Collaborative Causal Discovery with Atomic Interventions

Raghavendra Addanki
University of Massachusetts, Amherst
raddanki@cs.umass.edu

Shiva Prasad Kasiviswanathan
Amazon
kasivisw@gmail.com

Abstract

We introduce a new Collaborative Causal Discovery problem, through which we model a common scenario in which we have multiple independent entities each with their own causal graph, and the goal is to simultaneously learn all these causal graphs. We study this problem without the causal sufficiency assumption, using Maximal Ancestral Graphs (MAG) to model the causal graphs, and assuming that we have the ability to actively perform independent single vertex (or atomic) interventions on the entities. If the M underlying (unknown) causal graphs of the entities satisfy a natural notion of clustering, we give algorithms that leverage this property, and recovers all the causal graphs using roughly logarithmic in M number of atomic interventions per entity. These are significantly fewer than n atomic interventions per entity required to learn each causal graph separately, where n is the number of observable nodes in the causal graph. We complement our results with a lower bound and discuss various extensions of our collaborative setting.

1 Introduction

In this paper, we introduce a new model for *causal discovery*, the problem of learning all the causal relations between variables in a system. Under certain assumptions, using just observational data, some ancestral relations as well as certain causal edges can be learned, however, many observationally equivalent structures cannot be distinguished [Zhang, 2008a]. Given this issue, there has been a growing interest in learning causal structures using the notion of an *intervention* described in the Structural Causal Models (SCM) framework introduced by Pearl [2009].

As interventions are expensive (require carefully controlled experiments) and performing multiple interventions is time-consuming, an important goal in causal discovery is to design algorithms that utilize simple (preferably, single variable) and fewer interventions [Shanmugam et al., 2015]. However, when there are latents or unobserved variables in the system, in the worst-case, it is not possible to learn the exact causal DAG without intervening on every variable at least once. Furthermore, multivariable interventions are needed in presence of latents [Addanki et al., 2020].

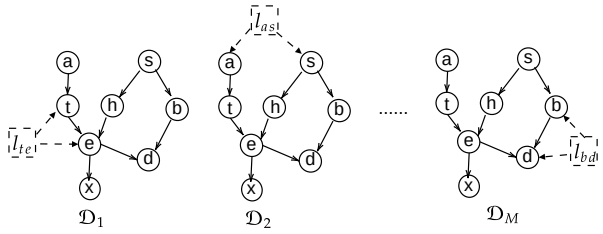


Figure 1: Examples of M causal graphs constructed from Lung Cancer dataset [Lauritzen and Spiegelhalter, 1988]. Here, the causal graphs differ only in the presence of latents (nodes with dotted square box), but they could differ elsewhere too.

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

On the other hand, in a variety of applications, there is no one true causal structure, different entities participating in the application might have different causal structures [Gates and Molenaar, 2012, Ramsey et al., 2011, Joffe et al., 2012]. For example, see figure 1. In these scenarios, generating a single causal graph by pooling data from these different entities might lead to flawed conclusions [Saeed et al., 2020]. Allowing for interventions, we propose a new model for tackling this problem, referred here as *Collaborative Causal Discovery*, which in its simplest form states that: given a collection of entities, each associated with an individual unknown causal graph and generating their own independent data samples, learn all the causal graphs while minimizing the number of *atomic* (single variable) interventions for every entity. An underlying assumption is that each entity on its own generates enough samples in both the observational and interventional settings so that conditional independence tests can be carried out *accurately* on each entity separately. To motivate this model of collaborative causal discovery, let us consider two different scenarios.

(a) Consider a health organization interested in controlling incidence of a particular disease. The organization has a set of M individuals (entities) whose data it monitors and can advise interventions on. Each individual is an independent entity that generates its own set of separate data samples[1]. In a realistic scenario, it is highly unlikely that all the M individuals share the same causal graph (e.g., see Figures 3a and 3b from Joffe et al. [2012] in Appendix A). It would be beneficial for the organization to collaboratively learn all the causal graphs together. The challenge is, *a priori* the organization does not know the set of possible causal graphs or which individual is associated with which graph from this set.

(b) An alternate setting is where, we have M companies (entities) wanting to work together to improve their production process. Each company generates their own data (e.g., from their machines) which they can observe and intervene on [Nguyen et al., 2016]. Again if we take the M causal graphs (one associated with each company) it is quite natural to expect some variation in their structure, more so because we do not assume *causal sufficiency* (i.e., we allow for latents). Since interventions might need expensive and careful experimental organization, each company would like to reduce their share of interventions.

The collaborative aspect of learning can be utilized if we assume that there is some underlying (unknown) clustering/grouping of the causal graphs on the entities.

Our Contributions. We formally introduce the collaborative causal discovery problem in Section 2. We assume that we have a collection of M entities that can be partitioned into k clusters such that any pair of entities belonging to two different clusters are separated by large distance (see Definition 2.1) in the causal graphs. Due to presence of latents variables, we use a family of mixed graphs known as *maximal ancestral graphs* (MAGs) to model the graphs on observed variables. Each entity is associated with a MAG.

In this paper, we focus on designing algorithms that have *worst-case* guarantees on the number of atomic interventions needed to recover (or approximately recover) the MAG of each entity. We assume that there are M MAGs one for each entity over the same set of n nodes. Learning a MAG with atomic interventions, in worst case requires n interventions (see Proposition 3.2). We show that this bound can be substantially reduced if the M MAGs satisfy the property that every pair of MAGs from different clusters have *at least* αn nodes whose direct causal relationships are different. We further assume that entities belonging to same cluster have similar MAGs in that every pair of them have *at most* βn ($\beta < \alpha$) nodes whose direct causal relationships are different. We refer to this clustering of entities as (α, β)-clustering (Definition 2.2). A special but important case is when $\beta = 0$, in which case all the entities belonging to the same cluster have the same causal MAG (referred to as α-clustering, Definition 2.3). An important point to notice is that while we assume there is a underlying clustering on the entities, it is *learnt* by our algorithms. Similar assumptions are common for recovering the underlying clusters, in many areas, for e.g., crowd-sourcing applications [Ashtiani et al., 2016, Awasthi et al., 2012].

We first start with the observation that under (α, β)-clustering, even entities belonging to the same cluster could have a different MAG, which makes exact recovery hard without making a significant number of interventions per entity. We present an algorithm that using at most $O(\Delta \log(M/\delta)/(\alpha - \beta)^2)$ many interventions per entity, with probability at least $1 - \delta$ (over only the randomness of the algorithm), can provably recover an *approximate* MAG for each entity. The approximation is such

[1] As is common in causal discovery, for the underlying conditional independence tests, the data is assumed to be i.i.d. samples from the interventional/observational distributions.

that for each entity we generate a MAG that is at most βn node-distance from the true MAG of that entity (see Section 3). Here, Δ is the maximum undirected degree of the causal MAGs. Our idea is to first recover the underlying clustering of entities by using a randomized set of interventions. Then, we distribute the interventions across the entities in each cluster, thereby, ensuring that the number of interventions per entity is small. By carefully combining the results learnt from these interventions we construct the approximate MAGs. For the number of interventions, the linear dependence on Δ is not uncommon for learning causal graphs [Kocaoglu et al., 2017]. Moreover, most real-world causal bayesian networks are known to have small maximum degrees (see section 5).

Under the slightly more restrictive α-clustering assumption, we present algorithms that can *exactly* recover all the MAGs using at most $\min\left\{O(\Delta \log(M/\delta)/\alpha), O(\log(M/\delta)/\alpha + k^2)\right\}$ many interventions per entity (see Section 4). Again, randomization plays an important role in our approach.

Complementing these upper bounds, we give a lower bound using Yao's minimax principle [Yao, 1977] that shows for any (randomized or deterministic) algorithm $\Omega(1/\alpha)$ interventions per entity is required for this causal discovery problem. This implies the $1/\alpha$ dependence in our upper bound in the α-clustering case is optimal.

Finally, a note about parameters. The (α, β)-clustering is universal, in the sense that *any* collection of MAGs will satisfy the (α, β)-clustering property for some value of α, β (with $\alpha > \beta$). Ideally, we would like in our problem instance, α to be close to 1 and β to be close to 0. In most real-world applications, we would also expect k to be relatively small and $M \gg n, k$.

In Section 5, we show experiments on data generated from both real and synthetic networks with added latents and demonstrate the efficacy of our algorithms for learning the underlying clustering and the MAGs.

Related Work. A number of algorithms, working under various assumptions, for learning causal graph (or a causal DAG) using interventions have been proposed in the literature, e.g., [Eberhardt, 2007, Hyttinen et al., 2013, Hu et al., 2014, Shanmugam et al., 2015, Kocaoglu et al., 2017, Ghassami et al., 2018, Lindgren et al., 2018, Acharya et al., 2018, Bello and Honorio, 2018, Kocaoglu et al., 2019, Greenewald et al., 2019, Jaber et al., 2020, Addanki et al., 2020, 2021, Tadepalli and Russell, 2021]. Saeed et al. [2020] consider a model where the observational data is from a mixture of causal DAGs, and outline ideas that recover a *union graph* (up to Markov equivalence) of these DAGs, without any interventions. Our setting is not directly comparable to theirs, as we have entities generating data and doing conditional independence tests independently (no pooling of data from entities), but show stronger guarantees for recovering causal graphs, assuming atomic interventions.

2 Our Model and Problem Statement

In this section, we introduce the collaborative causal discovery problem. We start with some notations.

Notation. Following the SCM framework [Pearl, 2009], we represent the set of random variables of interest by $V \cup L$ where V represents the set of endogenous (observed) variables that can be measured and L represents the set of exogenous (latent) variables that cannot be measured. We do not deal with selection bias in this paper. Let $|V| = n$.

We assume that the causal Markov condition and faithfulness holds for both the observational and interventional distributions [Hauser and Bühlmann, 2012]. We use conditional independence (CI) tests of the form $u \perp\!\!\!\perp v \mid Z$ or $u \perp\!\!\!\perp v \mid \mathrm{do}(w), Z$, for some $u, v, w \in V$ and $Z \subseteq V$ (See Appendix A for more details).

Throughout this paper, unless otherwise specified, a path between two nodes is an undirected path. A path of only directed edges is called a directed path. u is called an ancestor of v and v a descendant of u if $u = v$ or there is a directed path from u to v. A directed cycle occurs in G when $u \to v$ is in G and v is an ancestor of u.

Our Model. We assume that we have access to M entities labeled $1, \ldots, M$, each of which can independently generate their own observational and interventional data. Each entity i has an associated causal DAG $\mathcal{D}_i$ over $V \cup L_i$, where L_i represents the latent variables of entity i. In modeling the problem of causal discovery, complications arise in at least two ways:

(i) Latents. We allow some variables (called latents) in the causal DAG to be unobservable. As regular DAGs are not sufficient to represent the observed distribution when there are latents, we use *ancestral graphical models* that have been proposed as an elegant and useful surrogate for DAG models with latent variables [Richardson and Spirtes, 2002].

A mixed graph containing directed ($\leftarrow$) and bidirected ($\leftrightarrow$) edges is said to be *ancestral* if it has no directed cycles, and whenever there is a bidirected edge $u \leftrightarrow v$, then there is no directed path from u to v or from v to u. An ancestral graph on V (observables) is said to be *maximal*, if, for every pair of nonadjacent vertices u, v, there exists a set $Z \subset V$ with $u, v \notin Z$ such that u and v are m-separated (similar to d-separation, see Definition A.2) conditioned on Z. Every DAG with latents (and selection variables) can be transformed into a unique maximal ancestral graph (MAG) over the observed variables [Richardson and Spirtes, 2002].

(ii) Uniqueness. Secondly, with just observational data, if the MAGs $\mathcal{M}_1, \ldots, \mathcal{M}_M$ are Markov equivalent, then, without additional strong assumptions they cannot be distinguished, even if they are all structurally different. To overcome the problem of being not identifiable within an equivalence class, we allow for interventions on observed variables. In particular, we focus on atomic interventions in this paper, which are the simplest and most commonly used intervention type, modeled through the do-operator [Pearl, 1995]. As it turns out, Maximal Ancestral Graphs (MAGs) are uniquely identifiable using atomic interventions.[2]

Our objective will be to minimize these interventions. In particular, since each of these entities independently generate their own data, so we aim to reduce the number of interventions needed per entity. In causal discovery, minimizing the number of interventions while ensuring that they are of small size is an active research area [Pearl, 1995, Shanmugam et al., 2015, Ghassami et al., 2018, 2019].

Given the M entities, let $\mathcal{M}_i$ denote the MAG associated with entity i (the MAG constructed from the DAG $\mathcal{D}_i$). Our goal is to collaboratively learn all these MAGs $\mathcal{M}_1, \ldots, \mathcal{M}_M$ while minimizing the maximum number of interventions per entity.

To facilitate this learning, we make a natural underlying clustering assumption that partitions the entities based on their respective MAGs such that: (i) any two entities belonging to the same cluster have MAGs that are "close" to each other, (ii) any two entities belonging to different clusters have MAGs that are "far" apart. Before stating this assumption formally, we need some definitions.

For MAG $\mathcal{M}_i = (V, E_i)$, we denote the children (through outgoing edges), parent (through incoming edges), and spouse (through bidirected edges) of a node $u \in V$ as

$$\text{ch}_i(u) = \{v \mid u \to v \in E_i\},\ \text{pa}_i(u) = \{v \mid u \leftarrow v \in E_i\},\ \text{sp}_i(u) = \{v \mid u \leftrightarrow v \in E_i\}. \tag{1}$$

Also, define an incidence set for a vertex $u \in V$ which contains an entry (v, type) for every node v adjacent to u as

$$N_i(u) = \left\{ \begin{array}{ll} (v, \text{tail}) & \text{if } u \to v \in E_i \\ (v, \text{head}) & \text{if } u \leftarrow v \in E_i \\ (v, \text{bidirected}) & \text{if } u \leftrightarrow v \in E_i \end{array} \right\}. \tag{2}$$

Note that $|N_i(u)|$ is the undirected degree of u in $\mathcal{M}_i$. We now define a distance measure between MAGs that captures structural similarity between them.

Definition 2.1. *Given two MAGs $\mathcal{M}_i = (V, E_i)$ and $\mathcal{M}_j = (V, E_j)$, define the node-difference as the set:* $\text{diff}(\mathcal{M}_i, \mathcal{M}_j) = \{u \in V \mid N_i(u) \neq N_j(u)\}$, *and the node-distance as the cardinality of this set:* $d(\mathcal{M}_i, \mathcal{M}_j) = |\text{diff}(\mathcal{M}_i, \mathcal{M}_j)| = |\{u \in V \mid N_i(u) \neq N_j(u)\}|$.

Intuitively, the node distance captures the number of nodes whose incidence relationships differ. It is easy to observe that the node distance is a distance metric, and captures a strong structural similarity between the graphs. Two graphs $\mathcal{M}_i, \mathcal{M}_j$ are identical iff $d(\mathcal{M}_i, \mathcal{M}_j) = 0$. For e.g., in Figure 2, we have two MAGs that satisfy $d(\mathcal{M}_{12}, \mathcal{M}_{13}) = 2$ as $\text{diff}(\mathcal{M}_{12}, \mathcal{M}_{13}) = \{x, z\}$, where $d(\mathcal{M}_{12}, \mathcal{M}_{21}) = 3$ as $\text{diff}(\mathcal{M}_{12}, \mathcal{M}_{21}) = \{x, y, z\}$. We are now ready to define a simple clustering property on MAGs.

[2]However, in the presence of latents, even with power of atomic interventions, the structure of a causal DAG is not uniquely identifiable. (see, e.g., In Figure 4 in Appendix A). Similarly, we can show that using single vertex interventions, we also cannot exactly recover a wider class of acyclic graphs like ADMGs (Acyclic Directed Mixed Graphs).

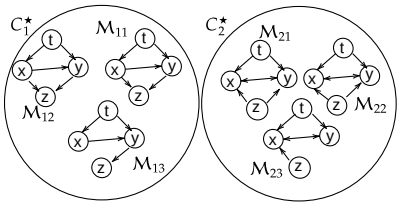


Figure 2: MAGs with $(\alpha = 0.75, \beta = 0.5)$-clustering. Every pair of graphs in $C_1^\star$ and $C_2^\star$ differ in at least $3(= 0.75 \times 4)$ nodes, while pairs of graphs within clusters differ by at most $2(= 0.5 \times 4)$ nodes.

Algorithm 1 IDENTIFY-OUTNBR $(\mathcal{U}_i, u)$

Input: node $u \in V$, PAG $\mathcal{U}_i$ of entity i
Output: $\text{ch}_i(u)$
$\text{ch}_i(u) = \{v \mid u \to v \in \mathcal{U}_i\}$
for $v \in \Gamma_i(u)$ such that $u \circ\!\!-\!\!\circ v$ or $u \circ\!\!\to v \in \mathcal{U}_i$ **do**
  if $u \not\perp\!\!\!\perp v \mid \text{do}(u)$ **then**
    $\text{ch}_i(u) \leftarrow \text{ch}_i(u) \cup \{v\}$
  end if
end for
Return $\text{ch}_i(u)$

Algorithm 2 IDENTIFY-BIDIRECTED $(\mathcal{U}_i, u)$

Input: node $u \in V$, PAG $\mathcal{U}_i$ of entity i
Output: $\text{sp}_i(u)$
$\text{sp}_i(u) = \{v \mid u \leftrightarrow v \in \mathcal{U}_i\}$
for $v \in \Gamma_i(u)$ such that $u \circ\!\!-\!\!\circ v$ or $u \leftarrow\!\!\circ v$ or $u \circ\!\!\to v \in \mathcal{U}_i$ **do**
  if $u \perp\!\!\!\perp v \mid \text{do}(u)$ and $u \perp\!\!\!\perp v \mid \text{do}(v)$ **then**
    $\text{sp}_i(u) \leftarrow \text{sp}_i(u) \cup \{v\}$
  end if
end for
Return $\text{sp}_i(u)$

Definition 2.2 ((α, β)-clustering)**.** *Let $\mathcal{M}_1, \ldots, \mathcal{M}_M$ be a set of M MAGs. We say that this set of MAGs satisfy the (α, β)-clustering property, with $\alpha > \beta \geq 0$, if there exists a partitioning of $[M]$ into sets (clusters) $C_1^\star, \ldots, C_k^\star \subset [M]$ (for some $k \in \mathbb{N}$) such that for all $(i, j) \in [M] \times [M]$:*

(i) if i and j belong to same set (cluster), then $d(\mathcal{M}_i, \mathcal{M}_j) \leq \beta n$;
(ii) if i and j belong to different sets (clusters), then $d(\mathcal{M}_i, \mathcal{M}_j) \geq \alpha n$.

