Learning Coarsened Causal DAG Models

Alex Markham*[1] **Pratik Misra**[2]

[1]Department of Mathematical Sciences, University of Copenhagen
[2]Department of Mathematics, Technical University of Munich

Abstract

Graphical models are a powerful tool for representing causal relationships among jointly distributed random variables, especially concerning data from across different experimental settings. However, it is not always practical or desirable to estimate a causal model at the granularity of features in a particular dataset. There is a growing body of research on *causal abstraction* to address such problems. We provide graphical identifiability results and propose an algorithm for directly and efficiently learning abstract causal graphs from data, as well as theoretical insights about the lattice structure of this search space. As proof of concept, we apply our algorithm to synthetic data as well as a real dataset containing measurements from protein-signaling networks. *This is a work-in-progress.*

1 INTRODUCTION

Discovering causal relationships is one of the fundamental goals of scientific research. Nevertheless, the causal relationships of interest are not always between features of a given dataset. For example, in neuroscience, one may have data describing the interactions of groups (or even individual) neurons and wish to learn from this data a causal model over cognitive or behavioral states [Grosse-Wentrup et al., 2024].

This is a very general and difficult problem, so we restrict our interest here to particular cases where there is data from some assumed low-level or fine-grained causal DAG model, and we offer a formalization for what it means for a DAG with fewer nodes to be a high-level or coarse-grained abstraction of this underlying causal DAG model. Our approach is based on the intuition that variables with similar causes and similar effects (graphically, similar ancestors and descendants) are in some sense redundant and can be abstracted away by clustering them together while retaining only the important causal relationships.

There is a growing number of similarly motivated works on consistent transformations of causal models [Rubenstein et al., 2017] and causal abstractions [Beckers and Halpern, 2019, Beckers et al., 2020, Beckers, 2021, Otsuka and Saigo, 2022] that lay a ground work for formally describing such problems. There is also some work on learning abstractions [Massidda et al., 2024], on directly learning a reduced DAG according to its marginal independences [Deligeorgaki et al., 2023], on using partitions to simplify learning large graphs [Gu and Zhou, 2020], and on clustering nodes in a DAG while preserving the identifiability of specified causal estimands [Tikka et al., 2023]. We hope to add to this growing body of research by taking a unique graphical perspective on the problem, which naturally lends itself to a structured description of the entire space of valid graphical abstractions as well as algorithms for directly learning such abstractions from data.

Contributions We list our contributions as follows:

- formalization turning search space of DAG models into search space of causal abstractions
- efficient search of this space
- nonparametric method, flexible learning framework for specific downstream tasks
- bridging fields, working at intersection of causal machine learning and combinatorics/algebraic statistics.

We also list current shortcomings and areas of ongoing work:

- the proposed algorithm is constraint-based, and thus prone to compounding sequential errors
- the statistical tests used show poor empirical performance, even on simple datasets, likely due to low statistical power

*Correspondence: alex.markham@causal.dev

Accepted for the 1st Workshop on Causal Abstractions and Representations (CAR) at UAI 2025 (CAR 2025).

- the theoretical results assume causal sufficiency, and it is unclear how badly they break down when this assumption is violated.

The paper proceeds as follows: Section 2 formalizes the problem of causal abstraction in terms of graphical structure (which we call coarsening) and outlines our general approach to thinking about the resulting search space; based on these insights, Section 3 presents a flexible oracle-based learning algorithm with promising theoretical properties—replacing the oracles with standard statistical tests results in a practical algorithm that is consistent in the large-sample limit. We implement the practical algorithm and apply it to synthetic and real data in Section 4. We conclude in Section 5 with a discussion of current shortcomings as well as plans for ongoing and future work.

2 COARSENING CAUSAL DAGS

We begin by introducing some general notation and preliminaries that are commonly used in the graphical models literature. A directed acyclic graph (DAG) is denoted by $G = (V, E)$, where V is the vertex set indexed by $\{1, 2, \ldots, n\}$ and E is a set of directed edges which do not form a directed cycle. In a DAG G, $\mathrm{pa}_G(A), \mathrm{ch}_G(A), \mathrm{an}_G(A)$ and $\mathrm{de}_G(A)$ respectively denote the parents, children, ancestors and descendants of the vertex set $A \subset V$.

