Applications of Common Entropy for Causal Inference

Murat Kocaoglu
MIT-IBM Watson AI Lab, IBM Research
murat@ibm.com

Sanjay Shakkottai
The University of Texas at Austin
shakkott@austin.utexas.edu

Alexandros G. Dimakis
The University of Texas at Austin
dimakis@austin.utexas.edu

Constantine Caramanis
The University of Texas at Austin
constantine@utexas.edu

Sriram Vishwanath
The University of Texas at Austin
sriram@austin.utexas.edu

Abstract

We study the problem of discovering the simplest latent variable that can make two observed discrete variables conditionally independent. The minimum entropy required for such a latent is known as common entropy in information theory. We extend this notion to Rényi common entropy by minimizing the Rényi entropy of the latent variable. To efficiently compute common entropy, we propose an iterative algorithm that can be used to discover the trade-off between the entropy of the latent variable and the conditional mutual information of the observed variables. We show two applications of common entropy in causal inference: First, under the assumption that there are no low-entropy mediators, it can be used to distinguish causation from spurious correlation among almost all joint distributions on simple causal graphs with two observed variables. Second, common entropy can be used to improve constraint-based methods such as PC or FCI algorithms in the small-sample regime, where these methods are known to struggle. We propose a modification to these constraint-based methods to assess if a separating set found by these algorithms are valid using common entropy. We finally evaluate our algorithms on synthetic and real data to establish their performance.

1 Introduction

Understanding the causal workings of a system from data is essential in many fields of science and engineering. Recently, there has been increasing interest in causal inference in the machine learning (ML) community. While most of ML has traditionally been relying solely on correlations in the data, it is now widely accepted that distinguishing causation from correlation is useful even for simple predictive tasks. This is because causal relations are more robust to the changes in the dataset and can help with generalization, while an ML system relying solely on correlations might suffer when these correlations change in the environment the system is deployed in [12].

A causal graph is a directed acyclic graph that depicts the causal workings of the system under study [38]. Since it indicates the causes of each variable, it can be seen as a qualitative summary of the underlying mechanisms. Learning the causal graph is the first step for most of the causal inference tasks, since inference algorithms rely on the causal structure. Causal graphs can be learned from

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

randomized experiments [17, 14, 44, 28, 27]. In settings where performing experiments are costly or infeasible, one needs to resort to observational methods, i.e., make best use of observational data, potentially under assumptions about the data generating mechanisms.

There is a rich literature on learning causal graphs from observational data [47, 54, 11, 1, 35, 46, 20, 50, 9]. *Score-based* methods optimize a regularized likelihood function in order to discover the causal graph. Under certain assumptions, these methods are consistent; they obtain a causal graph that is in the correct equivalence class for the given data [34, 10]. However, score-based methods are applicable only in the causally sufficient setting, i.e., when there are no latent confounders. A variable is called a latent confounder if it is not observable and causes at least two observed nodes. *Constraint-based* methods directly recover the equivalence class in the form of a mixed graph [1, 34, 47, 35, 55]: They test the conditional independence (CI) constraints in the data and use them to infer as much as possible about the causal graph. Despite being well-established for graphs with or without latents, constraint-based methods are known to work well only with an abundance of data. Early errors in CI statements might lead to drastically different graphs due to the sequential nature of these algorithms. A third class of algorithms can be described as those imposing assumptions in order to identify the graphs which are otherwise not identifiable [20, 46, 39, 21, 26]. Most of this literature focuses on the cornerstone case of two variables X, Y where constraint and score-based approaches are unable to identify if X causes Y or Y causes X, simply because they are indistinguishable without additional assumptions. This literature contains a wide range of assumptions that we summarize in Section 7.2.

Information theory has been shown to provide tools that can be useful for causal discovery [8, 42, 30, 15, 52, 26, 48]. In this work, we explore the uses of *common entropy* for learning causal graphs from observational data. To define common entropy, first consider the following problem: Given the joint probability distribution of two discrete random variables X, Y, we want to construct a third random variable Z such that X and Y are independent conditioned on Z. Without any constraints this can be trivially achieved: Simply picking $Z = X$ or $Z = Y$ ensures that $X \perp\!\!\!\perp Y \,|Z$. However, this trivial solution requires Z to be as *complex* as X or Y. We then ask the following question: *is there a simple Z that makes X, Y conditionally independent?* In this work, we use Rényi entropy of the variable as a notion of its complexity. Then the problem becomes identifying Z with the smallest Rényi entropy such that $X \perp\!\!\!\perp Y \,|Z$. Shannon entropy of this Z is called the *common entropy* of X and Y [31].

We demonstrate two uses of common entropy for causal discovery. The first is in the setting of two observed variables. Suppose we observe two correlated variables X, Y. Figure 1 shows some causal graphs that can induce correlation between X, Y. Note that *Latent Graph* differs from the others in that X does not have any causal effect on Y. Then distinguishing the latent graph from the others is important to understand whether an intervention on one of the variables will cause a change in the other. We show that if the latent confounder is *simple*, i.e., has small entropy then one can distinguish the latent graph from the triangle and direct graphs using common entropy. To identify the latent graph, we assume that the correlation is not induced only by a simple mediator, which eliminates the mediator graph. We show that this is a realistic assumption using simulated and real data.

Second, we show that common entropy can be used to improve constraint-based causal discovery algorithms in the small sample regime. For such algorithms, correctly identifying *separating sets*, i.e., sets of variables that can make each pair conditionally independent, is crucial. Our key observation is that, for a given pair of variables common entropy provides us with an information-theoretic lower bound on the entropy of any separating set. Therefore, it can be used to reject incorrect separating sets. We present our modification on the PC algorithm, which is called the *EntropicPC* algorithm.

To the best of our knowledge, the only result for finding common entropy is given in [31], where they identify its analytical expression for binary variables. They also note that the problem is difficult in the general case. To address this gap, in Section 2 we propose an iterative algorithm to approximate common entropy. We also generalize the notion of common entropy to *Rényi common entropy*.

Our contributions can be summarized as follows:

- In Section 2, we introduce the notion of Rényi common entropy. We propose a practical algorithm for finding common entropy and prove certain guarantees. Readers interested only in the applications of common entropy to causal inference can skip this section.
- In Section 3, under certain assumptions, we show that Rényi$_0$ common entropy can be used to distinguish latent graph from the triangle and direct graphs in Figure 1. We also show this identifiability result via Rényi$_1$ common entropy for binary variables, and propose a

(a) Latent Graph (b) Triangle Graph (c) Direct Graph (d) Mediator Graph

Figure 1: Different graphs that explain correlation between the observed X, Y. Z, M are unobserved.

conjecture for the general case. In Section 5.2, we validate one of our key assumptions in real and synthetic data. In Section 5.3, we validate our conjecture via real and synthetic data.

- In Section 4, we propose *EntropicPC*, a modified version of the PC algorithm that uses common entropy to improve sample efficiency. In Section 5.5, we demonstrate significant performance improvement for EntropicPC compared to the baseline PC algorithm. We also illustrate that EntropicPC discovers some of the edges missed by PC in ADULT data [13].
- In Section 5, in addition to the above, we provide experiments on the performance of our algorithm for finding common entropy, as well as its performance on distinguishing the latent graph from the triangle graph on synthetic data.

Notation: Support of a discrete random variable X is shown as $\mathcal{X}$. $p(.)$ and $q(.)$ are reserved for discrete probability distribution functions (pmfs). $[n] \coloneqq \{1, 2, \ldots, n\}$. $p(Y|x)$ is shorthand for the conditional distribution of Y given $X = x$. Shannon entropy, or entropy in short, is $H(X) = -\sum_x p(x) \log(p(x))$. Rényi entropy of order r is $H_r(X) = \frac{1}{1-r} \log\left(\sum_x p^r(x)\right)$. It can be shown that Rényi entropy of order 1 is identical to Shannon entropy. $D = (\mathcal{V}, \mathcal{E})$ is a directed acyclic graph with vertex set $\mathcal{V}$ and edge set $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. Pa_i represents the set of parents of vertex X_i in the graph and pa_i a specific realization. If D is a causal graph, the joint distribution between the variables (vertices of the graph) factorizes relative to the graph as $p(x_1, x_2, \ldots,) = \prod_i p(x_i|pa_i)$.

2 Rényi Common Entropy

We introduce the notion of Rényi common entropy, which generalizes the common entropy of [31].

Definition 1. *Rényi common entropy of order r or Rényi$_r$ common entropy of two random variables X, Y with probability distribution $p(x, y)$ is shown by $G_r(X, Y)$ and is defined as follows:*

$$\begin{aligned} G_r(X, Y) \coloneqq \min_{q(x,y,z)} \quad & H_r(Z) \\ \textit{s.t.} \quad & I(X; Y|Z) = 0; \sum_z q(x, y, z) = p(x, y), \forall x, y; \sum q(.) = 1; q(.) \geq 0 \end{aligned} \tag{1}$$

Rényi common entropy lower bounds the Rényi entropy of any variable that makes the observed variables conditionally independent. We focus on two special cases: Rényi$_0$ and Rényi$_1$ common entropies. Among all variables Z such that $X \perp\!\!\!\perp Y \,|Z$, Rényi$_0$ common entropy is the logarithm of the minimum number of states of Z and Rényi$_1$ common entropy is its minimum entropy.

In Section 3, we show that Rényi$_0$ common entropy can be used for distinguishing the latent graph from the triangle or direct graphs in Figure 1. Since we expect Rényi$_0$ common entropy to be sensitive to finite-sample noise in practice, we focus on Rényi$_1$ common entropy. Rényi$_1$ common entropy, or simply common entropy, was introduced in [31], where authors derived the analytical expression for two binary variables. They also remark that finding common entropy for non-binary variables is difficult. We propose an iterative update algorithm to approximate common entropy in practice, by assuming that we have access to the joint distribution between X, Y.

LatentSearch: An Algorithm for Calculating Rényi$_1$ Common Entropy

In this section, our objective is to solve a relaxation of the Rényi$_1$ common entropy problem in (1). Instead of enforcing conditional independence as a hard constraint of the optimization problem, we introduce conditional mutual information as a regularizer to the objective function. This allows us to discover a trade-off between two factors, the entropy $H(Z)$ of the third variable and the residual dependency between X, Y after conditioning on Z, measured by $I(X; Y|Z)$. We then have the loss

$$\mathcal{L} = I(X; Y|Z) + \beta H(Z). \tag{2}$$

Algorithm 1 LatentSearch: Iterative Update Algorithm

Input: Supports of $x, y, z, \mathcal{X}, \mathcal{Y}, \mathcal{Z}$, respectively. $\beta \geq 0$ used in (2). Observed joint $p(x, y)$. Initialization $q_1(z|x, y)$. Number of iterations N.
Output: Joint distribution $q(x, y, z)$
for $i \in [N]$ **do**
  Form the joint:
  $q_i(x, y, z) \leftarrow q_i(z|x, y)p(x, y), \forall x, y, z.$
  Calculate:
  $q_i(z|x) \leftarrow \frac{\sum_{y\in\mathcal{Y}} q_i(x,y,z)}{\sum_{y\in\mathcal{Y},z\in\mathcal{Z}} q_i(x,y,z)}, \quad q_i(z|y) \leftarrow \frac{\sum_{x\in\mathcal{X}} q_i(x,y,z)}{\sum_{x\in\mathcal{X},z\in\mathcal{Z}} q_i(x,y,z)}, \quad q_i(z) \leftarrow \sum_{x\in\mathcal{X},y\in\mathcal{Y}} q_i(x, y, z)$
  Update:
  $q_{i+1}(z|x, y) \leftarrow \frac{1}{F(x,y)} \frac{q_i(z|x)q_i(z|y)}{q_i(z)^{1-\beta}}$, where $F(x, y) = \sum_{z\in\mathcal{Z}} \frac{q_i(z|x)q_i(z|y)}{q_i(z)^{1-\beta}}$.
return $q(x, y, z) \coloneqq q_{N+1}(z|x, y)p(x, y)$

Rather than searching over $q(x, y, z)$ and enforcing the constraint $\sum_z q(x, y, z) = p(x, y), \forall x, y$, we can search over $q(z|x, y)$ and set $q(x, y, z) = q(z|x, y)p(x, y)$. Therefore we have $\mathcal{L} = \mathcal{L}(q(z|x, y))$. The support size of Z determines the number of optimization variables. Proposition 5 in [31] shows that without loss of generality, we can assume $|\mathcal{Z}| \leq |\mathcal{X}||\mathcal{Y}|$. In general, $\mathcal{L}$ is neither convex nor concave. Although first order methods (e.g., gradient descent) can be used to find a stationary point, as we empirically observe the convergence is slow and the performance is very sensitive to the step size. To this end, we propose a multiplicative update algorithm *LatentSearch* in Algorithm 1. Given $p(x, y)$, *LatentSearch* starts from a random initialization $q_0(z|x, y)$, and at each step i iteratively updates $q_i(z|x, y)$ to $q_{i+1}(z|x, y)$ to minimize the loss (2). Specifically, in the i^{th} step it marginalizes the joint $q_i(x, y, z)$ to get $q_i(z|x)$, $q_i(z|y)$, and $q_i(z)$, and imposing a scaled product form on these marginals, updates the joint to return $q_{i+1}(x, y, z)$. This decomposition and the update rule are motivated by the partial derivatives associated with the Lagrangian of the loss function (2) (See Section 7.3). More formally, as we show in the following theorem, after convergence *LatentSearch* outputs a stationary point of the loss function. For the proof, please see Sections 7.3, 7.4.

Theorem 1. *The stationary points of LatentSearch are also stationary points of the loss in (2). Moreover, for $\beta = 1$, LatentSearch converges to either a local minimum or a saddle point of (2), unless it is initialized at a local maximum.*

Therefore, if the algorithm converges to a solution, it outputs either a local minimum, local maximum or a saddle point. We observe in our experiments that the algorithm always converges for $\beta \leq 1$.

For each β, *LatentSearch* outputs a distribution $q(.)$ from which $H(Z)$ can be calculated. When using *LatentSearch* to approximate Rényi$_1$ common entropy, we will run it for multiple β values and pick the distribution $q(.)$ with the smallest $H(Z)$ such that $I(X; Y|Z) \leq \theta$ for a practical threshold θ to declare conditional independence. See Figure 3b for a sample output of *LatentSearch* for multiple β values in the $I - H$ plane. See also lines $6 - 7$ of Algorithm 2 for an algorithmic description.