Under this definition, all the M MAGs could be different. See, e.g., Figure 2. With right setting of $\alpha > \beta$ we can capture any set of possible M MAGs. Therefore, an algorithm such as FCI [Spirtes et al., 2000], that constructs PAGs might not be able to recover the clusters, as all the PAGs could be different, and the node-distance between PAGs does not correlate well with the node-distance between corresponding MAGs (e.g., see Figure 5 in Appendix A). We use the PAGs generated by FCI as a starting point for all our algorithms and further refine them. We assume that PAGs generated are correct (see Appendix B.1 for additional details). With this discussion, we introduce our collaborative causal graph learning problem as follows:

Assumption: MAGs $\mathcal{M}_1, \ldots, \mathcal{M}_M$ (associated with entities $1, \ldots, M$ respectively) satisfying the (α, β)-clustering property
Access to each entity: Through conditional independence (CI) tests on observational and interventional distributions. Each entity generates their own (independent) data samples.
Goal: Learn $\mathcal{M}_1, \ldots, \mathcal{M}_M$ while minimizing the max. number of interventions per entity.

An interesting case of the Definition 2.2 is when $\beta = 0$.

Definition 2.3 (α-clustering)**.** *We say a set of MAGs $\mathcal{M}_1, \ldots, \mathcal{M}_M$ satisfy the α-clustering property, if and only if they satisfy $(\alpha, 0)$-clustering property.*

Note that α-clustering is a natural property, wherein each cluster is associated with a single unique MAG, and all entities in the cluster have the same MAGs, and same conditional independences.

3 Causal Discovery under (α, β)-Clustering Property

In this section, we present our main algorithm for collaboratively learning causal MAGs under the (α, β)-clustering property. Missing details from this section are presented in Appendix C.

Definition 3.1 (Partial Ancestral Graph (PAG))**.** *Let $[\mathcal{M}_i]$ denote the Markov equivalence class of the MAG $\mathcal{M}_i$ and represented by the Partial Ancestral Graph (or PAG) $\mathcal{U}_i = (V, \widehat{E}_i)$. Edges $\widehat{E}_i$ have three kinds of endpoints given by arrowheads $(\leftarrow)$, circles $(\circ\!\!-)$ and tails $(-)$.*

All our algorithms are randomized, and succeed with high probability over the randomness introduced by the algorithm. The idea behind all our algorithms is to first learn the true clusters $C_1^\star, \ldots, C_k^\star$ using very few interventions. Once the true clusters are recovered, the idea is to distribute the interventions across the entities in each cluster and merge the results learned to recover the MAGs (Section 3.2). For our algorithms, a lower bound for α and upper bound for β is sufficient. In practice, a clustering of the PAGs (generated from FCI algorithm) can provide guidance about these bounds on α, β, or if we have additional knowledge that $\alpha \in [1-\epsilon, 1]$ and $\beta \in [0, \epsilon]$ for some constant $\epsilon > 0$, then, we can use binary search, that increases our intervention bounds by $\log^2(n\epsilon)/(1-2\epsilon)^2$ factor. It is important to note that none of our algorithms require the knowledge of k

Helper Routines. Let $\Gamma_i(u)$ denote all nodes that are adjacent to u in the PAG $\mathcal{U}_i$, i.e., $\Gamma_i(u) = \{v \mid (u,v) \in \widehat{E}_i\}$. Given the PAG $\mathcal{U}_i$, Algorithm IDENTIFY-OUTNBR identifies all the outgoing neighbors of any node u in $\mathcal{M}_i$. We look at edges of the form $u \circ\!\!-\!\!\circ v$ or $u \circ\!\!\rightarrow v$ in $\mathcal{U}_i$ incident on u, and identify if $u \to v$ using the CI-test $u \perp\!\!\!\perp v \mid \mathrm{do}(u)$. This is based on the observation that any node v that is a descendant of u (including $\mathrm{ch}_i(u)$) satisfies $u \not\perp\!\!\!\perp v \mid \mathrm{do}(u)$. Algorithm IDENTIFY-BIDIRECTED identifies all the bidirected edges incident on u. If there is an edge of the form $u \circ\!\!-\!\!\circ v$ or $u \leftarrow\!\!\circ v$ or $u \circ\!\!\rightarrow v$ in the PAG, and $v \notin \mathrm{ch}_i(u)$ and $u \notin \mathrm{ch}_i(v)$, then it must be a bidirected edge.

Using these helper routines, we give an Algorithm RECOVERG (in Appendix B) that recovers any MAG $\mathcal{M}_i$ using n atomic interventions. Complementing this, we show that n interventions are also required. The missing details are presented in Appendix B.

Proposition 3.2. *There exists a causal MAG $\mathcal{M}$ such that every adaptive or non-adaptive algorithm requires $\Omega(n)$ many atomic interventions to recover $\mathcal{M}$.*

3.1 Recovering the Clusters

From the (α, β)-clustering definition, we know that a pair of entities belonging to the same cluster have higher structural similarity between their MAGs than a pair of entities across different clusters. Let us start with a simplifying assumption that $\beta = 0$ (i.e., α-clustering). So, all the MAGs are separated by a distance of at least αn. We make the observation that to identify that two MAGs, say $\mathcal{M}_i$ and $\mathcal{M}_j$ belong to different clusters, it suffices to find a node u from the node-difference set $\mathrm{diff}(\mathcal{M}_i, \mathcal{M}_j)$ and checking their neighbors using Algorithms IDENTIFY-OUTNBR and IDENTIFY-BIDIRECTED. We argue that (see Claim D.3, Appendix D.2), with probability at least $1-\delta$, we can identify one such node $u \in \mathrm{diff}(\mathcal{M}_i, \mathcal{M}_j)$ by sampling $2\log(M/\delta)/\alpha$ nodes uniformly from V as $|\mathrm{diff}(\mathcal{M}_i, \mathcal{M}_j)| = d(\mathcal{M}_i, \mathcal{M}_j) \geq \alpha n$.[3] However, this approach will not succeed when $\beta \neq 0$ because now we have MAGs in the same cluster that are also separated by non-zero distance.

Overview of Algorithm (α, β)-BOUNDEDDEGREE. We now build upon the above idea, to recover the true clusters $C_1^\star, \ldots, C_k^\star$ when $\beta \neq 0$. As identifying a node $u \in \mathrm{diff}(\mathcal{M}_i, \mathcal{M}_j)$ is not sufficient, we maintain a count of the number of nodes among the sampled set of nodes S that the pair of entities i, j have the same neighbors, i.e., $\textsc{Count}(i,j) = \sum_{u \in S} \mathbf{1}\{N_i(u) = N_j(u)\}$. Based on a carefully chosen threshold value for the $\textsc{Count}(i,j)$, that arises through the analysis of our randomized algorithm, we classify whether a pair of entities belong to the same cluster correctly.

Overall, the idea here is to construct a graph $\mathcal{P}$ on entities (i.e., the node set of $\mathcal{P}$ is $[M]$). We include an edge between two entities i and j if $\textsc{Count}(i,j)$ is above the threshold $(1-(\alpha+\beta)/2)|S|$. Using Lemma 3.3, we show that this threshold corresponds to the case where if the entities are from same true clusters, then the COUNT value corresponding to the pair is higher than the threshold; and if they are from different clusters it will be smaller, with high probability. This ensures that every entity is connected only to the entities belonging to the same true cluster. We return the connected components in $\mathcal{P}$ as our clusters.

Theoretical Guarantees. In Algorithm (α, β)-BOUNDEDDEGREE, we construct a uniform sample S of size $O(\log(M/\delta)/(\alpha-\beta)^2)$, and identify all the neighbors of S for every entity $i \in [M]$. As we use IDENTIFY-BIDIRECTED to identify all the bi-directed edges, the total number of interventions

[3] For theoretical analysis, our intervention targets are randomly chosen, even with the knowledge available from PAGs, because in the worst-case the PAGs might contain no directed edges to help decide which nodes to intervene on. In practice, though if we already know edge orientations from PAG we do not have to relearn them, and a biased sampling based on edges uncertainties in PAGs might be a reasonable approach.

Algorithm 3 (α, β)-BOUNDEDDEGREE

Input: $\alpha > 0$, $\beta \geq 0$ ($< \alpha$), confidence parameter $\delta > 0$, PAGs $\mathcal{U}_1, \ldots, \mathcal{U}_M$ of M entities
Output: Partition of $[M]$ into clusters
Let S denote a uniform sample of $\frac{4\log(M/\delta)}{(\alpha-\beta)^2}$ nodes from V selected with replacement.
for every entity $i \in [M]$ and $u \in S$ **do**
    $\text{ch}_i(u) \leftarrow$ IDENTIFY-OUTNBR$(\mathcal{U}_i, u)$
    $\text{sp}_i(u) \leftarrow$ IDENTIFY-BIDIRECTED$(\mathcal{U}_i, u)$
    $\text{pa}_i(u) \leftarrow \Gamma_i(u) \setminus (\text{ch}_i(u) \cup \text{sp}_i(u))$
    Construct $N_i(u)$ (defined in (2))
end for
Let $\mathcal{P}$ denote an empty graph on set of entities $[M]$
for every pair of entities i, j **do**
    Let $\text{COUNT}(i,j) = \sum_{u \in S} \mathbf{1}\{N_i(u) = N_j(u)\}$
    if $\text{COUNT}(i,j) \geq \left(1 - \frac{\alpha+\beta}{2}\right)|S|$ **then**
        Include an edge between i and j in $\mathcal{P}$
    end if
end for
Return connected components in $\mathcal{P}$

used by an entity for this step is at most $\Delta \cdot |S|$. Combining all the above, using the next lemma, we show that with high probability Algorithm (α, β)-BOUNDEDDEGREE recovers all the true clusters.

Lemma 3.3. *If the underlying MAGs $\mathcal{M}_1, \ldots, \mathcal{M}_M$ satisfy (α, β)-clustering property with true clusters $C_1^\star, \ldots, C_k^\star$ and have maximum undirected degree Δ. Then, the Algorithm (α, β)-*BOUNDEDDEGREE *recovers the clusters $C_1^\star, \ldots, C_k^\star$ with probability at least $1 - \delta$. Every entity $i \in [M]$ uses at most $4(\Delta + 1)\log(M/\delta)/(\alpha - \beta)^2$ many atomic interventions.*

3.2 Learning Causal Graphs from (α, β)-Clustering

In this section, we outline an approach to recover a close approximation of the causal MAGs of all the entities, after correctly recovering the clusters using Algorithm (α, β)-BOUNDEDDEGREE. First, we note that since the (α, β)-clustering allows the MAGs even in the same cluster to be different, the problem of exactly learning all the MAGs is challenging (with a small set of interventions) as causal edges learnt for an entity may not be relevant for another entity in the same cluster.

In the scenarios mentioned in the introduction, we expect the clusters to be more homogeneous, with many entities in the same cluster sharing the same MAG. We provide an overview of Algorithm (α, β)-RECOVERY that recovers one such MAG called *dominant MAG* for every cluster. Consider a recovered cluster $C_a^\star$, and a partitioning $S_a^1, S_a^2, \cdots$ of MAGs such that all MAGs in a partition S_a^i are equal for all i. We call the MAG $\mathcal{M}_a^{\text{dom}}$ corresponding to the largest partition S_a^{dom} as the *dominant MAG* of $C_a^\star$. The dominant MAG of a cluster is parameterized by $\gamma_a = |S_a^{\text{dom}}|/|C_a^\star|$ (fraction of the MAGs in the cluster that belong to the largest partition). We defer additional details of Algorithm (α, β)-RECOVERY to Appendix C.1.

Overview of Algorithm (α, β)-RECOVERY. After recovering the clustering using Algorithm (α, β)-BOUNDEDDEGREE, our goal is to learn the causal graphs. Using Algorithm (α, β)-RECOVERY, we show that we can learn these graphs approximately up to a distance approximation of βn.

In a cluster $C_a^\star$, we construct a partitioning of MAGs such that two MAGs belong to a partition if they are equal. The MAG corresponding to the largest partition is called the *dominant MAG*. Using our algorithm, we learn the dominant MAG correctly and return it as an output. As all the MAGs in the cluster satisfy (α, β)-clustering property, the dominant MAG is within a distance of βn from the true MAG and therefore is a good approximation of the true MAG.

For learning the dominant MAG, there are two steps. First, we select a node uniformly at random for every entity and intervene on the node and its neighbors to learn all the edges incident on the node. Next, we construct the dominant MAG by combining the neighborhoods of each individual node. Let u be any node and T_u denote the set of all entities which intervened on u in the first step. Now, among all the neighborhoods identified by the entities in T_u, we do not know which of them correspond to

that of the dominant MAG. In order to identify this, we use a threshold-based approach and assign a score to every entity in T_u. The score of an entity i is the number of entities in T_u that has the same neighborhood of u as that of entity i. Finally, we select the entity with the maximum score and assign the neighborhood of the entity as the neighborhood of u for the dominant MAG (Lines 12-15 in Algorithm (α, β)-RECOVERY). We argue that if the cluster size is large (see Theorem 3.4), the neighborhoods of nodes using entities with maximum scores are equal to that of the dominant MAG. This is because the dominant MAG has the largest partition size, and if a sufficiently large number of entities (across all partitions) are assigned node u, then, many of them will be entities from the dominant MAG partition.

As the entities satisfy (α, β)-clustering property, for all entities the recovered MAGs (dominant MAGs) are close to the true MAGs, and within a distance of at most βn. Note that any MAG from the cluster is within a distance of at most βn due to (α, β)-clustering property, but naively generating a valid MAG from a cluster will require n interventions on one entity from Proposition 3.2. Our actual guarantee is somewhat stronger, as in fact, for the entities whose MAGs are dominant in their cluster, we do recover the exact MAGs. We have the result:

Theorem 3.4. *Suppose $\mathcal{M}_1, \mathcal{M}_2, \cdots \mathcal{M}_M$ satisfy (α, β) clustering property. If $\gamma_a > 1/2$ and $C_a^\star = \Omega(n \log(n/M\delta)(2\gamma_a - 1)^2)$ for all $a \in [k]$, then, Algorithm (α, β)-RECOVERY recovers graphs $\widehat{\mathcal{M}}_1, \cdots \widehat{\mathcal{M}}_M$ such that for every entity $i \in [M]$, we have $d(\mathcal{M}_i, \widehat{\mathcal{M}}_i) \leq \beta n$ with probability $1 - \delta$. Every entity uses at most $(\Delta + 1) + 4(\Delta + 1)\log(M/\delta)/(\alpha - \beta)^2$ many atomic interventions.*

4 Causal Discovery under α-Clustering Property

In the previous section, we discussed the more general (α, β)-clustering scenario where we manage to construct a good approximation to all the MAGs. Now, we show that we can in fact recover all the MAGs exactly, if we make a stronger assumption. Suppose the MAGs $\mathcal{M}_1, \ldots, \mathcal{M}_M$ of the M entities satisfy the α-clustering property (Defn. 2.3). Firstly, we can design an algorithm similar to Algorithm (α, β)-BOUNDEDDEGREE (see Algorithm α-BOUNDEDDEGREE, Appendix D.3) that recovers the causal MAGs exactly with $O(\Delta \log(M/\delta)/\alpha)$ many interventions per entity, succeeding with probability $1 - \delta$. Note that this has a better $1/\alpha$ term in the intervention bound, instead of $1/\alpha^2$ (when $\beta = 0$) term arising in Theorem 3.4. In absence of latents, we can further improve it to $O(\log(M/\delta)/\alpha)$ many interventions per entity (see Algorithm NOLATENTS, Appendix D.2).

In this section, we present another approach (Algorithm α-GENERAL) with an improved result that requires fewer number of interventions, even when Δ is big, provided that each cluster has at least $\Omega(n \log(M/\delta))$ entities. Missing details of Algorithm α-GENERAL are in Appendix D.4.

Overview of Algorithm α-GENERAL. First, using a similar approach as Algorithm (α, β)-BOUNDEDDEGREE, we construct a uniform sample $S \subseteq V$, and find all the outgoing neighbors of nodes in S, for every entity $i \in [M]$. Then, we construct a graph on entities denoted by $\mathcal{P}$, where we include an edge between a pair of entities if the outgoing neighbors of the set of sampled nodes S, and the set of neighbors in PAGs associated with the entities (obtained from FCI) are the same. However, due to the presence of bidirected edges, it is possible that the connected components of $\mathcal{P}$ may not represent the true clusters $C_1^\star, \ldots, C_k^\star$.

We make the observation that a pair of entities i, j that have an edge in this $\mathcal{P}$ and from different true clusters, can differ only if there is a node u such that u has a bidirected edge $u \leftrightarrow v$ in $\mathcal{M}_i$, and a directed edge $u \leftarrow v$ in $\mathcal{M}_j$ (or vice-versa). Intervening on both u and v will separate these entities, our main idea is to ensure that this happens. First, we show how to *detect* if there are at least two true clusters in any connected component of $\mathcal{P}$. Then, we identify all the entities belonging to these two clusters and remove the edges between these entities in $\mathcal{P}$ and continue.