The graphical model $\mathcal{M}_G$ corresponding to a DAG G is the collection of probability distributions that factorize according to the DAG. The model $\mathcal{M}_G$ can also be specified by the conditional independence statements implied by G. These independences correspond to the Markov properties associated to the DAG. Thus, different DAGs can specify the same model if they have the same set of conditional independence statements. This causes the key problem of structural identifiability, where one aims to infer the true DAG from observational data. On the other hand, when the number of variables increases, the graphical approach to analyzing the model may become impractical due to the overwhelming number of conditional independence statements. A natural way to reduce the size of the DAG and still preserve some of its essential properties is by obtaining a coarsened DAG by combining the vertices of the original DAG.

The concept of *coarsening* is obtained by considering partitions of different sizes of the vertex set of a DAG G. Each partition is then considered as a new vertex to obtain a coarsened DAG G' of G. Different types of coarsening can be obtained depending on how the the vertex set is partitioned. We will first define the most general type of coarsening, where there is no constraints on the partition.

2.1 GENERAL COARSENING

Definition 1. *Given a DAG $G = (V, E)$, we define a* graphical coarsening *to be any DAG $G' = (V', E')$ (with $|V'| < |V|$) for which there exists a surjection $\pi : V \to V'$ that is monotonic with respect to the partial orders induced by G and G', i.e., for which $u \leq_G v \implies \pi(u) \leq_{G'} \pi(v)$.*

Example 2. *Let $G = (V, E)$ be the DAG $1 \to 2 \to 3 \to 4$, and V' be the coarsening $\{1, 2, \{3, 4\}\}$. Here, the surjective map π maps $1 \mapsto 1, 2 \mapsto 2, 3 \mapsto \{3, 4\}, 4 \mapsto \{3, 4\}$. This gives us G' as the DAG $1 \to 2 \to \{3, 4\}$. Observe that π is monotonic with respect to the partial orders induced by G and G' as $1 \leq_G 2 \leq_G 3 \leq_G 4$ implies $1 \leq_{G'} 2 \leq_{G'} \{3, 4\}$. Similarly, if we consider V'' to be the coarsening $\{1, 3, \{2, 4\}\}$, then although there is a surjective map between V and V'', there is no DAG $G'' = (V'', E'')$ for which the map is monotonic with respect to the partial orders of G and G''.*

Note: For each DAG G, every valid coarsening corresponds to a unique DAG G', where $A \to B$ is an edge in G' only if:

- there exists some $a \in A$ and $b \in B$ such that $a \to b$ is an edge in G and,
- there exists no edge in G of the form $b \to a$ for any $b \in B$ and $a \in A$.

The motivation behind coarsening causal DAGs is that the coarsened DAGs do not introduce any new conditional independence statements, which is proven in the lemma below. Thus, one could potentially identify the *important* conditional independence statements and implement a causal discovery algorithm on these statements to obtain a coarsened CPDAG of the true DAG.

Lemma 3. *Let G be a DAG and G' be any coarsening of G. Then every conditional independence statement of G' is satisfied by G.*

Proof. Let G be a DAG with $[n]$ vertices and G' be a coarsening of G with vertices $A_1, A_2, \ldots, A_m$. Now, let $A_i \perp\!\!\!\perp A_j | \{A_{k_1}, A_{k_2}, \ldots, A_{k_p}\}$ be a d-separation statement in G'. We need to show that for any $a_i \in A_i$ and $a_j \in A_j$, a_i is d-separated from a_j by $A_{k_1} \cup A_{k_2} \cup A_{k_p}$ in G. In order to prove this, we first construct an intermediate graph $\hat{G}$ by using the structure of G' in a specific way such that G is a subgraph of $\hat{G}$. As every d-separation statement of $\hat{G}$ would hold in G, it is enough to show that a_i is d-separated from a_j by $A_{k_1} \cup A_{k_2} \cup \ldots \cup A_{k_p}$ in $\hat{G}$.

