G)} 1.$$

Even though the set of AMOs could have exponential size (e.g., if G is an undirected complete graph K_n), Wienöbst et al. [2023] gives a polynomial (in n) time algorithm for computing the cardinality. They establish the following recurrence in the case where G is an undirected chordal graph

$$\#\mathsf{AMO}(G) = \sum_{S \in \Delta(G) \cup \Pi(G)} \underbrace{|\mathsf{PFP}(S, \Delta(G))|}_{\triangleq \Phi(S)} \prod_{H \in C_G(S)} \#\mathsf{AMO}(H).$$

Here $C_G(S)$ denotes the set of undirected chordal components of the chain graph that they obtain by orienting some of the edges of G using *Maximum Label Search* (MLS in short) (Algorithm 4 in Wienöbst et al. [2023]). This gives a recursive algorithm for counting the AMOs of an undirected chordal graph G (where they also have to compute $\Phi(S)$ efficiently for each S, for which they use a different form).

Then, Wienöbst et al. [2023] counts the consistent extensions of a CPDAG G as

$$\#\mathsf{EXT}(G) \triangleq \prod_{H \in \mathsf{UCCG}(G)} \#\mathsf{AMO}(H).$$

A.4 Online Prediction with Experts

The framework of prediction with experts is setup as follows; the formulation we follow here is based on Chapter 2 of Cesa-Bianchi and Lugosi [2006]. The goal is to design an algorithm $\mathcal{A}$ that predicts an unknown sequence $x^{(1)}, x^{(2)}, \ldots$ of elements of an *outcome space* $\mathcal{X}$. The algorithm's predictions $\widehat{P}_1, \widehat{P}_2, \ldots$ belong to a *decision space* $\mathcal{D}$ (which in our case will be the probability simplex on $\mathcal{X}$). The algorithm $\mathcal{A}$ makes its predictions sequentially, and the quality of its predictions is benchmarked against a set of reference predictions called *experts*. At each time instant t, $\mathcal{A}$ has access to the set of expert predictions $\mathcal{E}$, where $\mathcal{E}$ is a fixed subset of $\mathcal{D}$. $\mathcal{A}$ makes its own prediction $\widehat{P}_t$ based on the expert predictions. Finally, after the prediction $\widehat{P}_t$ is made, the true outcome $x^{(t)}$ is revealed, and the algorithm $\mathcal{A}$ incurs a loss $\ell(\widehat{P}_t, x^{(t)})$, where $\ell : \mathcal{D} \times \mathcal{X} \to \mathbb{R}$ is a non-negative *loss function*.

Note that $x^{(1)}, x^{(2)}, \ldots$ form an arbitrary sequence. Hence, naturally, the loss incurred by the algorithm depends on how well the experts fit these outcomes.

Now we are ready to define the notion of regret.

Definition A.21 (Regret)**.** Given the notation above, the *cumulative regret* (or simply *regret*) over T steps is defined as:

$$\mathsf{Reg}_T(\mathcal{A}; \mathcal{E}) = \sum_{t=1}^{T} \ell(\widehat{P}_t, x^{(t)}) - \min_{E \in \mathcal{E}} \sum_{t=1}^{T} \ell(E, x^{(t)})$$

Since $\mathcal{E}$ is usually fixed, we often use the shorthand $\mathsf{Reg}_T(\mathcal{A})$. The *average regret* is defined as $\frac{1}{T}\mathsf{Reg}_T(\mathcal{A})$. The total T steps are also sometimes referred to as the *horizon* of the algorithm.

In this work, we apply the well-studied online *Randomized Weighted Majority* (RWM) algorithm in our online interventional learning framework. The RWM algorithm maintains a time-varying *distribution* over a set $\mathcal{E}$ of experts, and at each time step t, predicts a single randomly chosen expert $\widehat{P}_t \in \mathcal{E}$ before seeing the outcome $x^{(t)} \in \mathcal{X}$. The algorithm then suffers a loss $\ell(\widehat{P}_t, x^{(t)})$ determined by a *loss function* $\ell : \mathcal{E} \times \mathcal{X} \to \mathbb{R}_{\geqslant 0}$.

The regret bound below for RWM assumes the loss function $\ell(\cdot, \cdot)$ is bounded.

Lemma A.22 (Cesa-Bianchi and Lugosi [2006], Lemma 4.2)**.** *With $N = |\mathcal{E}|$ experts and a loss function bounded in $[-L_{\max}, L_{\max}]$, for any sequence of outcomes $x^{(1)}, \ldots, x^{(T)}$, the expected regret of the* RWM *forecaster with the parameter $\eta = \sqrt{8(\log N)/T}$ is bounded as follows:*

$$\mathbb{E}[\mathsf{Reg}_T(\mathsf{RWM}; \mathcal{E})] \leqslant L_{\max}\sqrt{\frac{T \log N}{2}}.$$

Moreover, with probability at least $1 - \delta$, the following holds:

$$\mathsf{Reg}_T(\mathsf{RWM}; \mathcal{E})] \leqslant L_{\max}\left(\sqrt{\frac{T \log N}{2}} + \sqrt{\frac{T}{2}\log\frac{1}{\delta}}\right).$$

Algorithm 4: RWM forecaster

Input : Experts $\mathcal{E} = \{E_1, \ldots, E_N\}$, parameter η, horizon T.

$w_{i,0} \leftarrow 1$ for each $i \in [N]$.
for $t \leftarrow 1$ *to* T **do**
  Sample $\widehat{P}_t \in \mathcal{E}$ where $\Pr[\widehat{P}_t = E_i] = \frac{w_{i,t-1}}{\sum_{j=1}^N w_{j,t-1}}$.
  Output $\widehat{P}_t$.
  Observe outcome $x^{(t)}$.
  for $i \in [N]$ **do**
    $w_{i,t} \leftarrow w_{i,t-1} \cdot \exp\left(-\eta \cdot \ell(E_i, x^{(t)})\right)$.

B Online Causal Discovery

In this section, we prove the technical lemmas used to prove Theorem 2.1, which we restate here.

Theorem 2.1. *Suppose $\mathcal{I}$ is a finite set of intervention targets which are seen in the online interventional phase, and $\mu(I) \triangleq \frac{1}{T}\sum_{t=1}^T \mathbb{I}[I_t = I]$ denotes the relative frequency of intervention I among $I_1, \ldots, I_T$. Then, for any class of DAGs $\mathcal{C}$, if $T \geqslant \widetilde{\Omega}\left(\frac{n^3 d}{\varepsilon^2}\right)$:*

$$\mathop{\mathbb{E}}_{x^{(1:T)}} \frac{1}{T}\sum_{t=1}^T d_{\mathrm{KL}}\left(P^*_{\mathrm{do}(I_t)} \,\middle\|\, \widehat{P}_{D_t, I_t}\right) \leqslant \min_{D \in \mathcal{C}} \sum_{I \in \mathcal{I}} \mu(I)\, d_{\mathrm{KL}}\left(P^*_{\mathrm{do}(I)} \,\middle\|\, \widehat{P}_{D,I}\right) + \varepsilon. \tag{4}$$

The proof relies on the following general lemma, which relates the time-average of the step-wise KL losses of the interventional predictions to the regret of an arbitrary online prediction algorithm w.r.t the log loss.

Lemma B.1 (Online causal discovery w.r.t known interventions)**.** *Suppose $\mathcal{I}$ is a finite set of intervention targets which are seen in the online interventional phase, and $\mu(I) \triangleq \frac{1}{T}\sum_{t=1}^T \mathbb{I}[I_t = I]$ denotes the relative frequency of intervention I among $I_1, \ldots, I_t$. Consider an online learning algorithm $\mathcal{A}$ that predicts DAGs $D_1, \ldots, D_T \in \mathcal{C}$ for a class of DAGs $\mathcal{C}$ under the loss given by Eq.* (2)*. Then*

$$\mathop{\mathbb{E}}_{x^{(1:T)}} \frac{1}{T}\sum_{t=1}^T d_{\mathrm{KL}}\left(P^*_{\mathrm{do}(I_t)} \,\middle\|\, \widehat{P}_{D_t, I_t}\right) \leqslant \frac{1}{T}\mathop{\mathbb{E}}_{x^{(1:T)}}\left[\mathrm{Regret}_T(\mathcal{A}; \mathcal{C})\right] + \min_{D \in \mathcal{C}} \sum_{I \in \mathcal{I}} \mu(I)\, d_{\mathrm{KL}}\left(P^*_{\mathrm{do}(I)} \,\middle\|\, \widehat{P}_{D,I}\right) \tag{10}$$