More formally, let $T_1, \ldots, T_{k'}$ be the partition of $[M]$ provided by the k' connected components of $\mathcal{P}$ and some of these can contain more than one true cluster, hence $k' \leq k$ and we focus on detecting such events. Let $\pi : [M] \to V$ denote a mapping from the set of entities to the nodes in V such that $\pi(i)$ is chosen uniformly at random from V for every entity i. For every entity i, we intervene on the node $\pi(i)$. To detect that there are at least two clusters in a given subset T_a of entities, we show that there are two entities i, j with an edge in $\mathcal{P}$ and for some node $u \in S$, we can identify the neighbor $v \in \Gamma_i(u) \cap \Gamma_j(u)$ such that $u \leftrightarrow v$ is an edge in $\mathcal{M}_i$ and $u \leftarrow v$ is an edge in $\mathcal{M}_j$ (or vice-versa). As there are at least $\Omega(n \log(M/\delta))$ entities in each of these two true clusters in T_a, for some $i, j \in T_a$, we can show that $\pi(i) = \pi(j) = v$ with probability at least $1 - \delta$.

After detecting the event that a component T_a of $\mathcal{P}$ contains entities from at least two different true clusters (say, $C_b^\star$ and $C_c^\star$) due to an edge (u, v) as above, we intervene on v for every entity in T_a. By intervening on v (and $u \in S$), we can separate all entities in T_a that belong to true clusters $C_b^\star$ and $C_c^\star$, and remove edges between such entity pairs from $\mathcal{P}$. We repeat this above procedure of refining $\mathcal{P}$. In each iteration, we will have removed all edges between every pair of entities belonging to at least two different true clusters. Since there are at most k^2 different true cluster pairs, after k^2 iterations the connected components remaining correspond to the true clusters (with high probability). This can be done without knowing the value of k, by checking whether the connected components in $\mathcal{P}$ change or not after each iteration of the above idea.

Going from Clustering to MAGs. The idea of going from clusters to MAGs is simple and based on distributing the interventions across the entities in the cluster. Since under α-clustering, entities belonging to a cluster share the same MAG, combining the results is relatively simpler (see Appendix D.1). Combining the guarantees of α-GENERAL and α-BOUNDEDDEGREE, we have:

Theorem 4.1. *If MAGs $\mathcal{M}_1, \ldots, \mathcal{M}_M$ satisfy α-clustering property with true clusters $C_1^\star, \ldots, C_k^\star$ such that $\min_{b\in[k]} |C_b^\star| = \Omega(n \log(M/\delta))$. Then, there is an algorithm that exactly learns all these MAGs with probability at least $1 - \delta$. Every entity $i \in [M]$ uses $\min\left\{O(\Delta \log(M/\delta)/\alpha), O(\log(M/\delta)/\alpha + k^2)\right\}$ many atomic interventions.*

Lower Bound on the Number of Interventions. We now give a lower bound on the number of atomic interventions needed for every algorithm that recovers the true clusters on the MAGs $\mathcal{M}_1, \mathcal{M}_2, \cdots \mathcal{M}_M$. Since a lower bound under α-clustering is also a lower bound under (α, β)-clustering, we work with the α-clustering property here. First, we show that to identify whether a given pair of entities i, j belong to the same true cluster or not, every (randomized or deterministic) algorithm must make $\Omega(1/\alpha)$ interventions for both i and j.

Our main idea here is to use the famous Yao's minimax theorem [Yao, 1977] to get lower bounds on randomized algorithms. Yao's theorem states that an *average case* lower bound on a deterministic algorithm implies a *worst case* lower bound on randomized algorithms. To show a lower bound using Yao's minimax theorem, we construct a distribution μ on MAG pairs and show that every deterministic algorithm requires $\Omega(1/\alpha)$ interventions for distinguishing a pair of MAGs drawn from μ. The construction of this distribution is presented in Appendix D.5. We summarize the result:

Theorem 4.2. *Suppose the underlying MAGs $\mathcal{M}_1, \ldots, \mathcal{M}_M$ satisfy α-clustering property. In order to recover the clusters with probability $2/3$, every (randomized or deterministic) algorithm requires $\Omega(1/\alpha)$ interventions for every entity in $[M]$.*

5 Experimental Evaluation

In this section, we provide an evaluation of our approaches on data generated from real and synthetic causal networks for learning MAGs satisfying (α, β)-clustering property. We defer additional details, results, and evaluation for α-clustering to Appendix E.

Causal Networks. We consider the following real-world Bayesian networks from the *Bayesian Network Repository* which cover a wide variety of domains: *Asia* (Lung cancer) (8 nodes, 8 edges), *Earthquake* (5 nodes, 4 edges), *Sachs* (Protein networks) (11 nodes, 17 edges), and *Survey* (6 nodes, 6 edges). For the synthetic data, we use Erdös-Rényi random graphs (10 nodes). We use the term "causal network" to refer to these ground-truth Bayesian networks.

Parameters. For each causal network, we start from the corresponding DAG, and generate M MAGs (one for each entity) split into k clusters that satisfy the (α, β)-clustering property through random changes to the graph. We also randomly introduce two latents in each graph, and account for them in MAG constructions. For more details, refer Appendix E. We set number of entities $M = 40$, number of clusters $k = 2$, $\alpha = 0.60, \beta = 0.20$, and dominant MAG parameter $\gamma = 0.90$ for both the clusters. For the synthetic data generated using Erdös-Rényi model, we use $n = 10$, probability of edge 0.3.

Evaluation of Clustering. First, we focus on recovering the clustering using Algorithm (α, β)-BOUNDEDDEGREE. As a baseline, we employ the well-studied FCI algorithm based on purely observational data [Spirtes et al., 2000]. After recovering the PAGs corresponding to the MAGs using FCI, we cluster them by constructing a similarity graph (similar to (α, β)-BOUNDEDDEGREE) defined on the set of entities (refer Appendix E for more details). For Algorithm (α, β)-BOUNDEDDEGREE, we first construct a sample S, and perform various interventions based on the set S for every entity

Causal Network	FCI			(α, β)-BOUNDEDDEGREE (Alg. 3)			Maximum # Interventions
	Precision	Recall	Accuracy	Precision	Recall	Accuracy	
Earthquake	0.57 ± 0.18	0.94 ± 0.013	0.58 ± 0.18	0.78 ± 0.24	0.92 ± 0.03	0.77 ± 0.23	4
Survey	0.62 ± 0.21	0.94 ± 0.013	0.62 ± 0.2	0.64 ± 0.23	0.97 ± 0.02	0.63 ± 0.23	5
Asia	0.57 ± 0.18	0.94 ± 0.013	0.58 ± 0.18	0.92 ± 0.14	0.95 ± 0.03	0.91 ± 0.14	5
Sachs	0.52 ± 0.12	0.94 ± 0.01	0.52 ± 0.12	0.89 ± 0.20	0.96 ± 0.02	0.88 ± 0.19	6
Erdös-Rényi	0.62 ± 0.21	0.94 ± 0.02	0.62 ± 0.21	1.0 ± 0.00	0.95 ± 0.02	0.97 ± 0.013	6

Table 1: In this table, we present the precision, recall and accuracy values obtained by our Algorithm (α, β)-BOUNDEDDEGREE and using FCI. Each cell includes the mean value along with the standard deviation computed over 10 runs. The last column represents the maximum number of interventions per entity including both Algorithms (α, β)-BOUNDEDDEGREE and (α, β)-RECOVERY.

to obtain the clusters. We also implemented another baseline algorithm (GREEDY) that uses interventions, based on a greedy idea that selects nodes to set S in Algorithm (α, β)-BOUNDEDDEGREE by considering nodes in increasing order of their degree in the PAGs returned by FCI. We use this ordering to minimize the no. of interventions as we intervene on every node in S and their neighbors.

Metrics. We use the following standard metrics for comparing the clustering performance: *precision* (fraction of pairs of entities correctly placed in a cluster together to the total number of pairs placed in a cluster together), *recall* (fraction of pairs of entities correctly placed in a cluster together to the total number of pairs in the same ground truth clusters), and *accuracy* (fraction of pairs of entities correctly placed or not placed in a cluster to the total number of pairs of entities).

Results. In Table 1, we compare Algorithm (α, β)-BOUNDEDDEGREE to FCI on the clustering results. For Algorithm (α, β)-BOUNDEDDEGREE, we use a sample S of size 1, and observe in Figure 8 (Appendix E), that this corresponds to about 3 interventions per entity. With increase in sample size, we observed that the results were either comparable or better. We observe that our approach leads to considerably better performance in terms of the accuracy metric with an average difference in mean accuracy of about 0.25. This is because FCI recovers partial graphs, and clustering based on the partial information results in poor accuracy. Because of the presence of a dominant MAG with in each cluster, we observe that the corresponding entities are always assigned to the same cluster, resulting in high recall for both (α, β)-BOUNDEDDEGREE and FCI. We observe a higher value of precision for our algorithms, because FCI is unable to correctly classify the MAGs that are different from the dominating MAG.

Algorithm (α, β)-BOUNDEDDEGREE outperforms the GREEDY baseline for the same sample(S) size. For example, on the *Earthquake* and *Survey* causal networks, Algorithm (α, β)-BOUNDEDDEGREE obtains the mean accuracy values of 0.77 and 0.63 respectively, while GREEDY for the same number of interventions obtained an accuracy of only 0.487 and 0.486 respectively. For the remaining networks, the accuracy values of GREEDY are almost comparable to our Algorithm (α, β)-BOUNDEDDEGREE.

After clustering, we recover the dominant MAGs using Algorithm (α, β)-RECOVERY, and observe that the additional interventions needed are bounded by the maximum degree of the graphs (see Theorem 3.4). This is represented in the last column in Table 1. We observe that our *collaborative* algorithms use fewer interventions for dominant MAG recovery compared to the number of nodes in each graph. E.g., in the Erdös-Rényi setup, the number of nodes $n = 10$, whereas we use at most 6 interventions per entity. Thus, compared to the worst-case, cutting the number of interventions for each entity by 40%.

6 Conclusion

We introduce a new model for causal discovery to capture practical scenarios where are multiple entities with different causal structures. Under natural clustering assumption(s), we give efficient provable algorithms for causal learning with atomic interventions and demonstrate its empirical performance. Our model can be extended to the setting where all interventions are non-adaptive, and we plan to study it as part of future work. An interesting future direction would be to use interventional equivalence classes of DAGs as part of the model, instead of the clustering assumption. This might require extending the interventional equivalence between DAGs studied in [Hauser and Bühlmann, 2012, Katz et al., 2019] to the setting without the causal sufficiency assumption and exploit that for learning.

Acknowledgements

The work was partially supported by NSF grants 1934846, 1908849, 1637536 and an Adobe Faculty Research grant.

References

Jayadev Acharya, Arnab Bhattacharyya, Constantinos Daskalakis, and Saravanan Kandasamy. Learning and testing causal models with interventions. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 9469–9481, 2018.

Raghavendra Addanki, Shiva Kasiviswanathan, Andrew McGregor, and Cameron Musco. Efficient intervention design for causal discovery with latents. In *International Conference on Machine Learning*, pages 63–73. PMLR, 2020.

Raghavendra Addanki, Andrew McGregor, and Cameron Musco. Intervention efficient algorithms for approximate learning of causal graphs. In *Algorithmic Learning Theory*, pages 151–184. PMLR, 2021.

Hassan Ashtiani, Shrinu Kushagra, and Shai Ben-David. Clustering with same-cluster queries. In *Advances in Neural Information Processing Systems*, 2016.

Pranjal Awasthi, Avrim Blum, and Or Sheffet. Center-based clustering under perturbation stability. *Information Processing Letters*, 112(1-2):49–54, 2012.

Kevin Bello and Jean Honorio. Computationally and statistically efficient learning of causal bayes nets using path queries. *Advances in Neural Information Processing Systems*, 31:10931–10941, 2018.

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

Frederick Eberhardt. Causation and intervention. *PhD Thesis, Carnegie Mellon University*, 2007.

Kathleen M Gates and Peter CM Molenaar. Group search algorithm recovers effective connectivity maps for individuals in homogeneous and heterogeneous samples. *NeuroImage*, 63(1):310–319, 2012.

AmirEmad Ghassami, Saber Salehkaleybar, Negar Kiyavash, and Elias Bareinboim. Budgeted experiment design for causal structure learning. In *International Conference on Machine Learning*, pages 1724–1733. PMLR, 2018.

AmirEmad Ghassami, Saber Salehkaleybar, and Negar Kiyavash. Interventional experiment design for causal structure learning. *arxiv preprint arxiv 1910.05651*, 2019.

Kristjan Greenewald, Dmitriy Katz, Karthikeyan Shanmugam, Sara Magliacane, Murat Kocaoglu, Enric Boix Adsera, and Guy Bresler. Sample efficient active learning of causal trees. In *Advances in Neural Information Processing Systems*, pages 14279–14289, 2019.

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

Huining Hu, Zhentao Li, and Adrian R Vetta. Randomized experimental design for causal graph discovery. In *Advances in neural information processing systems*, pages 2339–2347, 2014.

Antti Hyttinen, Frederick Eberhardt, and Patrik O Hoyer. Experiment selection for causal discovery. *The Journal of Machine Learning Research*, 14(1):3041–3071, 2013.

Amin Jaber, Murat Kocaoglu, Karthikeyan Shanmugam, and Elias Bareinboim. Causal discovery from soft interventions with unknown targets: Characterization and learning. *Advances in Neural Information Processing Systems*, 2020.

Michael Joffe, Manoj Gambhir, Marc Chadeau-Hyam, and Paolo Vineis. Causal diagrams in systems epidemiology. *Emerging themes in epidemiology*, 9(1):1–18, 2012.

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.

Dmitriy Katz, Karthikeyan Shanmugam, Chandler Squires, and Caroline Uhler. Size of interventional markov equivalence classes in random dag models. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3234–3243. PMLR, 2019.

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.

Murat Kocaoglu, Amin Jaber, Karthikeyan Shanmugam, and Elias Bareinboim. Characterization and learning of causal graphs with latent variables from soft interventions. *Advances in Neural Information Processing Systems*, 2019.

Steffen L Lauritzen and David J Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 50(2):157–194, 1988.

Erik Lindgren, Murat Kocaoglu, Alexandros G Dimakis, and Sriram Vishwanath. Experimental design for cost-aware learning of causal graphs. In *Advances in Neural Information Processing Systems*, pages 5279–5289, 2018.

Florence Jessie MacWilliams and Neil James Alexander Sloane. *The theory of error-correcting codes*, volume 16. Elsevier, 1977.

Matey Neykov, Sivaraman Balakrishnan, and Larry Wasserman. Minimax optimal conditional independence testing. *The Annals of Statistics*, 49(4):2151–2177, 2021.

Dang Trinh Nguyen, Quoc Bao Duong, Eric Zamai, and Muhammad Kashif Shahzad. Fault diagnosis for the complex manufacturing system. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 230(2):178–194, 2016.

Judea Pearl. Causal diagrams for empirical research. *Biometrika*, 82(4):669–688, 1995.

Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge university press, 2009.

JD Ramsey, Peter Spirtes, and Clark Glymour. On meta-analyses of imaging data and the mixture of records. *NeuroImage*, 57(2):323–330, 2011.

Thomas Richardson and Peter Spirtes. Ancestral graph markov models. *The Annals of Statistics*, 30 (4):962–1030, 2002.

Basil Saeed, Snigdha Panigrahi, and Caroline Uhler. Causal structure discovery from distributions arising from mixtures of dags. In *International Conference on Machine Learning*, pages 8336–8345. PMLR, 2020.

Karthikeyan Shanmugam, Murat Kocaoglu, Alexandros G Dimakis, and Sriram Vishwanath. Learning causal graphs with small interventions. In *Advances in Neural Information Processing Systems*, pages 3195–3203, 2015.

Peter Spirtes, Christopher Meek, and Thomas Richardson. An algorithm for causal inference in the presence of latent variables and selection bias. *Computation, causation, and discovery*, 21:1–252, 1999.

Peter Spirtes, Clark N Glymour, Richard Scheines, David Heckerman, Christopher Meek, Gregory Cooper, and Thomas Richardson. *Causation, prediction, and search*. MIT press, 2000.

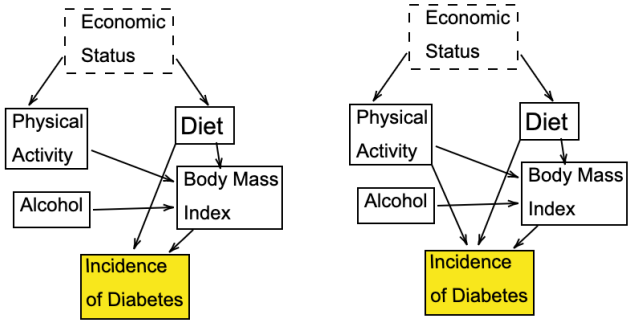
Prasad Tadepalli and Stuart J Russell. Pac learning of causal trees with latent variables. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 9774–9781, 2021.

Andrew Chi-Chin Yao. Probabilistic computations: Toward a unified measure of complexity. In *18th Annual Symposium on Foundations of Computer Science*, pages 222–227. IEEE, 1977.

Jiji Zhang. Causal reasoning with ancestral graphs. *Journal of Machine Learning Research*, 9(Jul): 1437–1474, 2008a.

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, 2008b.

Appendix for "Collaborative Causal Discovery with Atomic Interventions"



(a) Causal DAG $\mathcal{D}_1$ (b) Causal DAG $\mathcal{D}_2$

Figure 3: Two possible diabetes incidence graphs for an individual from [Joffe et al., 2012] differing in the causal edge between *Physical Activity* and *Incidence of Diabetes*. The observed variables include: *Diet, Body Mass Index (BMI), Physical Activity, Alcohol (consumption), Incidence of Diabetes*, and the unobserved variable (latent) is *Economic Status*. The variable *Incidence of Diabetes* is observable but can't be intervened on, this is not an issue as it has no outgoing edges in the graphs. In this paper, we do not know the underlying causal graphs or which individuals share the same graph. As intervening on variables such as *Diet, BMI* might need expensive and careful experimental organization, we ask the following question – given a collection of independent entities (in this diabetes example, they can refer to a collection of people), can we collaboratively learn each entity's causal graphs while minimizing the number of interventions per entity?