We construct $\hat{G}$ in the following way: $\hat{G}$ is a graph on $[n]$ vertices. For every vertex set $A_1, A_2, \ldots, A_p$ in G', we add the appropriate edges in G (directed according to a linear extension of the partial order induced by G) such that each

A_i becomes a complete subgraph (clique). Now, if $A_1 \rightarrow A_2$ is an edge in G', we know by the Definition 1 that there exists some $a_1 \in A_1$ and $a_2 \in A_2$ such that $a_1 \rightarrow a_2$ is an edge in G. Further, it is also known that $a_2' \rightarrow a_1'$ is not an edge in G for any $a_2' \in A_2$ and $a_1' \in A_1$. So, for every edge $A_1 \rightarrow A_2$ in G', we complete the construction of $\hat{G}$ by adding the edges $a_1 \rightarrow a_2$ for all $a_1 \in A_1$ and $a_2 \in A_2$.

It is clear from the construction that $\hat{G}$ is obtained by only adding edges to G, implying that G is indeed a subgraph of $\hat{G}$. We now focus on the conditional independence statements obtained from the directed local Markov property in G', i.e., $A_i \perp\!\!\!\perp A_j | \mathrm{pa}_{G'}(A_j)$ where A_i and A_j are non adjacent vertices and A_j is not a descendant of A_i. For any $A_l \in \mathrm{pa}_{G'}(A_j)$, $a_l \rightarrow a_j$ is an edge in $\hat{G}$ for all $a_l \in A_l$ and $a_j \in A_j$. This implies that $\{\cup A_l : A_l \in \mathrm{pa}_{G'}(A_j)\} \subseteq \mathrm{pa}_{\hat{G}}(a_j)$. Similarly, for every $a_l \in \mathrm{pa}_{\hat{G}}(a_j)$ and for any $a_j \in A_j$, either $a_l \in A_j$ or there exists some $A_l \in \mathrm{pa}_{G'}(A_j)$ such that $a_l \in A_l$. As A_j is a clique in $\hat{G}$, we pick the source node in A_j, say a_s. This gives us that every parent of a_s in $\hat{G}$ lies in some $A_l \in \mathrm{pa}_{G'}(A_j)$, implying that $a_i \perp\!\!\!\perp a_s | \{\cup A_l : A_l \in \mathrm{pa}_{G'}(A_j)\}$ for all $a_i \in A_i$.

We now pick the vertex a_w in A_j which comes second in the ordering, i.e., $\mathrm{pa}_{\hat{G}}(a_w) \cap A_j = \{a_s\}$. From here, we get the directed local Markov property that $a_i \perp\!\!\!\perp a_w | \{a_s \cup A_l : A_l \in \mathrm{pa}_{G'}(A_j)\}$ for all $a_i \in A_i$. However, using the contraction and decomposition axioms [Pearl, 1988, Chapter 3, Theorem 1] on the two statements gives us that $a_i \perp\!\!\!\perp a_w | \{\cup A_l : A_l \in \mathrm{pa}_{G'}(A_j)\}$ for all $a_i \in A_i$. We continue this process by using the ordering of the vertices in A_j to get the remaining conditional independence statements. □

2.2 PARTITION LATTICE OF GENERAL COARSENING

For a given DAG G on n vertices, every valid coarsening can be seen as a non trivial partition of $[n]$. Furthermore, the coarsening of G into a DAG with a single vertex (i.e., all the vertices clustered together) is also a valid coarsening of G. Thus, it makes sense to view every valid coarsening of G as a point on the *partition lattice* of $[n]$. A lattice is a partially ordered set where any two elements $\{a, b\}$ have a unique *meet* (greatest lower bound denoted by $a \wedge b$) and *join* (least upper bound denoted by $a \vee b$). Example of the entire partition lattice of 4 elements can be seen in the Appendix.

As each valid coarsening of G corresponds to a point in the partition lattice of n elements, they form a poset in the partition lattice with the topmost element $\{1, 2, \ldots, n\}$ (G' is a single vertex) and the lowermost element $\{1\}, \{2\}, \ldots, \{n\}$ (G' equal to G). We call this poset the *coarsening poset* of G. Thus, moving down in this poset corresponds to a partition refinement and addition of certain CI statements obtained from the refinement. In the Theorem below, we show that the coarsening poset of a DAG is indeed a sublattice of the partition refinement lattice. We use this sublattice structure to develop a oracle-based algorithm in Section 3, for which we discuss practical implementations and apply in Section 4.