Proof. By Jensen's inequality and the concavity of the minimum operator, we can push the expectation inside the minimum for the baseline term:

$$\frac{1}{T}\,\mathbb{E}\left[\mathrm{Regret}_T(\mathcal{A};\mathcal{C})\right] \geqslant \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\ell_t(D_t)\right] - \min_{D\in\mathcal{C}}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\ell_t(D)\right]$$

We can then analyze the first term of this expression, substituting the loss defined in Equation (2), and use the fact that D_t is determined by $x^{(1)},\ldots,x^{(t-1)}$ and the randomness of the algorithm (before seeing $x^{(t)}$) to get:

$$\begin{aligned}
\frac{1}{T}\sum_{t=1}^{T}\mathop{\mathbb{E}}_{x^{(1:t)}\sim P^*_{\mathrm{do}(I_1)}}\left[\ell_t(D_t)\right] &= \frac{1}{T}\sum_{t=1}^{T}\mathop{\mathbb{E}}_{x^{(1)}:x^{(t-1)}}\mathop{\mathbb{E}}_{x\sim P^*_{\mathrm{do}(I_t)}}\left[-\log\widehat{P}_{D_t,I_t}(x)\,\middle|\,x^{(1)},\ldots,x^{(t-1)}\right] \\
&= \frac{1}{T}\sum_{t=1}^{T}\mathop{\mathbb{E}}_{x^{(1)},\ldots,x^{(t-1)}}\left[\mathrm{H}\left(P^*_{\mathrm{do}(I_t)},\widehat{P}_{D_t,I_t}\right)\right] \\
&= \mathop{\mathbb{E}}_{x^{(1)},\ldots,x^{(T)}}\left[\frac{1}{T}\sum_{t=1}^{T}\mathrm{H}\left(P^*_{\mathrm{do}(I_t)},\widehat{P}_{D_t,I_t}\right)\right],
\end{aligned}$$

where $\mathrm{H}(P,Q)$ is the cross entropy.

We can apply the exact same logic to the expression inside the minimizer in the second term, noting that D is a fixed DAG independent of the samples and the algorithm's randomness, to get:

$$\frac{1}{T}\,\mathbb{E}\left[\mathrm{Regret}_T(\mathcal{A};\mathcal{C})\right] \geqslant \mathop{\mathbb{E}}_{x^{(1)},\ldots,x^{(T)}}\left[\frac{1}{T}\sum_{t=1}^{T}\mathrm{H}\left(P^*_{\mathrm{do}(I_t)},\widehat{P}_{D_t,I_t}\right)\right] - \min_{\mathcal{D}}\frac{1}{T}\sum_{t=1}^{T}\mathrm{H}\left(P^*_{\mathrm{do}(I_t)},\widehat{P}_{D,I_t}\right).$$

By the definition of $\mu(I)$ as the relative frequency of I in $I_1,\ldots,I_t$, we can rewrite:

$$\frac{1}{T}\,\mathbb{E}\left[\mathrm{Regret}_T(\mathcal{A};\mathcal{C})\right] \geqslant \mathop{\mathbb{E}}_{x^{(1)},\ldots,x^{(T)}}\left[\frac{1}{T}\sum_{t=1}^{T}\mathrm{H}\left(P^*_{\mathrm{do}(I_t)},\widehat{P}_{D_t,I_t}\right)\right] - \min_{\mathcal{D}}\sum_{I\in\mathcal{I}}\mu(I)\,\mathrm{H}\left(P^*_{\mathrm{do}(I)},\widehat{P}_{D,I}\right).$$

Finally, adding and subtracting $\frac{1}{T}\sum_{t=1}^{T}\mathrm{H}(P^*_{\mathrm{do}(I_t)}) = \sum_{I\in\mathcal{I}}\mu(I)\,\mathrm{H}(P^*_{\mathrm{do}(I)})$ changes the cross-entropies to KL divergences:

$$\frac{1}{T}\,\mathbb{E}\left[\mathrm{Regret}_T(\mathcal{A};\mathcal{C})\right] \geqslant \mathop{\mathbb{E}}_{x^{(1:T)}}\left[\frac{1}{T}\sum_{t=1}^{T}d_{\mathrm{KL}}\left(P^*_{\mathrm{do}(I_t)}\,\middle\|\,\widehat{P}_{D_t}\right)\right] - \min_{D\in\mathcal{C}}\sum_{I\in\mathcal{I}}\mu(I)\,d_{\mathrm{KL}}\left(P^*_{\mathrm{do}(I)}\,\middle\|\,\widehat{P}_{D}\right), \tag{11}$$

which proves the lemma. □

Constructing $\widehat{P}$ from the true P^*

As described in the introduction, we assume that the true observational distribution P^* is known in the online interventional phase. But to get the sample complexity using RWM regret bounds when taking $\widehat{P}=P^*$ as in the overview, we would need to assume that $\widehat{P}=P^*$ is such that $\widehat{P}(x)\geqslant\tau$ for all x, which need not always be the case. We now describe the change we need to make to obtain $\widehat{P}$ from P^* in such a way that we can lower-bound the probabilities of $\widehat{P}$ while maintaining closeness in KL divergence.

We define two operations on probability distributions that will be useful in constructing $\widehat{P}$. These were first defined in [Bhattacharyya et al., 2025, version 1 preprint]. Let $P\in\Delta([k])$ for some positive integer k, and without loss of generality $P(1)\leqslant P(2)\leqslant\cdots\leqslant P(k)$. Note that $P(k-1)\leqslant 1/2$.

Definition B.2. Given $0<\tau<1/k^2$, if $P(i)\geqslant\tau$ for $1\leqslant i<k$, define $\mathsf{Clip}_\tau(P)=P$. Otherwise, let m such that $P(m)<\tau$ but $P(m+1)\geqslant\tau$, and define $\mathsf{Clip}_\tau(P)=Q$ where:

$$Q(i)=\begin{cases}\tau & \text{if } 1\leqslant i\leqslant m\\ P(i) & \text{if } m+1\leqslant i<k\\ 1-\sum_{j=1}^{k-1}Q(j) & \text{if } i=k\end{cases}$$

Note that $\mathsf{Clip}_\tau(P)(k) = P(k) - \sum_{i=1}^m (\tau - P(i)) \geqslant \frac{1}{k} - (k-1)\tau \geqslant \tau$ for $\tau \leqslant \frac{1}{2k^2}$.

Lemma B.3. *For any $P, P' \in \Delta([k])$, if $\tau \leqslant 1/2k^2$ and $Q = \mathsf{Clip}_\tau(P)$:*

$$d_{\mathrm{KL}}(P'\|Q) \leqslant d_{\mathrm{KL}}(P'\|P) + 2k^2\tau.$$

Proof. By definition, $Q(i) \geqslant P(i)$ for $i < k$ and $Q(k) \geqslant P(k) - k\tau$. Hence,

$$\begin{aligned} d_{\mathrm{KL}}(P'\|\mathsf{Clip}_\tau(P)) - d_{\mathrm{KL}}(P'\|P) &= \sum_{i=1}^k P'(i) \log \frac{P(i)}{Q(i)} \leqslant \log \frac{P(k)}{Q(k)} \leqslant \log \frac{P(k)}{P(k) - k\tau} \leqslant \log \frac{1/k}{1/k - k\tau} \\ &\leqslant 2k^2\tau \end{aligned}$$

where we used the facts that $P(k) \geqslant 1/k$, $-\log(1-x) \leqslant 2x$ for $x \leqslant 1/2$, and $k^2\tau \leqslant 1/2$. □

Now we define the bucketing operation. Below, let $Q = \mathsf{Clip}_\tau(P)$ so that $\tau \leqslant Q(1) \leqslant Q(2) \leqslant \cdots \leqslant Q(k-1) \leqslant 1/2$.