A Missing Details from Section 2

A.1 Maximal Ancestral Graphs

Ancestral graphical models were introduced motivated by the need to represent data generating processes that involve latent variables. In this paper, we work with a class of graphical models, the maximal ancestral graph (MAG), which are a generalization of DAGs and are closed under marginalization and conditioning [Richardson and Spirtes, 2002]. A maximal ancestral graph (MAG) is a (directed) mixed graph that may contain two kinds of edges: directed edges ($\rightarrow$) and bi-directed edges ($\leftrightarrow$). Before defining a MAG, we need some preliminaries.

Consider a mixed graph $\mathcal{G}$. Given an path $\pi = \langle u, \ldots, w, \ldots, v \rangle$, w is a collider on π if the two edges incident to w in π are both into w, that is, have an arrowhead into w; otherwise it is called a non-collider on π. Let S be any subset of nodes in the graph $\mathcal{G}$. An *inducing path* relative to S is a path on which every node not in S (except for the endpoints) is a collider on the path and every collider is an ancestor of an endpoint of the path.

Definition A.1. *A mixed graph is called a maximal ancestral graph (MAG) if*

1. *The mixed graph is* ancestral*, i.e., it has no directed cycles, and whenever there is a bidirected edge $u \leftrightarrow v$, then there is no directed path from u to v or v to u.*
2. *There is no inducing path between any two non-adjacent nodes.*

It is straightforward to extend the notion of d-separation in DAGs to mixed graphs using the notion of m-separation [Richardson and Spirtes, 2002].

Definition A.2. *In a mixed graph, a path π between nodes u and v is m-connecting relative to a (possibly empty) set of nodes Z with $u, v \notin Z$ if*

1. *every non-collider on π is not a member of Z;*
2. *every collider on π is an ancestor of some member of Z.*

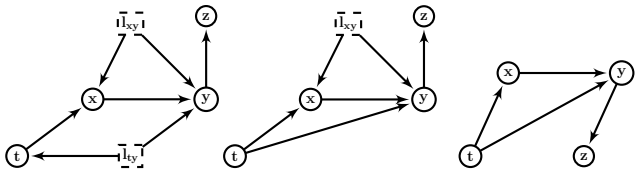
u and v are said to be m-separated by Z if there is no m-connected path between u and v relative to Z.

Conversion of a DAG to a MAG. The following construction gives us a MAG $\mathcal{M}$ from a DAG $\mathcal{D}$:

1. for each pair of variables $u, v \in V$, u and v are adjacent in $\mathcal{M}$ if and only if there is an inducing path between them relative to L in $\mathcal{D}$. The skeleton or the undirected graph constructed from PAG $\mathcal{U}$ (obtained using FCI [Spirtes et al., 2000]) by ignoring the directions of edges captures all the edges in $\mathcal{M}$.
2. for each pair of adjacent variables u, v in $\mathcal{M}$, orient the edge as $u \to v$ in $\mathcal{M}$ if u is an ancestor of v in $\mathcal{D}$; orient it as $u \leftarrow v$ in $\mathcal{M}$ if v is an ancestor of u in $\mathcal{D}$; orient it as $u \leftrightarrow v$ in $\mathcal{M}$ otherwise.

Several DAGs can lead to the same MAG (See Figure 4c). Essentially a MAG represents a set of DAGs that have the exact same d-separation structures and ancestral relationships among the observed variables. By construction, the MAG is unique for a given DAG.

As a further evidence to the claim that interventions are required, see Figure 5, that gives an example of two MAGs separated by a distance of $\frac{n}{2}$ and have the same Partial Ancestral Graph identified by FCI [Zhang, 2008b].



(a) DAG $\mathcal{D}_1$ (b) DAG $\mathcal{D}_2$ (c) MAG for $\mathcal{D}_1$ and $\mathcal{D}_2$

Figure 4: Different DAGs with same MAG. It is easy to observe that, no single vertex interventions can differentiate $\mathcal{D}_1$ from $\mathcal{D}_2$.

Conditional Independence (CI) Tests. Conditional independence tests are an important building block in causal discovery.

(i) CI-test in observational distribution: Given $u, v \in V$, $Z \subset V$ check whether u is independent of v given Z, denoted by $u \perp\!\!\!\perp v \mid Z$.

(ii) CI-test in interventional distribution: Given $u, v \in V$, $Z \subset V$, and $w \in V$, check whether u is independent of v given Z in the interventional distribution of w, denoted by $u \perp\!\!\!\perp v \mid Z, \mathrm{do}(w)$ where $\mathrm{do}(w)$ is the intervention on the variable w.

The convergence rates of CI tests are well-known [Neykov et al., 2021] which can be used to obtain the required sample size bounds for any of the PAG estimation procedures for the desired Type 1 error bound (omitted here). Note that in our experiments (Section 5), we do run CI tests on actual data samples generated by our model.

B Helper Routines

Claim B.1. *Suppose $\mathcal{D}_i$ is the DAG and $\mathcal{M}_i$ is the corresponding MAG for some entity $i \in [M]$. Then, $u \not\perp\!\!\!\perp v \mid \mathrm{do}(u)$ iff u is an ancestor of v in the graph $\mathcal{D}_i$.*

Proof. We follow a proof similar to Lemma 1 in [Kocaoglu et al., 2017]. If u is an ancestor of v in the graph $\mathcal{D}_i$ using the path π_{uv}, then, in the mutilated graph corresponding to $\mathrm{do}(u)$, the path π_{uv} remains intact. From d-separation [Pearl, 2009], π_{uv} can only be blocked by conditioning on one of the nodes that are not end points. As we do not condition on any variables in the CI-test $u \perp\!\!\!\perp v \mid \mathrm{do}(u)$ and therefore do not block the path π_{uv}, we have $u \not\perp\!\!\!\perp v \mid \mathrm{do}(u)$.

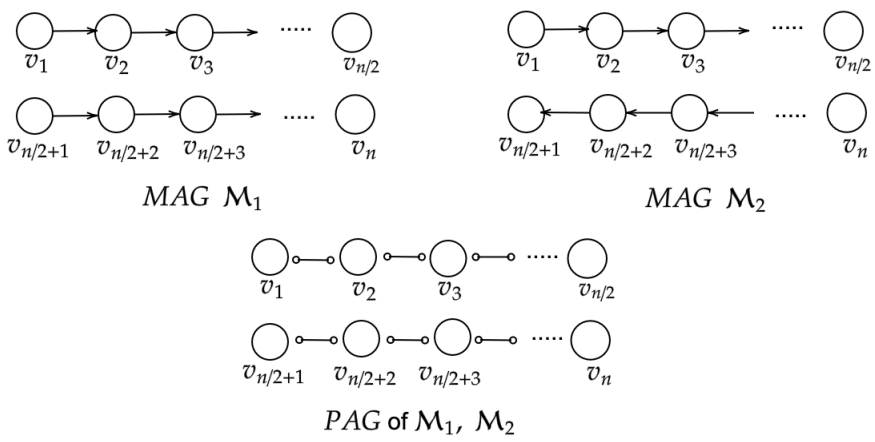


Figure 5: An example of MAGs $\mathcal{M}_1$ and $\mathcal{M}_2$ with large distance $d(\mathcal{M}_1, \mathcal{M}_2)$ but generating the same PAG.

Now, we consider the other direction. If $u \not\perp\!\!\!\perp v \mid \mathrm{do}(u)$, then, there is at least a path π_{uv} between u and v that is not blocked. In the mutilated graph corresponding to the interventional distribution $\mathrm{do}(u)$, the incoming edges into the node u are removed. In the path π_{uv}, the edge incident on u is an outgoing edge. If there is a collider on π_{uv}, we have blocked the path by not conditioning on it (from d-separation). As the path is not blocked, it implies that there is no collider on the path. Therefore, the path π_{uv} is a directed path from u to v. Hence, the claim. □

Claim B.2. *Given an entity $i \in [M]$, and a node $u \in V$, Algorithm* IDENTIFY-OUTNBR *identifies all outgoing edges of u in $\mathcal{M}_i$ ($ch_i(u)$) correctly using an intervention on u.*

Proof. We know that $\mathcal{U}_i = (V, \widehat{E}_i)$ represents the partial ancestral graph of $\mathcal{M}_i$. We observe that any outgoing edge (u, v) incident on a node u in the PAG $\mathcal{U}_i$ can be of the form $u \circ\!\!-\!\!\circ v$ or $u \circ\!\!\rightarrow v$. Otherwise, we already know that the edge is not an outgoing edge from u. We claim that we can identify an outgoing edge (u, v) from a node u correctly, if CI-test returns $u \not\perp\!\!\!\perp v \mid \mathrm{do}(u)$ for every $v \in \Gamma_i(u)$ satisfying the condition mentioned above. From Claim B.1, we have that $u \not\perp\!\!\!\perp v \mid \mathrm{do}(u)$ iff u is an ancestor of v in $\mathcal{D}_i$, which implies $u \to v$ is present in $\mathcal{M}_i$ and $v \in \mathrm{ch}_i(u)$. □

Claim B.3. *Given an entity $i \in [M]$, and a node $u \in V$, Algorithm* IDENTIFY-BIDIRECTED *identifies all bidirected edges incident on u in $\mathcal{M}_i$ ($sp_i(u)$) correctly using atomic interventions on all nodes in $\Gamma_i(u)$.*

Proof. We observe that any bi-directed edge (u, v) incident on a node $u \in V$ in the PAG $\mathcal{U}_i$ can be of the form $u \circ\!\!-\!\!\circ v$ or $u \leftarrow\!\!\circ v$ or $u \circ\!\!\rightarrow v$. Otherwise, we already know that the edge is not a bi-directed edge incident at u. In Algorithm IDENTIFY-BIDIRECTED, for every neighbor v of u in the PAG $\mathcal{U}_i$ satisfying the above condition, we check if $u \perp\!\!\!\perp v \mid \mathrm{do}(u)$ and $v \perp\!\!\!\perp u \mid \mathrm{do}(v)$ is satisfied. From Claim B.1, we know that if $u \not\perp\!\!\!\perp v \mid \mathrm{do}(u)$, then u is an ancestor of v in $\mathcal{D}_i$ (similarly, v is an ancestor of u in $\mathcal{D}_i$ if $u \not\perp\!\!\!\perp v \mid \mathrm{do}(v)$). So, if $u \perp\!\!\!\perp v \mid \mathrm{do}(u)$ and $u \perp\!\!\!\perp v \mid \mathrm{do}(v)$, then, u is not an ancestor of v or vice-versa, which implies $u \leftrightarrow v$ is present in $\mathcal{M}_i$, i.e., $v \in \mathrm{sp}_i(u)$. As we perform an intervention for every neighbor of u in $\mathcal{U}_i$, we have the claim.

□

Algorithm RECOVERG. For every $u \in V$, first identify outgoing neighbors using Algorithm IDENTIFY-OUTNBR and then identify all the bidirected edges incident on u using Algorithm IDENTIFY-BIDIRECTED.

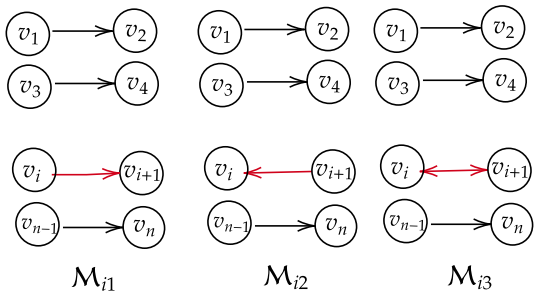


Figure 6: The MAGs used in the proof of Proposition B.5.

Lemma B.4. *Algorithm* RECOVERG *recovers all edges of $\mathcal{M}_i$, for an entity $i \in [M]$ using n atomic interventions.*

Proof. Given an entity $i \in [M]$, we obtain the partial ancestral graph $\mathcal{U}_i$ from observational data. Using Algorithm RECOVERG, we create interventions for every node $u \in V$. For every node u, we correctly identify all the outgoing neighbors of u using Algorithm IDENTIFY-OUTNBR (Claim B.2) and all the bidirected edges using Algorithm IDENTIFY-BIDIRECTED (Claim B.3). Therefore, we have recovered all edges of $\mathcal{M}_i$ using n atomic interventions. □

Proposition B.5. *[Proposition 3.2 restated] There exists a causal MAG $\mathcal{M}$ such that every adaptive or non-adaptive algorithm requires n many atomic interventions to recover $\mathcal{M}$.*

Proof. Suppose the set of nodes of an unknown MAG $\mathcal{M}$ is given by $V = \{v_1, v_2, \cdots v_n\}$. We denote ALG by any adaptive or non-adaptive *deterministic* algorithm that recovers $\mathcal{M}$ using the set of interventions $\mathcal{S} \subseteq V$. For the sake of contradiction, let ALG recover $\mathcal{M}$ correctly and v_i be the vertex that has not been intervened on, i.e., $v_i \notin \mathcal{S}$.

Construct the MAGs $\mathcal{M}_{i1}, \mathcal{M}_{i2}, \mathcal{M}_{i3}$ with edges $E_{i1} = \{v_1 \to v_2, v_3 \to v_4, \cdots, v_i \to v_{i+1}, \cdots, v_{n-1} \to v_n\}, E_{i2} = \{v_1 \to v_2, v_3 \to v_4, \cdots, v_i \leftarrow v_{i+1}, \cdots, v_{n-1} \to v_n\}, E_{i3} = \{v_1 \to v_2, v_3 \to v_4, \cdots, v_i \leftrightarrow v_{i+1}, \cdots, v_{n-1} \to v_n\}$ respectively (see Figure 6).

Upon termination, ALG will have recovered one of the MAGs $\mathcal{M}_{i1}$, $\mathcal{M}_{i2}$ or $\mathcal{M}_{i3}$. As $v_i \notin \mathcal{S}$, we will argue that the true MAG is different from the recovered MAG. We consider two cases:

1. If $v_{i+1} \in \mathcal{S}$. First, we observe that for all three MAGs $\mathcal{M}_{i1}, \mathcal{M}_{i2}$ and $\mathcal{M}_{i3}$, the CI-test $v_i \not\perp\!\!\!\perp v_{i+1}$. For MAGs $\mathcal{M}_{i1}$ and $\mathcal{M}_{i2}$, we have $v_i \perp\!\!\!\perp v_{i+1} \mid \mathrm{do}(v_{i+1})$ while $v_i \not\perp\!\!\!\perp v_{i+1} \mid \mathrm{do}(v_{i+1})$ for MAG $\mathcal{M}_{i2}$. As these are the only possible CI-tests for vertices v_i and v_{i+1}, the algorithm ALG cannot differentiate between $\mathcal{M}_{i1}$ and $\mathcal{M}_{i3}$. If ALG recovers $\mathcal{M}_{i1}$, then, we can set $\mathcal{M}$ to be $\mathcal{M}_{i3}$. This is a contradiction.

2. If $v_{i+1} \notin \mathcal{S}$. We observe that for all three MAGs $\mathcal{M}_{i1}, \mathcal{M}_{i2}$ and $\mathcal{M}_{i3}$, the CI-test $v_i \not\perp\!\!\!\perp v_{i+1}$, and it is the only possible CI-test involving vertices v_i and v_{i+1}. Therefore, the algorithm ALG cannot differentiate between $\mathcal{M}_{i1}, \mathcal{M}_{i2}$ and $\mathcal{M}_{i3}$. If ALG recovers $\mathcal{M}_{i1}$, then, we can set $\mathcal{M}$ to be $\mathcal{M}_{i2}, \mathcal{M}_{i3}$ and similarly for other cases. This is a contradiction.

Therefore, to recover $\mathcal{M} \in \{\mathcal{M}_{i1}, \mathcal{M}_{i2}, \mathcal{M}_{i3}\}$ correctly, we must have $v_i \in \mathcal{S}$. As i is chosen arbitrarily, and for every i we can construct the MAGs $\mathcal{M}_{i1}, \mathcal{M}_{i2}, \mathcal{M}_{i3}$, such that any adaptive or non-adaptive deterministic algorithm requires interventions on every node.

We can extend the proof to include *randomized* algorithms, with success probability strictly greater than $1/2$, by observing that when $v_i \notin \mathcal{S}$, ALG has at least two MAGs among $\mathcal{M}_{i1}, \mathcal{M}_{i2}, \mathcal{M}_{i3}$ that it cannot differentiate (as argued using two cases above).

□

B.1 Handling Uncertainty in PAG Estimation

We assume throughout this paper that the initial PAGs (fed to our algorithms) are estimated correctly from observational data. We outline some reasons behind such an assumption.

(a) PAG estimation is a very well-studied problem in causal discovery from both a theoretical and practical perspective. Well known algorithms for recovering PAGs, such as FCI (Fast Causal Inference), are known to be sound and complete (see [Spirtes et al., 1999] and [Zhang, 2008b]). Also, recent variations of FCI such as Really Fast Causal Inference (RFCI) have sped up the FCI procedure [Colombo et al., 2012]. Today FCI/RFCI procedures are commonly used in practice, with various implementations available [Kalisch et al., 2012].

(b) Note, for all our algorithms and bounds, all that we require from the PAGs is that they have the correct (undirected) skeleton as their corresponding MAGs, i.e., we could just ignore all the directed edges in the initial PAGs and replace them with edges before using them in our algorithms, and this would not change our results.