Theorem 4. *For any DAG $G = (V, E)$, the poset of all the valid coarsenings of G is a lattice, and more specifically a sublattice of the partition refinement lattice of $[n]$ (where $n = |V|$).*

Proof. Let G' and G'' be two valid coarsening of G, with vertex sets $A = \{A_1, A_2, \ldots, A_k\}$ and $B = \{B_1, B_2, \ldots, B_m\}$. In other words, let A and B be two valid partitions of $[n]$. In order to conclude that the poset is a sublattice of the partition refinement lattice, we need to show that both $A \wedge B$ and $A \vee B$ are valid coarsenings of G. From the partition refinement lattice, we know that $A \wedge B = \{A_i \cap B_j : i = 1, 2, \ldots, k, j = 1, 2, \ldots, m\}$. Now, let $a \in A_i \cap B_j$ and $b \in A_{i'} \cap B_{j'}$ such that $a \rightarrow b \in G$. Note that i can be equal to i' or j can be equal to j' but both cannot be equal simultaneously. If there also exists some $a_1 \in A_i \cap B_j$ and $b_1 \in A_{i'} \cap B_{j'}$ such that $b_1 \rightarrow a_1 \in G$, then that would imply either A or B (or both) is not a valid partition of G. This is because if $i \neq i'$ (similarly if $j \neq j'$), then we would have $a, a_1 \in A_i$ and $b, b_1 \in A_{i'}$ with both edges $a \rightarrow b$ and $b_1 \rightarrow a_1$ lying in G. Thus, $A \wedge B$ corresponds to a valid coarsening of G, implying that the poset is a finite meet-semilattice.

As the coarsening poset is a subposet of the partition refinement lattice, we know that the coarsening poset also has a unique upper bound. This allows us to use [Stanley, 2011, Proposition 3.3.1], which states that if a poset is a finite meet-semilattice that is bounded above, then it is a lattice which has a well defined join. Thus, we can conclude that the coarsening poset of any DAG is indeed a sublattice of the partition refinement lattice. □

Theorem 5. *An abstraction poset of a DAG $G = ([n], E)$ is a distributive lattice only if the DAG has a directed $(n-1)$-path.*

Proof. Suppose the longest directed path in G is $p = (v_1, v_2, \ldots, v_m)$ with $m < n$. Extend the partial order induced by G to a total order $t = (t_1 = v_1, t_2, \ldots, t_n = v_m)$. Consider a vertex $v^* = t_i \notin p$, and let $v^- = \max_{j<i}\{t_j \in p\}$ while $v^+ = \min_{j>i}\{t_j \in p\}$. Then $\{v, v'\}, \{v, v^*\}, \{v', v^*\}$ and $\{v, v', v^*\}$ are all valid partitions. This sublattice does not satisfy distributivity (it is just the partition refinement lattice on 3 nodes), completing this direction of the proof. □

Theorem 5 tells us that DAGs that are sparse (in the sense of not connecting all nodes by at least a single long path, which can perhaps be related more explicitly to average degree [Meyniel, 1973]) do not results in distributive abstraction lattices. This is a somewhat negative result, because distributive lattices have nice algebraic properties (e.g., a relation to Gröbner bases) that would allow use of computational algebraic tools.

2.3 MARGINAL COARSENING

In Section 2.1, we defined the general coarsening of a DAG, where we had no constraint on the type of coarsening. However, we can also define constraint-based coarsening where each coarsening has to preserve certain properties of the DAG. One such constraint is the *marginal coarsening*, which is defined as follows:

Definition 6. *Given a DAG $G = (V, E)$, we define a* marginal coarsening *to be any graphical coarsening $G' = (V', E')$ whose partition function π preserves the marginal d-separation statements of G via $u \perp v \in I(G) \implies \pi(u) \perp \pi(v) \in I(G')$.*

In particular, marginal coarsenings are a special type of general coarsening where every marginal independence statement of G is preserved. This has a similar motivation to general coarsening, where we are instead interested in learning only the marginal independencies of G, which is useful for obtaining potential targets to perform interventional experiments. For instance, equivalence classes can be constructed using using the marginal independencies and can also be used to implement causal discovery algorithms [Deligeorgaki et al. [2023]].