Definition B.4. Given Q as above, and a parameter $\gamma > 0$, define $\mathsf{Bucket}_\gamma(Q)$ to be a distribution R such that:

$$R(i) = \begin{cases} \tau \cdot (1+\gamma)^{\lfloor \log_{1+\gamma}(Q(i)/\tau) \rfloor} & \text{if } 1 \leqslant i < k \\ 1 - \sum_{j=1}^{k-1} R(j) & \text{if } i = k \end{cases}$$

Lemma B.5. *Suppose $0 < \tau < 1/2k^2$ and $0 \leqslant \gamma < 1$. For any $P', P \in \Delta([k])$, if $Q = \mathsf{Clip}_\tau(P)$, and $R = \mathsf{Bucket}_\gamma(Q)$, then:*

$$d_{\mathrm{KL}}(P'\|R) \leqslant d_{\mathrm{KL}}(P'\|Q) + \gamma \leqslant d_{\mathrm{KL}}(P'\|P) + 2k^2\tau + \gamma.$$

Proof. The second inequality follows from Lemma B.3; so we focus on the first. Note that $R(i) \leqslant Q(i)$ for $i < k$, and so, $R(k) \geqslant Q(k)$.

$$d_{\mathrm{KL}}(P'\|R) - d_{\mathrm{KL}}(P'\|Q) = \sum_{i=1}^k P'(i) \log \frac{Q(i)}{R(i)} \leqslant \sum_{i=1}^{k-1} P'(i) \log \frac{Q(i)}{R(i)} \leqslant \log(1+\gamma) \leqslant \gamma.$$

□

For future reference, note that using the notation above, $R(k) \geqslant Q(k) \geqslant \tau$ (for $\tau \leqslant \frac{1}{2k^2}$).

Given a DAG D, for a distribution $Q \in \Delta([k]^n)$, a node $i \in [n]$, and an assignment $\sigma \in [k]^{\mathsf{Pa}(i)}$ to the parents of the node i in D, let $Q_{\mathsf{Pa}(i)} \in \Delta([k]^{\mathsf{Pa}(i)})$ be the marginal distribution of Q on $\mathsf{Pa}(i)$, and let $Q_{i,\sigma} \in \Delta([k])$ denote the conditional distribution $Q(X_i | X_{\mathsf{Pa}(i)} = \sigma)$ of Q on node i, conditioned on its parents having value(s) σ. Note that here, Q itself need not be a Bayes net on D.

Lemma B.6 (Clipping and bucketing to lower-bound probabilities)**.** *For any $P^* \in \Delta([k]^n)$, and for any DAG D with indegree $\leqslant d$, with*

$$P^*_D(x) \triangleq \prod_{i \in [n]} P^*(x_i | x_{\mathsf{Pa}(i)}),$$

define

$$\widehat{P}_D(x) \triangleq \prod_{i \in [n]} \mathsf{Bucket}_\gamma \mathsf{Clip}_\tau \left(P^*_{i, x_{\mathsf{Pa}(i)}} \right)$$

with $\tau = \varepsilon/(4nk^2)$, $\gamma = \varepsilon/2n$, and the notation $P^*_{i,\sigma}$ as described above.

Then, for any distribution Q on $[k]^n$, $d_{\mathrm{KL}}(Q\|\widehat{P}_D) \leqslant d_{\mathrm{KL}}(Q\|P^*_D) + \varepsilon$. Also, by construction, $\widehat{P}_D$ will satisfy $\widehat{P}_D(x) \geqslant \left(\frac{\varepsilon}{4nk^2}\right)^n$ for each $x \in [k]^n$.

Proof. Let $R_{i,\sigma} \triangleq \mathsf{Bucket}_\gamma \mathsf{Clip}_\tau \left(P^*_{i,\sigma}\right)$, so that $\widehat{P}_D(x) = \prod_{i \in [n]} R_{i, x_{\mathsf{Pa}(i)}}$.

Using Lemma B.5 with $\tau = \varepsilon/(4nk^2)$ and $\gamma = \varepsilon/2n$, $d_{\mathrm{KL}}(Q_{i,\sigma}\|R_{i,\sigma}) \leqslant d_{\mathrm{KL}}(Q_{i,\sigma}\|P^*_{i,\sigma}) + \varepsilon/n$ for each i, σ. We now use the following fact:

Fact B.7 (Lemma 3.3 in Bhattacharyya et al. [2023])**.** Suppose $P, Q \in \Delta([k]^n)$, and P is a Bayes net on a DAG D. Then the following holds:

$$d_{\mathrm{KL}}(Q\|P) = J_{Q,D} + \sum_{i\in[n]} \sum_{\sigma\in[k]^{\mathsf{Pa}(i)}} Q_{\mathsf{Pa}(i)}(\sigma) \cdot d_{\mathrm{KL}}(Q_{i,\sigma}\|P_{i,\sigma}), \tag{12}$$

where $J_{Q,D}$ is a constant that depends on Q and D but not on P.

Therefore, we can say that:

$$\begin{aligned} d_{\mathrm{KL}}(Q\|\widehat{P}_D) &= J_{Q,D} + \sum_{i\in[n]} \sum_{\sigma\in[k]^{\mathsf{Pa}(i)}} Q_{\mathsf{Pa}(i)}(\sigma) \cdot d_{\mathrm{KL}}(Q_{i,\sigma}\|R_{i,\sigma}) \\ &\leqslant J_{Q,D} + \sum_{i\in[n]} \sum_{\sigma\in[k]^{\mathsf{Pa}(i)}} Q_{\mathsf{Pa}(i)}(\sigma) \cdot (d_{\mathrm{KL}}(Q_{i,\sigma}\|P^*_{i,\sigma}) + \varepsilon/n) \\ &= d_{\mathrm{KL}}(Q\|P^*_D) + \varepsilon. \end{aligned}$$

□

Proof of Theorem 2.1

We run Algorithm 1 with $\widehat{P}$ (used to compute the log losses and weights for each DAG D) constructed from the true observational distribution P^* as described in Lemma B.6. We can then use the RWM expected regret bound (Lemma A.22), with $L_{\max} = n \log\left(\frac{4nk^2}{\varepsilon}\right)$ (since $\widehat{P}(x) \geqslant \left(\frac{\varepsilon}{4nk^2}\right)^n$ from Lemma B.6) and $N = \mathsf{EXT}_d(C) \leqslant (n!)^{d+1}$, to get

$$\frac{1}{T}\,\mathbb{E}[\mathsf{Reg}_T(\mathsf{RWM}; \mathsf{EXT}_d(C))] \leqslant n \log\left(\frac{4nk^2}{\varepsilon}\right)\sqrt{\frac{(d+1)n\log n}{2T}}.$$

Substituting the bound $T \geqslant \frac{2n^3 d \log^3\left(\frac{4nk}{\varepsilon}\right)}{\varepsilon^2}$ into this expression and using Lemma B.1 proves the theorem. □

C Online-to-Batch Causal Discovery

C.1 Online-to-Batch Causal Discovery with Known Interventions

In this section, we prove the technical lemmas used to prove Theorem 2.2, which we restate here.

Theorem 2.2 (RWM Online-to-batch Learning w.r.t Seen Interventions)**.** *Suppose Algorithm 1 is run for T rounds, and suppose that at each time step $t \in [T]$, the intervention target I_t is sampled independently from a fixed but unknown distribution μ on a finite set of interventions $\mathcal{I}$, and subsequently the sample $x^{(t)}$ is sampled from $P^*_{\mathrm{do}(I_t)}$. Let D_τ denote the DAG output by sampling τ uniformly at random from $[T]$ and choosing the round τ prediction of Algorithm 1.*

Then, if $T \geqslant \widetilde{\Omega}\left(\frac{n^3 d}{\varepsilon^2}\right)$, then we have

$$\mathbb{E}\left[\sum_{I\in\mathcal{I}} \mu(I)\, d_{\mathrm{KL}}\left(P^*_{\mathrm{do}(I)} \,\middle\|\, \widehat{P}_{D_\tau,I}\right)\right] \leqslant \min_{D\in\mathsf{EXT}_d(C)} \sum_{I\in\mathcal{I}} \mu(I)\, d_{\mathrm{KL}}\left(P^*_{\mathrm{do}(I)} \,\middle\|\, \widehat{P}_{D,I}\right) + \varepsilon, \tag{5}$$

where the outer expectations are taken over the random draws of the interventions $I_{1:T}$, the samples $x^{(1:T)}$, and the algorithm's internal randomization for the predictions $D_{1:T}$ as well as the choice τ.

The proof of this theorem again relies on the following lemma, whose proof is similar to that of Lemma B.1. The proof of Theorem 2.2 from Lemma C.1 is exactly analogous to the proof of Theorem 2.1 in the previous section, using the bound $T \geqslant \frac{2n^3 d \log^3\left(\frac{4nk}{\varepsilon}\right)}{\varepsilon^2} \in \widetilde{\Omega}\left(\frac{n^3 d}{\varepsilon^2}\right)$ to get the expected KL error $\leqslant \varepsilon$.