(c) Finally, we could even relax our assumptions and tolerate error even in skeleton estimation. The idea is simple, and we sketch it here. Suppose the MAGs $\mathcal{M}_1, \mathcal{M}_2, \cdots \mathcal{M}_M$ satisfy the α-clustering assumption with true clusters $C_1^\star, C_2^\star, \cdots C_k^\star$. Now consider the setting where we have errors in the PAG skeleton estimation. Let $\mathcal{U}_1, \mathcal{U}_2, \cdots \mathcal{U}_M$ be the true skeletons of the MAGs $\mathcal{M}_1, \mathcal{M}_2, \cdots \mathcal{M}_M$. Consider for each MAG $\mathcal{M}_i$, a corrupted counterpart $\mathcal{M}_i^{\text{corr}}$, with the guarantee that $d(\mathcal{M}_i, \mathcal{M}_i^{\text{corr}}) \leq \beta/2n$. These corrupted MAGs are only constructed for the sake of proof, and are not actually present. Assume that the skeleton estimation is not precise and instead of $\mathcal{U}_1, \mathcal{U}_2, \cdots \mathcal{U}_M$, it produces the skeletons $\mathcal{U}_1^{\text{corr}}, \mathcal{U}_2^{\text{corr}}, \cdots \mathcal{U}_M^{\text{corr}}$, associated with these corrupted MAGs $\mathcal{M}_1^{\text{corr}}, \mathcal{M}_2^{\text{corr}}, \cdots \mathcal{M}_M^{\text{corr}}$. By triangle inequality, it is easy to observe that the MAGs satisfy $(\alpha - \beta, \beta)$-clustering assumption. If $\beta < \alpha/2$, then, using Algorithm (α, β)-BOUNDEDDEGREE on $\mathcal{U}_1^{\text{corr}}, \mathcal{U}_2^{\text{corr}}, \cdots \mathcal{U}_M^{\text{corr}}$ with parameter α replaced by $\alpha - \beta$ will guarantee that we recover the true clusters $C_1^\star, C_2^\star, \cdots C_k^\star$. This follows because any pair of entities i, j that were originally in the same true cluster will still remain together in the same cluster, even under corruption, as their corrupted MAGs will be at most $\beta n < \alpha/2n$ distance apart. Similarly, if i, j belonged to different true clusters then they will still remain in different clusters, even under corruption, as their corrupted MAGs will be $> \alpha/2n$ distance apart. Also, if the corrupted MAGs satisfy the conditions in Theorem 3.4, we can recover the dominant MAG. With the right set of parameters, this argument can also be extended starting from an (α, β)-clustering.

C Discovery under (α, β)-Clustering

In this section, we present an algorithm that recovers the underlying clusters $C_1^\star, C_2^\star, \cdots, C_k^\star$ provided they satisfy (α, β)-clustering property. After recovering the clusters, in Section C.1, we give an algorithm that recovers an approximate MAG for every entity with only few additional interventions.

Firstly, using the next lemma, we show that the threshold used by Algorithm (α, β)-BOUNDEDDEGREE correctly identifies whether two entities belong to the same true cluster or not. This implies that our algorithm (α, β)-BOUNDEDDEGREE recovers the clusters with high probability.

Lemma C.1 (Lemma 3.3 Restated)**.** *If the underlying MAGs $\mathcal{M}_1, \ldots, \mathcal{M}_M$ satisfy (α, β)-clustering property with true clusters $C_1^\star, \ldots, C_k^\star$ and have maximum undirected degree Δ. Then, the Algorithm (α, β)-BOUNDEDDEGREE recovers the clusters $C_1^\star, \ldots, C_k^\star$ with probability at least $1 - \delta$. Every entity $i \in [M]$ uses at most $4(\Delta + 1)\log(M/\delta)/(\alpha - \beta)^2$ many atomic interventions.*

Proof. Let $\text{COUNT}(i, j) = \sum_{u \in S} \mathbf{1}\{N_i(u) = N_j(u)\}$ for distinct entities i, j. If i, j belong to the same true cluster $C_t^\star$ for some $t \in [k]$, we have :

$$\mathbf{E}[\text{COUNT}(i, j)] = \mathbf{E}\left[\sum_{u \in S} \mathbf{1}\{N_i(u) = N_j(u)\}\right] \geq (1 - \beta)|S|$$

Using Hoeffding's inequality, with probability at least $1 - \exp\left(-\frac{\Lambda^2}{2|S|}\right)$

$$\text{COUNT}(i, j) \geq \mathbf{E}[\text{COUNT}(i, j)] - \frac{\Lambda}{2}$$

If i, j belong to different true clusters, then, we have :

$$\mathbf{E}[\textsc{Count}(i,j)] = \mathbf{E}\left[\sum_{u \in S} \mathbf{1}\{N_i(u) = N_j(u)\}\right] \leq (1-\alpha)|S|$$

Using Hoeffding's inequality, with probability at least $1 - \exp\left(-\frac{\Lambda^2}{2|S|}\right)$

$$\textsc{Count}(i,j) < \mathbf{E}[\textsc{Count}(i,j)] + \frac{\Lambda}{2}$$

Set $\Lambda = |S|(\alpha - \beta)$ and $|S| = \frac{4 \log M/\delta}{(\alpha-\beta)^2}$.

Using union bound for every pair of entities in $[M]$, we have with probability at least $1 - \delta$:

$$\text{if entities } i, j \in C_t^\star \text{ (belong to the same true cluster) : } \textsc{Count}(i,j) \geq \left(1 - \frac{\alpha+\beta}{2}\right)|S| \text{ and}$$

$$\text{if entities } i, j \notin C_b^\star \; \forall b \in [k] \text{ (do not belong to the same true cluster) : } \textsc{Count}(i,j) < \left(1 - \frac{\alpha+\beta}{2}\right)|S|$$

Therefore, every pair of entities from same true cluster satisfy the condition that COUNT value is larger than $(1 - \frac{\alpha+\beta}{2})|S|$ and will include an edge in $\mathcal{P}$, while we do not include an edge between pair of entities from different clusters. The resulting graph $\mathcal{P}$, will have k connected components and Algorithm (α, β)-BOUNDEDDEGREE will return the true clusters correctly.

As we intervene on all the neighbors of every node in S, it will increase the interventions for every entity by a multiplicative $\Delta + 1$ factor. For an entity i, the total number of interventional distributions constructed is

$$\sum_{u \in S}(1 + |\Gamma_i(u)|) \leq |S|(\Delta+1) = 4(\Delta+1)\log(M/\delta)/(\alpha-\beta)^2 \text{ as } \max_{i \in [M], w \in V} |\Gamma_i(w)| \leq \Delta.$$

This completes the proof. □

C.1 Learning Causal Graphs from Clusters

In this section, we give the Algorithm (α, β)-RECOVERY that returns an approximate causal graph for every entity $i \in [M]$. We also include the brief overview from Section 3.2 for clarity.

Overview of Algorithm (α, β)-RECOVERY. Consider a cluster $C_a^\star$. We recover the dominant MAG of this cluster, $\mathcal{M}_a^{\text{dom}}$, by recovering all the neighbors of every node and carefully merging them. Our idea is to assign a node, selected uniformly at random, to every entity in $C_a^\star$, and recover the neighborhood of the node using Algorithms IDENTIFY-OUTNBR and IDENTIFY-BIDIRECTED. If the clusters are large such that $|C_a^\star| \gg n$ (see Theorem 3.4 for a precise bound), we can show a large number of entities T_u are assigned node u, and many of them will share the dominant MAG. We maintain a count $\textsc{NCount}(i,u)$ of the number of times the entity i agrees with other entities in T_u about neighbors of u, and guarantee (with high probability) that the entity with the highest count will be that of dominant MAG. After merging the neighbors recovered for every node, we assign the resulting graph to every entity in the cluster.

Algorithm 4 (α, β)-RECOVERY

Input: $\alpha > 0$, $\beta \geq 0$ ($< \alpha$), confidence parameter $\delta > 0$, PAGs $\mathcal{U}_1, \ldots, \mathcal{U}_M$ of M entities
Output: $\widehat{\mathcal{M}}_1, \widehat{\mathcal{M}}_2, \cdots \widehat{\mathcal{M}}_M$ representing set of M MAGs.
Obtain clusters $C_1^\star, C_2^\star, \cdots, C_k^\star$ using Algorithm 3.
for every cluster $C_a^\star$ where $a \in [k]$ **do**
  Let $\widehat{\mathcal{M}}_a^{\text{dom}}$ be an empty graph on the set of nodes V.
  For every entity $i \in C_a^\star$, select a node $u \in V$ uniformly at random and assign it to u represented by the set T_u.
  for every node $u \in V$ **do**
    for every entity $i \in T_u$ **do**
      $\text{ch}_i(u) \leftarrow$ IDENTIFY-OUTNBR$(\mathcal{U}_i, u)$
      $\text{sp}_i(u) \leftarrow$ IDENTIFY-BIDIRECTED$(\mathcal{U}_i, u)$.
      $\text{pa}_i(u) \leftarrow \Gamma_i(u) \setminus (\text{ch}_i(u) \cup \text{sp}_i(u))$.
      Construct $N_i(u)$ (defined in (2)) and calculate $\text{NCOUNT}(i, u) = \sum_{j \in T_u : j \neq i} \mathbf{1}\{N_i(u) = N_j(u)\}$
    end for
    Let $u_{\max} \leftarrow \arg\max_{i \in T_u} \text{NCOUNT}(i, u)$.
    Set neighbors of u in $\widehat{\mathcal{M}}_a^{\text{dom}}$ to the set $N_{u_{\max}}(u)$.
  end for
  For every entity $i \in C_a^\star$, set $\widehat{\mathcal{M}}_i = \widehat{\mathcal{M}}_a^{\text{dom}}$.
end for
Return $\widehat{\mathcal{M}}_1, \widehat{\mathcal{M}}_2, \cdots \widehat{\mathcal{M}}_M$

Consider the cluster $C_a^\star$ for some $a \in [k]$. In the next claim, we show that if the size of $C_a^\star$ is sufficiently large, then, each node $u \in V$ is assigned a large number of entities by (α, β)-RECOVERY using the set T_u.

Claim C.2. *Consider a cluster $C_a^\star$ such that $\gamma_a > 1/2$ and $|C_a^\star| \geq \frac{8n \log(nM/\delta)}{(2\gamma_a - 1)^2}$. Let T_u denote the set of entities assigned to node u in Algorithm 4. Then, we have with probability $1 - \delta$, $|T_u| \geq \frac{4 \log(nM/\delta)}{(2\gamma_a - 1)^2}$ for every node $u \in V$.*

Proof. For a node $u \in V$, and cluster $C_a^\star$, we have:

$$\mathbf{E}[T_u] = \frac{|C_a^\star|}{n} \geq \frac{8 \log(nM/\delta)}{(2\gamma_a - 1)^2}.$$

Using Chernoff bound, with probability at least $1 - \exp\left(-\log(nM/\delta)/(2\gamma_a - 1)^2\right) \geq 1 - \delta/nM$, we have:

$$T_u \geq \frac{\mathbf{E}[T_u]}{2} \geq \frac{4 \log(nM/\delta)}{(2\gamma_a - 1)^2}.$$

Applying union bound for every node $u \in V$ and $a \in [k]$, gives us the claim. □

Consider a partitioning of $C_a^\star$ given by $S_a^1, S_a^2, \cdots S_a^t$ where each S_a^i for any $i \in [t]$ represents the maximal collection of MAGs that are equal. Formally, we have:

$$S_a^i = \{\mathcal{M}_p \mid \mathcal{M}_p \in C_a^\star \text{ and } \mathcal{M}_p = \mathcal{M}_q \quad \forall \mathcal{M}_q \in S_a^i\}.$$

Let $|S_a^{\text{dom}}| \geq |S_a^i|$ for every partition $i \in [t]$ and dom_a denote an entity in S_a^{dom}. We define:

$$G_a(u) = \{j \mid j \in C_a^\star \text{ and } N_i(u) = N_{\text{dom}_a}(u)\} \text{ and } B_a(u) = C_a^\star \setminus G_a(u).$$

We can observe that:

$$|G_a(u)| \geq |S_a^{\text{dom}}| \text{ and } |B_a(u)| \leq |C_a^\star| - |S_a^{\text{dom}}|.$$

Conditioned on the previous claim that each set T_u for all $u \in V$ is large, we argue that for any pair of entities $i, j \in C_a^\star$ where $\mathcal{M}_i = \mathcal{M}_a^{\text{dom}}$, and $\mathcal{M}_j \neq \mathcal{M}_a^{\text{dom}}$, the NCOUNT value calculated by (α, β)-RECOVERY of entity i for the node u is always larger than that of entity j. Intuitively, after assigning the entities to nodes, we observe that for every node $u \in V$, the set T_u contains a large number of entities with the dominant MAG, i.e., $|T_u \cap G_a(u)|$ is large. Because dominant MAGs share the same neighborhood (as they represent the same graph), we can show that the NCOUNT value of dominant MAG is larger than any other MAG in the cluster. We formalize this statement using the following lemma.

Lemma C.3. *For every $a \in [k], u \in V$ and any pair of entities $i, j \in C_a^\star$ that satisfy $i \in G_a(u)$ and $j \in B_a(u)$, we have with probability $1-\delta$,*

$$\text{NCount}(i,u) > \text{NCount}(j,u).$$

Proof. From Algorithm 4, we know that $\text{NCount}(i,u) = \sum_{j \neq i, j \in T_u} \mathbf{1}\{N_i(u) = N_j(u)\}$ for an entity $i \in T_u$ and a node $u \in V$. Consider the case $i \in G_a(u)$. Then, we have:

$$\begin{aligned}
\mathbf{E}[\text{NCount}(i,u)] &= \mathbf{E}\left[\sum_{j \neq i, j \in T_u} \mathbf{1}\{N_i(u) = N_j(u)\}\right] \\
&= \mathbf{E}\left[\sum_{j \neq i, j \in T_u} \mathbf{1}\{N_{\text{dom}_a}(u) = N_j(u)\}\right] \\
&= \mathbf{E}[|T_u \cap G_a(u)|] \\
&\geq \frac{|S_a^{\text{dom}}|}{n} = \frac{|S_a^{\text{dom}}|}{|C_a^\star|} \cdot \frac{|C_a^\star|}{n} = \gamma_a \cdot \frac{|C_a^\star|}{n}
\end{aligned}$$

Using Hoeffding's inequality, with probability at least $1 - \exp\left(-\frac{\Lambda^2}{2|T_u|}\right)$

$$\text{NCount}(i,u) \geq \mathbf{E}[\text{NCount}(i,u)] - \frac{\Lambda}{2} \geq \gamma_a \cdot \frac{|C_a^\star|}{n} - \frac{\Lambda}{2}$$

If $i \in B_a(u)$, then, we have :

$$\begin{aligned}
\mathbf{E}[\text{NCount}(i,u)] &= \mathbf{E}\left[\sum_{j \neq i, j \in T_u} \mathbf{1}\{N_i(u) = N_j(u)\}\right] \\
&= \mathbf{E}\left[\sum_{j \neq i, j \in T_u \cap G_a(u)} \mathbf{1}\{N_i(u) = N_{\text{dom}_a}(u)\} + \sum_{j \neq i, j \in T_u \cap B_a(u)} \mathbf{1}\{N_i(u) = N_j(u)\}\right] \\
&= \mathbf{E}\left[\sum_{j \neq i, j \in T_u \cap B_a(u)} \mathbf{1}\{N_i(u) = N_j(u)\}\right] \\
&= \mathbf{E}[|T_u \cap B_a(u)|] \\
&\leq \frac{|C_a^\star| - |S_a^{\text{dom}}|}{n} = \left(1 - \frac{|S_a^{\text{dom}}|}{|C_a^\star|}\right) \cdot \frac{|C_a^\star|}{n} = (1-\gamma_a) \cdot \frac{|C_a^\star|}{n}
\end{aligned}$$

Using Hoeffding's inequality, with probability at least $1 - \exp\left(-\frac{\Lambda^2}{2|T_u|}\right)$

$$\text{NCount}(i,u) < \mathbf{E}[\text{Count}(i,u)] + \frac{\Lambda}{2} < (1-\gamma_a) \cdot \frac{|C_a^\star|}{n} + \frac{\Lambda}{2}$$

Set $\Lambda = \frac{|C_a^\star|}{n}(2\gamma_a - 1)$ and $|T_u| \geq \frac{4\log(nM/\delta)}{(2\gamma_a - 1)^2}$. Then, for any pair of entities $i, j \in C_a^\star$ such that $i \in G_a(u)$ and $j \in B_a(u)$, we have, with a probability $1 - \delta/nM^2$:

$$\text{NCount}(i,u) > \text{NCount}(j,u).$$

Using union bound for every pair of entities in $[M]$ and $u \in V$, with probability at least $1-\delta$, we have the final claim. □

From the previous Lemma C.3, we know that NCount values are always larger for the dominant MAG partition, and therefore merging the neighborhoods of all the nodes gives us the dominant MAG. As dominant MAG is within a distance of at most $\beta \cdot n$ from every MAG in the cluster, the dominant MAG returned is a sufficiently good approximation of the true MAG. We formalize this using the following statement.

Theorem C.4 (Theorem 3.4 Restated). *Suppose $\mathcal{M}_1, \mathcal{M}_2, \cdots \mathcal{M}_M$ satisfy (α, β) clustering property. If $\gamma_a > 1/2$ and $C_a^\star = \Omega(\frac{n \log(n/M\delta)}{(2\gamma_a - 1)^2})$ for all $a \in [k]$, then, Algorithm (α, β)-RECOVERY recovers graphs $\widehat{\mathcal{M}}_1, \cdots \widehat{\mathcal{M}}_M$ such that for every entity $i \in [M]$, we have $d(\mathcal{M}_i, \widehat{\mathcal{M}}_i) \leq \beta n$ with probability $1 - \delta$. Moreover, every entity uses at most $(\Delta + 1) + \frac{4(\Delta+1)\log(M/\delta)}{(\alpha-\beta)^2}$ many atomic interventions.*

Proof. From Lemma C.3, we have that $\text{NCOUNT}(i, u) > \text{NCOUNT}(j, u)$, which implies $u_{\max} \in G_a(u)$. Using Algorithm 4, every entity i in the cluster $C_a^\star$ is assigned the graph $\widehat{\mathcal{M}}_i = \mathcal{M}_a^{\text{dom}}$. From the definition of $(\alpha, \beta)-$clustering property, we have that all entities $i \in C_a^\star$ are such that $d(\mathcal{M}_i, \widehat{\mathcal{M}}_i) = d(\mathcal{M}_i, \mathcal{M}_a^{\text{dom}}) \leq \beta n$.