In the following lemma, we give a complete characterization of all the possible marginal coarsening of G.

Lemma 7. *Let G be a DAG and $i \perp\!\!\!\perp j$ in G. Then any partition refinement of $V(G)$ containing i is either singleton or can only contain those k for which $k \perp\!\!\!\perp j$.*

Proof. As $i \perp\!\!\!\perp j$ in G, there does not exist any trek between i and j in G. This means that every path between i and j is blocked by some collider. Now, let k be some vertex in G such that i and k are grouped together in a valid marginal coarsening (partition) G'. We name this vertex of G' as P_i. This implies that i and k are not marginally independent in G (as grouping i and k together would remove any independence statement between i and k in G'). So, there exists a trek $\mathcal{T}(i,k)$ between i and k. If we assume that k is not marginally independent of j in G, then there also exists a trek $\mathcal{T}(k,j)$ between k and j as well. Let $\mathcal{T}(i,k)_R$ and $\mathcal{T}(k,j)_L$ correspond to the right and left part of the treks $\mathcal{T}(i,k)$ and $\mathcal{T}(k,j)$, respectively. Thus, the only way $\mathcal{T}(i,k) \cup \mathcal{T}(k,j)$ does not form a trek between i and j is that the path $\mathcal{T}(i,k)_R \cup \mathcal{T}(k,j)_L$ has a collider at some vertex l in the intersection of $\mathcal{T}(i,k)_R$ and $\mathcal{T}(k,j)_L$ (Note that l can be equal to k as well). Without loss of generality we can assume that $\mathcal{T}(i,k)_R \cap \mathcal{T}(k,j)_L$ is equal to $l \to \ldots \to k$. Now, let $p \to l$ be an edge $\mathcal{T}(k,j)_L$ (Note that p can be equal to j as well). Then we know that $i \perp\!\!\!\perp p$ in G (else we can replace l with p and repeat the argument), and hence p cannot lie in P_i. If l lies in P_i, then the marginal independence $i \perp\!\!\!\perp p$ is lost in G' as any vertex containing p has to be adjacent to P_i in G'. Similarly, if l does not lie in P_i, then there must exist a directed path from P_l to P_i in G'. This also results in G' losing the marginal independence $i \perp\!\!\!\perp p$, which is a contradiction to the construction of G'. □

The marginal coarsening poset also forms a lattice which could be used to construct an analogous marginal version of Algorithm 1.

3 LEARNING COARSENED CAUSAL MODELS

3.1 ORACLE-BASED ALGORITHM

Algorithm 1 DAG learning by recursive partition refinement

```
function REPARE(partition-DAG $\mathcal{D} = (\{\pi_{[k]}\}, E)$)
    $\pi^*, \{\pi'_r, \pi'_s\} :=$ REFINE($\{\pi_{[k]}\}$) ▷ Pick a part to refine
    if $\pi^* = \emptyset$ then ▷ Stop recursion when no part to refine
        return
    $\{\pi'_{[k+1]}\} := \{\pi'_r, \pi'_s\} \cup \{\pi_{[k]}\} \setminus \pi^*$ ▷ Initialize nodes of new DAG
    $E' := E \cap \{\pi'_{[k+1]}\}^2$ ▷ Initialize edges of new DAG
    for $\pi' \in \{\pi'_r, \pi'_s\}$ do ▷ Update edges into/out of refined part
        for $\pi \in \mathrm{pa}_{\mathcal{D}}(\pi^*)$ do
            if ADJACENT($\pi, \pi'$) then
                append $\pi \to \pi'$ to $E'$
        for $\pi \in \mathrm{ch}_{\mathcal{D}}(\pi^*)$ do
            if ADJACENT($\pi, \pi'$) then
                append $\pi' \to \pi$ to $E'$
    if PARENT($\pi'_r, \pi'_s$) then ▷ Update edges within refined part
        append $\pi'_r \to \pi'_s$ to $E'$
    else if PARENT($\pi'_s, \pi'_r$) then
        append $\pi'_s \to \pi'_r$ to $E'$
    $\mathcal{D}' := (\{\pi'_{[k+1]}\}, E')$ ▷ New DAG based on partition refinement and updated edges
    return concatenation of $\mathcal{D}'$ and REPARE($\mathcal{D}'$)
```