3 Identifying Correlation without Causation via Rényi Common Entropy

Suppose we observe two discrete random variables X, Y to be statistically dependent. Reichenbach's common cause principle states that X and Y are either causally related, or there is a common cause that induces the correlation.[1] If the correlation is *only* due to a common cause, intervening on either variable will have no effect on the other. Therefore, it is important for policy decisions to identify this case of *correlation without causation*. Specifically, we want to distinguish latent graph from the triangle or direct graphs in Figure 1. Since our goal is not to identify the causal direction between X and Y, we use triangle, direct and mediator graphs to refer to either direction. We show that, under certain assumptions, Rényi common entropy can be used to solve this identification problem.

Our key assumption is that the latent confounders, if they exist, have small Rényi entropy. In other words, in Figure 1 $H_r(Z) \leq \theta_r$ for some θ_r. We consider two cases: Rényi$_0$ and Rényi$_1$ entropies. $H_0(Z) \leq \theta_0$ is equivalent to upper bounding the support size of Z. $H_1(Z) \leq \theta_1$ upper bounds the Shannon entropy of Z. In general, $H_1(Z) \leq \theta$ can be seen as a relaxation of $H_0(Z) \leq \theta$ as the latter

[1] We assume no selection bias in this work, which can also induce spurious correlation.

implies the former but not vice verse. Accordingly, we show stronger results for Rényi$_0$, whereas we leave the most general identifiability result of Rényi$_1$ as a conjecture. We also quantify how small the confounder's Rényi entropy should be for identifiability.

Note that bounding the Rényi entropy of the latent confounder in the latent graph bounds the Rényi common entropy of the observed variables. Therefore, in order to distinguish the latent graph from the triangle and direct graphs, we need to obtain lower bounds on the Rényi common entropy of a typical pair X, Y when data is generated from the triangle or direct graphs.

We first establish bounds on the Rényi$_0$ common entropy for the triangle and direct graphs, which hold for almost all parametrizations. To measure the fraction of causal models for which our bound is valid, we use a natural probability measure on the set of joint distributions by sampling each conditional uniformly randomly from the probability simplex:

Definition 2 (Uniform generative model (UGM))**.** *For any causal graph, consider the following generative model for the joint distribution $p(x_1, x_2, \ldots) = \prod_i p(x_i|pa_i)$, where $X_i \in \mathcal{X}_i, \forall i$: For all i and pa_i, let the conditional distribution $p(X_i|pa_i)$ be sampled independently and uniformly randomly from the probability simplex in $|\mathcal{X}_i|$ dimensions.*

The following theorem uses the measure induced by UGM to show that for almost all distributions obtained from the triangle or direct graph, Rényi$_0$ common entropy of the observed variables is large.

Theorem 2. *Consider the random variables X, Y, Z with supports $[m], [n], [k]$, respectively. Let $p(x, y, z)$ be a pmf sampled from the triangle or the direct graphs according to UGM. Then with probability 1, $G_0(X, Y) = \log(\min\{m, n\})$.*

Now consider the latent graph where Z is the true confounder. We clearly have that $G_0(X, Y) \leq H_0(Z)$ since Z indeed makes X, Y conditionally independent. In other words, $G_0(X, Y)$ is upper bounded in the latent graph whereas it is lower bounded in the triangle and the direct graphs by Theorem 2. Therefore, as long as the correlation cannot be explained by the mediator graph, $G_0(X, Y)$ can be used as a parameter to identify the latent graph. In order to formalize this identifiability statement, we need two assumptions with parameters (r, θ):

Assumption 1 (r, θ)**.** *Consider any causal model with observed variables X, Y. Let Z represent the variable that captures all latent confounders between X, Y. Then $H_r(Z) < \theta$.*

Assumption 2 (r, θ)**.** *Consider a causal model where X causes Y. If X causes Y only through a latent mediator Z, i.e., $X \to Z \to Y$, then $H_r(Z) \geq \theta$.*

Assumption 1 states that the collection of latent confounders, represented by Z, has to be "simple", which is quantified by its Rényi entropy. This assumption can also be interpreted as relaxing the causal sufficiency assumption by allowing *weak* confounding. Assumption 2 states that if the correlation is induced only due to a mediator, this mediator cannot have low Rényi entropy. Even though this assumption might seem restrictive, we provide evidence on both real and synthetic data in Section 5.2 to show it indeed often holds in practice. We have the following corollary:

Corollary 1. *Consider the random variables X, Y with supports $[m], [n]$, respectively. For $\theta = \log(\min\{m, n\})$, latent graph can be identified with probability 1 under UGM, Assumption 1,2$(0, \theta)$.*

Corollary 1 indicates that, when the latent confounder has less than $\min\{m, n\}$ number of states, we can infer that the true causal graph is the latent graph from observational data under Assumptions 1 and 2. However, using common entropy, we cannot distinguish triangle graph from the direct graph. Also note that the identifiability result holds for *almost all* parametrizations of these graphs, i.e., the set of parameters where it does not hold has Lebesgue measure zero.

Next, we investigate if Rényi$_1$ common entropy can be used for the same goal. Finding Rényi$_1$ common entropy in general is challenging. For binary X, Y we can use the analytical expression of [31] to show that $G_1(X, Y)$ is almost always larger than $H(Z)$ asymptotically for the triangle graph:

Theorem 3. *Consider the random variables X, Y, Z with supports $[2], [2], [k]$, respectively. Let $p(x, y, z)$ be a pmf sampled from the triangle graph according to UGM except $p(z)$, which can be arbitrary. Then $\lim_{H(Z)\to 0} \mathbb{P}(G_1(X, Y) > H(Z)) = 1$, where $\mathbb{P}$ is the probability measure induced by UGM.*

In Section 7.8, we provide simulations for binary and ternary Z to demonstrate the behavior for small non-zero $H(Z)$. Then, we have the following asymptotic identifiability result using common entropy:

Algorithm 2 InferGraph: Identifying the Latent Graph

Input: k : Support size of Z, $p(x,y)$, $T : I(X;Y|Z)$ threshold, $\{\beta_i\}_{i\in[N]}$, $\theta : H(Z)$ threshold.
Randomly initialize N distributions $q_0^{(i)}(z|x,y), i \in [N]$.
for $i \in [N]$ **do**
    $q^{(i)}(x,y,z) \leftarrow$ *LatentSearch*$(q_0^{(i)}(z|x,y), \beta_i)$.
    Calculate $I^{(i)}(X;Y|Z)$ and $H^{(i)}(Z)$ from $q^{(i)}(x,y,z)$.
$S = \{i : I^{(i)}(X;Y|Z) \leq T\}$.
if $\min(\{H^{(i)}(Z) : i \in S\}) > \theta$ or $S = \emptyset$ **then**
    return Triangle or Direct Graph
else
    return Latent Graph

Algorithm 3 EntropicPC (for $F = False$) and EntropicPC-C (for $F = True$)

Input: CI Oracle $\mathcal{C}$ for $\mathcal{V} = \{X_1, \dots X_n\}$. Common entropy oracle $\mathcal{B}$. Entropy oracle H. Flag F.
Form the complete undirected graph $D = (\mathcal{V}, \mathcal{E})$ on node set $\mathcal{V}$.
$l \leftarrow -1$. $maysep(X_i, X_j) \leftarrow True, \forall i, j$.
while $\exists i, j$ s.t. $(X_i, X_j) \in \mathcal{E}$ **and** $|adj_D(X_i) \backslash \{X_j\}| > l$ **and** $maysep(X_i, X_j) = True$. **do**
    $l \leftarrow l + 1$
    for All i, j s.t. $(X_i, X_j) \in \mathcal{E}$ and $|adj_D(X_i) \backslash \{X_j\}| \geq l$ **do**
        while $(X_i, X_j) \in \mathcal{E}$ and $\exists S \subseteq adj_D(X_i) \backslash \{X_j\}$ s.t. $|S| = l$ **and** $maysep(X_i, X_j) = True$. **do**
            Pick a new $S \subseteq adj_D(X_i) \backslash \{X_j\}$ s.t. $|S| = l$.
            if $\mathcal{C}(X, Y|Z) = True$ **then**
                if $H(S) \geq \mathcal{B}(X_i, X_j)$ **then**
                    $\mathcal{E} \leftarrow \mathcal{E} - \{(X_i, X_j)\}$.
                    $sepset(X_i, X_j) \leftarrow S$.
                else if $F = True$ **then**
                    $maysep(X_i, X_j) \leftarrow False$
                else
                    if $\mathcal{B}(X_i, X_j) \geq 0.8 \min\{H(X_i), H(X_j)\}$ **then**
                        $maysep(X_i, X_j) \leftarrow False$
Orient unshielded colliders according to separating sets $\{sepset(X_i, X_j)\}_{i,j\in[n]}$ [11].
Orient as many of the remaining undirected edges as possible by repeatedly applying the Meek rules [35].
Return: $D = (\mathcal{V}, \mathcal{E})$.

Corollary 2. *For binary X, Y under UGM, Assumption 1,2$(1, \theta)$ if the entropy upper bound θ is known, the fraction of causal models for which latent graph can identified goes to 1 as θ goes to 0.*

For the general case, we conjecture that when the data is sampled from the triangle or the direct graphs, $G_1(X, Y)$ scales with $\min\{H(X), H(Y)\}$.

Conjecture 1. *Consider the random variables X, Y, Z with supports $[m], [n], [k]$, respectively. Let $p(x, y, z)$ be a pmf sampled from the triangle or direct graphs according to UGM except $p(z)$, which can be arbitrary. Then, there exists a constant $\alpha = \Theta(1)$ such that with probability $1 - (\min\{m, n\})^{-c}$ $G_1(X, Y) > \alpha \min\{H(X), H(Y)\}$ for some constant $c = c(\alpha)$.*

According to Conjecture 1, we expect that for most of the parametrizations of the triangle and direct graphs, common entropy of the observed variables should be lower-bounded by the entropies of the observed variables, up to a scaling by a constant. It is easy to see that under assumptions similar to those in Corollaries 1 and 2, Conjecture 1 implies identifiability of the latent graph. In Section 5, we conduct experiments to support the conjecture and identify α. We conclude this section by formalizing how *LatentSearch* can be used in Algorithm 2, under Assumption $1(1, \theta)$, $2(1, \theta)$. Conjecture 1 suggests that, in Algorithm 2, we can set $\theta = \alpha \min\{H(X), H(Y)\}$ for some $\alpha < 1$.

4 Entropic Constraint-Based Causal Discovery

A causal graph imposes certain CI relations in the data. Constraint-based causal discovery methods utilize CI statements to reverse-engineer the underlying causal graph. Consider a causal graph over a set $\mathcal{V}$ of observed variables. Constraint-based methods identify a set $S_{X,Y} \subset \mathcal{V}$ as a *separating set*

for every pair X, Y if the CI statement $X \perp\!\!\!\perp Y \,|S_{X,Y}$ holds in the data. Starting with a complete graph, edges between pairs are removed if they are separable by some set. Separating sets are later used to orient parts of the graph, which is followed by a set of orientation rules [47, 55] .

Despite being grounded theoretically in the large sample limit, in practice these methods require a large number of samples and are very sensitive to noise: An incorrect CI statement early on might lead to a drastically different causal graph at the end due to their sequential nature. Another issue is that the distribution should be faithful to the graph, i.e., any connected pair should be dependent [47, 29].

To help alleviate some of these issues, we propose a simple modification to the existing constraint-based learning algorithms using common entropy. Our key observation is that the common entropy of two variables provide an information-theoretic lower bound on the entropy of *any* separating set. In other words, common entropy provides us with a necessary condition for a set $S_{X,Y}$ to be a valid separating set: $X \perp\!\!\!\perp Y \,|S_{X,Y}$ only if $H(S_{X,Y}) \geq G_1(X, Y)$. Accordingly, we can modify any constraint-based method to ensure this condition. We only lay out our modifications on the PC algorithm. It can be trivially applied to other methods such as modern variants of PC and FCI.

We propose two versions: *EntropicPC* and the conservative version *EntropicPC-C*. In both, $S_{X,Y}$ is accepted only if $H(S_{X,Y}) \geq G_1(X, Y)$. The difference is how they handle pairs X, Y that are deemed CI despite that $H(S_{X,Y}) < G_1(X, Y)$. EntropicPC-C concludes that the data for X, Y is unreliable and simply declares them non-separable by any set. EntropicPC only does this when common entropy is large, i.e., $G_1(X, Y) \geq 0.8 \min\{H(X), H(Y)\}$; otherwise it searches for another set that may satisfy $H(S_{X,Y}) \geq G_1(X, Y)$. 0.8 is chosen based on our experiments in Section 5.

We provide the pseudo-code in Algorithm 3. It is easy to see that both algorithms are sound in the sample limit. The case of $S = \emptyset$ in **line 10** is of special interest. Setting $H(\emptyset) = 0$ is not reasonable with finitely many samples since the common entropy of independent variables will not be estimated as exactly zero. To address this, in simulations $H(\emptyset)$ is set to $0.1 \min\{H(X), H(Y)\}$ in line 10. This and the choice of 0.8 as the coefficient in **line 16** can be seen as hyper-parameters to be tuned.

5 Experiments

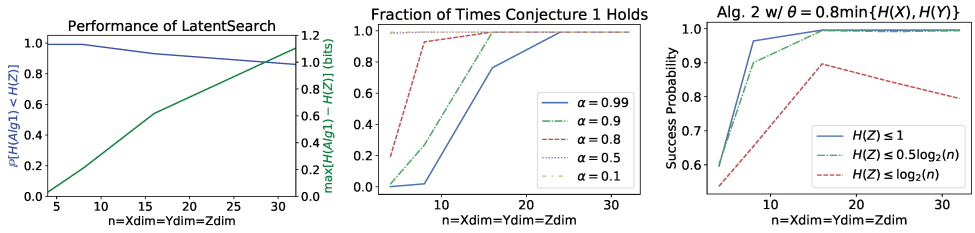
5.1 Performance of LatentSearch

We evaluate how well *LatentSearch* performs by generating data from the latent graph and comparing the entropy it recovers with the entropy of the true confounder Z in Figure 2a. In the generated data, we ensure $H(Z)$ is bounded above by 1 for all n. This makes the task harder for the algorithm for larger n. The left axis shows the fraction of times *LatentSearch* recovers a latent with entropy smaller than $H(Z)$. The right axis shows the worst-case performance in terms of the entropy gap between the algorithm output and true entropy. We generated 100 random distributions for each n. The same number of iterations is used for the algorithm for all n. As expected, performance slowly degrades as n is increased, since $H(Z) < 1, \forall n$. We conclude that *LatentSearch* performs well within the range of n values we use in this paper. Further research is needed to make *LatentSearch* adapt to n.