Using Algorithm 4 we assign every entity to a single node $u \in V$, and perform at most $\Delta + 1$ interventions to identify all the neighbors of u for every entity in T_u. Therefore, we perform at most $\Delta + 1$ interventions per entity. For obtaining clusters, from Lemma 3.3, we know that every entity performs at most $\frac{4(\Delta+1)\log(M/\delta)}{(\alpha-\beta)^2}$ interventions. Hence, the theorem. □

D Discovery under α-Clustering Property

D.1 From α-Clustering to Learning Causal Graphs

Suppose that the underlying MAGs $\mathcal{M}_1, \mathcal{M}_2, \cdots, \mathcal{M}_M$ satisfy the α-clustering property, our algorithms are based on first accurately recovering these clusters. The idea of going from clusters to MAGs is simple and is based on distributing the interventions across the entities in the cluster. We now discuss a meta-algorithm that returns the associated causal MAG of every entity given the true clustering. Our meta-algorithm takes as input the true clusters $C_1^\star, C_2^\star, \cdots, C_k^\star$ and recovers the MAGs associated with each of them. In any cluster $C_b^\star$ such that $|C_b^\star| < n$, our meta-algorithm uses an additional $\lceil n/|C_b^\star| \rceil$ many interventions for each entity in $C_b^\star$. For clusters satisfying $|C_b^\star| \geq n$, it uses an extra intervention per entity.

Meta-Algorithm. Consider a true cluster $C_b^\star$ ($b \in [k]$). Construct a mapping ϕ that partitions the n nodes in V among all the entities in $C_b^\star$, such that no entity is assigned to more than $\lceil n/|C_b^\star| \rceil$ many nodes. By definition, all entities in $C_b^\star$ have the same PAG. Let $\mathcal{U}$ be the common PAG. Construct a MAG $\mathcal{M}$ from $\mathcal{U}$ as follows. Consider an edge (u, v) in $\mathcal{U}$. Let $u = \phi(i)$ and $v = \phi(j)$ where the entities $i, j \in C_b^\star$ are such that we intervene on node u in entity i and node v in entity j (i could be equal to j). Now, if $v \in \text{ch}_i(u)$, we add $u \to v$ into the graph $\mathcal{M}$, else if $u \in \text{ch}_j(v)$, we add $u \leftarrow v$, and $u \leftrightarrow v$ otherwise. We assign graph $\mathcal{M}$ for every entity in $C_b^\star$. Repeating this procedure for every $C_b^\star$ generates the M MAGs, one for each entity.

Lemma D.1. *Suppose there is an Algorithm $\mathcal{A}$ that recovers the true clusters $C_1^\star, C_2^\star, \cdots, C_k^\star$ of the underlying MAGs $\mathcal{M}_1, \mathcal{M}_2, \cdots, \mathcal{M}_M$ satisfying α-clustering property such that every entity $i \in [M]$ uses at most $f(M)$ interventions. Then, there is an algorithm that can learn all the MAGs $\mathcal{M}_1, \mathcal{M}_2, \cdots, \mathcal{M}_M$ such that every entity $i \in [M]$ uses at most $f(M) + \lceil n/\Upsilon \rceil$ many interventions, where $\Upsilon = \min_{b \in [k]} C_b^\star$.*

Proof. Consider a cluster $C_b^\star$ for $b \in [k]$. As the mapping ϕ assigns every entity at most $\lceil n/|C_b^\star| \rceil$ many nodes to intervene on, we have that every entity in $C_b^\star$ uses at most $\lceil n/|C_b^\star| \rceil$ additional interventions. Therefore, over all true clusters, every entity uses at most $f(M) + \lceil n/\Upsilon \rceil$ many interventions.

Consider any cluster $C_b^\star$. The mapping ϕ in the Meta-Algorithm is well-defined and satisfies the claim that for every node $u \in V$, there exists an entity in $C_b^\star$ for which we construct an interventional distribution $\text{do}(u)$. Therefore, for cluster $C_b^\star$, we have n interventional distributions one for every node in V, and we use Algorithm RECOVERG to learn the MAG for this cluster (i.e., MAG for all the entities in $C_b^\star$). Repeating this for every cluster $C_1^\star, C_2^\star, \cdots, C_k^\star$, we obtain all the MAGs $\mathcal{M}_1, \mathcal{M}_2, \cdots, \mathcal{M}_M$. □

If the clusters are of size at least n, i.e., $\min_{b \in [k]} |C_b^\star| \geq n$, then, we have the following corollary from Lemma D.1.

Corollary D.2. *Suppose there is an Algorithm $\mathcal{A}$ that recovers the true clusters $C_1^\star, C_2^\star, \cdots, C_k^\star$ of the underlying MAGs $\mathcal{M}_1, \mathcal{M}_2, \cdots, \mathcal{M}_M$ satisfying the α-clustering property such that every entity $i \in [M]$ uses at most $f(M)$ interventions. Suppose $\min_{b\in[k]} |C_b^\star| \geq n$. Then, there is an algorithm that can learn all the MAGs $\mathcal{M}_1, \mathcal{M}_2, \cdots, \mathcal{M}_M$ such that every entity $i \in [M]$ uses at most $f(M) + 1$ many interventions.*

D.2 Discovery without Latents

In this section, we present a randomized algorithm that recovers (with high probability) all the M MAGs $\mathcal{M}_1, \ldots, \mathcal{M}_M$ when the underlying data generating process for each of these entities do not have any latents (i.e., causal DAGs $\mathcal{D}_1, \ldots, \mathcal{D}_M$ satisfy causal sufficiency). This translates into the fact that the MAGs $\mathcal{M}_1, \ldots, \mathcal{M}_M$ do not have bidirected edges.

We first make the observation that to identify that two graphs, say $\mathcal{M}_i$ and $\mathcal{M}_j$ belong to different clusters, it suffices to find a node u from the node-difference set $\mathrm{diff}(\mathcal{M}_i, \mathcal{M}_j)$ and checking their outgoing neighbors using Algorithm IDENTIFY-OUTNBR. We argue that, with probability at least $1-\delta$, we can identify one such node $u \in \mathrm{diff}(\mathcal{M}_i, \mathcal{M}_j)$ by sampling $2\log(M/\delta)/\alpha$ nodes uniformly from V as $|\mathrm{diff}(\mathcal{M}_i, \mathcal{M}_j)| = d(\mathcal{M}_i, \mathcal{M}_j) \geq \alpha n$.

In Algorithm NOLATENTS, we obtain a sample of nodes S and construct interventional distribution for every entity in $[M]$, and for every node in S. After finding the outgoing neighbors for every entity i and node in S, we construct a graph $\mathcal{P}$ on entities (i.e., the node set of $\mathcal{P}$ is $[M]$). We include an edge between two entities if they share the same outgoing neighbors for every $u \in S$. This ensures that every entity is connected only to the entities belonging to the same true cluster, and we return the connected components in $\mathcal{P}$ as our clusters.

Algorithm 5 NOLATENTS

Input: $\alpha > 0$, confidence parameter $\delta > 0$, PAGs $\mathcal{U}_1, \ldots, \mathcal{U}_M$ of M entities.
Output: Partition of $[M]$ into clusters
Let S denote a uniform sample of $\frac{2\log M/\delta}{\alpha}$ nodes from V selected with replacement.
for every entity $i \in [M]$ and $u \in S$ **do**
    $\mathrm{ch}_i(u) \leftarrow$ IDENTIFY-OUTNBR(i, u)
end for
Let $\mathcal{P}$ denote an empty graph on set of entities $[M]$
for every pair of entities i, j **do**
    if $\mathrm{ch}_i(u) = \mathrm{ch}_j(u)$ and $\Gamma_i(u) = \Gamma_j(u)$ for every $u \in S$ **then**
        Add an edge between entities i and j in $\mathcal{P}$
    end if
end for
Return connected components in $\mathcal{P}$

Claim D.3. *Let S denote a set of $2\log(M/\delta)/\alpha$ nodes sampled with replacement uniformly from V. Then, for every pair of entities i, j that belong to different true clusters, we have with probability at least $1-\delta$, $u \in \mathrm{diff}(\mathcal{M}_i, \mathcal{M}_j)$ for some $u \in S$.*

Proof. Let S denote a set of sampled nodes such that $|S| = 2\log(M/\delta)/\alpha$. Therefore, we have

$$\Pr_{u\sim V}[u \in \mathrm{diff}(\mathcal{M}_i, \mathcal{M}_j)] \geq \alpha, \text{ and}$$

$$\Pr_{S\sim V}[\,\forall u \in S \ : u \notin \mathrm{diff}(\mathcal{M}_i, \mathcal{M}_j)] \leq (1-\alpha)^{|S|}$$

$$\leq e^{-\alpha|S|} \leq \frac{\delta}{M^2}.$$

Using union bound for every pair of entities in $[M]$ that belong to two different clusters, we have:

$$\forall i, j \in [M],\ \Pr_{S\sim V}[\forall\, u \in S, u \notin \mathrm{diff}(\mathcal{M}_i, \mathcal{M}_j)] \leq \delta.$$

Therefore, for every pair of entities $i, j \in [M]$ belonging to different true clusters, there exists $u \in S$ such that :

$$\Pr[u \in \mathrm{diff}(\mathcal{M}_i, \mathcal{M}_j)] \geq 1-\delta.$$

□

Lemma D.4. *Assume causal sufficiency. If MAGs $\mathcal{M}_1, \ldots, \mathcal{M}_M$ satisfy α-clustering property with true clusters $C_1^\star, \ldots, C_k^\star$, then Algorithm* NoLatents *exactly recovers the clusters $C_1^\star, \ldots, C_k^\star$ with probability at least $1-\delta$. Every entity $i \in [M]$ uses $2\log(M/\delta)/\alpha$ many atomic interventions.*

Proof. Consider two entities i, j and their corresponding MAGs $\mathcal{M}_i$ and $\mathcal{M}_j$ respectively. We first observe that if the PAGs of these two entities are different then they belong to different clusters. Now consider the case where the PAGs for both these entities are the same, i.e., $\mathcal{U}_i = \mathcal{U}_j$. Now if i and j belong to different true clusters, then we claim that it suffices to find a node u from the node-difference set $\text{diff}(\mathcal{M}_i, \mathcal{M}_j) = \{u \mid N_i(u) \neq N_j(u)\}$ to notice this fact. As there are no latents (causal sufficiency), we can identify whether $u \in \text{diff}(\mathcal{M}_i, \mathcal{M}_j)$, by checking only the outgoing neighbors of u for entities i, j, i.e., $\text{diff}(\mathcal{M}_i, \mathcal{M}_j) = \{u \mid \text{ch}_i(u) \neq \text{ch}_j(u)\}$. When we identify such a node u, the set of outgoing neighbors of node u are different for entities i, j, and therefore must belong to different true clusters (by α-clustering property). In order to identify at least one node $u \in \text{diff}(\mathcal{M}_i, \mathcal{M}_j)$, we use sampling.

Let S denote the set of sampled nodes (with replacement) from V such that $|S| = 2\log(M/\delta)/\alpha$. In Algorithm NoLatents, we construct interventional distributions for every node $u \in S$, for every entity $i \in [M]$. Using these interventional distributions we obtain the outgoing neighbors of nodes in S using Algorithm Identify-OutNbr.

From Claim D.3, we have that for every pair of entities i, j belonging to different true clusters, there exists $u \in S$ such that:

$$\Pr[\text{ch}_i(u) \neq \text{ch}_j(u)] = \Pr[u \in \text{diff}(\mathcal{M}_i, \mathcal{M}_j)] \geq 1 - \delta.$$

This implies that, with probability at least $1-\delta$, for every i, j pair we have the following: in the entity graph $\mathcal{P}$ there would not be an edge between i, j if they belong to different true clusters, and there would be an edge if they belong to the same true cluster. The resulting graph $\mathcal{P}$, will have k connected components and Algorithm NoLatents will return the true clusters correctly.

Hence, with probability at least $1-\delta$, we can recover all the true clusters $C_1^\star, \ldots, C_k^\star$ using Algorithm NoLatents. □

Theorem D.5. *Assume causal sufficiency. If MAGs $\mathcal{M}_1, \ldots, \mathcal{M}_M$ satisfy α-clustering property with true clusters $C_1^\star, \ldots, C_k^\star$ then Algorithm* NoLatents *exactly recovers these clusters with probability at least $1-\delta$. Furthermore, if $\min_{b\in[k]} |C_b^\star| \geq n$, then there is an algorithm that exactly learns all these MAGs with probability at least $1-\delta$. Every entity $i \in [M]$ uses $2\log(M/\delta)/\alpha + 1$ many atomic interventions.*

Proof. From Lemma D.4, we can recover the clusters correctly with probability at least $1-\delta$. Using the Meta-Algorithm discussed in Section D.1, we can learn the graphs of every entity with a single additional intervention (see Corollary D.2). This establishes the result. □

D.3 Discovery with Latents: Bounded Degree MAGs

Throughout this section, we let :

$$\Delta = \max_{i\in[M], u\in V} |\Gamma_i(u)|.$$

We now discuss an algorithm that recovers clusters $C_1^\star, C_2^\star \cdots, C_k^\star$ using ideas developed in Section D.2 but now with latents in the system. In the presence of latents, the collection of MAGs $\mathcal{M}_1, \ldots, \mathcal{M}_M$ are mixed graphs that also contain bidirected edges, which introduces issues, as bidirected edges cannot be detected easily. For example, two entities i and j might be such that $u \leftrightarrow v$ could be present in $\mathcal{M}_i$ and $u \leftarrow v$ could be present in $\mathcal{M}_j$, in which case intervening on just u alone will not suffice to distinguish i from j, we need interventions on both u and v. This is the idea behind Algorithm α-BoundedDegree, which identifies all the outgoing and bidirected edges incident on the set of sampled nodes (say S), for every entity in $[M]$. Since from this we can compute all neighboring relations of u ($N_i(u)$), Algorithm α-BoundedDegree then checks whether these neighborhoods are the same or not for every node $u \in S$. We can now leverage the α-clustering property to argue that this process succeeds with probability at least $1-\delta$.

As we use Algorithm IDENTIFY-BIDIRECTED, to find all bidirected edges incident on a node $u \in S$, we use an additional $O(\Delta)$ atomic interventions (per entity) where $\Delta = \max_{i\in[M],u\in V} \Gamma_i(u)$ is the maximum undirected degree in the PAGs $\mathcal{U}_1, \ldots, \mathcal{U}_M$.

Algorithm 6 α-BOUNDEDDEGREE

Input: $\alpha > 0$, confidence parameter $\delta > 0$, PAGs $\mathcal{U}_1, \ldots, \mathcal{U}_M$ of M entities
Output: Partition of $[M]$ into clusters
Let S denote a uniform sample of $\frac{2\log M/\delta}{\alpha}$ nodes from V selected with replacement.
for every entity $i \in [M]$ and $u \in S$ **do**
  $\text{ch}_i(u) \leftarrow$ IDENTIFY-OUTNBR(i, u)
  $\text{sp}_i(u) \leftarrow$ IDENTIFY-BIDIRECTED(i, u)
  $\text{pa}_i(u) \leftarrow \Gamma_i(u) \setminus (\text{ch}_i(u) \cup \text{sp}_i(u))$
  Construct $N_i(u)$ (defined in (2))
end for
Let $\mathcal{P}$ denote an empty graph on set of entities $[M]$
for every pair of entities i, j **do**
  if $N_i(u) = N_j(u)$ for every $u \in S$ **then**
    Include an edge between i and j in $\mathcal{P}$
  end if
end for
Return connected components in $\mathcal{P}$

Lemma D.6. *If the underlying MAGs $\mathcal{M}_1, \ldots, \mathcal{M}_M$ satisfy α-clustering property with true clusters $C_1^\star, \ldots, C_k^\star$, then Algorithm α-BOUNDEDDEGREE exactly recovers the clusters $C_1^\star, \ldots, C_k^\star$ with probability at least $1 - \delta$. Every entity $i \in [M]$ uses at most $2(\Delta + 1)\log(M/\delta)/\alpha$ many atomic interventions.*

Proof. We follow a proof idea similar to Lemma D.4. Again if two entities i, j have different PAGs then they belong to different true clusters.

Consider two entities i, j belonging to different true clusters but having the same PAG. Again it suffices to find a node u from the node-difference set $\text{diff}(\mathcal{M}_i, \mathcal{M}_j) = \{u \mid N_i(u) \neq N_j(u)\}$ to conclude that they belong to different clusters.

As there are latents (causal sufficiency), we cannot identify whether $u \in \text{diff}(\mathcal{M}_i, \mathcal{M}_j)$, by checking only the outgoing neighbors of u for entities i, j, and have to check the set of bidirected edges incident on u as well. We can identify all the bidirected edges incident on u for both i, j using Algorithm IDENTIFY-BIDIRECTED. Identifying such a node u, whose set of neighbors of node u are different for entities i, j, provides a certificate that i, j belong to different true clusters (α-clustering property). In order to identify at least one node $u \in \text{diff}(\mathcal{M}_i, \mathcal{M}_j)$, we use sampling.

Let S denote the set of sampled nodes (with replacement) from V such that $|S| = 2\log(M/\delta)/\alpha$. In Algorithm α-BOUNDEDDEGREE, we construct interventional distributions for every node $u \in S$ and all the neighbors in the PAG given by $\Gamma_i(u)$, for every entity $i \in [M]$. From these interventional distributions, we can compute $N_i(u)$ and $N_j(u)$ for all the nodes $u \in S$ (using Algorithms IDENTIFY-OUTNBR and IDENTIFY-BIDIRECTED).

From Claim D.3, we have that for every pair of entities i, j belonging to different true clusters, there exists $u \in S$ such that:

$$\Pr[N_i(u) \neq N_j(u)] = \Pr[u \in \text{diff}(\mathcal{M}_i, \mathcal{M}_j)] \geq 1 - \delta.$$

Hence, with probability at least $1 - \delta$, we can recover all the true clusters using Algorithm α-BOUNDEDDEGREE.