Definition 8. *A* refine-oracle*, with respect to a ground truth DAG $\mathcal{D}^* := ([n], E^*)$, takes as input a partition $\{\pi_{[k]}\}$ such that $\exists \mathcal{D} := (\{\pi_{[k]}\}, E) \preceq \mathcal{D}^*$ (i.e., there's an order-reflecting map from $\mathcal{D}$ to $\mathcal{D}^*$), and outputs a partition refinement $\{\pi'_{[k']}\}$ such that $\mathcal{D} \preceq \exists \mathcal{D}' = (\{\pi'\}^{k'}, E') \preceq \mathcal{D}^*$. [note that necessarily $k < k'$—i.e., the refinement is nontrivial]*

Definition 9. *An* adjacent-oracle*, with respect to a ground truth DAG* $\mathcal{D}^* := ([n], E^*)$*, takes as input parts* $\pi_i, \pi_j \in \{\pi_{[k]}\}$ *such that* $\exists \mathcal{D} := (\{\pi_{[k]}\}, E) \preceq \mathcal{D}^*$ *and determines whether or not* π_i *and* π_j *are adjacent in* $\mathcal{D}^*$ *(i.e,* $\exists u \in \pi_i$ *and* $v \in \pi_j$ *such that* $u \to v$ *or* $v \to u$ *is in* $\mathcal{D}^*$*) and correspondingly outputs* `True` *or* `False`*.*

Definition 10. *A* parent-oracle*, with respect to a ground truth DAG* $\mathcal{D}^* := ([n], E^*)$*, takes as input parts* $\pi_i, \pi_j \in \{\pi_{[k]}\}$ *such that* $\exists \mathcal{D} := (\{\pi_{[k]}\}, E) \preceq \mathcal{D}^*$ *and determines whether or not* $\pi_i \to \pi_j$ *in* $\mathcal{D}^*$ *and correspondingly outputs* `True` *or* `False`*.*

Theorem 11. *Given partition-, adjacent-, and parent-oracles with respect to a ground truth DAG* $\mathcal{D}^*$ *on* n *nodes, Algorithm 1 correctly identifies* $\mathcal{D}^*$ *after* $n-1$ *recursive calls, when started on the trivial partition-DAG* $\mathcal{D}^{\{[n]\}} = (\{[n]\}, \emptyset)$*. More specifically,* $\mathtt{REPARE}(\mathcal{D}^{\{[n]\}}) = (\mathcal{D}^{\{\pi_{[2]}\}}, \mathcal{D}^{\{\pi'_{[3]}\}}, \ldots, \mathcal{D}^{\{\pi''_{[n]}\}} = \mathcal{D}^*)$*.*

proof sketch. The idea is to do induction over the lattice structure. The base case is the supremum of the lattice (a trivially valid coarsening for any DAG) and we show that, assuming the input is an element of the $k-n+1$th level of the lattice, then the output of nth iteration of the algorithm is an element of the $k-n$th level of the lattice. The refine-oracle relates the element of the lattice to the surjection in Definition 1, while the the adjacent- and parent-oracles relate ensure that the surjection is monotonic, as required. Hence, the output of the $n-1$th iteration of the algorithm must be the infimum of the lattice, i.e., $\mathcal{D}^*$.

□

3.2 PRACTICAL ALGORITHM

In order to apply Algorithm 1 to data, the functions `REFINE` and `ADJACENT` can use statistical tests. While there are a multitude of options for these tests, we focus here on a choice of two simple, widely applicable nonparametric tests: the two-sample Kolmogorv-Smirnov test for distribution equality and a mutual information-based independence test.