5.2 Validating Assumption 2: No Low-Entropy Mediator

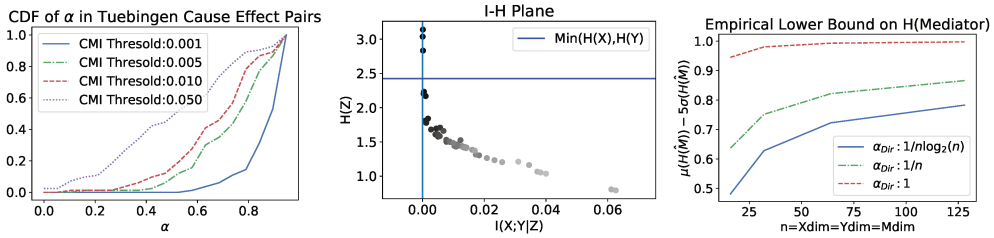
We conducted synthetic and real experiments to validate the assumption that, in practice, it is unlikely for cause-effect pairs to only have low-entropy mediators. First, in Figure 3c we generated data from $X \to Z \to Y$ and evaluated $H(Z)$. $p(X)$ is sampled uniformly from the probability simplex. $p(Z|x), \forall x$ are sampled from Dirichlet with parameter α_{Dir}. It is observed that mediator entropy scales with $\log_2(n)$ for all cases. This supports Assumption 2 by asserting that for most causal models, unless the mediator has a constant number of states, its entropy is close to $H(X), H(Y)$.

Second, in Figure 3a, we run *LatentSearch* on the real cause-effect pairs from Tuebingen dataset [36]. Our goal is to test if the causation can be solely due to low-entropy mediators: If it is, then common entropy should be small since mediator can make the observed variables conditionally independent. We used different thresholds for conditional mutual information for declaring two variables conditionally independent. Investigating typical $I - H$ plots for this dataset (see (b)), we conclude that 0.001 is a suitable CMI threshold for this dataset. From the empirical cdf of $\alpha := \frac{G_1(X,Y)}{\min\{H(X),H(Y)\}}$ across the dataset, we identified that for most pairs $G_1(X, Y) \geq 0.8 \min\{H(X), H(Y)\}$. This indicates that if the causation is solely due to a mediator, it must have entropy of at least $0.8 \min\{H(X), H(Y)\}$.



(a) (b) (c)

Figure 2: **(a)** [Section 5.1] Performance of *LatentSearch* (Alg. 1) on synthetic data. **(b)** [Section 5.3] Fraction of times Conjecture 1 holds in synthetic data for different values of α. **(c)** [Section 5.4] Performance of Algorithm 2 on synthetic data for different confounder entropies.



(a) (b) (c)

Figure 3: **(a)** [Section 5.3,5.2] Empirical cumulative distribution function for α from Conjecture 1 from Tuebingen Data for various conditional mutual information thresholds. **(b)** Tradeoff curve discovered by *LatentSearch* for pair 7 of Tuebingen. Each point is the output of *LatentSearch* for a different value of β. The sharp elbow around $I(X;Y|Z) = 0$ hints that the suitable conditional mutual information threshold for this pair is ≈ 0.001. **(c)** Entropy of mediator in synthetic data. $y-$axis shows the mean minus 5 times the st-dev. of entropy as a fraction of $\log_2(n)$ as an empirical lower bound. Results show that, for almost all models, entropy of the mediator scales with $\log_2(n)$.

5.3 Validating Conjecture 1

In Figure 2b, for each n we sample 1000 distributions from the triangle graph with $H(Z) \leq 1$. Even with a 1 bit latent, we observe that $G_1(X, Y)$ scales with $\min\{H(X), H(Y)\}$. In fact the result hints at the even stronger statement that that for any α, $\exists n_0(\alpha)$ where the conjecture holds $\forall n \geq n_0(\alpha)$.

Experiments with the Tuebingen data, as explained in Section 5.2 and Figure 3a, similarly supports Conjecture 1: If a pair is causally related, common entropy typically scales with $\min\{H(X), H(Y)\}$.

5.4 Identifiability via Algorithm 2

In Figure 2c we uniformly mixed data from triangle and latent graphs with different entropy bounds on the confounder. Since in a practical problem, true $H(Z)$ will not be available, we used $0.8 \min\{H(X), H(Y)\}$ as the entropy threshold θ in Algorithm 2. The results indicate that even if the true $H(Z)$ scales with n (e.g., $0.5 \log_2(n)$), we can still distinguish latent from triangle graph. $\leq$ can be interpreted as $\approx$ due to our sampling method of $p(z)$, as detailed in Section 7.10.

5.5 Evaluating EntropicPC

In this section, to illustrate the performance improvement provided by using common entropy, we compare the order-independent version of PC algorithm [11] with our proposed modification EntropicPC and its conservative version EntropicPC-C as described in Algorithm 3 on both synthetic graphs and data generated from them and on ADULT dataset. We use *pcalg* package in R [24, 18]

Synthetic Graphs/Data: In Figure 4 we randomly sampled joint distributions from 100 Erdös-Rényi graphs on 10 nodes (see Section 7.12 for 3-node graphs) each with 8 states, and obtained datasets with $1k, 5k, 80k$ samples. $80k$ is chosen to mimic the infinite-sample regime where we expect identical

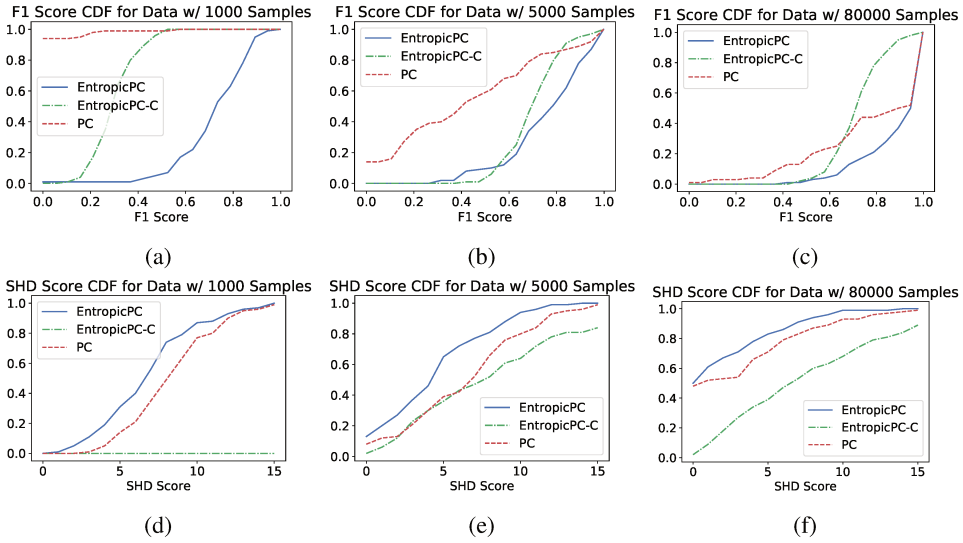


Figure 4: [Section 5.5] Performance of EntropicPC, EntropicPC-C and PC in synthetic data.

performance with PC. **(a), (b), (c)** shows the empirical cumulative distribution function (CDF) of F1 score for identifying edges in either direction correctly (skeleton). Since $F_1 = 1$ is ideal, the best-case CDF is a step function at 1. Accordingly, lower the CDF curve the better. **(d), (e), (f)** shows the empirical CDF of Structural Hamming Distance (SHD) [49] between the true essential graphs, and the output of each algorithm using the same synthetic data. Since $SHD = 0$ is ideal, the best-case CDF is a step function at 0, Accordingly, higher the CDF curve the better. EntropicPC provides very significant improvements to skeleton discovery as indicated by the F1 score and moderate improvement to identifying the true essential graph as indicated by SHD in the small-sample regime.

On ADULT Dataset: We compare the performance of EntropicPC with the baseline PC algorithm that we used for our modifications. Entropic PC identifies the following additional edges that are missed by the PC algorithm: It discovers that *Marital Status, Relationship, Education, Occupation* causes *Salary*. Even though there is no ground truth, at the very least, we expect *Education* and *Occupation* to be causes of *Salary*, whereas PC outputs *Salary* as an isolated node, implying it is not caused by any of the variables. We believe the reason it that for every neighbor, there is some conditioning set and a configuration with very few number of samples. This can easily be interpreted as conditional independence by a CI tester. EntropicPC alleviates this problem by concluding that there is significant dependence between these variables by checking their common entropy. From both real and synthetic simulations, we conclude that common entropy is more robust to small number of samples than CI testing. For space restriction, causal graphs are shown in Figure 8 in Appendix.

6 Conclusions

In this paper, we showed that common entropy, an information-theoretic quantity, can be used for causal discovery. We introduced the notion of Rényi common entropy and proposed a practical algorithm for approximately calculating Rényi$_1$ common entropy. Next, we showed theoretically that common entropy can be used to identify if the observed variables are causally related under certain assumptions. Finally, we showed that common entropy can be used to improve the existing constraint-based causal discovery algorithms. We evaluated our results on synthetic and real data.

Acknowledgments and Disclosure of Funding

This work was supported in part by NSF Grants SATC 1704778, CCF 1763702, 1934932, AF 1901292, 2008710, 2019844, 1731754, 1646522, 1609279, ONR, ARO W911NF-17-1-0359, research gifts by Western Digital, WNCG IAP, computing resources from TACC, the Archie Straiton Fellowship.

Broader Impact

This work lays the foundations for using the information-theoretic notion of common entropy within the context of discovering causal relations from data.

Problems that require discovering causal relations from observational data are prominent across many different fields. Causality is also central to the development of AI. Therefore, we expect this work to have a positive impact by providing a new methodology and identifying settings in which causality can be inferred using this framework.

In terms of the negative effects, we do not foresee an immediate negative effect that may arise because of this work. The only risks would be due to the risks associated with having stronger machine learning models, and better AI that could be misused or exploited.

References

[1] 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.

[2] Sanjeev Arora, Rong Ge, Yonatan Halpern, David Mimno, Ankur Moitra, David Sontag, Yichen Wu, and Michael Zhu. A practical algorithm for topic modeling with provable guarantees. In *International Conference on Machine Learning*, pages 280–288, 2013.

[3] Mohammad Taha Bahadori, Krzysztof Chalupka, Edward Choi, Robert Chen, Walter F Stewart, and Jimeng Sun. Causal regularization. *arXiv preprint arXiv:1702.02604*, 2017.

[4] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[5] Matthew Brand. An entropic estimator for structure discovery. In *Advances in Neural Information Processing Systems*, pages 723–729, 1999.

[6] Ruichu Cai, Jie Qiao, Kun Zhang, Zhenjie Zhang, and Zhifeng Hao. Causal discovery from discrete data using hidden compact representation. In *Advances in Neural Information Processing Systems*, pages 2671–2679, 2018.

[7] Richard Caron and Tim Traynor. The zero set of a polynomial, 2005. [Online].

[8] Rafael Chaves, Lukas Luft, Thiago O Maciel, David Gross, Dominik Janzing, and Bernhard Schölkopf. Inferring latent structures via information inequalities. *arXiv preprint arXiv:1407.2256*, 2014.

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

[10] David Maxwell Chickering and Christopher Meek. Finding optimal bayesian networks. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, pages 94–102, 2002.

[11] Diego Colombo and Marloes H Maathuis. Order-independent constraint-based causal structure learning. *The Journal of Machine Learning Research*, 15(1):3741–3782, 2014.

[12] Pim de Haan, Dinesh Jayaraman, and Sergey Levine. Causal confusion in imitation learning. In *Advances in Neural Information Processing Systems*, pages 11693–11704, 2019.

[13] Dua Dheeru and Efi Karra Taniskidou. UCI machine learning repository, 2017.

[14] Frederick Eberhardt. Phd thesis. *Causation and Intervention (Ph.D. Thesis)*, 2007.

[15] Jalal Etesami and Negar Kiyavash. Discovering influence structure. In *IEEE ISIT*, 2016.

[16] Eric Gaussier and Cyril Goutte. Relation between plsa and nmf and implications. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 601–602. ACM, 2005.

[17] Alain Hauser and Peter Bühlmann. Characterization and greedy learning of interventional markov equivalence classes of directed acyclic graphs. *Journal of Machine Learning Research*, 13(1):2409–2464, 2012.

[18] Alain Hauser and Peter Bühlmann. Characterization and greedy learning of interventional Markov equivalence classes of directed acyclic graphs. *Journal of Machine Learning Research*, 13:2409–2464, 2012.

[19] Thomas Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 289–296. Morgan Kaufmann Publishers Inc., 1999.

[20] Patrik O Hoyer, Dominik Janzing, Joris Mooij, Jonas Peters, and Bernhard Schölkopf. Nonlinear causal discovery with additive noise models. In *Proceedings of NIPS 2008*, 2008.

[21] Dominik Janzing, Jonas Peters, Joris Mooij, and Bernhard Schölkopf. Identifying confounders using additive noise models. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 249–257. AUAI Press, 2009.

[22] Dominik Janzing and Bernhard Scholkopf. Causal inference using the algorithmic markov condition. *IEEE Transactions on Information Theory*, 56(10):5168–5194, 2010.

[23] Dominik Janzing, Eleni Sgouritsa, Oliver Stegle, Jonas Peters, and Bernhard Schölkopf. Detecting low-complexity unobserved causes. *arXiv preprint arXiv:1202.3737*, 2012.

[24] Markus Kalisch, Martin Mächler, Diego Colombo, Marloes H. Maathuis, and Peter Bühlmann. Causal inference using graphical models with the R package pcalg. *Journal of Statistical Software*, 47(11):1–26, 2012.

[25] D Kaltenpoth and J Vreeken. We are not your real parents: Telling causal from confounded by mdl. In *SIAM International Conference on Data Mining (SDM)*, 2019.

[26] Murat Kocaoglu, Alexandros G. Dimakis, Sriram Vishwanath, and Babak Hassibi. Entropic causal inference. In *AAAI'17*, 2017.

[27] Murat Kocaoglu, Amin Jaber, Karthikeyan Shanmugam, and Elias Bareinboim. Characterization and learning of causal graphs with latent variables from soft interventions. In *Advances in Neural Information Processing Systems*, pages 14369–14379, 2019.