For an entity i, the total number of interventional distributions constructed is

$$\sum_{u\in S}(1 + |\Gamma_i(u)|) \leq |S|(\Delta + 1) = 2(\Delta + 1)\log(M/\delta)/\alpha \text{ as } \max_{i\in[M],w\in V} |\Gamma_i(w)| \leq \Delta.$$

□

Theorem D.7. *If the underlying MAGs $\mathcal{M}_1, \ldots, \mathcal{M}_M$ satisfy α-clustering property with true clusters $C_1^\star, \ldots, C_k^\star$, then Algorithm α-BOUNDEDDEGREE exactly recovers these clusters with probability at least $1-\delta$. Furthermore, if $\min_{b\in[k]} |C_b^\star| \geq n$, then there is an algorithm that exactly learns all these MAGs with probability at least $1-\delta$. Every entity $i \in [M]$ uses at most $2(\Delta+1)\log(M/\delta)/\alpha + 1$ many atomic interventions.*

Proof. From Lemma D.6, we can recover the clusters correctly with probability at least $1-\delta$. Using the Meta-Algorithm discussed in Section D.1, and from Corollary D.2, we can obtain an algorithm to learn the graphs of every entity with an additional intervention per entity. This completes the proof. □

D.4 Missing Details from Section 4

In Algorithm α-GENERAL, we obtain all the outgoing neighbors of the sampled set of nodes S. Then, we construct a graph on set of entities, $[M]$ such that an edge between a pair of entities i, j is included if they share same PAGs, i.e., $\mathcal{U}_i = \mathcal{U}_j$ and same outgoing neighbors for every node in S. However, it is possible that the graph $\mathcal{P}$ can contain more than one true cluster. In the next lemma, we show that we can detect this, and remove all the edges between entities belonging to two different clusters using 2 interventions.

Lemma D.8. *Suppose a component T_a in $\mathcal{P}_{\text{itr}}$ for some $\text{itr} \geq 1$ contains all the entities from two true clusters $C_b^\star, C_c^\star$. If $\min_{r\in[k]} |C_r^\star| \geq \Omega(n \log M/\delta)$, then, we can identify, with a probability $1-\delta/2k^2$, all the pairs of entities $i', j' \in T_a$ such that $i' \in C_b^\star$ and $j' \in C_c^\star$ (or vice-versa) using at most 2 interventions for every entity in T_a .*

Proof. We claim that if a component T_a containing $C_b^\star$ and $C_c^\star$ exists, then, we can identify a pair of entities i, j that are joined by an edge in $\mathcal{P}_{\text{itr}}$ such that $i \in C_b^\star$ and $j \in C_c^\star$ or vice-versa.

It suffices to find a node u from the node-difference set $\text{diff}(\mathcal{M}_i, \mathcal{M}_j) = \{u \mid N_i(u) \neq N_j(u)\}$ to conclude that they belong to different clusters. From Claim D.3, we know that when $|S| = 2\log(2M/\delta)/\alpha$, we can identify such a $u \in \text{diff}(\mathcal{M}_i, \mathcal{M}_j)$ with a probability $1-\delta/2$. We make the observation that a pair of entities i, j that have an edge in this $\mathcal{P}_{\text{itr}}$ and from different true clusters, can differ only if there is a node $u \in \text{diff}(\mathcal{M}_i, \mathcal{M}_j)$ such that u has a bidirected edge $u \leftrightarrow v$ in $\mathcal{M}_i$, and a directed edge $u \leftarrow v$ in $\mathcal{M}_j$ (or vice-versa). Intervening on both u and v will separate these entities, our main idea is to ensure that this happens.

Consider a mapping $\pi : [M] \to V$ where $\pi(i)$ is assigned a node from V selected uniformly at random. Using this mapping, we ensure that there are two entities $i \in T_a \cap C_b^\star, j \in T_a \cap C_c^\star$ joined by an edge, such that $\pi(i) = \pi(j) = v$ and $u \leftrightarrow v$ in $\mathcal{M}_i$, $u \leftarrow v$ in $\mathcal{M}_j$ (or vice-versa) for some $u \in S$. We have:

$$\Pr[\text{ for any } i \in T_a,\ \pi(i) \neq v] = 1 - \frac{1}{n}, \text{ and}$$
$$\Pr[\forall i \in C_b^\star\ : \pi(i) \neq v] = \left(1 - \frac{1}{n}\right)^{|C_b^\star|}.$$

Similarly, we have

$$\Pr[\forall j \in C_c^\star\ : \pi(j) \neq v] = \left(1 - \frac{1}{n}\right)^{|C_c^\star|}.$$

$$\begin{aligned}\Pr[\forall\, i \in C_b^\star, j \in C_c^\star \text{ such that } \pi(i) \neq v,\ \pi(j) \neq v] &= \left(1 - \frac{1}{n}\right)^{|C_b^\star|+|C_c^\star|} \\ &\leq \frac{\delta}{2M^2} \leq \frac{\delta}{2k^2} \\ \Rightarrow \Pr[\exists i \in C_b^\star, \exists j \in C_c^\star\ : \pi(i) = \pi(j) = v] &\geq 1 - \frac{\delta}{2k^2}.\end{aligned}$$

As we intervene on $\pi(i)$ for every entity $i \in T_a$, we know that there exists $i \in C_b^\star, j \in C_c^\star$, both in T_a and that are assigned v by π. Therefore, we can separate i, j and remove the edge from $\mathcal{P}_{\text{itr}}$. Now, we create an intervention on $\pi(i) = \pi(j) = v$ for every entity in T_a and separate all the entity pairs

(i', j') joined by an edge in $\mathcal{P}_{\text{itr}}$ that satisfy: $u \leftarrow v$ in $\mathcal{M}_{i'}$ and $u \leftrightarrow v$ in $\mathcal{M}_{j'}$ (or vice-versa). As we use at most two interventions for every entity in T_a, the lemma follows. □

Algorithm 7 α-GENERAL

Input: $\alpha > 0$, confidence parameter $\delta > 0$, PAGs $\mathcal{U}_1, \ldots, \mathcal{U}_M$ of M entities $\mathcal{U}_1, \ldots, \mathcal{U}_M$)
Output: Partition of $[M]$ into clusters
Let S denote a uniform sample of $\frac{2\log(2M/\delta)}{\alpha}$ nodes from V selected with replacement.
for every entity $i \in [M]$ and $u \in S$ **do**
  $\text{ch}_i(u) \leftarrow$ IDENTIFY-OUTNBR(i, u)
end for
Let $\mathcal{P}$ denote an empty graph on the set of entities $[M]$.
for every pair of entities i, j **do**
  if $\text{ch}_i(u) = \text{ch}_j(u)$ and $\Gamma_i(u) = \Gamma_j(u)$ $\forall u \in S$ **then**
    Include an edge between i and j in $\mathcal{P}$
  end if
end for
$\text{itr} \leftarrow 1$, $\mathcal{P}_0 \leftarrow \mathcal{P}$
while TRUE **do**
  $\mathcal{P}_{\text{itr}} \leftarrow \mathcal{P}_{\text{itr}-1}$
  Let $T_1, T_2, \cdots$ denote the components in $\mathcal{P}_{\text{itr}}$.
  For all $i \in [M]$, obtain interventional distribution on $\pi(i)$ picked u.a.r from V.
  if $\exists$ edge $(i, j) \in \mathcal{P}_{\text{itr}}$ in component T_a such that $\pi(i) = \pi(j)$ **then**
    Let $v = \pi(i) = \pi(j)$
    if $v \in \text{sp}_i(u), v \notin \text{sp}_j(u)$ (or vice-versa) for some $u \in S$ **then**
      Intervene on v for every entity in T_a.
      Remove edge (i', j') from $\mathcal{P}_{\text{itr}}$ if $v \in \text{sp}_{i'}(u)$, $v \notin \text{sp}_{j'}(u)$ (or vice-versa) for every $i', j' \in T_a$
    end if
  end if
  if the set of edges in $\mathcal{P}_{\text{itr}}$ are *same* as the set of edges in $\mathcal{P}_{\text{itr}-1}$ **then**
    Return connected components in $\mathcal{P}_{\text{itr}}$
  end if
  $\text{itr} \leftarrow \text{itr} + 1$
end while

Lemma D.9. *If the underlying MAGs satisfy α-clustering property with true clusters $C_1^\star, \ldots, C_k^\star$ such that $\min_{b\in[k]} C_b^\star = \Omega(n \log(M/\delta))$ entities, the Algorithm α-*GENERAL *exactly recovers the clusters $C_1^\star, \ldots, C_k^\star$ with probability at least $1 - \delta$. Every entity $i \in [M]$ uses at most $O(\log(M/\delta)/\alpha + k^2)$ many atomic interventions.*

Proof. From Claim D.3, with probability at least $1 - \delta/2$, we have that the set of sampled nodes S (where $|S| = 2\log(2M/\delta)/\alpha$) satisfy that for every pair of entities from different clusters there is a node $u \in S$ that can be used to identify that they belong to different clusters. Using Lemma D.8, we have that, in every iteration itr, we remove all the edges in $\mathcal{P}_{\text{itr}}$ between entities that are part of the same component but from different true clusters. After k^2 iterations, we would have separated all the pairs of entities between all the true clusters. In Algorithm α-GENERAL, we return the connected components in $\mathcal{P}_{\text{itr}}$ when there is no change in the set of edges between entities between $\mathcal{P}_{\text{itr}-1}$ and $\mathcal{P}_{\text{itr}}$.

From Lemma D.8, we have that, every entity performs at most $|S| + 2k^2$ interventions. As there are at most k^2 iterations, and from Lemma D.8, each iteration fails with probability at most $\delta/2k^2$, using union bound, we have that at least one of the iterations fails with probability at most $\delta/2$.

Finally, using union bound for failure probability of calculating S correctly, and failing in at least one of the iterations, we have, with probability at least $1 - \delta$, Algorithm α-GENERAL recovers the true clusters. □

From Lemma D.9, we know that we can recover the clusters correctly with probability at least $1-\delta$. Using the Meta-Algorithm discussed in Appendix D.1, and from Corollary D.2, we can obtain an algorithm to learn the graphs of every entity with an additional intervention per entity. Combining it with guarantees obtained by Algorithm α-BOUNDEDDEGREE in Theorem D.7, gives us the following result.

Theorem D.10 (Theorem 4.1 Restated)**.** *If MAGs $\mathcal{M}_1, \ldots, \mathcal{M}_M$ satisfy α-clustering property with true clusters $C_1^\star, \ldots, C_k^\star$ such that $\min_{b\in[k]} |C_b^\star| = \Omega(n \log(M/\delta))$. Then, there is an algorithm that exactly learns all these MAGs with probability at least $1-\delta$. Every entity $i \in [M]$ uses $\min\left\{O(\Delta \log(M/\delta)/\alpha), O(\log(M/\delta)/\alpha + k^2)\right\}$ many atomic interventions.*

D.5 Lower Bound on the Number of Interventions

In this section, we present a lower bound for the number of interventions required by every entity to recover true clusters. First, we state Yao's minimax theorem, which will be used to prove the lower bound.

Theorem D.11 (Yao's minimax theorem [Yao, 1977])**.** *Let $\mathcal{X}$ be a set of inputs to a problem and $\mathcal{A}$ the set of all possible deterministic algorithms that solve the problem. For any algorithm $A \in \mathcal{A}$ and $x \in \mathcal{X}$, let cost(A, x) denote real-valued measure of cost of an algorithm A on input x. Let ν, μ be distributions over $\mathcal{A}$ and $\mathcal{X}$ respectively. Then,*

$$\max_{x\in\mathcal{X}} \mathop{\mathbf{E}}_{A\sim_\nu \mathcal{A}}[\textit{cost}(A, x)] \geq \min_{a\in\mathcal{A}} \mathop{\mathbf{E}}_{X\sim_\mu \mathcal{X}}[\textit{cost}(a, X)]$$

Informally, the theorem states that to prove lower bounds on the cost of *any* randomized algorithm, we have to find some distribution μ on inputs, such that *every* deterministic algorithm $A \in \mathcal{A}$ has high cost.

Outline of the Lower Bound. Our distribution μ places a probability of $1/2$ for pairs of MAGs that have distance zero and a probability of $1/2$ equally distributed among all pairs of MAGs with distance equal to αn. This ensures that both the events considered are equally likely, and we show that to distinguish them, with success probability at least $2/3$ (over the distribution μ), every deterministic algorithm must make $\Omega(1/\alpha)$ interventions for both the MAGs. Then, we use Yao's theorem to translate this into a worst case lower bound for any randomized algorithm. In particular, this means that any algorithm that is based on recovering the clusters to construct the MAGs will require $\Omega(1/\alpha)$ interventions for every entity in $[M]$.

Details. For the lower bound, consider the case when $M = 2$, and assuming causal sufficiency, where we wish to identify the clusters of two MAGs $\mathcal{M}_1, \mathcal{M}_2$. We observe that a lower bound on the number of interventions required for every entity in the case of identifying two clusters will also extend for the general case of identifying k clusters with latents.

Consider two MAGs $\mathcal{M}_1, \mathcal{M}_2$ on a node set V, with the promise that either $d(\mathcal{M}_1, \mathcal{M}_2) = 0$ or $d(\mathcal{M}_1, \mathcal{M}_2) = \alpha n$, and the goal is to identify which case holds. Note that in the first case the two entities are in the same cluster ($k = 1$), and in the second case they are in different clusters ($k = 2$).

Let $V = \{v_1, \ldots, v_n\}$ be the set of observable nodes of these MAGs. Consider the node difference set of the MAGs $\mathcal{M}_1, \mathcal{M}_2$ given by $\text{diff}(\mathcal{M}_1, \mathcal{M}_2)$ and let $e \in \{0, 1\}^n$ denote its characteristic vector where lth coordinate of e is 1 iff $v_l \in \text{diff}(\mathcal{M}_1, \mathcal{M}_2)$. We can observe that, under the above promise, e is either 0^n or has exactly αn ones. Therefore, we have reduced our problem to that of finding whether the vector e contains all zeros or not. Using this reduction, we focus on establishing a lower bound for this modified problem.

We want to check if a given n-dimensional binary vector is a zero vector, i.e., 0^n or not, with a promise that if it is not a zero vector, then, it contains αn coordinates with 1 in them. Using Lemma D.12, we show that $\Omega\left(\frac{1}{\alpha}\right)$ queries to co-ordinates of x are required, for any randomized or deterministic algorithm to distinguish between these two cases.

Lemma D.12. *Suppose we are given a vector $x \in \{0, 1\}^n$ with the promise that either $x = 0^n$ or x contains αn ones. In order to distinguish these two cases with probability more than $2/3$, every randomized or deterministic algorithm must make at least $\Omega(1/\alpha)$ queries to the coordinates of the vector x.*

Proof. It is easy to see that every deterministic algorithm for this problem requires $(1-\alpha)n+1$ queries. For obtaining a lower bound on the number of queries of any randomized algorithm, we use Yao's minimax theorem Yao [1977]. To do so, we construct an input distribution μ on $\{0,1\}^n$ and show that every deterministic algorithm on the worst case requires at least q queries while succeeding with a probability $2/3$. From Yao's minimax theorem (Thm D.11), this implies that every randomized algorithm requires at least q queries to output the correct answer with probability of success $2/3$. We construct μ by using a probability of $1/2$ for 0^n vector and a probability of $1/2$ equally distributed among all vectors in $\{0,1\}^n$ containing exactly αn ones.

Suppose a deterministic algorithm (denoted by ALG) is used to identify whether $x = 0^n$ or not. Let $\mathcal{E}(x)$ denote the event that the ALG answers correctly on $x \in \{0,1\}^n$, $Q(x)$ denote the set of queries used by ALG such that $|Q(x)| = q$ and $L(x)$ denote the coordinates of x that are non-zero.

Consider the event $\mathcal{E}(x)$ when ALG answers correctly. We can write it as :

$$\begin{aligned}&\Pr_{x\sim\mu}[\mathcal{E}(x)]\\&= \Pr[\mathcal{E}(x) \mid Q(x)\cap L(x)\neq\phi]\Pr[Q(x)\cap L(x)\neq\phi] + \Pr[\mathcal{E}(x)\mid Q(x)\cap L(x)=\phi]\Pr[Q(x)\cap L(x)=\phi].\end{aligned}$$

We calculate the probability that the coordinates queried are not part of the non-zero coordinates of x, given by $Q(x)\cap L(x)=\phi$:

$$\begin{aligned}&\Pr_{x\sim\mu}[Q(x)\cap L(x)=\phi]\\&= \Pr[Q(x)\cap L(x)=\phi \mid x=0^n]\Pr[x=0^n] + \Pr[Q(x)\cap L(x)=\phi\mid x\neq 0^n]\Pr[x\neq 0^n]\\&= \frac{1}{2}\left(\Pr[Q(x)\cap L(x)=\phi\mid x=0^n] + \Pr[Q(x)\cap L(x)=\phi\mid x\neq 0^n]\right)\\&= \frac{1}{2}\left(1+\frac{\binom{n-q}{\alpha n}}{\binom{n}{\alpha n}}\right) = \frac{1+\tau}{2} \quad \text{, where } \tau = \frac{\binom{n-q}{\alpha n}}{\binom{n}{\alpha n}}.\end{aligned}$$

Now, we calculate the probability that ALG answers correctly when the queries $Q(x)$ all return zero. We upper bound this probability by considering the case when ALG answers 'yes', and the case when ALG answers 'no' separately. It is easy to observe that $\mathcal{E}(x)$ is correct when ALG ='yes' iff $x = 0^n$. Therefore, we have:

$$\begin{aligned}\Pr[\mathcal{E}(x)\mid Q(x)\cap L(x)=\phi] &\leq \max\left\{\overbrace{\frac{\Pr[\mathcal{E}(x),Q(x)\cap L(x)=\phi]}{\Pr[Q(x)\cap L(x)=\phi]}}^{\text{ALG answers 'yes'}}, \overbrace{\frac{\Pr[\mathcal{E}(x),Q(x)\cap L(x)=\phi]}{\Pr[Q(x)\cap L(x)=\phi]}}^{\text{ALG answers 'no'}}\right\}\\&\leq \max\left\{\frac{1/2}{(1+\tau)/2}, \frac{\tau/2}{(1+\tau)/2}\right\} \leq \frac{1}{1+\tau}.\end{aligned}$$

$$\begin{aligned}\Pr_{x\sim\mu}[\mathcal{E}(x)] &= \Pr[\mathcal{E}(x)\mid Q(x)\cap L(x)\neq\phi]\Pr[Q(x)\cap L(x)\neq\phi] + \Pr[\mathcal{E}(x)\mid Q(x)\cap L(x)=\phi]\Pr[Q(x)\cap L(x)=\phi]\\&\leq \Pr[Q(x)\cap L(x)\neq\phi] + \Pr[\mathcal{E}(x)\mid Q(x)\cap L(x)=\phi]\Pr[Q(x)\cap L(x)=\phi]\\&\leq \left(1-\frac{1+\tau}{2}\right) + \frac{1}{1+\tau}\frac{1+\tau}{2} = 1-\frac{\tau}{2}.\end{aligned}$$

We know the probability of success for ALG is at least $\frac{2}{3}$. Therefore, we have $\Pr_{x\sim\mu}[\mathcal{E}(x)] \geq \frac{2}{3}$, which implies $\tau \leq \frac{2}{3}$.