Lemma C.1 (Online-to-batch Learning w.r.t seen interventions)**.** *Suppose the intervention targets $I_1, \dots, I_T$ are drawn i.i.d. from a fixed but unknown distribution μ over a finite set of targets $\mathcal{I}$, and we have an online learning algorithm $\mathcal{A}$ that predicts DAGs $D_1, \dots, D_T \in \mathcal{C}$ for a class of DAGs $\mathcal{C}$ under the loss given by Eq.* (2)*. Let D_τ be the DAG output by the algorithm in round τ, where $\tau \sim \mathsf{Unif}([T])$. Then*

$$\mathbb{E}\left[\sum_{I\in\mathcal{I}} \mu(I)\, d_{\mathrm{KL}}\left(P^*_{\mathrm{do}(I)} \,\middle\|\, \widehat{P}_{D_\tau,I}\right)\right] \leqslant \frac{1}{T}\,\mathbb{E}\left[\mathrm{Regret}_T(\mathcal{A};\mathcal{C})\right] + \min_{D\in\mathcal{C}} \sum_{I\in\mathcal{I}} \mu(I)\, d_{\mathrm{KL}}\left(P^*_{\mathrm{do}(I)} \,\middle\|\, \widehat{P}_{D,I}\right) \tag{13}$$

where the outer expectations are taken over the random draws of the interventions $I_{1:T}$, the samples $x^{(1:T)}$, and the algorithm's internal randomization for predictions $D_{1:T}$ as well as the choice τ.

Proof. From the definition of regret, taking the expectation over all random variables yields:

$$\frac{1}{T}\,\mathbb{E}\left[\mathrm{Regret}_T(\mathcal{A};\mathcal{C})\right] = \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\ell_t(D_t)\right] - \frac{1}{T}\,\mathbb{E}\left[\min_{D\in\mathcal{C}}\sum_{t=1}^{T}\ell_t(D)\right]$$

By Jensen's inequality and the concavity of the minimum operator, we can push the expectation inside the minimum for the baseline term:

$$\frac{1}{T}\,\mathbb{E}\left[\mathrm{Regret}_T(\mathcal{A};\mathcal{C})\right] \geqslant \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\ell_t(D_t)\right] - \min_{D\in\mathcal{C}}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\ell_t(D)\right]$$

Now, we analyze the expected loss of the algorithm's prediction at step t. Crucially, the model D_t is chosen based solely on the history up to time $t-1$. Because I_t and $x^{(t)}$ are drawn i.i.d. from μ and $P^*_{\mathrm{do}(I_t)}$ respectively, they are conditionally independent of D_t. Isolating the current step:

$$\begin{aligned}
\mathbb{E}\left[\ell_t(D_t)\right] &= \mathop{\mathbb{E}}_{I_{1:t-1},x^{(1:t-1)}}\left[\mathop{\mathbb{E}}_{I_t\sim\mu,\, x^{(t)}\sim P^*_{\mathrm{do}(I_t)}}\left[-\log \widehat{P}_{D_t,I_t}(x^{(t)}) \,\middle|\, I_{1:t-1}, x^{(1:t-1)}\right]\right] \\
&= \mathop{\mathbb{E}}_{I_{1:t-1},x^{(1:t-1)}}\left[\sum_{I\in\mathcal{I}}\mu(I)\,\mathrm{H}\left(P^*_{\mathrm{do}(I)}, \widehat{P}_{D_t,I}\right)\right] \\
&= \mathbb{E}\left[\sum_{I\in\mathcal{I}}\mu(I)\,\mathrm{H}\left(P^*_{\mathrm{do}(I)}, \widehat{P}_{D_t,I}\right)\right]
\end{aligned}$$

where $\mathrm{H}(P,Q)$ is the cross-entropy. By substituting this back into the regret bound and applying the exact same logic to the fixed baseline D (which trivially does not depend on the history), we obtain:

$$\frac{1}{T}\,\mathbb{E}\left[\mathrm{Regret}_T(\mathcal{A};\mathcal{C})\right] \geqslant \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\sum_{I\in\mathcal{I}}\mu(I)\,\mathrm{H}\left(P^*_{\mathrm{do}(I)}, \widehat{P}_{D_t,I}\right)\right] - \min_{D\in\mathcal{C}}\sum_{I\in\mathcal{I}}\mu(I)\,\mathrm{H}\left(P^*_{\mathrm{do}(I)}, \widehat{P}_{D,I}\right)$$

Because the algorithm outputs D_τ by drawing τ uniformly at random from $\{1,\dots,T\}$, the average over t inside the expectation is exactly equivalent to taking the expectation over τ. Letting the outer expectation absorb the expectation over the random variable τ:

$$\frac{1}{T}\,\mathbb{E}\left[\mathrm{Regret}_T(\mathcal{A};\mathcal{C})\right] \geqslant \mathbb{E}\left[\sum_{I\in\mathcal{I}}\mu(I)\,\mathrm{H}\left(P^*_{\mathrm{do}(I)}, \widehat{P}_{D_\tau,I}\right)\right] - \min_{D\in\mathcal{C}}\sum_{I\in\mathcal{I}}\mu(I)\,\mathrm{H}\left(P^*_{\mathrm{do}(I)}, \widehat{P}_{D,I}\right)$$

Finally, subtracting the true entropy $\sum_{I\in\mathcal{I}}\mu(I)\mathrm{H}(P^*_{\mathrm{do}(I)})$ from both the algorithm's cross-entropy and the benchmark's cross-entropy converts them into KL-divergences, completing the proof:

$$\frac{1}{T}\,\mathbb{E}\left[\mathrm{Regret}_T(\mathcal{A};\mathcal{C})\right] \geqslant \mathbb{E}\left[\sum_{I\in\mathcal{I}}\mu(I)\, d_{\mathrm{KL}}\left(P^*_{\mathrm{do}(I)} \,\middle\|\, \widehat{P}_{D_\tau,I}\right)\right] - \min_{D\in\mathcal{C}}\sum_{I\in\mathcal{I}}\mu(I)\, d_{\mathrm{KL}}\left(P^*_{\mathrm{do}(I)} \,\middle\|\, \widehat{P}_{D,I}\right)$$

□

C.2 Generalization to Unseen Interventions

A central goal of our online causal discovery framework is to learn a model capable of predicting the effects of entirely novel, unseen interventions. In this section, we provide strict, finite-sample out-of-distribution (OOD) generalization guarantees by combining our regret bounds with the structural identifiability of causal graphs.

We begin by formally defining the conditions required to uniquely identify the true causal graph G^*. It is well known that purely observational data can only identify a graph up to its Markov Equivalence Class, represented by a Completed Partially Directed Acyclic Graph (CPDAG) [Andersson et al., 1997]. To orient the remaining undirected edges in the CPDAG, interventional data is required. Following Choo et al. [2022], who build upon the foundational graph-theoretic conditions established by Hyttinen et al. [2013] and Shanmugam et al. [2015], we define a verifying set as follows:

Definition C.2 (Verifying Set)**.** Let G^* be a causal DAG and let its CPDAG represent its observational Markov Equivalence Class. A collection of interventional targets $\mathcal{V}_{\text{struct}}$ is a *verifying set* for G^* if, for every undirected edge $\{u, v\}$ in the CPDAG, there exists an intervention $I \in \mathcal{V}_{\text{struct}}$ such that I contains exactly one of u or v.

Any interventional dataset whose support contains a verifying set is sufficient to uniquely identify G^*. To bound the finite-sample generalization error, we must account for the probability that the algorithm outputs an incorrect graph structure. We define the *minimum structural margin* $\Delta > 0$ as the smallest possible error any incorrect graph $D \in \mathcal{C} \backslash \{G^*\}$ makes on the verifying set:

$$\Delta \triangleq \min_{D \neq G^*} \max_{I \in \mathcal{V}_{\text{struct}}} d_{\text{KL}} \left(P^*_{\text{do}(I)} \,\middle\|\, \widehat{P}_{D,I} \right) > 0 \tag{14}$$

Furthermore, to quantify the difficulty of transferring predictions to a completely novel regime, let I' be an unseen target intervention. We define the *maximum covariate shift constant* $C_{\text{shift}}(I')$ as the worst-case density ratio of the parent configurations, evaluated strictly over the marginal support of the parents under the unseen regime relative to the observational regime:

$$C_{\text{shift}}(I') \triangleq \max_{v \notin I'} \sup_{pa \in \text{supp}\left(P^*_{\text{do}(I')}(\mathsf{Pa}^*(v))\right)} \frac{P^*_{\text{do}(I')}(\mathsf{Pa}^*(v) = pa)}{P^*(\mathsf{Pa}^*(v) = pa)} < \infty \tag{15}$$

With these definitions established, we can state and prove our generalization theorem for Algorithm 1.