[28] Murat Kocaoglu, Karthikeyan Shanmugam, and Elias Bareinboim. Experimental design for learning causal graphs with latent variables. In *Advances in Neural Information Processing Systems*, pages 7018–7028, 2017.

[29] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[30] Ioannis Kontoyiannis and Maria Skoularidou. Estimating the directed information and testing for causality. *IEEE Trans. Inf. Theory*, 62:6053–6067, 2016.

[31] Gowtham Ramani Kumar, Cheuk Ting Li, and Abbas El Gamal. Exact common information. In *IEEE International Symposium on Information Theory (ISIT), 2014*, pages 161–165. IEEE, 2014.

[32] Ciaran M Lee and Robert W Spekkens. Causal inference via algebraic geometry: feasibility tests for functional causal structures with two binary observed variables. *Journal of Causal Inference*, 5(2), 2017.

[33] Furui Liu and Laiwan Chan. Causal discovery on discrete data with extensions to mixture model. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 7(2):21, 2016.

[34] Christopher Meek. *Graphical Models: Selecting causal and statistical models*. PhD thesis.

[35] Christopher Meek. Causal inference and causal explanation with background knowledge. In *Proceedings of the eleventh international conference on uncertainty in artificial intelligence*, 1995.

[36] Joris M Mooij, Jonas Peters, Dominik Janzing, Jakob Zscheischler, and Bernhard Schölkopf. Distinguishing cause from effect using observational data: methods and benchmarks. *The Journal of Machine Learning Research*, 17(1):1103–1204, 2016.

[37] Joris M. Mooij, Oliver Stegle, Dominik Janzing, Kun Zhang, and Bernhard Schölkopf. Probabilistic latent variable models for distinguishing between cause and effect. In *Proceedings of NIPS 2010*, 2010.

[38] Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, 2009.

[39] Jonas Peters and Peter Bühlman. Identifiability of gaussian structural equation models with equal error variances. *Biometrika*, 101:219–228, 2014.

[40] Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference using invariant prediction: identification and confidence intervals. *Statistical Methodology, Series B*, 78:947 – 1012, 2016.

[41] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. Causal inference on discrete data using additive noise models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33:2436–2450, 2011.

[42] Christopher Quinn, Negar Kiyavash, and Todd Coleman. Directed information graphs. *IEEE Trans. Inf. Theory*, 61:6887–6909, 2015.

[43] Eleni Sgouritsa, Dominik Janzing, Jonas Peters, and Bernhard Schölkopf. Identifying finite mixtures of nonparametric product distributions and causal inference of confounders. *arXiv preprint arXiv:1309.6860*, 2013.

[44] Karthikeyan Shanmugam, Murat Kocaoglu, Alex Dimakis, and Sriram Vishwanath. Learning causal graphs with small interventions. In *NIPS 2015*, 2015.

[45] Madhusudana Shashanka, Bhiksha Raj, and Paris Smaragdis. Sparse overcomplete latent variable decomposition of counts data. In *Advances in neural information processing systems*, pages 1313–1320, 2008.

[46] S Shimizu, P. O Hoyer, A Hyvarinen, and A. J Kerminen. A linear non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7:2003—2030, 2006.

[47] Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*. A Bradford Book, 2001.

[48] Bastian Steudel and Nihat Ay. Information-theoretic inference of common ancestors. *Entropy*, 17(4):2304–2327, 2015.

[49] Ioannis Tsamardinos, Laura E Brown, and Constantin F Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine learning*, 65(1):31–78, 2006.

[50] Thomas Verma and Judea Pearl. An algorithm for deciding if a set of observed independencies has a causal explanation. In *Proceedings of the Eighth international conference on uncertainty in artificial intelligence*, 1992.

[51] Simon Webb. Central slices of the regular simplex. *Geometriae Dedicata*, 61(1):19–28, 1996.

[52] Mirjam Weilenmann and Roger Colbeck. Analysing causal structures with entropy. *Proc. R. Soc. A*, 473(2207):20170483, 2017.

[53] Aaron Wyner. The common information of two dependent random variables. *IEEE Transactions on Information Theory*, 21(2):163–179, 1975.

[54] Jiji Zhang. *Causal inference and reasoning in causally insufficient systems*. PhD thesis, Citeseer, 2006.

[55] Jiji Zhang. On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias. *Artificial Intelligence*, 172(16-17):1873–1896, 2008.

7 Appendix

7.1 Detailed Background

Let $D = (\mathcal{V}, \mathcal{E})$ be a directed acyclic graph on the set of vertices $\mathcal{V} = \{V_1, V_2, \ldots, V_n\}$ with directed edge set $\mathcal{E}$. Each directed edge $e_k \in \mathcal{E}$ is a tuple $e_k = (V_i, V_j)$. Let $\mathbb{P}$ be a joint distribution over a set of variables labeled by $\mathcal{V}$. D is called a valid Bayesian network for the distribution $\mathbb{P}$, if $\mathbb{P}$ factorizes with respect to the graph D as $\mathbb{P}(V_1, V_2, \ldots V_n) = \prod_i \mathbb{P}(V_i|pa_i)$, where pa_i are the set of parents of vertex V_i in graph D. In a Bayesian network D that is valid for $\mathbb{P}$, if three vertices X, Y, Z satisfy a graphical criterion called the *d-separation* on D, then $X \perp\!\!\!\perp Y \,|Z$ in $\mathbb{P}$. The faithfulness assumption allows us to infer dependence relations based on d-separation: $\mathbb{P}$ is said to be faithful to graph D when the following holds: Any three variables X, Y, Z that are not d-separated are conditionally dependent, i.e., $X \not\perp\!\!\!\perp Y \,|Z$ in $\mathbb{P}$.

Note that the edges in a Bayesian network do not carry a physical meaning: They simply indicate how a joint distribution can be factorized. *Causal Bayesian networks* (or causal graphs) [38] on the other hand capture the causal relations between variables: They extend the notion of Bayesian networks to different experimental, the so called interventional settings. *An intervention* is an experiment that changes the workings of the underlying system and sets the value of a variable, shown as $do(X = x)$. Causal Bayesian networks allow us to calculate the joint distributions under these experimantal conditions, called the interventional distributions[2].

In this paper, we work with the causal graphs given in Figure 1. From the d-separation principle, we see that the latent graph satisfies $X \perp\!\!\!\perp Y \,|Z$, whereas under the faithfulness condition, $X \not\perp\!\!\!\perp Y \,|Z$ in the triangle graph or the direct graph. Checking the existence of such a latent variable can help us recover the true causal graph as we discover in the next sections. We work with discrete ordinal or categorical variables. Suppose the support sizes of the observed variables X and Y are m and n, respectively. The joint distribution can be represented with an $m \times n$ non-negative matrix whose entries sum to 1. We assume that we have access to this joint distribution.

We use $[n]$ to represent the set $\{1, 2, \ldots, n\}$ for any $n \in \mathbb{N}$. Capital letters represent random variables, lowercase letters represent realizations[3]. Letters X, Y are reserved for the observed variables, whereas Z is used for the latent variable. To represent the probability mass function over three variables X, Y, Z, we use $p(x, y, z) \coloneqq \mathbb{P}(X = x, Y = y, Z = z)$ and similarly for any conditional $p(z|x, y) \coloneqq \mathbb{P}(Z = z|X = x, Y = y)$. For a function $q(x, y, z)$ that is understood to be a probability mass function, we use shorthand notation for marginals and conditionals such as $q(x, y)$ and $q(x|z)$ to represent the functions obtained from $q(x, y, z)$ via standard operations on probability distributions. Lowercase boldface letters are used for vectors and uppercase boldface letters are used for matrices. We also use $p(Z|x, y)$ to represent the conditional distribution $\mathbb{P}(Z|X = x, Y = y)$ (Similarly for $p(Z|x), p(Z|y)$). card(X) stands for the support size of X. Rényi entropy of order α of a random variable X is defined as $H_\alpha(X) = \frac{1}{1-\alpha} \log \sum_i p_i^\alpha$. Rényi entropy of order 0 gives the support size of a random variable. It can be shown that in the limit as $\alpha \to 1$, Rényi entropy becomes Shannon entropy, defined as $H_1(X) = -\sum_x p(x) \log_2(p(x))$ in bits. In a graph D with nodes labeled as $\{X_i\}_i$, pa_i stands for the set of parents of X_i in D. $Dir(\alpha)$ stands for Dirichlet distribution with parameter α.

7.2 Detailed Related Work

Latent Variable Discovery: Latent variables have been used to model and explain dependence between observed variables in different communities under different names. Probabilistic latent semantic analysis (pLSA) [19] aims at constructing a variable that explains dependence. However the objective is not to minimize entropy of the constructed variable. Latent Dirichlet allocation (LDA) is another framework which is widely used in topic modeling [4, 2]. Although LDA encourages sparsity of topics, this does not correspond to minimizing the support size of the constructed latent variable. Factorizing the joint distribution matrix between two observed variables via NMF with generalized KL divergence loss recovers solutions to the pLSA problem [16]. Similar to pLSA, NMF does not have an incentive to discover low-entropy latents.

[2] For a formal introduction to Pearl's framework please see [47, 38].

[3] In some proofs, x_i is used to represent the probability that the variable X takes the value i for simplicity.

Perhaps the most relevant to ours in the machine learning literature are the two papers in the Bayesian setting [5, 45]. They use low-entropy priors on the latent variable's distribution while performing inference. However their approach is different and their methods cannot be used to discover the tradeoff between conditional mutual information and the entropy of the latent variable. In [45], the authors use low-entropy prior as a proxy for discovering latent factors with sparse support.

Finding the latent variable with smallest entropy that renders the two observed variables conditonally independent is closely related to some of the problems in information theory: Wyner's common information [53] is defined as the minimum rate of the source from which the observed variables X, Y can be reconstructed using additional random bits. Wyner allows multiple channel uses and is interested in the approximate reconstruction of the observed joint distribution. This can be seen as approximate reconstruction of the joint distribution when we raise the dimension of X and Y via cartesian product with itself. [31] considers finding the source with the minimum rate for the exact recovery of the observed joint distribution, but still in the asymptotic regime of multiple channel uses. They also introduce the notion of common entropy and obtain an analytical expression for binary variables, which we utilize in this work.

Learning Causal Graphs with Latents: Learning causal graphs with latent variables has been extensively studied in the literature. In graphs with many observed variables, some of the edges can be recovered from the observational data (for example through algorithms that employ conditional independence (CI) tests such as IC* [38] and FCI [47]). However, latent variables make the CI tests less informative, by inducing spurious correlations between the observed variables. For example for the graphs in Figure 1 CI tests on the observed variables is not informative of the causal structure.

Identifiability of causal structures without latent variables from data has been studied extensively in the literature under various assumptions [20, 41, 37, 39, 40, 3, 15, 26]. Our approach can be seen as an extension of [26]: There, the authors assume that the exogenous variables have small Rényi entropy and suggest an algorithm to distinguish the causal graph $X \to Y$ from $X \leftarrow Y$. However, their approach cannot be used in the presence of latent variables. In the presence of latents [6] considers a setup similar to [26], where the hidden variable has small support size, however also assumes the mapping to the hidden variable is deterministic. In [23], authors identify a condition on $p(Y|X)$ which implies that there does not exist any latent variable Z with small support which can make X, Y conditionally independent. For discrete variables, this assumption implies that the conditionals $p(Y|x)$ lie on the boundary of the probability simplex, which corresponds to the joint probability matrix to be sparse in a structured way. In the continuous variable setting, [43] propose using kernel methods to detect latent confounders. [52] and [8] analyzes the discoverability of causal structures with latents using the entropic vector of the variables. Finally, related work also includes [21] and [33], where the authors extend the additive noise model based approach in [20] to the case with a latent confounder. Algebraic geometry can be used to distinguish causal graphs as the set of distributions that can be encoded by a graph correspond to different algebraic varieties. However, these methods in general are not scalable beyond a few variables and a few number of states [32]. Authors in [22] propose using Kolmogorov complexity of the causal model and declare the graph with smaller complexity to be the true graph. [25] uses description length as a proxy to Kolmogorov complexity to identify the latent confounders. In [48], the authors use information inequalities to infer which subsets of a set of observed variables must have latent confounders, along with an associated lower bound on the entropy of these confounders. In our setting of two observed variables, this gives the trivial bound of $H(Z) \geq I(X;Y)$ for any latent confounder Z.

7.3 Proof of Theorem 1: Stationarity

In this section, we show the first part of Theorem 1, i.e., that the stationarity points of the algorithm are also stationary points of the given loss function. We write the objective function more explicitly in terms of the optimization variables $q(z|x,y)$:

$$\mathcal{L}(q(\cdot|\cdot,\cdot)) = \sum_{x,y,z} q(x,y,z) \log\left(\frac{q(x,y|z)}{q(x|z)q(y|z)}\right) - \beta \sum_{z} q(z)\log(q(z)) \tag{3}$$

$$= \sum_{x,y,z} p(x,y)q(z|x,y) \log\left(\frac{q(z|x,y)}{q(z|x)q(z|y)}\right) + (1-\beta)\sum_{z} q(z)\log(q(z)) + I(X;Y) \tag{4}$$

by Bayes rule and assuming that $q(z|x,y)$ and $p(x,y)$ are strictly positive.