Let $H(x)$ denote the binary entropy function. Using the bound from (MacWilliams and Sloane [1977], Page 309)

$$\sqrt{\frac{a}{8b(a-b)}}2^{aH(b/a)} \leq \binom{a}{b} \leq \sqrt{\frac{a}{2\pi b(a-b)}}2^{aH(b/a)}.$$

We have

$$\tau = \frac{\binom{n-q}{\alpha n}}{\binom{n}{\alpha n}} \leq \sqrt{\frac{8(n-q)(n-\alpha n)}{2\pi n(n-q-\alpha n)}} 2^{(n-q)H(\frac{\alpha n}{n-q})-nH(\alpha)}$$
$$= \sqrt{\frac{4}{\pi}\left(1+\frac{q\alpha}{n-q-\alpha n}\right)} 2^{(n-q)(H(\frac{\alpha n}{n-q})-H(\alpha))-qH(\alpha)}.$$

We observe that $q \leq (1-\alpha)n+1$ for any algorithm, as we can identify whether $x = 0^n$ or not trivially by querying more than $(1-\alpha)n+1$ coordinates. Therefore,

$$\tau \leq \sqrt{\frac{4}{\pi}(1-q\alpha)} 2^{(n-q)(H(\frac{\alpha n}{n-q})-H(\alpha))-qH(\alpha)}$$
$$\leq \sqrt{\frac{4}{\pi}} 2^{-q\alpha\log e/2+(n-q)(H(\frac{\alpha n}{n-q})-H(\alpha))-qH(\alpha)}$$

Using $\frac{\alpha n}{n-q} \geq \alpha$ and mean-value theorem, we have:

$$(n-q)\left(H\left(\frac{\alpha n}{n-q}\right)-H(\alpha)\right) \leq q\alpha H'\left(\frac{\alpha n}{n-q}\right)$$
$$= q\alpha\log\left(\frac{n-q}{\alpha n}-1\right)$$
$$\leq q\alpha\log(1-\alpha) - q\alpha\log\alpha.$$

Substituting the above expression and expanding $H(\alpha)$, we have :

$$\tau \leq \sqrt{\frac{4}{\pi}} 2^{-q\alpha\log e/2+q\alpha\log(1-\alpha)-q\alpha\log\alpha+q\alpha\log\alpha+(1-\alpha)q\log(1-\alpha)}$$
$$\leq \sqrt{\frac{4}{\pi}} 2^{-q\alpha\log e/2+q\log(1-\alpha)}$$
$$\leq \sqrt{\frac{4}{\pi}} 2^{-q\alpha\log e/2-q\alpha} \leq \frac{2}{3}.$$

Therefore, for ALG to succeed with probability at least $2/3$, we have

$$q \geq \Omega\left(\frac{1}{\alpha}\right).$$

Using this with Yao's minimax theorem (Thm D.11), we get that with every randomized algorithm needs $\Omega(1/\alpha)$ queries to succeed on this problem with probability at least $2/3$. □

For the above problem of identifying whether a vector is zero or not, we can replace each coordinate query by an intervention on the corresponding node for the two entities (due to the equivalency between the two as explained above). Therefore, from Lemma D.12, we have the following corollary about recovering the clusters.

Corollary D.13. *Suppose we are given two MAGs $\mathcal{M}_1$ and $\mathcal{M}_2$ corresponding to two entities, with the promise that either $d(\mathcal{M}_1, \mathcal{M}_2) = 0$ or $d(\mathcal{M}_1, \mathcal{M}_2) = \alpha n$. In order to distinguish these two cases with probability at least $2/3$, every (randomized or deterministic) algorithm must make at least $\Omega(1/\alpha)$ interventions on both the entities.*

Theorem D.14 (Theorem 4.2 Restated)**.** *Suppose the underlying MAGs $\mathcal{M}_1, \ldots, \mathcal{M}_M$ satisfy α-clustering property. In order to recover the clusters with probability $2/3$, every (randomized or deterministic) algorithm requires $\Omega(1/\alpha)$ interventions for every entity in $[M]$.*

Proof. From Corollary D.13, we have that to identify whether two MAGs belong to the same cluster or not, we have to make at least $\Omega(1/\alpha)$ interventions for every entity. Therefore, to recover all the clusters, we have to make at least $\Omega(1/\alpha)$ many interventions for every entity $i \in [M]$. □

Note. The lower bound for the number of interventions per entity is for the first step of identifying the underlying clustering. Similar to our upper bounds, our lower bound considers the worst-case, where the MAGs satisfy the α-clustering property are all Markov equivalent. This implies that the PAGs obtained using FCI are identical and will not be helpful in identifying the clusters. In practice, the information available in the PAGs could be useful to reduce the number of interventions.

E Experimental Evaluation

In this section, we provide additional details about the experimental evaluation discussed in Section 5. For ensuring reproducibility, the entire experiment setup is submitted as part of the supplementary material.

E.1 Learning MAGs under (α, β)-clustering property

(Synthetic) Data Generation. We use following process for each of the five considered causal network (*Asia*, *Earthquake*, *Sachs*, *Survey*, and *Erdős-Renyi*). We construct causal MAGs for M entities distributed among the clusters $C_1^\star, C_2^\star, \cdots C_k^\star$ equally, i.e., $|C_i^\star| = M/k$ for all $i \in [k]$. In our experiments, we set $k = 2$ (i.e., two clusters), and start with $k = 2$ DAGs that are sufficiently far apart. To do so, we create two copies of the original causal network $\mathcal{D}$, and denote the DAG copies by $\mathcal{D}_1$ and $\mathcal{D}_2$. For each of the DAGs $\mathcal{D}_1$ and $\mathcal{D}_2$, we select a certain number of pairs of nodes randomly, and include a latent variable between them, that has a causal edge to both the nodes. In our experiments, we used 2 latents per DAG. This results in two new DAGs $\mathcal{D}_1'$ and $\mathcal{D}_2'$. To ensure αn node distance between clusters, we modify $\mathcal{D}_2'$ using random changes until the two MAGs corresponding to the DAGs $\mathcal{D}_1'$ and $\mathcal{D}_2'$ are separated by a distance of αn. These two MAGs, denoted by $\mathcal{M}_1^{\text{dom}}$ and $\mathcal{M}_2^{\text{dom}}$ form the dominant MAG for each of the two clusters.

Then, we create $(1-\gamma)M/k = (1-\gamma)M/2$ copies of the dominant MAG and assign it to distinct entities in each cluster. Consider cluster $C_1^\star$ with dominant MAG $\mathcal{M}_1^{\text{dom}}$, and corresponding DAG $\mathcal{D}_1'$. Note that each cluster has $M/k = M/2$ entities. For the remaining entities in $C_1^\star$, we start with $\mathcal{D}_1$ and include 2 latent variables between randomly selected pairs of nodes. Then, we repeat the previous procedure, of performing a series of random insertions or deletions of edges to the DAG until the distance between the corresponding MAG and $\mathcal{M}_1^{\text{dom}}$ increases to βn. We follow the same procedure for cluster $C_2^\star$ with dominant MAG $\mathcal{M}_2^{\text{dom}}$. Note that in this construction different entities could differ both in latents and their observable graphs. This construction ensures the entities satisfy (α, β)-clustering property. As an example, see Figure 7 containing two dominant MAGs of the Causal Network *Earthquake*.

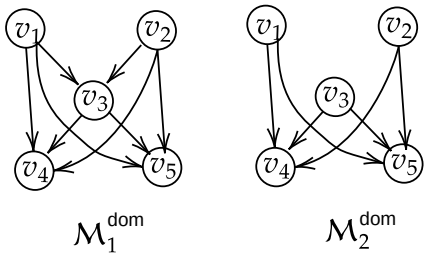


Figure 7: Dominant MAGs of the causal network *Earthquake* constructed using the described procedure.

Sample Set S Size. For Algorithm (α, β)-BOUNDEDDEGREE, we use different sample sizes S ranging from 1 to 3. In Figure 8, we plot the mean value of the maximum number of interventions per entity with change in sample set size.

With increase in sample set size, our Algorithm (α, β)-BOUNDEDDEGREE requires more interventions (see Lemma C.1) and we observe the same in Figure 8. We chose the smallest size $|S| = 1$ in our experiments, as increasing the size will increase the number of interventions but did not lead to much improved clustering results. As a sample set of size 1 roughly corresponds to around 3 interventions (across all causal networks), we use that for results presented in Table 1.

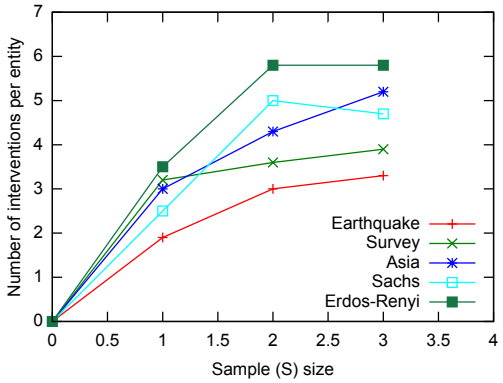


Figure 8: Sample size vs. maximum number of interventions per entity used by Algorithm (α, β)-BOUNDEDDEGREE.

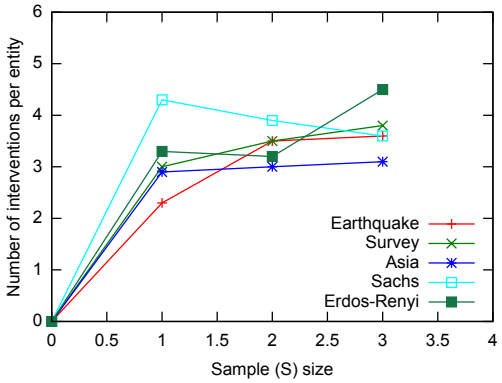


Figure 9: Sample size vs. maximum number of interventions per entity used by Algorithm α-BOUNDEDDEGREE.

Causal Network	FCI Precision	FCI Recall	FCI Accuracy	α-BOUNDEDDEGREE (Alg. 6) Precision	α-BOUNDEDDEGREE (Alg. 6) Recall	α-BOUNDEDDEGREE (Alg. 6) Accuracy	Maximum # Interventions
Earthquake	0.79 ± 0.25	0.98 ± 0.02	0.79 ± 0.25	1 ± 0.00	1.0 ± 0.0	1.00 ± 0.00	3
Survey	0.79 ± 0.25	0.98 ± 0.02	0.79 ± 0.25	0.89 ± 0.20	1.0 ± 0.00	0.89 ± 0.20	4
Asia	0.84 ± 0.23	0.98 ± 0.02	0.84 ± 0.23	0.89 ± 0.20	1.0 ± 0.00	0.89 ± 0.20	4
Sachs	1.0 ± 0.00	1.0 ± 0.00	1.0 ± 0.00	0.79 ± 0.25	1.0 ± 0.00	0.79 ± 0.25	5
Erdös-Rényi	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	5

Table 2: In this table, we present the precision, recall and accuracy values obtained by Algorithm α-BOUNDEDDEGREE and FCI. Each cell includes the mean value along with the standard deviation computed over 10 runs. The last column contains the maximum number of interventions per entity required (including both Algorithm α-BOUNDEDDEGREE and the Meta-algorithm) for recovering the DAGs.

Construction of Clusters from FCI Output. Our first focus is on recovering the true clustering using Algorithm (α, β)-BOUNDEDDEGREE. As a baseline, we employ the well-studied FCI algorithm [Spirtes et al., 2000]. We know that FCI returns the Partial Ancestral Graph(PAG) corresponding to the causal MAG using only the observational data. After recovering the PAGs corresponding to the MAGs using FCI, we cluster them by constructing a weighted graph (similar to Algorithm (α, β)-BOUNDEDDEGREE) defined on the set of entities. For every pair of entities i, j, we calculate the number of nodes n_{ij} that share the same neighborhood using the PAGs associated with them, and assign the weight of the edge as n_{ij}. This weight captures the similarity between two entities, and whether they belong to the same cluster or not. Now, we use minimum-k-cut algorithm to partition the set of entities into k components or clusters. In Algorithm (α, β)-BOUNDEDDEGREE, we first construct a sample S, and perform various interventions based on the set S for every entity to finally obtain the k clusters.

Setup. We used a personal Apple Macbook Pro laptop with 16GB RAM and Intel i5 processor for conducting all our experiments. We use the FCI algorithm implemented in [Kalisch et al., 2012]. For every causal network, each experiment took less than 10 minutes to finish all the 10 runs.

E.2 Learning MAGs under α-clustering property

(Synthetic) Data Generation. We use following process for each of the five considered causal network (*Asia*, *Earthquake*, *Sachs*, *Survey*, and *Erdős-Renyi*). We construct causal DAGs for M entities distributed among the clusters $C_1^\star, C_2^\star, \cdots C_k^\star$ equally, i.e., $|C_i^\star| = M/k$ for all $i \in [k]$. Again we set $k = 2$. For each of the DAGs $\mathcal{D}_1$ and $\mathcal{D}_2$, we select a certain number of pairs of nodes randomly, and include a latent variable between them, that has a causal edge to both the nodes. In our experiments, we used 2 latents per DAG. This results in two new DAGs $\mathcal{D}_1'$ and $\mathcal{D}_2'$. To ensure αn node distance between clusters, we modify $\mathcal{D}_2'$ using random changes until the two DAGs $\mathcal{D}_1'$ and $\mathcal{D}_2'$ are separated by a distance of αn and are Markov equivalent. Without Markov equivalence,

we observe that FCI always recovers the underlying clusters correctly in the α-clustering case.[4] However, existence of Markov equivalent DAGs is a well-known problem in real-world graphs, a popular example to illustrate this comes the " breathing dysfunction" causal graph in Fig. 3 in [Zhang, 2008b]. We create $M/2$ copies of the each of the two DAGs $\mathcal{D}'_1$ and $\mathcal{D}'_2$ and assign it to distinct entities in each of the two clusters.

Parameters. We present the following settings for the model parameters, α is at least 0.60, $M = 40$. For the synthetic data generated using Erdös-Rényi model, we use $n = 10$, probability of edge $p = 0.30$. We ran all of our experiments for 10 times with the stated values and report the results.

Sample Set S Size. For Algorithm α-BOUNDEDDEGREE, we again tried different set S sizes ranging from 1 to 3. In Figure 9, we plot the mean value of the maximum number of interventions per entity with increase in sample size. It has a same trend as with (α, β)-clustering (Figure 9). A sample size of 1 roughly corresponds to around 3 interventions, and we use that for the results presented in Table 2.

Evaluation of Clustering. We start by results on recovering the clustering using Algorithm α-BOUNDEDDEGREE. As a baseline, we again employ the well-studied FCI algorithm [Spirtes et al., 2000]. After recovering the PAGs corresponding to the DAGs using FCI, we cluster them by constructing a similarity graph (similar to the case of (α, β)-clustering discussed previously) defined on the set of entities. For Algorithm α-BOUNDEDDEGREE, we first construct a sample S, and perform various interventions based on the set S for every entity to finally obtain the k clusters. We also implemented another baseline algorithm (GREEDY) that uses interventions, based on a greedy idea that selects nodes to set S in Algorithm α-BOUNDEDDEGREE by considering nodes in increasing order of their degree in the PAGs returned by FCI. We use this ordering to minimize the number of interventions as we intervene on every node in S and their neighbors. We use the same metrics as the (α, β)-clustering case.

Results. In Table 2, we compare Algorithm α-BOUNDEDDEGREE to FCI on the clustering results. For Algorithm α-BOUNDEDDEGREE, we use a sample S of size 1, and observe in Figure 9, that this corresponds to about 2 interventions per entity. With increase in sample size, we observed that the results were either comparable or better. We observe that our approach leads to considerably better performance in terms of the accuracy metric with an average difference in mean accuracy of about 0.20. We observe that entities belonging to the same true cluster are always assigned to the same cluster, resulting in high recall for both Algorithm α-BOUNDEDDEGREE and FCI. Further, the higher value of precision for our algorithm is because FCI is unable to correctly detect that there are two clusters, as the DAGs are Markov Equivalent which means that they result in the same PAGs.

Algorithm α-BOUNDEDDEGREE outperforms the GREEDY baseline for the same sample (S) size. For example, on the *Earthquake* and *Survey* causal networks, Algorithm α-BOUNDEDDEGREE obtains the mean accuracy values of 1.0 and 0.89 respectively, while GREEDY for the same number of interventions obtained an accuracy of only 0.74 and 0.64 respectively. On the remaining causal networks, the accuracy values of GREEDY are almost comparable to our Algorithm α-BOUNDEDDEGREE.

After clustering, we recover the DAGs using the Meta-algorithm described in Section D.1, and observe that only one additional intervention is needed. In the last column in Table 2, we report the maximum number of interventions for recovering DAGs, which includes both the interventions used by the Algorithm α-BOUNDEDDEGREE and the Meta-algorithm. We observe that our *collaborative* approach uses fewer interventions for MAG recovery compared to the number of nodes in each causal network. For example, in the Erdös-Rényi setup, the number of nodes $n = 10$, whereas we use at most 5 interventions per entity. Thus, compared to the worst-case, cutting the number of interventions for each entity by 50%.

[4] Again this is not true for (α, β)-clustering, as shown by our experiments results for that case, because now difference in PAGs between two entities does not automatically imply that those two entities must belong to different clusters.