Theorem 2.3. *Assume the true causal graph $G^* \in \mathcal{C}$, and that the training distribution μ has support over both the empty set $\varnothing$ and a verifying set $\mathcal{V}_{\text{struct}}$ for G^*. Let $\mu_{\min} \triangleq \min_{I \in \mathcal{V}_{\text{struct}} \cup \{\varnothing\}} \mu(I)$. Run Algorithm 1 and output D_τ by sampling $\tau \sim [T]$ uniformly, in the same setup as Theorem 2.2. Then, for any unseen target intervention I' with a finite covariate shift constant $C_{\text{shift}}(I')$, we have*

$$\mathbb{E}\left[d_{\text{KL}} \left(P^*_{\text{do}(I')} \,\middle\|\, \widehat{P}_{D_\tau, I'} \right) \right] \leqslant \varepsilon, \tag{6}$$

provided the number of rounds $T \geqslant \mathsf{poly}\left(n, \log k, \frac{1}{\varepsilon}, \frac{1}{\Delta}, C_{\text{shift}}(I'), \frac{1}{\mu_{\min}}\right)$.

Proof. The proof involves bounding (see Theorem C.3) the expected KL loss of D_τ by $\left(C_{\text{shift}}(I') + \frac{M}{\Delta}\right) \frac{B_\tau}{\mu_{\min}}$, where M is the largest possible KL divergence $d_{\text{KL}} \left(P^*_{\text{do}(I)} \,\middle\|\, \widehat{P}_{D,I} \right)$ for any $\widehat{P}_{D,I}$ and B_τ is the expected average regret bound for the the online prediction algorithm.

Since each $\widehat{P}_{D,I}$ will satisfy $-\log \widehat{P}_{D,I}(x) \leqslant n \log \frac{4nk^2}{\varepsilon}$ for all x due to the construction of $\widehat{P}$ (Lemma B.6), we will get the theorem by using the RWM regret bounds for B_τ and taking $M = n \log \frac{4nk^2}{\varepsilon}$.

To bound B_τ, we can use the RWM regret bounds (see Lemma A.22) with appropriate parameters $N = (n!)^{d+1}$ and $L_{\max} = n \log \left(\frac{4nk^2}{\varepsilon} \right)$ as in the proof of Theorem 2.1 to get:

$$\frac{1}{T} \mathbb{E}[\mathsf{Reg}_T(\mathsf{RWM}; \mathsf{EXT}_d(C))] \leqslant n \log \left(\frac{4nk^2}{\varepsilon} \right) \sqrt{\frac{(d+1) n \log n}{2T}} \triangleq B_\tau.$$

Thus, when

$$T \geqslant \left(C_{\text{shift}}(I') + \frac{M}{\Delta}\right)^2 \frac{2(d+1)n^2 \log^3\left(\frac{4nk^2}{\varepsilon}\right)}{\mu_{\min}^2 \varepsilon^2},$$

we get the expected KL loss $\leqslant \left(C_{\text{shift}}(I') + \frac{M}{\Delta}\right) \frac{B_\tau}{\mu_{\min}}$ (see Theorem C.3) $\leqslant \varepsilon$. □

The general theorem for bounding the out-of-distribution generalization error in KL is as follows:

Theorem C.3 (OOD Generalization)**.** *Assume the true causal graph $G^* \in \mathcal{C}$, let $B_\tau \triangleq \frac{1}{T}\mathbb{E}\left[\text{Regret}_T(\mathcal{A}; \mathcal{C})\right]$ denote the expected average regret, and let M be the maximum bounded KL-divergence for any prediction in the hypothesis class. Assume the training distribution μ has support over both the empty set $\varnothing$ and a verifying set $\mathcal{V}_{\text{struct}}$ for G^*, and let $\mu_{\min} \triangleq \min_{I \in \mathcal{V}_{\text{struct}} \cup \{\varnothing\}} \mu(I)$. Then, for any unseen target intervention I' with a finite covariate shift constant $C_{\text{shift}}(I')$ and any finite horizon T, the expected KL-divergence is bounded by:*

$$\mathbb{E}\left[d_{\text{KL}}\left(P^*_{\text{do}(I')} \,\middle\|\, \widehat{P}_{D_\tau, I'}\right)\right] \leqslant \left(C_{\text{shift}}(I') + \frac{M}{\Delta}\right) \frac{B_\tau}{\mu_{\min}} \tag{16}$$

Proof. For notational brevity throughout this proof, we let $d_{\text{KL}}(I)$ denote the KL-divergence $d_{\text{KL}}\left(P^*_{\text{do}(I)} \,\middle\|\, \widehat{P}_{D_\tau, I}\right)$.

We partition the expected unseen error into two disjoint events: the algorithm successfully recovering the true structure ($D_\tau = G^*$), and the algorithm failing ($D_\tau \neq G^*$).

$$\mathbb{E}\left[d_{\text{KL}}(I')\right] = \mathbb{E}\left[d_{\text{KL}}(I') \cdot \mathbb{I}\{D_\tau = G^*\}\right] + \mathbb{E}\left[d_{\text{KL}}(I') \cdot \mathbb{I}\{D_\tau \neq G^*\}\right]$$

Bounding the Probability of Structural Failure. Because $\mathcal{V}_{\text{struct}}$ is a verifying set, any graph $D_\tau \neq G^*$ must suffer a KL-divergence of at least Δ on at least one intervention $I \in \mathcal{V}_{\text{struct}}$. Therefore, the weighted sum of errors across the verifying set is bounded below by $\mu_{\min}\Delta$ whenever $D_\tau \neq G^*$. By Lemma 1, the expectation of this sum is bounded by B_τ:

$$\mathbb{E}\left[\sum_{I \in \mathcal{V}_{\text{struct}}} \mu(I) d_{\text{KL}}(I)\right] \geqslant \mathbb{E}\left[\mu_{\min}\Delta \cdot \mathbb{I}\{D_\tau \neq G^*\}\right] = \mu_{\min}\Delta \cdot P(D_\tau \neq G^*)$$

Because the total error over μ is bounded by B_τ, we apply Markov's inequality to obtain:

$$P(D_\tau \neq G^*) \leqslant \frac{B_\tau}{\mu_{\min}\Delta}$$

Thus, the failure case is bounded by the maximum divergence M times this probability:

$$\mathbb{E}\left[d_{\text{KL}}(I') \cdot \mathbb{I}\{D_\tau \neq G^*\}\right] \leqslant M \frac{B_\tau}{\mu_{\min}\Delta}$$

Bounding the Success Case via Covariate Shift. When $D_\tau = G^*$, $\mathsf{Pa}_{D_\tau}(v) = \mathsf{Pa}^*(v)$. By the chain rule of KL-divergence for Bayesian networks, the divergence under the unseen regime decomposes into the sum of local mechanisms for the un-intervened nodes. We change the measure from the unseen distribution to the purely observational distribution P^* using $C_{\text{shift}}(I')$:

$$\begin{aligned}
\mathbb{E}\left[d_{\text{KL}}(I') \mid D_\tau = G^*\right] &= \sum_{v \notin I'} \mathop{\mathbb{E}}_{\mathsf{Pa}^*(v) \sim P^*_{\text{do}(I')}} \left[d_{\text{KL}}\left(P^*(v \mid \mathsf{Pa}^*(v)) \,\middle\|\, \widehat{P}_{D_\tau}(v \mid \mathsf{Pa}^*(v))\right)\right] \\
&\leqslant C_{\text{shift}}(I') \sum_{v \notin I'} \mathop{\mathbb{E}}_{\mathsf{Pa}^*(v) \sim P^*} \left[d_{\text{KL}}\left(P^*(v \mid \mathsf{Pa}^*(v)) \,\middle\|\, \widehat{P}_{D_\tau}(v \mid \mathsf{Pa}^*(v))\right)\right]
\end{aligned}$$

Notice that the right-hand summation is strictly bounded by the total graph KL-divergence under the observational regime $\varnothing$ (which sums over all nodes $v \in V$). Because all local KL-divergences are non-negative:

$$\sum_{v \notin I'} \cdots \leqslant d_{\text{KL}}\left(P^* \,\middle\|\, \widehat{P}_{D_\tau, \varnothing}\right)$$

Taking the expectation and bounding the observational error by the average regret yields:

$$\mathbb{E}\left[d_{\text{KL}}(I') \cdot \mathbb{I}\{D_\tau = G^*\}\right] \leqslant C_{\text{shift}}(I')\,\mathbb{E}\left[d_{\text{KL}}(\varnothing)\right] \leqslant C_{\text{shift}}(I') \frac{B_\tau}{\mu(\varnothing)} \leqslant C_{\text{shift}}(I') \frac{B_\tau}{\mu_{\min}}$$

Combining the Bounds. Summing the bounds for both events yields the final result:

$$\mathbb{E}\left[d_{\mathrm{KL}}\left(P^{*}_{\mathrm{do}(I')} \,\middle\|\, \widehat{P}_{D_\tau, I'}\right)\right] \leqslant \left(C_{\mathrm{shift}}(I') + \frac{M}{\Delta}\right) \frac{B_\tau}{\mu_{\min}}$$

which completes the proof. □

D Weighted Counting and Sampling of AMOs in a CPDAG

Let C be a CPDAG on $[n]$ vertices. The set of consistent extensions of C is denoted $\mathsf{EXT}(C)$, where each $D \in \mathsf{EXT}(C)$ is a fully Directed Acyclic Graph (DAG) that perfectly represents a causal structure in the Markov Equivalence Class of C. The set of consistent extensions of C with indegree $\leqslant d$ is denoted by $\mathsf{EXT}_d(C)$.