Our objective then is

$$\begin{aligned} \underset{q(z|x,y)}{\text{minimize}} \quad & \mathcal{L}(q(z|x,y)) \\ \text{subject to} \quad & \sum_z q(z|x,y) = 1, \ \forall x, y, \\ & q(z|x,y) \geq 0, \ \forall z, x, y. \end{aligned} \tag{5}$$

We can write the Lagrangian, which we represent with $\bar{\mathcal{L}}$, as

$$\begin{aligned} \bar{\mathcal{L}} = & \sum_{x,y,z} p(x,y) q(z|x,y) \log\left(\frac{q(z|x,y)}{q(z|x)q(z|y)}\right) + I(X;Y) + (1-\beta)\sum_z q(z)\log(q(z)) \\ & + \sum_{x,y} \delta_{x,y}\left(\sum_z q(z|x,y) - 1\right) \end{aligned} \tag{6}$$

In order to find the stationary points of the loss, we take its first derivative and set it to zero. To compute the partial derivatives, notice that $q(z|x), q(z|y), q(z)$ are linear functions of $q(z|x,y)$ (use Bayes rule and marginalization). We can then easily write the partial derivatives of these quantities with respect to $q(z|x,y)$ as follows:

$$\begin{aligned} \frac{\partial q(z|x)}{\partial q(z|x,y)} &= \frac{\partial \sum_{y'} q(z|x,y')p(y'|x)}{\partial q(z|x,y)} = p(y|x), \\ \frac{\partial q(z|y)}{\partial q(z|x,y)} &= \frac{\partial \sum_{x'} q(z|x',y)p(x'|y)}{\partial q(z|x,y)} = p(x|y) \\ \frac{\partial q(z)}{\partial q(z|x,y)} &= \frac{\partial \sum_{x',y'} q(z|x',y')p(x',y')}{\partial q(z|x,y)} = p(x,y). \end{aligned}$$

Using these expressions we have the following.

$$\begin{aligned} \frac{\partial \bar{\mathcal{L}}}{\partial q(z|x,y)} &= p(x,y)\left[1 + \log(q(z|x,y)) - (1 + \log(q(z|x))\right. \\ &\quad \left. - (1 + \log(q(z|y))) \quad + (1-\beta)(1 + \log(q(z))) + \delta_{x,y}\right] \\ &= p(x,y)\left[-\beta + \delta_{x,y} + \log\left(\frac{q(z|x,y)q(z)^{1-\beta}}{q(z|x)q(z|y)}\right)\right] \end{aligned}$$

Assuming $p(x,y) > 0$, any stationary point then satisfies

$$q(z|x,y) = \left(\frac{1}{2}\right)^{\delta_{x,y}-\beta} \frac{q(z|x)q(z|y)}{q(z)^{1-\beta}} \tag{7}$$

Since $q(z|x,y)$ is a probability distribution, we have

$$\sum_z q(z|x,y) = \left(\frac{1}{2}\right)^{\delta_{x,y}-\beta} \sum_z \frac{q(z|x)q(z|y)}{q(z)^{1-\beta}} = 1 \tag{8}$$

Defining $F(x,y) \coloneqq \left(\frac{1}{2}\right)^{\delta_{x,y}-\beta}$, we have

$$F(x,y) = \frac{1}{\sum_z \frac{q(z|x)q(z|y)}{q(z)^{1-\beta}}}. \tag{9}$$

From the algorithm description, any stationary point of Algorithm 1 should satisfy

$$q(z|x,y) = F(x,y)\frac{q(z|x)q(z|y)}{q(z)^{1-\beta}}, \tag{10}$$

for the same $F(x,y)$ defined above. Therefore a point is a stationary point of the loss function if and only if it is a stationary point of LatentSearch (Algorithm 1). □

7.4 Proof of Theorem 1: Convergence

In this section, we show the latter statement in Theorem 1, i.e., *LatentSearch* converges to a local minimum or a saddle point. We can rewrite the loss as

$$\mathcal{L}(q(\cdot|\cdot,\cdot)) = \sum_{x,y,z} q(x,y,z)\log\left(\frac{q(x,y|z)}{q(x|z)q(y|z)}\right) - \beta\sum_{z} q(z)\log(q(z)) \tag{11}$$

$$= \sum_{x,y,z} p(x,y)q(z|x,y)\log\left(\frac{q(z|x,y)}{q(z|x)q(z|y)}\right) + I(X;Y) + (1-\beta)\sum_{z} q(z)\log(q(z)), \tag{12}$$

If we substitute $\beta = 1$, we obtain

$$\mathcal{L}(q(\cdot|\cdot,\cdot)) = \sum_{x,y,z} p(x,y)q(z|x,y)\log\left(\frac{q(z|x,y)}{q(z|x)q(z|y)}\right) + I(X;Y). \tag{13}$$

Our optimization problem can be written as

$$\begin{aligned} \underset{q(z|x,y)}{\text{minimize}} \quad & \mathcal{L}(q(z|x,y)) \\ \text{subject to} \quad & \sum_{z} q(z|x,y) = 1, \forall x,y. \end{aligned} \tag{14}$$

Notice that $\mathcal{L}(q(z|x,y))$ is not convex or concave in $q(z|x,y)$. However we can rewrite the minimization as follows:

$$\begin{aligned} \underset{q(z|x,y)}{\text{minimize}} \quad \underset{r(z|x),s(z|y)}{\text{minimize}} \quad & \sum_{x,y,z} p(x,y)q(z|x,y) \\ & \log\left(\frac{q(z|x,y)}{r(z|x)s(z|y)}\right) + I(X;Y) \\ \text{subject to} \quad & \sum_{z} q(z|x,y) = 1, \forall x,y \\ & \sum_{z} r(z|x) = 1, \forall x, \\ & \sum_{z} s(z|y) = 1, \forall y. \end{aligned}$$

To see that (15) is equivalent to (14), notice that the optimum for the inner minimization is $r^*(z|x) = q(z|x)$ and $s^*(z|x) = q(z|y)$. This is due to the fact that (15) is convex in $r(z|x)$ and $s(z|y)$ and concave in $t(z)$, which can be seen through the partial derivatives of the Lagrangian:

$$\min_{q(z|x,y)} \quad \min_{r(z|x),s(z|y)} \quad \sum_{x,y,z} p(x,y)q(z|x,y) + \log\left(\frac{q(z|x,y)}{r(z|x)s(z|y)}\right) + I(X;Y) \tag{15}$$

$$+ \sum_{x,y} \delta_{x,y}\left(\sum_{z} q(z|x,y) - 1\right) + \sum_{x} \eta_x\left(\sum_{z} r(z|x) - 1\right) \tag{16}$$

$$+ \sum_{x} \nu_y\left(\sum_{z} s(z|y) - 1\right) \tag{17}$$

Let $\bar{\mathcal{L}}$ be defined as

$$\bar{\mathcal{L}} = \sum_{x,y} \delta_{x,y}\left(\sum_{z} q(z|x,y) - 1\right) + \sum_{x} \eta_x\left(\sum_{z} r(z|x) - 1\right) + \sum_{x} \nu_y\left(\sum_{z} s(z|y) - 1\right) \tag{18}$$

For fixed $q(z|x,y), s(z|y)$, we have

$$\frac{\partial \bar{\mathcal{L}}}{\partial r(z|x)} = -\frac{p(x)q(z|x)}{r(z|x)} + \eta_x$$
$$\frac{\partial^2 \bar{\mathcal{L}}}{\partial r(z|x)^2} = \frac{p(x)q(z|x)}{r(z|x)^2}.$$

Therefore $\bar{\mathcal{L}}$ is convex in $r(z|x)$ and the optimum can be obtained by setting the first derivative to zero. Then we have

$$r^*(z|x) = \frac{p(x)q(z|x)}{\eta_x}, \forall x, z. \tag{19}$$

Since we have $\sum_z r^*(z|x) = \frac{p(x)}{\eta_x} \sum_z q(z|x) = 1$, we obtain $r^*(z|x) = q(z|x)$. Similarly, we can show that $s^*(z|x) = q(z|y)$. Notice that this inner minimization is exactly the same as the first update of Algorithm 1.

We can also show that $\mathcal{L}$ is convex in the variables r, s jointly: This can be seen through the fact that $\frac{\partial^2}{\partial r(z|x)s(z|y)}\mathcal{L} = 0$ and the Hessian is positive definite.

This concludes that (15) is equivalent to (5). Moreover, since the objective function is convex in $q(z|x,y)$ and also jointly convex in $r(z|x), s(z|y)$, we can switch the order of the minimization terms. Therefore, we can equivalently write

$$\min_{r(z|x),s(z|y)} \quad \min_{q(z|x,y)} \quad \sum_{x,y,z} p(x,y)q(z|x,y)\log\left(\frac{q(z|x,y)}{r(z|x)s(z|y)}\right) + I(X;Y) + \bar{\mathcal{L}} \tag{20}$$

Let us analyze the inner minimization in this equivalent formulation for fixed $r(z|x), s(z|x)$. Similarly, we can take the partial derivative as follows:

$$\begin{aligned}\frac{\partial \bar{\mathcal{L}}}{\partial q(z|x,y)} &= p(x,y)\left[1 + \log(q(z|x,y)) - \log(r(z|x)) - \log(s(z|x)) + \delta_{x,y}\right] \\ &= p(x,y)\left[1 + \delta_{x,y} + \log\left(\frac{q(z|x,y)}{r(z|x)s(z|y)}\right)\right] \\ \frac{\partial^2 \bar{\mathcal{L}}}{\partial q(z|x,y)^2} &= p(x,y)\left[\frac{1}{q(z|x,y)}\right].\end{aligned}$$

Notice that $\frac{\partial^2 \bar{\mathcal{L}}}{\partial q(z|x,y)^2} > 0$. Hence $\bar{\mathcal{L}}$ is convex in $q(z|x,y)$. Then the optimum can be obtained by setting the first derivative to zero. We have

$$p(x,y)\left[1 + \delta_{x,y} + \log\left(\frac{q(z|x,y)}{r(z|x)s(z|y)}\right)\right] = 0, \tag{21}$$

or equivalently

$$q(z|x,y) = \left(\frac{1}{2}\right)^{1+\delta_{x,y}} r(z|x)s(z|y). \tag{22}$$

Note that if we define

$$F(x,y) \coloneqq \sum_z r(z|x)s(z|y),$$

since $\sum_z q(z|x,y) = \left(\frac{1}{2}\right)^{1+\delta_{x,y}} \sum_z r(z|x)s(z|y) = 1$, we can write

$$q(z|x,y) = \frac{1}{F(x,y)} r(z|x)s(z|y). \tag{23}$$

This is exactly the same as the second update of LatentSearch (Algorithm 1) if $r(z|x) = q(z|x), s(z|y) = q(z|y)$.

Therefore, if $q_i(z|x,y)$ is the current conditional at iteration i, the next update of LatentSearch (Algorithm 1) is equivalent to first solving the inner minimization of (15) thereby assigning $r(z|x) =$

$q_i(z|x), s(z|y) = q_i(z|y)$, then switching the order of the minimization operations, and solving the inner minimization of (20), therefore assigning $q_{i+1}(z|x,y) = \frac{1}{F(x,y)} q_i(z|x) q_i(z|y)$. In each of this two-step optimization iteration, either loss function goes down, or it does not change. If it does not change, the algorithm has converged. Otherwise, it cannot go down indefinitely since loss (2) is lower bounded as $I(X;Y|Z) \geq 0$ and $H(Z) \geq 0$ and therefore has to converge. This proves convergence of the algorithm to either a local minimum or a saddle point. The converged point cannot be a local maximum since it is arrived at after a minimization step. □

7.5 Proof of Theorem 2

We first show the result for distinguishing latent graph from the triangle graph.

Since X, Y are discrete variables, we can represent the joint distribution of X, Y in matrix form. Let $\mathbf{M} = [p(x,y)]_{(x,y)\in[m]\times[n]}$. With a slight abuse of notation, let $\mathbf{z} \coloneqq [z_1, z_2, \dots z_k]$ be the probability mass (row) vector of variable Z, i.e., $\mathbb{P}[Z = i] = \mathbf{z}[i] = z_i$. Similarly, let $\mathbf{x}_z \coloneqq [x_{z,1}, x_{z,2}, \dots x_{z,k}]$ be the conditional probability mass vector of X conditioned on $Z = z$, i.e., $\mathbb{P}[X = i|Z = z] = \mathbf{x}_z[i] = x_{z,i}$. Finally, let $\mathbf{y}_{z,x} \coloneqq [y_{z,x,1}, y_{z,x,2}, \dots y_{z,x,n}]$ be the conditional probability mass vector of Y conditioned on $X = x$ and $Z = z$. We can write the matrix $\mathbf{M}$ as follows:

$$\mathbf{M} = \sum_{i=1}^{k} z_i \begin{bmatrix} x_{i,1}\mathbf{y}_{i,1} \\ x_{i,2}\mathbf{y}_{i,2} \\ \vdots \\ x_{i,m}\mathbf{y}_{i,m} \end{bmatrix} \tag{24}$$

Now suppose for the sake of contradiction that there exists such a $q(x,y,z)$ such that $\sum_z q(x,y,z) = p(x,y)$ and $X \perp\!\!\!\perp Y\,|Z$. Then $\mathbf{M}$ admits a factorization of the form

$$\mathbf{M} = \sum_{i=1}^{k} z'_i \begin{bmatrix} x'_{i,1}\mathbf{y}'_{i,1} \\ x'_{i,2}\mathbf{y}'_{i,2} \\ \vdots \\ x'_{i,m}\mathbf{y}'_{i,m} \end{bmatrix}, \tag{25}$$

where $x'_{i,j}, \mathbf{y}'_{i,j}, z'_i$ are due to the joint $q(x,y,z)$ and are potentially different form their counterparts in (24). Notice that since $X \perp\!\!\!\perp Y\,|Z$, we have $\mathbf{y}'_{i,j} = \mathbf{y}'_{i,l}, \forall (j,l) \in [k] \times [m]$. Therefore the matrices

$$\begin{bmatrix} x'_{i,1}\mathbf{y}'_{i,1} \\ x'_{i,2}\mathbf{y}'_{i,2} \\ \vdots \\ x'_{i,m}\mathbf{y}'_{i,m} \end{bmatrix}, \tag{26}$$

are rank 1 $\forall i \in [k]$. Therefore, $\mathbf{M}$ has NMF rank at most k. Since matrix rank is upper bounded by the NMF rank, $\text{rank}(M) \leq k$. Therefore, there exists a $q(x,y,z)$ such that $\sum_z q(x,y,z) = p(x,y)$ and $X \perp\!\!\!\perp Y\,|Z$ *only if* $\text{rank}(M) \leq k$. In fact, it is easy to show that this is an if and only if relation: Any NMF of the joint distribution corresponds to a latent confounder and and latent confounder corresponds to an NMF of the joint distribution. Next, we show that under the generative model described in the theorem statement, this happens with probability zero.

We have the following lemma:

Lemma 1. *Let $\{\mathbf{x}_i : i \in [n]\}$ be a set of vectors sampled independently, uniformly randomly from the simplex S_{n-1} in n dimensions. Then, $\{\mathbf{x}_i : i \in [n]\}$ are linearly independent with probability 1.*

Proof. If $\mathbf{x}_i$ are linearly dependent, then there exists a set $\{\alpha_i : i \in [n]\}$ such that $\sum_{i=1}^{n} \alpha_i \mathbf{x}_i = 0$. Let $j = \arg\max\{i \in [n] : \alpha_i > 0\}$. Equivalently $\mathbf{x}_j$ is in the range of the set of vectors $\{\mathbf{x}_i : i \in [j-1]\}$. Therefore, we can write

$$\begin{aligned} &\mathbb{P}\left[\{\mathbf{x}_i : i \in [n]\} \text{ are linearly independent }\right] \\ &\qquad \leq \sum_{i=2}^{n} \mathbb{P}\left[\mathbf{x}_i \in R(\mathbf{x}_1, \dots, \mathbf{x}_{i-1})\right], \end{aligned} \tag{27}$$

where $R(\mathbf{x}_1, \ldots, \mathbf{x}_{i-1})$, is the range of the vectors $\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}$, i.e., the vector space spanned by $\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}$.