We consider a general score-decomposable weight function over the DAGs in $\mathsf{EXT}_d(C)$:

$$W(D) = \prod_{i=1}^{n} \phi_i(i, \mathsf{Pa}_D(i)) \tag{17}$$

where $\phi_i(i, R)$ is an arbitrary non-negative *local weight* for node i given parent set R. This abstraction allows our algorithm to have greater generality: ϕ_i could represent the cumulative exponentiated log-loss for online learning, or standard Bayesian network scoring criteria (e.g., BIC scores) for offline maximum-likelihood structure learning.

The central computational problem we consider in this section is to compute the partition function (the sum of weights of all consistent extensions with indegree $\leqslant d$ for a given parameter d):

$$Z(C) \triangleq \sum_{D \in \mathsf{EXT}_d(C)} W(D) \tag{18}$$

After having done this, we show how to sample a DAG $D \in \mathsf{EXT}_d(C)$ with probability proportional to $W(D)$. As shown in Section 2, this is sufficient to obtain our main result.

We assume access to an oracle $\mathsf{WO}(v, R)$ that efficiently evaluates the local weight function $\phi_v(v, R)$ for any node v and chosen parent set R in $O(1)$ time.

Definition D.1 (Weight Oracle)**.** A *weight oracle* $\mathsf{WO}(v, R)$, given a node $v \in [n]$, and a choice for the parent set $R \subseteq \mathsf{Nbr}_C(v)$ of v, returns $\phi_v(v, R)$. We assume all other informations and parameters required to compute $\phi_v(v, R)$ are stored internally inside the oracle and has only polynomial size.

While we are invoking the weight oracle WO, we will append additional information in our invocations to WO. For clarity of presentation, we are choosing this presentation of WO.

D.1 Computing the Partition Function

The undirected edges of a CPDAG C form a set of disjoint Undirected Chordal Chain Components, denoted $\mathsf{UCCG}(G)$. As established by Andersson et al. [1997], orienting the edges within one chain component into an Acyclic Moral Orientation (AMO) is independent of the orientations chosen for any other component. Any combination of local AMOs across all UCCGs yields a valid global extension $D \in \mathsf{EXT}(C)$ without introducing directed cycles or new v-structures.

Because the weight function factorizes node-wise, this structural independence implies that the partition function factors across the chain components. For a component $G \in \mathsf{UCCG}(C)$ within the ambient partially directed graph C, a local orientation $O \in \mathsf{AMO}(G)$ forms a directed subgraph $G \oplus O$. The conditional weight of this oriented subgraph given C is:

$$W(G \oplus O \mid C) \triangleq \prod_{v \in V(G)} \phi_v(v, \mathsf{Pa}_{G \oplus O}(v) \cup \mathsf{Pa}_C(v)). \tag{19}$$

The local partition function for the component G conditioned on C is the sum of weights over its valid AMOs:

$$Z(G \mid C) \triangleq \sum_{O \in \mathsf{AMO}(G)} W(G \oplus O \mid C). \tag{20}$$

Because the UCCGs partition the entire vertex set of C, the total partition function for the CPDAG simplifies to the product of the local partition functions:

$$Z(C) = w_{\text{dir}}(C) \prod_{G \in \mathsf{UCCG}(C)} Z(G \mid C). \tag{21}$$

where, to define $w_{\text{dir}}(C)$ we define $V_{\text{dir}} \subseteq V$ as the set of all vertices with no adjacent undirected edges in C, and $w_{\text{dir}}(C) = \prod_{v \in V_{\text{dir}}} \phi_v(v, \mathsf{Pa}_C(v))$.

D.1.1 The Weighted Counting Recursion

Let G be an undirected chordal graph. We denote its set of maximal cliques by $\Pi(G)$ and its set of minimal separators by $\Delta(G)$. For a chosen root block $S \in \Delta(G) \cup \Pi(G)$, we define $C_G(S)$ as the partially directed graph obtained by orienting the edges of G using the Maximum Label Search (MLS) algorithm initialized at S.

In Theorem 17 of Wienöbst et al. [2023], the authors proved that MLS identifies exactly the uniformly directed edges present in all AMOs of G where S acts as a topological prefix, and $C_G(S)$ can be computed in linear time $O(|V| + |E|)$. Furthermore, Corollary 18 of Wienöbst et al. [2023] establishes that the unoriented edges in $C_G(S)$ form disjoint undirected chordal graphs, denoted $\mathsf{UCCG}(C_G(S))$. Any combination of AMOs on these components extends the partial orientation to a valid AMO of the entire graph H.

Main recursion. Building on these properties, we compute $Z(G \mid C)$ by partitioning the space of AMOs based on their unique root block S. To avoid overcounting topologies that could be generated by a smaller minimal separator, we restrict the internal permutations of S to be prefix-free with respect only to the minimal separators that are strict subsets of S. We denote this subset as:

$$\Delta(S, G) \triangleq \{S' \in \Delta(G) \mid S' \subset S\} \tag{22}$$

For a fixed S, each prefix-free permutation $\sigma \in \mathsf{PFP}(S, \Delta(S, G))$ induces a unique acyclic orientation of the subgraph $G[S]$, denoted by $\mathcal{O}_\sigma$. Since the MLS algorithm guarantees all directed edges between S and the remaining components point away from S, the parent sets for nodes within S depend exclusively on this internal orientation $\mathcal{O}_\sigma$ and the ambient graph C. This establishes the exact weighted recurrence:

$$Z(G \mid C) = \sum_{S \in \Delta(G) \cup \Pi(G)} Z_{root}(S \mid C) \prod_{G_i \in \mathsf{UCCG}(C_G(S))} Z(G_i \mid C \oplus C_G(S)) \tag{23}$$

where the partition function of the root block sums the weights of all valid internal permutations:

$$Z_{root}(S \mid C) \triangleq \sum_{\sigma \in \mathsf{PFP}(S, \Delta(S,G))} W(G[S] \oplus \mathcal{O}_\sigma \mid C) \tag{24}$$

Lemma D.2 (Parent-Set Preservation)**.** *Let G be an undirected chordal graph. Suppose we choose a root clique $S \in \Delta(G) \cup \Pi(H)$ and a prefix-free permutation $\sigma \in PFP(S, \Delta(G))$. Let $C_G(S)$ denote the partially directed graph obtained by running the Maximum Label Search (MLS) algorithm starting with S, yielding undirected connected chordal components $G_1, \ldots, G_k$.*

For any component G_j and any vertex $v \in V(G_j)$, let $\mathsf{Pa}(v)$ denote the parent set of v in any full Acyclic Moral Orientation (AMO) of G that extends the partial orientation (S, σ). Then $\mathsf{Pa}(v)$ partitions exactly as:

$$\mathsf{Pa}(v) = \mathsf{Pa}_{ext}(v) \cup \mathsf{Pa}_{int}(v) \tag{25}$$

where:

1. *$\mathsf{Pa}_{ext}(v) \subseteq V(G) \backslash V(G_j)$ is completely fixed by the directed edges in $C_G(S)$, and is identical for all $v \in V(G_j)$.*
2. *$\mathsf{Pa}_{int}(v) \subseteq V(G_j)$ consists solely of vertices inside G_j.*

Proof. By the definition of the connected components $G_1, \ldots, G_k$, there are no unoriented edges between G_j and any other component G_ℓ, nor between G_j and the root block S.