Notice that $\dim(R(\mathbf{x}_1, \ldots, \mathbf{x}_{i-1})) < n-1, \forall i \leq n-1$. Therefore, codimension of $R(\mathbf{x}_1, \ldots, \mathbf{x}_{i-1})$ with respect to the simplex is non-zero $\forall i \leq n-1$. Therefore, the Lebesgue measure of $R(\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}) \cap S_{n-1}$ is zero with respect to the uniform measure over S_{n-1}. Hence, $\mathbb{P}\left[\mathbf{x}_i \in R(\mathbf{x}_1, \ldots, \mathbf{x}_{i-1})\right] = 0, \forall i \leq n-1$.

The above argument does not hold for the last term in the summation in (27). However, intersection of any $n-1$ dimensional vector space with the simplex S_{n-1} is an $n-2$ dimensional slice of the simplex [51]. Therefore, it has Lebesgue measure zero with respect to the uniform measure over the simplex. □

Corollary 3. *Let $\{\mathbf{x}_i : i \in [n]\}$ be a set of vectors sampled independently, uniformly randomly from the simplex S_{n-1} in n dimensions. Let $\{c_i \neq 0 : i \in [n]\}$ be arbitrary real scalars that are non-zero. Then, $\{c_i \mathbf{x}_i : i \in [n]\}$ are linearly independent with probability 1.*

Proof. The proof of Lemma 1 goes through since the span of a set of vectors does not change with scaling of the vectors. □

$\mathbf{M}$ is rank deficient if and only if its determinant is zero, i.e., $\det(M) = 0$. The determinant is a polynomial in $\{z_i : i \in [k]\}$. By induction, one can show that if a finite degree multivariate polynomial is not identically zero, the set of roots has zero Lebesgue measure (for example, see [7]). The uniform measure over the simplex is absolutely continuous with respect to Lebesgue measure. Hence, the set of roots of a finite degree multivariate polynomial has measure zero with respect to the uniform measure over the simplex.

To show that $\det(\mathbf{M})$ is not identically zero, it is sufficient to choose a set of $z_i's$ for which determinant is non-zero. First, observe that by Corollary 3, each matrix

$$\begin{bmatrix} x_{i,1}\mathbf{y}_{i,1} \\ x_{i,2}\mathbf{y}_{i,2} \\ \vdots \\ x_{i,m}\mathbf{y}_{i,m} \end{bmatrix} \tag{28}$$

is full rank with probability 1. Let $z_1 = 1$ and $z_j, \forall j \in \{2, 3, \ldots, k\}$. Then $\det(\mathbf{M}) \neq 0$ since $\mathbf{M}$ is full rank. Therefore, the determinant, which is a polynomial in $\{z_i : i \in [k]\}$ is not identically zero. This concludes the proof that with probability 1, $\text{rank}(\mathbf{M}) = n > k$.

If the distribution is generated from the direct graph, from Lemma 1, we know that the rows of conditional probability matrix are linearly independent. Since joint probability matrix can be obtained by scaling each row of this matrix with the probability values of X, and this operation does not change rank, joint probability matrix obtained from the direct graph is full rank with probability 1. Therefore non-negative rank of this matrix has to be n, concluding the proof. □

7.6 Proof of Corollary 1

The statement follows from the fact that the proposed generative model induces a non-zero probability measure on every joint distribution, which is the set of distributions that can be encoded from the *triangle graph* and any distribution that can be encoded by the *latent graph* requires $X \perp\!\!\!\perp Y \,|Z$, which we show in Theorem 2 happens with probability zero. □

7.7 Proof of Theorem 3

We give the proof for binary Z. The argument can be extended to when Z has any finite number of states.

We overload the notation and use z for the probability that random variable Z is 0. We have

$$p(Z = 0) = z, \qquad p(Z = 1) = 1 - z \tag{29}$$

$$p(X = 0|Z = z) = x_z \tag{30}$$

$$p(Y = 0|X = x, Z = z) = y_{x,z} \tag{31}$$

The conditional distributions from UGM can be sampled uniformly from the simplex via normalized exponential random variables, however in the case of binary variables, this is equivalent to sampling uniformly. Hence, we can assume $x_z, y_{x,z}$ are uniform random variables with support $[0, 1]$. Based on this generative model, we can calculate $p(x)$ and $p(y|x)$ as follows:

$$p(X = 0) = x_0 z + x_1(1 - z), \tag{32}$$
$$p(X = 1) = (1 - x_0)z + (1 - x_1)(1 - z)$$
$$p(Y = 0|X = 0) = \frac{y_{0,0}x_0 z + y_{0,1}x_1(1 - z)}{x_0 z + x_1(1 - z)} \tag{33}$$
$$p(Y = 0|X = 1) = \frac{y_{1,0}(1 - x_0)z + y_{1,1}(1 - x_1)(1 - z)}{(1 - x_0)z + (1 - x_1)(1 - z)} \tag{34}$$

We use the characterization of [31] for the minimum entropy Z that can make X, Y conditionally independent. Let $t = p(X = 0)$ and let $\alpha \coloneqq p(Y = 0|X = 0), \beta \coloneqq p(Y = 0|X = 1)$. We re-state their theorem for self-containment of our paper:

Theorem 4 ([31]). *Consider two binary random variables X, Y. Define $t \coloneqq p(X = 0), \alpha \coloneqq p(Y = 0|X = 0), \beta \coloneqq p(Y = 0|X = 1)$. Let $\alpha' = \min\{\alpha, \beta\}, \beta' \coloneqq \max\{\alpha, \beta\}$. Then of all $q(x, y, z')$ where $q(x, y) = p(x, y)$ and $X \perp\!\!\!\perp Y \,|Z$ minimum entropy Z' has entropy*

$$LB \coloneqq \min\{H_b(A), H_b(B)\}, \tag{35}$$
$$A = t\left(1 - \frac{\alpha'}{\beta'}\right), B = (1 - t)\left(1 - \frac{1 - \beta'}{1 - \alpha'}\right) \tag{36}$$

Note that in the generative model we are considering, the entries of $p(x, y)$ are random variables, which implies that LB is a random variable.

Consider a sequence z_n. Let Z_n be the binary random variable where $\mathbb{P}(Z_n = 0) = z_n$. Notice that $H(z_n)$ converges to zero if and only if z_n converges to either 0 or 1. Since the generative model is symmetric with respect to the conditionals $p(x|z = 0)$ compared to $p(x|z = 1)$ and $p(y|x, z = 0)$ compared to $p(y|x, z = 1)$, without loss of generality we can consider the case where z_n goes to 0.

Now suppose $0 < z_0 < 0.5$ and z_n is a monotonically decreasing sequence. When we substitute z_n for z in the generative model, we use the symbols in Theorem 4 with subscript n to distinguish them for different values of n.

The event that there does not exist a latent variable with small entropy that can make the observed variables independent is equivalent to the event that the lower bound is strictly greater than the entropy of the true latent variable:

$$\mathbb{P}(\mathcal{Q}_p = \emptyset) = \mathbb{P}(p(x, y) : \nexists q(x, y, z') \tag{37}$$
$$\text{s.t. } \sum_{z'} q(x, y, z') = p(x, y), \tag{38}$$
$$X \perp\!\!\!\perp Y \,|Z' \,, H(Z') \leq H(Z)) \tag{39}$$
$$= \mathbb{P}(LB > H(Z)) \tag{40}$$

We want to show that

$$\lim_{n\to\infty} \mathbb{P}(LB_n > H(Z_n)) = 1. \tag{41}$$

Define the following events:

$$\varepsilon^A_{z_n} \coloneqq \{\text{Event that } H_b(A_n) \leq H_b(z_n)\}. \tag{42}$$
$$\varepsilon^B_{z_n} \coloneqq \{\text{Event that } H_b(B_n) \leq H_b(z_n)\}. \tag{43}$$

By union bound

$$\mathbb{P}(LB_n \leq H(Z_n)) \leq \mathbb{P}(\varepsilon^A_{z_n}) + \mathbb{P}(\varepsilon^B_{z_n}) \tag{44}$$

We first investigate the term $\lim_{n\to\infty} \mathbb{P}(\varepsilon^A_z)$. By conditioning on the event that $A_n \leq 0.5$ and $A_n > 0.5$, we can reduce the comparison between $H_b(a), H_b(c)$ to a comparision between a and c.

Due to first applying the law of total probability and then Bayes rule, we have

$$\begin{aligned}\mathbb{P}(\varepsilon_{z_n}^A) &= \mathbb{P}\left(\varepsilon_{z_n}^A | A_n \leq 0.5\right)\mathbb{P}(A_n \leq 0.5) + \mathbb{P}\left(\varepsilon_{z_n}^A | A_n > 0.5\right)\mathbb{P}(A_n > 0.5) \\ &= \mathbb{P}\left(A_n \leq z_n | A_n \leq 0.5\right)\mathbb{P}(A_n \leq 0.5) + \mathbb{P}\left(A_n \geq 1 - z_n | A_n > 0.5\right)\mathbb{P}(A_n > 0.5) \\ &= \mathbb{P}\left(A_n \leq z_n\right)\mathbb{P}(A_n \leq 0.5 | A_n \leq z_n) + \mathbb{P}\left(A_n \geq 1 - z_n\right)\mathbb{P}(A_n > 0.5 | A_n \geq 1 - z_n) \\ &= \mathbb{P}\left(A_n \leq z_n\right) + \mathbb{P}\left(A_n \geq 1 - z_n\right)\end{aligned}$$

Define the following random variable:

$$S_n^A \coloneqq -z_n + t_n\left(1 - \frac{\alpha_n'}{\beta_n'}\right), \tag{45}$$

where the terms on the right hand side are as defined in Theorem 4. Then $\mathbb{P}\left(A_n \leq z_n\right) = \mathbb{P}\left(S_n^A \leq 0\right)$ and $\mathbb{P}\left(A_n \geq 1 - z_n\right) = \mathbb{P}\left(S_n^A \geq 1\right)$. We have

$$\mathbb{P}\left(S_n^A \leq 0\right) = \int_{-\infty}^{0} S_n^A d\mu, \tag{46}$$

where μ is the probability measure induced by the generative model. Note that $t_n \in [0,1], z_n \in (0, 0.5), \frac{\alpha_n'}{\beta_n'} \in (0,1]$, we have $|S_n| \leq 1$. Then from the dominated convergence theorem since $\int 1 d\mu = 1 < \infty$, we have

$$\lim_{n\to\infty} \int_{-\infty}^{0} S_n^A d\mu = \int_{-\infty}^{0} \lim_{n\to\infty} S_n^A d\mu. \tag{47}$$

We have

$$\lim_{n\to\infty} S_n^A = \lim_{n\to\infty} -z_n + t_n\left(1 - \frac{\alpha_n'}{\beta_n'}\right) \tag{48}$$

Both α_n and β_n are random variables supported on $[0,1]$. Moreover, since limit exists for α_n, β_n, it also exists for $\alpha_n' \coloneqq \min\{\alpha_n, \beta_n\}$, similarly it exists for β_n'. Therefore,

$$\lim_{n\to\infty} -z_n + t_n(1 - \frac{\alpha_n'}{\beta_n'}) = x_1\left(1 - \frac{\lim_n \alpha_n'}{\lim_n \beta_n'}\right) \tag{49}$$

$$= x_1\left(1 - \frac{\lim_n \min\{\alpha_n, \beta_n\}}{\lim_n \max\{\alpha_n, \beta_n\}}\right) \tag{50}$$

$$= x_1\left(1 - \frac{\min\{\lim_n \alpha_n, \lim_n \beta_n\}}{\max\{\lim_n \alpha_n, \lim_n \beta_n\}}\right) \tag{51}$$

$$= x_1\left(1 - \frac{\min\{y_{0,1}, y_{1,1}\}}{\max\{y_{0,1}, y_{1,1}\}}\right) \tag{52}$$

where the last equation follows from the equations (32)-(34). Finally, we have that

$$\int_{-\infty}^{0} x_1\left(1 - \frac{\min\{y_{0,1}, y_{1,1}\}}{\max\{y_{0,1}, y_{1,1}\}}\right) d\mu \tag{53}$$

$$= \mathbb{P}\left(x_1\left(1 - \frac{\min\{y_{0,1}, y_{1,1}\}}{\max\{y_{0,1}, y_{1,1}\}}\right) \leq 0\right) \tag{54}$$

$$= \mathbb{P}\left(x_1\left(1 - \frac{\min\{y_{0,1}, y_{1,1}\}}{\max\{y_{0,1}, y_{1,1}\}}\right) = 0\right) = 0 \tag{55}$$

$$\tag{56}$$

where the last two equations follow from the fact that $x_1\left(1 - \frac{\min\{y_{0,1}, y_{1,1}\}}{\max\{y_{0,1}, y_{1,1}\}}\right)$ is supported in the interval $[0,1]$ and has an absolutely continuous measure, which implies that measure of a single point is zero.

Similarly, we can calculate

$$\mathbb{P}\left(S_n^A \geq 1\right) = \int_{1}^{\infty} S_n^A d\mu = 0, \tag{57}$$

which leads to $\mathbb{P}\left(\varepsilon_{z_n}^A\right) = 0$.

Next, we consider the same analysis for $\mathbb{P}\left(\varepsilon_{z_n}^B\right)$. One difference is the replacement of t_n with $1 - t_n$ which does not affect the derivation except for the replacement of x_1 with $1 - x_1$. Moreover, in the numerator within the paranthesis in (55, $\min\{y_0, 1, y_{1,1}\}$ is replaced with $1 - \max\{y_0, 1, y_{1,1}\}$ and similarly in the denominator $\max\{y_0, 1, y_{1,1}\}$ is replaced with $1 - \min\{y_0, 1, y_{1,1}\}$. It follows that $\mathbb{P}\left(\varepsilon_{z_n}^B\right) = 0$. This implies that $\lim_{n\to\infty} \mathbb{P}\left(LB_n \leq H(Z_n)\right) = 0$ concluding the proof.