The set of edges between S and $V(G)\backslash S$ depends only on the choice of the clique S and is completely independent of the specific permutation σ chosen for S. Thus, the boundary edges are fixed before any internal orientations of G_j are considered (e.g., Lemma 11 in Wienöbst et al. [2023]).

Furthermore, during the MLS traversal, the set of previously visited neighbors of a vertex v exactly forms its parent set from outside the component. This implies that all edges crossing the boundary into G_j strictly act as incoming parent edges to the vertices in G_j. Consequently, no vertex $u \notin V(G_j)$ can ever act as a child of $v \in V(G_j)$ (e.g., Corollary 18 in Wienöbst et al. [2023]).

Finally, For any adjacent vertices a, b within the same undirected component G_j, their sets of previously visited neighbors are identical. Since G_j is a connected component, applying this property transitively implies that the external parent set $\mathsf{Pa}_{ext}(v)$ is uniform for all vertices $v \in V(G_j)$ (e.g., Corollary 18 in Wienöbst et al. [2023]).

Therefore, the external parent dependencies of G_j are completely resolved, uniform, and fixed by $C_G(S)$. Any subsequent orientation choices made strictly within G_j to form $\mathsf{Pa}_{int}(v)$ cannot alter $\mathsf{Pa}_{ext}(v)$ nor create new v-structures with the outside graph, allowing the local weights to safely factorize across components. □

Lemma D.2 (Parent-Set Preservation) guarantees that the external parent set for any component H_i is completely uniform and fixed by $C \oplus C_G(S)$. Therefore, the orientation choices within any component G_i do not affect the local weights of nodes in S or any other component G_j. This structural independence ensures that the weight of the global AMO factorizes exactly into the product of the weights of the root block and the subcomponents, justifying the weight-preserving nature of the bijection. Equation (23) is implied by the following bijection:

Lemma D.3. *Let H be an undirected chordal graph. There is a weight-preserving bijection between $\mathsf{AMO}(H)$ and the set of tuples:*

$$\left\{\left(\mathcal{O}_\sigma, E_{\mathrm{MLS}}(S), (\mathsf{AMO}(G_i))_{G_i \in \mathsf{UCCG}(C_G(S))}\right) : S \in \Delta(G) \cup \Pi(G), \sigma \in \mathsf{PFP}(S, \Delta(S,G))\right\}$$

where $\Delta(S,G) = \{S' \in \Delta(G) \mid S' \subset S\}$.

Proof. We prove this lemma by showing injectivity and surjectivity below.

Surjectivity: Combining an internal AMO of $G[S]$ induced by $\sigma \in \mathsf{PFP}(S, \Delta(S,G))$, the MLS-oriented edges $E_{\mathrm{MLS}}(S)$, and valid AMOs of the subcomponents $G_i \in \mathsf{UCCG}(C_G(S))$ yields a valid AMO of G. First, $\mathcal{O}_\sigma$ is acyclic by construction and moral because S is a subset of a clique. Second, the MLS algorithm inherently orients all edges between S and $V\backslash S$ outwards from S. These edges cannot form v-structures with nodes in $V\backslash S$ (because the parents residing in S form a clique and are therefore adjacent). By Theorem 17 of Wienöbst et al. [2023], the edges in $E_{\mathrm{MLS}}(S)$ are exactly the uniformly-directed edges in all AMOs of G prefixed by S, guaranteeing no new v-structures are formed.

Injectivity: Lemma 19 of Wienöbst et al. [2023] establishes that for any $M \in \mathsf{AMO}(G)$, there exists a unique set $S \in \Delta(G) \cup \Pi(G)$ that acts as the smallest common prefix among all topological orderings of M. Given this unique S, there is a corresponding unique prefix-free permutation $\sigma \in \mathsf{PFP}(S, \Delta(S,G))$ dictating the orientation of $G[S]$. The restriction to $\Delta(S,G)$ ensures that no ordering is counted under S if it could have been generated by a smaller minimal separator $S' \subset S$, preventing combinatorial overcounting. Removing S and the fixed edges $E_{\mathrm{MLS}}(S)$ leaves unique, independent AMOs on the remaining components $G_1, \ldots, G_k$.

For the weighted setting, because all edges between S and G_i point strictly toward G_i, the parent sets for nodes in S are bounded entirely within S and the ambient graph C. The parent dependencies factorize across the components, justifying the multiplicative step in the recurrence. □

D.1.2 Putting everything together

Evaluating the root partition function $Z_{root}(S \mid C)$ in (24) by explicitly summing over all prefix-free permutations requires $O(|S|!)$ operations. We can optimize this to $O(|S|2^{|S|})$ by adapting subset-lattice dynamic programming. While originating as a solution to the Traveling Salesperson Problem [Held and Karp, 1962], this subset-based DP framework was applied to Bayesian network structure learning by Koivisto and Sood [2004] and Silander and Myllymäki [2006].

Let $Z_{HK}(U)$ be the sum of weights of all valid topological orderings for a subset of nodes $U \subseteq S$. We build the subsets progressively. If a subset U exactly matches a forbidden separator $S' \in \Delta(S, G)$, we set $Z_{HK}(U) = 0$, effectively pruning all orderings that contain S' as a prefix. Otherwise, we sum over all possible choices for the last node $v \in U$:

$$Z_{HK}(U) = \sum_{v \in U} Z_{HK}(U \backslash \{v\}) \cdot \phi_v(v, (U \backslash \{v\}) \cup \mathsf{Pa}_C(v)) \tag{26}$$

Setting $Z_{root}(S \mid C) = Z_{HK}(S)$ establishes the exact weighted recurrence for the chain component:

$$Z(G \mid C) = \sum_{S \in \Delta(G) \cup \Pi(G)} Z_{HK}(S) \prod_{G_i \in \mathsf{UCCG}(C_G(S))} Z(G_i \mid C \oplus C_G(S)) \tag{27}$$

The recurrence in (27) is implemented in Algorithm 2.

Finally, having computed this local partition function $Z(G \mid C)$ for each top-level component $G \in \mathsf{UCCG}(C)$, the global partition function for the entire CPDAG is obtained by multiplying these independent results together:

$$Z(C) = \prod_{G \in \mathsf{UCCG}(C)} Z(G \mid C).$$

Theorem 3.1 (Computational Complexity of $Z(G)$)**.** *Let C be a CPDAG with n vertices, m edges, and undirected chordal chain components having a maximum treewidth* tw*. Assuming an $O(1)$ oracle* WO *for evaluating local weights, and utilizing dynamic programming memoization, Algorithm 2 computes the total partition function $Z(C)$ in time $\tilde{O}\left(n^2 m + n^2 \cdot 2^{\mathrm{tw}}\right)$.*

Proof of Theorem 3.1. The algorithm operates by recursively decomposing undirected chordal components H. To achieve the stated bound, the algorithm memoizes the return values of $Z(G \mid C)$.

The total number of distinct undirected connected chordal components G generated across all possible recursive branches of the MLS decomposition is bounded by $O(n)$. Therefore, with memoization, the algorithm evaluates at most $O(n)$ distinct subproblems.

For a given component G, finding $\Pi(G)$ and $\Delta(G)$ takes $O(n + m)$ time. Since G is a chordal graph, the number of maximal cliques and minimal separators is strictly bounded by n (Dirac [1961]). Thus, the outer loop considers at most $|\Delta(G) \cup \Pi(G)| \leqslant 2n$ candidate root blocks S.

For each candidate block S:

- Running the MLS algorithm to compute $C_G(S)$ takes linear time $O(n + m)$.
- Computing the root partition function $Z_{HK}(S)$ requires iterating over all subsets $U \subseteq S$. The size of any root block S is bounded by the maximum clique size $\omega = \mathrm{tw} + 1$. The number of subsets is $2^{|S|} \leqslant 2^{\mathrm{tw}+1}$. For each subset, iterating over its elements and making the $O(1)$ oracle call takes $O(|S|)$ time. The Held-Karp step thus takes $O((\mathrm{tw} + 1)2^{\mathrm{tw}+1})$ time.

Aggregating the work inside the loop for a single subproblem yields:

$$O\left(n \cdot \left[(n + m) + (\mathrm{tw} + 1)2^{\mathrm{tw}+1}\right]\right)$$

Multiplying the work per subproblem by the $O(n)$ total memoized subproblems gives the final time complexity:

$$O\left(n^2(n + m) + n^2 \cdot (\mathrm{tw} + 1)2^{\mathrm{tw}+1}\right)$$

□

Remark D.4. Note that if the any valid extension $D \in \mathsf{EXT}(G)$ is constrained to a maximum in-degree d, the treewidth is bounded by $\mathrm{tw} \leqslant d$, ensuring the runtime is bounded by $O(n^2 m + n^2 \cdot d2^d)$. In any valid AMO of a clique of size ω, the topological sink node of that clique will have exactly $\omega - 1$ parents within the clique, where ω is the maximum clique size. However, the maximum clique size in the CPDAG cannot exceed $d + 1$, implying $\mathrm{tw} \leqslant d$.

D.2 Exact Weighted Sampling

Theorem 3.2 (Computational Complexity of Exact Sampling)**.** *Given a CPDAG C with n nodes, m edges, and bounded indegree d, and assuming the partition functions $Z(G \mid C)$ have been computed and memoized via Algorithm 2, there exists an algorithm (Algorithm 3) to sample a random AMO $D \in \mathsf{EXT}_d(C)$ with exact probability $P(D) = \frac{W(D)}{Z(C)}$ in time bounded by $\tilde{O}\left(n^2(m + 2^d\right)$.*

Every DAG $D \in \mathsf{EXT}_d(C)$ can be constructed via a top-down generative process that mirrors the recursive decomposition. At each component G, we first sample a root block S^*. The probability of choosing a specific block S conditioned on the ambient graph C is proportional to the total partition function of all downstream DAGs that share this prefix:

$$P_{trans}(S \mid C) = \frac{Z_{HK}(S) \prod_{i=1}^{k} Z(G_i \mid C \oplus C_G(S))}{Z(G \mid C)} \tag{28}$$

Once a root block S^* is sampled, we sample its internal topological ordering. We trace backward through the Held-Karp subset lattice generated for S^*. Starting with the full set $U = S^*$, the probability of choosing a node $v \in U$ to be the topological sink (the last node) of U is exactly proportional to the weight of the sub-orderings that end with v:

$$P_{node}(v \mid U) = \frac{Z_{HK}(U \backslash \{v\}) \cdot \phi_v(v, (U \backslash \{v\}) \cup \mathsf{Pa}_C(v))}{Z_{HK}(U)} \tag{29}$$

We sample a sink node v^*, direct all remaining edges in U toward v^*, and iteratively repeat the process on $U \backslash \{v^*\}$ until the set is empty. This efficiently reconstructs a valid internal acyclic orientation $\mathcal{O}_{\sigma^*}$ strictly proportional to its weight. Algorithm 3 describes the overall implementation of the sampling procedure.

Proof of Theorem 3.2. The sampling algorithm visits a hierarchy of disjoint chordal components to construct the final DAG. Because each choice of a root block S^* and subsequent MLS orientation strictly reduces the unoriented graph, the total number of subcomponents instantiated and recursively visited to form a single global AMO is strictly bounded by n.

At any visited component G, the algorithm evaluates at most $|\Delta(G) \cup \Pi(G)| \leqslant 2n$ candidate root blocks. For each candidate block S, it looks up the memoized subcomponent partition functions in $O(1)$ time and re-evaluates the Held-Karp subset DP to find $Z_{HK}(S)$ and retain the intermediate subset table. As established in the proof of Algorithm 2 (see Remark D.4), executing the Held-Karp DP for a block bounded by the indegree takes $O((d+1)2^{d+1})$ time. Thus, evaluating the transition probabilities for the current component takes $O(n(n+m) + n(d+1)2^{d+1})$ time.

Once S^* is sampled, backtracing the Held-Karp lattice to find the internal permutation requires iterating over the decreasing subset U. The number of iterations is $|S^*| \leqslant d+1$, and each iteration evaluates at most $|S^*|$ weights. This backtracing takes $O(|S^*|^2) \leqslant O(d^2)$ time, which is strictly dominated by the Held-Karp computation.

Because the maximum number of recursive steps (visited components) is n, multiplying the dominant work per step by n yields a total time complexity of $O(n^2(n+m) + n^2(d+1)2^{d+1})$.

□

E The MLS Algorithm

We use the Maximal Label Search (MLS) algorithm from Berry et al. [2009] in our weighted counting and sampling algorithms. Before presenting the algorithm, we first define the notion of labeling structure.

Definition E.1 (Labeling Structure)**.** A *labeling structure* is a structure $(L, \preceq, l_0, \mathrm{Inc})$, where:

- L is a set (the set of labels),
- $\preceq$ is a partial order on L (which may be total or not, with $\prec$ denoting the corresponding strict order), which will be used to choose a vertex of maximal label,

- l_0 is an element of L, which will be used to initialize the labels,
- Inc is a mapping from $L \times \mathbb{Z}^+$ to L, which will be used to increment a label, and such that the following condition IC (Inclusion Condition) holds: IC : for any subsets I and I' of $\mathbb{Z}_2^+$, if $I \subset I'$ then $\text{lab}_S(I) \prec \text{lab}_S(I')$, where $\text{lab}_S(I) = \text{Inc}(\dots(\text{Inc}(\text{Inc}(l_0, i_1), i_2), \dots), i_k)$, where $I = \{i_1, i_2, \dots, i_k\}$ with $i_1 > i_2 > \dots > i_k$.

We present the pseudocode of the MLS algorithm below for completeness.

Algorithm 5: MLS (Maximal Label Search) algorithm

Input: A graph $G = (V, E)$ and a labeling structure $(L, \preceq, l_0, \text{Inc})$
Output: An ordering α on V

```
Initialize all labels as l_0; G' ← G ;
for i = n to 1 do
    Choose a vertex x of G' of maximal label ;
    α(i) ← x ;
    forall y ∈ N_{G'}(x) do
        label(y) ← Inc(label(y), i) ;
    Remove x from G' ;
```

F Additional Experiments

Code and results The code and detailed results for the experiments can be found on GitHub at https://github.com/philips-george/weighted-online-causal-discovery/.

Sachs protein-signalling experiment We use the CPDAG obtained from the accepted Sachs reference DAG, and evaluate held-out perturbation prediction: for each intervention target, half of its samples are held out for testing, while the training stream includes up to b samples from each intervention condition. The baseline is hill-climbing structure learning from the `bnlearn` Scutari [2010] library, using its intervention-aware modified BDe score (`hc + mbde`). We report held-out negative log-likelihood per scored node, excluding experimentally manipulated nodes, averaged over 5 intervention targets and 5 random splits. RWM substantially improves as interventional samples are added and is competitive with the `bnlearn hc + mbde` baseline, while maintaining a weighted distribution over DAGs instead of returning a single DAG.

Table 2: Sachs protein-signalling experiment results

Samples per intervention target (b)	**RWM mixture**	**bnlearn hc + mbde**
0	1.062 ± 0.110	1.034 ± 0.117
10	0.797 ± 0.057	0.814 ± 0.046
25	0.742 ± 0.049	0.743 ± 0.039
50	0.707 ± 0.044	0.700 ± 0.034
100	0.674 ± 0.037	0.657 ± 0.027

Table 3: Runtime benchmarks

Width (k)	n	**RWM**	**Restricted bnlearn hc**
2	2000	3.90 s	17.38 s
2	5000	11.51 s	369.20 s
3	2000	4.44 s	18.87 s
3	5000	21.75 s	504.16 s
4	2000	7.46 s	20.92 s
4	5000	29.56 s	542.73 s*

Note: Only 1/3 `bnlearn hc` runs completed at $n = 5000, k = 4$; the other two exceeded the timeout of 600 seconds.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract states some learning sample complexities (in concise form) which match the theorems, and the introduction section has the formal theorem statements, problem definitions, etc.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: It is a theoretical paper, and we give the problem statements and theorems precisely including all assumptions.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: All these things are included, with complete proofs in the supplementary appendices.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [N/A]

   Justification: The paper has no experimental results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [N/A]

Justification: The paper has no experimental results.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [N/A]

   Justification: The paper has no experimental results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [N/A]

   Justification: The paper has no experimental results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [N/A]

   Justification: The paper has no experimental results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The authors have reviewed the guidelines. We use no human subjects, no datasets, and the paper is entirely theoretical in nature and only considers abstract problems (no societal impact).

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: The work is entirely theoretical on an abstract problem, and there is no societal impact.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not propose datasets or models, only algorithms.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [N/A]

    Justification: The paper does not use existing assets, except for other papers which have been properly cited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets, other than the algorithms documented in the paper.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not use crowdsourcing/human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper is entirely theoretical and does not require such approvals.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: We have used LLMs only for drafting sections of the paper and have not used LLMs for core methodology and scientific rigor of our results.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.