□

7.7.1 Comments on Distinguishing Direct Graph from Latent Graph with Entropy

Consider the uniform generative model for the triangle graph. It is easy to see that in this case, in (36), t, α', β' become independent and uniformly distributed random variables with the compact support $[0, 1]$. One can calculate the distribution of this lower bound accordingly. This can be used to obtain the probability of identifiability between direct graph and the latent graph for a given upper bound on the entropy of the latent variable. We do not pursue this calculation here.

7.7.2 Extension to Z with k states

Consider the setting where Z has k states. We use the following notation in this section:

$$p(Z = i) = z^{(i)}. \tag{58}$$

First, note that the characterization of [31] is still applicable since they show increasing the dimension of Z to more than two states cannot reduce the minimum entropy. Similar to the above proof, we will assume a sequence of random variables Z_n. Let Z_n be a sequence of random variables with the pmf

$$p(Z_n = i) = z_n^{(i)}. \tag{59}$$

Note that $H(Z_n) \to 0$ if and only if $\exists i \in [k]$ such that $z_n^{(i)} \to 1$ and $(z_n^{(j)})_{j\neq i} \to \mathbf{0}$. In the following, we show that a similar analysis to the binary case goes through irrrespective of how $(z_n^{(j)})_{j\neq i}$ converges to the zero vector. Suppose without loss of generality $z_n^{(1)} \to 1$.

Due to the grouping rule of entropy, we have

$$H(Z_n) = H_b(z_n^{(1)}) + (1 - z_n^{(1)})H((w_n^i)_{2\leq i\leq k}), \tag{60}$$

where $w_n^{(i)} = \frac{z_n^{(i)}}{1-z_n^{(1)}}$. Let N be such that $z_N^{(1)} \geq 1 - \frac{\epsilon}{\log_2(k)}$. Then $z_n^{(1)} \geq 1 - \frac{\epsilon}{\log_2(k)}, \forall n > N$. Then we have $H(Z_n) \leq H_b(z_n^{(1)}) + \epsilon, \forall n \geq N$.

Now we can replicate the proof for the binary Z as follows. Let us define the events:

$$\begin{aligned}
\varepsilon_n^A &:= \{\text{Event that } H_b(A_n) \leq H_b(z_n^{(1)})\}.\\
\varepsilon_n^B &:= \{\text{Event that } H_b(B_n) \leq H_b(z_n^{(1)})\}.\\
\delta_n^A &:= \{\text{Event that } H_b(z_n^{(1)}) < H_b(A_n) \leq H(Z_n)\}.\\
\delta_n^B &:= \{\text{Event that } H_b(z_n^{(1)}) < H_b(B_n) \leq H(Z_n)\}.
\end{aligned}$$

By union bound

$$\mathbb{P}(LB_n \leq H(Z_n)) \leq \mathbb{P}(\varepsilon_n^A) + \mathbb{P}(\varepsilon_n^B) + \mathbb{P}(\delta_n^A) + \mathbb{P}(\delta_n^B) \tag{61}$$

We can write

$$\mathbb{P}(\delta_n^A) \leq \int_{H_b(z_n^{(1)})}^{H_b(z_n^{(1)})+\epsilon} H_b(A)d\mu. \tag{62}$$

It is easy to see that $\lim_{n\to\infty} \mathbb{P}(\delta_n^A) = 0$ since $\epsilon \to 0$ and the measaure induced by the generative model is absolutely continuous in the integral.

The rest of the analysis follows similarly to the binary case: We can obtain expressions for α, β using k terms instead of 2. In the limit, all but the term that contains $z_n^{(1)}$ go to zero and we can conclude the proof using the same arguments. □

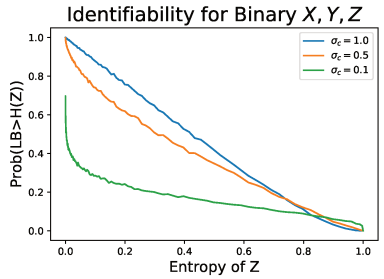


Figure 5: The measure of distributions from the triangle graph that does not fit to latent graph with a latent at least as simple as the true latent. As the entropy of the true latent goes to 0, this fraction goes to 1. This is precisely the measure of models which are identifiable with our entropic approach in the case of binary X, Y, Z.

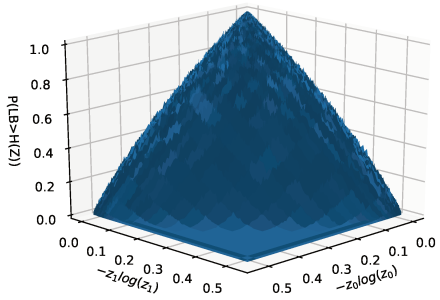


Figure 6: Fraction of distributions from the triangle graph that does not fit to latent graph with a latent at least as simple as the true latent for binary X, Y and ternary Z. As the entropy of the true latent goes to 0, this fraction goes to 1. This is precisely the fraction of models which are identifiable with our entropic approach in the case of binary X, Y, Z.

7.8 Entropic Identifiability for Binary Variables

For $H(Z) > 0$, we can approximate the fraction of identifiable causal models via simulations. We sample probability distributions from the uniform generative model. For each sample we check if the entropy of the true latent $H(Z)$ is strictly less than the common entropy of observed variables $G_1(X, Y)$. These are the models from triangle graph which cannot be fit onto latent graph with a low-entropy latent. Figure 5 shows this fraction. σ_c is the parameter of the Dirichlet distribution used to sample the conditional distributions from the Triangle graph. $\sigma_c = 1$ corresponds to uniform sampling model and smaller the σ_c more deterministic the conditional distributions are in the sampling model.

See Figure 6 for the corresponding plot for ternary Z.

7.9 $I(X;Y|Z)$ vs. $H(Z)$ Tradeoff Curve

Figure 7 shows the $I(X;Y|Z)$-$H(Z)$ tradeoff LatentSearch (Algorithm 1) obtains for a joint distribution sampled as follows: The distribution of Z as well as the conditional distributions $p(X|z), p(Y|z), \forall z$ are chosen uniformly at random over the simplex.

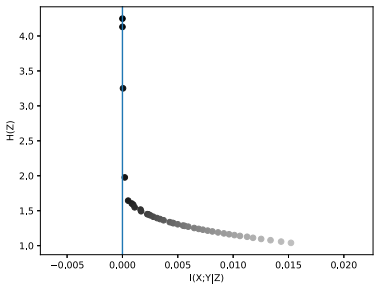


Figure 7: $I(X;Y|Z)$ vs. $H(Z)$ tradeoff curve obtained by LatentSearch (Algorithm 1) for an arbitrary joint $p(x,y)$ from the graph $X \leftarrow Z \rightarrow Y$. We observed that the curve's shape is consistent across many runs irrespective of the graph, although the crossing point where $I(X;Y|Z) = 0$ changes.

7.10 Complete Details of Experiments

In this section, we explain some of the key implementation details for the experiments in Section 5 that were left out from the main text due to space constraints.

Sampling a low-entropy latent: In many experiments, we sample distributions from either the latent graph or the triangle graph such that entropy of the latent confounder Z is small. For example, we enforce $H(Z) \leq \theta$ for varying thresholds θ in Figure 2c . We use a form of rejection sampling combined with sampling from Dirichlet distributions with low-entropy as follows:

Suppose, we need N samples where $H(Z) \leq \theta$. We initialize $\alpha^{(0)} = 1$ and obtain $10N$ samples from $Dir(\alpha^{(0)})$. If we have at least N samples where $H(Z) \leq \theta$, we are done. If not, we update α by halving it, i.e., $\alpha^{(1)} = 0.5\alpha^{(0)}$. The lower the α value, the lower-entropy distributions we will obtain from $Dir(\alpha)$. Then we repeat this process until iteration i, where at least N samples can be obtained from $10N$ samples using $Dir(\alpha^{(i)})$. We conclude by analyzing the histogram plots of $H(Z)$ that this method not only allows us to sample distributions where $H(Z) \leq \theta$ but also where $H(Z) \approx \theta$, providing us with a better control over the entropy of the latent confounder.

Choosing number of states of latent variable in LatentSearch: Recall that LatentSearch allows us to discover a tradeoff between $H(Z)$ and $I(X;Y|Z)$, which, combined with a $I(X;Y|Z)$ for conditional independence, can be used to approximate common entropy. Since only X, Y are observed, we do not know how many states k Z has. As pointed out in the main text, one can try all $k \leq mn$ without loss of generality, where m, n are the number of states of X, Y, respectively. However, in practice, this takes a long time. Furthermore, we identified that this is not necessary for the estimation of common entropy.

We observe that if we search over Z with very large number of states, e.g., $k = mn$, performance of LatentSearch does not improve compared to having $k = \min\{m, n\}$. This is because the number of optimization parameters increases significantly which may require many more iterations. It also slows down the algorithm. We observed that choosing $k = \max\{m, n\}$ provides the smallest entropy latents in practice. Therefore, we set $k = \max\{m, n\}$ in LatentSearch for our experiments.

Sampling DAGs for testing EntropicPC in Figure 4: We first sample Erdös-Rényi graphs with parameter 0.2. Since there are 10 nodes, this corresponds to an average degree of 2 per node. Note that these graphs are undirected. We need to make them directed and ensure there are no cycles. For this, we randomly picked a total order between the nodes and directed the edges respecting that total order. It can be easily shown that the resulting graphs have no cycles.

Sampling joint distributions for a given DAG in Figure 4: For every DAG we generate, we need to obtain a joint distribution from which we sample a dataset. To obtain a distribution for a given graph, we employ a method from Chickering and Meek [10]. It is known that constraint-based methods require the data to be faithful to the graph, i.e., every pair of variables that are connected in the graph should be dependent. This notion should also be true under any conditioning set. In practice, this does not always hold. Specifically, nodes that are far away from each other in the graph might be almost statistically independent. To ensure faithfulness in practice, Chickering and Meek

use a method to sample conditional distributions for a given DAG [10]. In summary, it ensures that parent-child relations are far from uniform. The details of their sampling method, which we also use, are as follows:

For a variable X with m states, they define the vector $\mathbf{v} = \frac{1}{T}(\frac{1}{1}, \frac{1}{2}, \ldots, \frac{1}{m}), T = \sum_{i=1}^{m} \frac{1}{i}$. For the j^{th} instantiation of the parent set of X, $pa_x^{(j)}$, they use the vector $\mathbf{v}_j$ which is the j-shifted version of $\mathbf{v}$. The shifts are cyclic in the sense that $\mathbf{v}_m = \mathbf{v}$. They later sample $p(X|pa_x^{(j)})$ from the Dirichlet distribution with parameter vector $\mathbf{v}_j$. When each coordinate parameter of Dirichlet is identical, the expected distribution is uniform. When Dirichlet is sampled with parameters $\mathbf{v}_j$ as given above, each coordinate has a different parameter. Indeed, the expected distribution becomes $\mathbf{v}_j$ rather than uniform. Therefore, this method ensures that $p(x|pa_x^j)$ is typically far from uniform, encouraging strong dependence between parents and the children in the graph.

Details specific to Figure 2a: We sampled 100 distributions from the latent graph for each value of n, where X, Y, Z all have n states. In each distribution, we ensure that $H(Z) \leq 1$ using the low-entropy sampling method described above. We use LatentSearch on 50 different values of β, uniformly spaced in the interval $[0, 0.1]$. We set the latent variable for LatentSearch to n number of states. We run LatentSearch for 500 iterations each time. We used the conditional mutual information threshold of 0.001: In other words, of the algorithm outputs for the 50 β values used, we pick the smallest entropy Z discovered by the algorithm among those that ensure $I(X;Y|Z) \leq 0.001$. We then compare this value with the entropy of the true latent confounder.

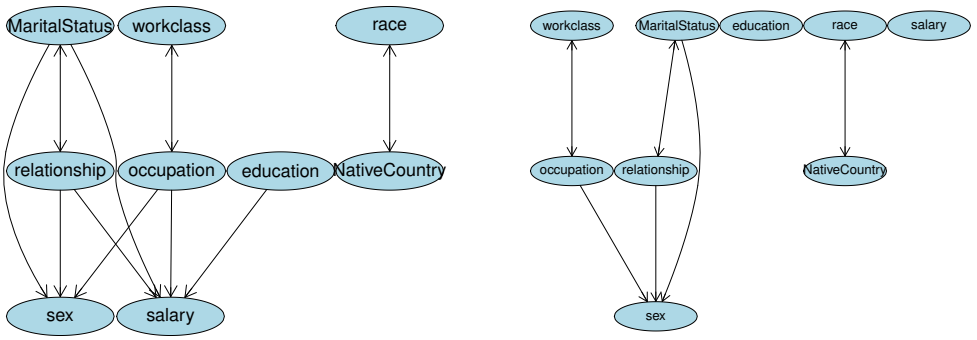
Details specific to Figure 2b: We sample 1000 distributions from the triangle graph. As mentioned in the main text, we use LatentSearch output to approximate common entropy. The settings for LatentSearch are the same as above, i.e., we use 50 β values uniformly spaced in the range $[0, 0.1]$, use n states for the latent and run the algorithm for 500 iterations. Entropy recovered by LatentSearch for a pair X, Y is then compared with $\min\{H(X), H(Y)\}$. The $y-$axis shows that fraction of times the reconstructed latent has entropy of at least $\alpha \min\{H(X), H(Y)$ for different values of α.

Details specific to Figure 2c: For this figure we sample 1000 distributions from both triangle graph and the latent graph for various upper bounds on the entropy of Z. Low-entropy sampling is done as explained before. Finally, Algorithm 2 is used to identify the true causal graph with $\theta = 0.8 \min\{H(X), H(Y)\}$. LatentSearch settings used within Algorithm 2 are as given previously.

Details specific to Figure 3a: Tuebingen dataset consists of around 100 real cause-effect pairs. We run LatentSearch to understand whether real cause-effect pairs can be made independent by low-entropy variables. As explained in the main text, we used different conditional independence thresholds. Visual inspection of $I - H$ curves suggest that 0.001 is a good threshold for this dataset. This can be done by checking, for the given range of β values, where the curve disengages form the $x = 0$ axis. We used 100 β values in the range $[0, 0.1]$ and run the LatentSearch algorithm for 1000 iterations for this experiment.

Details specific to Figure 3b: This figure is an example of the tradeoff curve LatentSearch discovers for various values of β. Each dot corresponds to a joint distribution $p(x, y, z)$ constructed by LatentSearch for a given value of β after a certain number of iterations. As can be seen from (2), smaller β values enforce smaller $I(X;Y|Z)$. The horizontal line indicates $\min\{H(X), H(Y)\}$. X, Y can always be separated with this much entropy since by definition $X \perp\!\!\!\perp Y\,|X$ and $X \perp\!\!\!\perp Y\,|Y$. Ideally, i.e., with infinite samples and infinitely many β values, the point that intersects the $x = 0$ line (i.e., the $y-$ axis) should give the common entropy. To account for finite-sample effects, we use a different horizontal line, which we call conditional mutual information threshold, as described before.

Details specific to Figure 3c: We sample from the graph $X \to Z \to Y$ and investigate $H(Z)$. Note that Z acts as a mediator if it is not observed. Our goal is to understand if it is typical to have low-entropy mediators. We set the dimensions of X, Y, Z to n. If Z has k states, $H(Z) \leq \log(k)$. Our goal is to demonstrate that unless k is a constant, $H(Z)$ scales similar to $H(X), H(Y)$. Most of the details of this experiment are provided in the main text. Note that when $\alpha_{Dir} \leq \frac{1}{n}$ for a distribution with n states, a sample from Dirichlet distribution typically looks very peaked, i.e., it has very high probability for one of the states, and very low probabilities for the rest. When such a low α_{Dir} is used to sample the conditional of $p(Z|x)$ for every value of x, X and Z are almost deterministically related, i.e., there is almost no additional entropy introduced in the system. We show that even then the entropy of the mediator scales. Larger α_{Dir} values will give distributions that are closer to uniform, which in turn will make Z close to uniform and have $\log(n)$ entropy.



(a) (b)

Figure 8: [Section 5.5] Output of **(a)** EntropicPC and **(b)** PC in ADULT Data.

Details specific to Figure 4: Our goal is to demonstrate how well output of EntropicPC approximates a true causal graph by checking graphical distances between the skeleton and essential graph. Essential graph is the mixed graph where undirected edges show the edges that cannot be learned whereas directed edges show the edges that can be learned. Structural Hamming Distance counts the number of edges that should be reverted, added or removed to change the output graph into the true graph. For skeleton discovery, we see edge discovery as a classification problem and calculate the F-score as an established summary of the classifier performance. The graph and distribution sampling are described above. We sample a dataset with 100000 variables and for each figure, we subsample varying number of samples from this dataset without replacement. This is repeated for 100 different graphs, and their corresponding distributions.

7.11 EntropicPC and PC on ADULT Dataset

Due to space constraints in the main text, we provide the outputs of PC and EntropicPC algorithms for the ADULT dataset in this section. The results are given in Figure 8.

Note that the bidirected edges represent undirected, i.e., unoriented edges. Even though the true causal graph is not known, we can easily conclude that EntropicPC discovers a much more reasonable graph: *salary* is caused by *education,occupation* whereas PC misses both edges. Both algorithms seems to suffer from unfaithful data - *sex* is not required to separate *marital-status* and *occupation* whereas we expect it to since it should be a source node. This drives both algorithms to orient *sex* as a collider.

7.12 EntropicPC on Line and Collider Graphs

To demonstrate effectiveness of EntropicPC compared to PC on the simplest possible graph, we conducted the experiments of Figure 4 on the line graph $X \to Y \to Z$ and the collider graph $X \to Y \leftarrow Z$. The results are given in Figures 9 and 10, respectively.

7.13 Comparing LatentSearch with EM, NMF and Gradient descent

7.13.1 Comparison to gradient descent

Instead of using LatentSearch for minimizing the loss in (2), one can use gradient descent. Even though the objective is not convex, gradient descend will still output a stationary point if it converges. However gradient descend comes with many practical issues, as we detail in the following, and support with experiments.

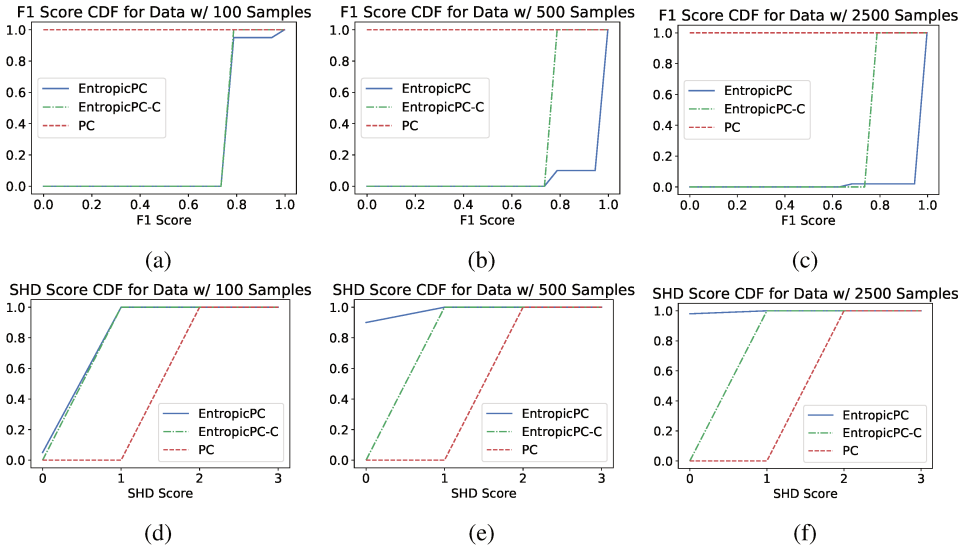


Figure 9: Performance of EntropicPC, EntropicPC-C and PC in the graph $X \to Y \to Z$. All three methods perform identically for 10000 samples (not shown). Both EntropicPC and EntropicPC-C consistently outperforms baseline PC.

First, we observed that iterative update step is slightly faster than the gradient descent step: Average time for iterative update:0.000063 seconds. Average time for gradient update: 0.000078 seconds. More importantly, gradient descent takes much longer to converge and does not even achieve the same performance.

As observed in Figure 11, gradient descent converges only after 350000 iterations, whereas we observed that iterative update converges after around 200 iterations. Based on the average update times, this corresponds to a staggering difference of 0.01 seconds for the iterative algorithm vs. 27.3 seconds for the gradient descent algorithm. Although these results are for when $n = m = k = 5$ states, we observed single iterative update to be faster than single gradient update, giving similar performance comparison results for $n = m = k = 80$ states.

The above result is for a constant step size of 0.001. With smaller step size, convergence slows down even further. With larger step size, gradient descent does not converge.

7.13.2 Comparison with EM algorithm

EM is the first algorithm suggested for solving the pLSA problem [19]. For the details of EM within this framework, please see [19]. However the EM algorithm for pLSA problem does not have any incentive to minimize the entropy of the latent factor.

In order to see how EM affects the entropy of the discovered latent variable, we run EM algorithm by initializing it at the points that are output by LatentSearch (Algorithm 1). Results are illustrated in Figure 12. We observe that the points obtained in the $I - H$ plane migrate towards $I(X;Y|Z) = 0$ line, while staying above what we believe is a fundamental lower bound curve. This step does not improve our algorithm, as it increases the entropy of the latent variables.

7.13.3 Comparison with NMF

Consider the joint distribution matrix $\mathbf{M}$. Suppose we find an approximation to this matrix as $\mathbf{M} \approx \mathbf{UV}$ where the common dimension of $\mathbf{U}, \mathbf{V}$ is k through NMF. This is equivalent to setting the dimension of the latent variable to k. This can be seen as a hard entropy threshold on the entropy of the latent factor since $H(Z) \leq \log(k)$. We can sweep through different dimensions and see how NMF performs compared to LatentSearch (Algorithm 1). Note that NMF is in general hard to solve. A commonly used approach is the iterative algorithm: Initialize $\mathbf{U}_0, \mathbf{V}_0$. Find the best $\mathbf{U}_1$ such that $\mathbf{M} \approx \mathbf{U}_1\mathbf{V}_0$. Then find the best $\mathbf{V}_1$ such that $\mathbf{M} \approx \mathbf{U}_1\mathbf{V}_1$ and iterate. In the experiments, we used this iterative algorithm together with l_1 loss.

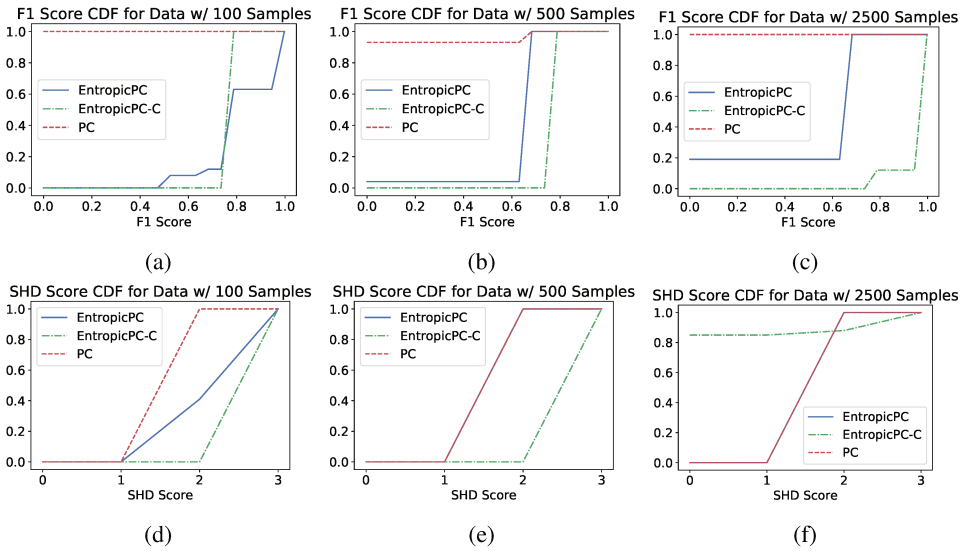


Figure 10: Performance of EntropicPC, EntropicPC-C and PC in the graph $X \to Y \leftarrow Z$. All three methods perform identically for 10000 samples (not shown). Different from the line graph, for collider graph in the very small sample regime, even though the proposed methods significantly outperforms PC in terms of skeleton discovery, PC gets a slight edge in performance in terms of SHD. Note that PC tends to declare variables independent in the small sample regime. It is likely that for 100 samples, PC often estimates empty graph, which has SHD of 2 from the essential graph. This explains the discrepancy between the performance for F1 and SHD scores for 100 samples.

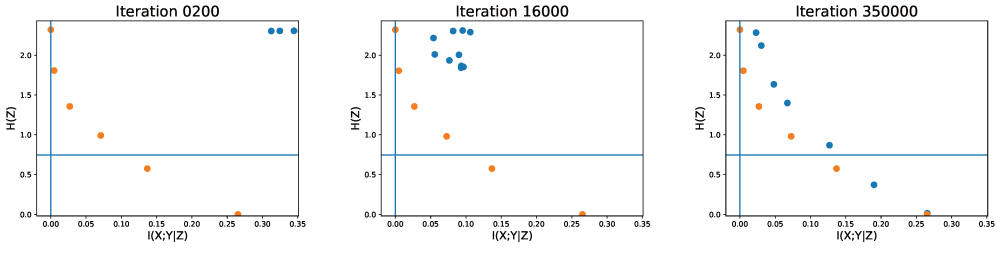
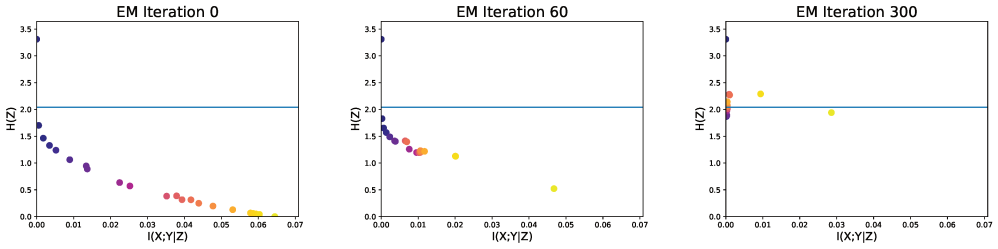
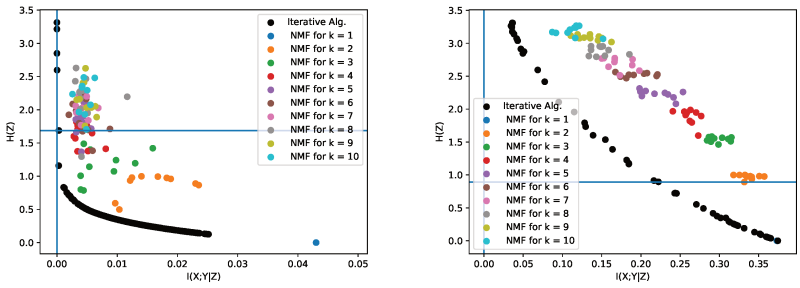


Figure 11: Comparison of the iterative algorithm with gradient descent. Blue points show the trajectory of gradient descent, whereas orange points show the trajectory for Algorithm 1 for 10 randomly initialized points with different β values in loss (2). Gradient descent takes 350,000 iterations to converge whereas iterative algorithm converges in about 200 iterations. Moreover, the points achieved by iterative algorithm are strictly better than gradient descent after convergence.



(a) (b) (c)

Figure 12: Applying EM to the output of iterative algorithm migrates points to $I(X;Y|Z=0)$ line: (a) Latent variables discovered by LatentSearch (Algorithm 1) shown on the $I(X;Y|Z) - H(Z)$ plane. (b,c) After applying EM algorithm on the points in (a) after 60 and 300 iterations. Observe that the points always remain above the line depicted by LatentSearch (Algorithm 1).



(a) Causal Graph $X \leftarrow Z \rightarrow Y$ (b) Causal Graph $X \leftarrow Z \rightarrow Y, X \rightarrow Y$

Figure 13: Comparison of the iterative algorithm to NMF for when $|X| = |Y| = 20, |Z| = 10$. When the true model comes from the causal graph $X \leftarrow Z \rightarrow Y$ in (a), iterative algorithm successfully finds latent variables that with entropy at most true latent entropy (shown as blue horizontal line), whereas NMF cannot achieve the same performance, irrespective of the dimension restriction to the latent variable. In (b) data comes from the causal model $X \leftarrow Z \rightarrow Y, X \rightarrow Y$. Although neither algorithm can identify a latent factor that makes X, Y conditionally independent (vertical blue line), iterative algorithm finds strictly better latent factors in terms of both small entropy and conditional mutual information between X, Y.