`REFINE` with Kolmogorv-Smirnov test. In place of the refine-oracle, we use KS tests between observational and interventional datasets. See [Hodges, 1958] for details and [Virtanen et al., 2020] for an implementation. The number of interventional settings and their targets dictates how refined of a coarsening is learned.

Algorithm 2 Practical alternative to the `REFINE` oracle.

```
1: function KS-REFINE($\{\pi_{[k]}\}$)
2:     use (observational) reference dataset $R$ and interventional datasets $\mathcal{I} = \{I^1, \ldots, I^n\}$
3:     define $\pi'_r = \{i \in \pi^* \mid \mathrm{KS}(R_i, I^*_i)$ accepts $H_0\}$
       and $\pi'_s = \{i \in \pi^* \mid \mathrm{KS}(R_i, I^*_i)$ rejects $H_0\}$
4:     pick a part $\pi^* \in \{\pi_{[k]}\}$ and intervention $I^* \in \mathcal{I}$
       such that $\pi'_r$ and $\pi'_s$ are non-empty, if they exist
5:     return $\pi^*, \{\pi'_r, \pi'_s\}$
```

`ADJACENT` with mutual information test. In place of the adjacent/parent-oracle, we use a mutual information test. See [Runge, 2018] for details and the `tigramite` Python package for an implementation.

Algorithm 3 Practical alternative to the `ADJACENT` oracle.

```
function MI-ADJACENT($\pi, \pi'$)
    use pooled data $D = R \cup I^1 \cup \ldots \cup I^n$
    return TRUE if $\mathrm{MI}(D_\pi, D'_\pi)$ rejects $H_0$ else
    FALSE
```

Note that interventional information (i.e., π'_r vs π'_s in Line 3 of Algorithm 2) induces a causal order, so `MI-ADJACENT` can also be used to replace the parent-oracle.

Asymptotic consistency.

Corollary 12. *When using* `IKS-REFINE` *and* `MI-ADJACENT`*, Algorithm 1 learns the correct coarsening in the large sample limit.*

4 EXPERIMENTAL RESULTS

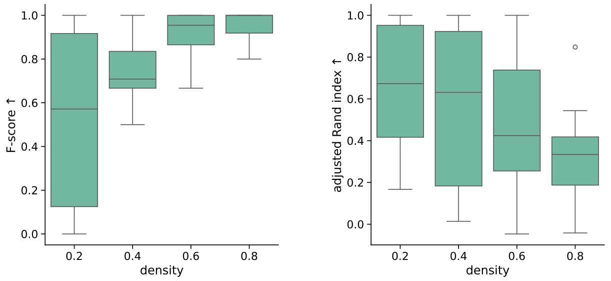
As proof of concept, we apply `RePaRe` to synthetic data sampled from random DAG models as well as to a real-world interventional dataset.[1]

4.1 SYNTHETIC DATA

We use the `sempler` package [Gamella et al., 2022] to generate random DAG models with varying densities and to sample datasets from these models across observational and interventional settings. The models are over 10 nodes and are linear with additive Gaussian noise, and the data is standardized before learning.

We evaluate the coarsening learned by `RePaRe` using two metrics: F-score[Manning, 2009, Eq. (8.6)] and the adjusted

[1] An open source implementation of the algorithm as well as scripts for reproducing all of the following experiments can be found at https://codeberg.org/alex-markham/repare.



(a) F-score reported for evaluating edge recovery of `MI-ADJACENT`.

(b) Adjusted Rand index reported for evaluating node partitioning of `IKS-REFINE`.

Figure 1: Evaluation on synthetic data, averaged over 10 seeds, as ground truth graph density varies.

Rand index (ARI) [Hubert and Arabie, 1985, Eq. (5)]. Considering the consistency of our method, as established in Theorem 11 and Corollary 12, we are interested here in determining performance on finite samples. In particular, we are interested in how overall performance is affected by the practical choices of statistical tests (Section 3.2): F-score gives a sense of `MI-ADJACENT`'s performance on edge recovery, while ARI measures performance of `IKS-REFINE` on partitioning.

The results are shown in Figure 1, indicating decent edge recovery that increases with density (Figure 1a) but highly variable partitioning performance that decreases with density (Figure 1b).

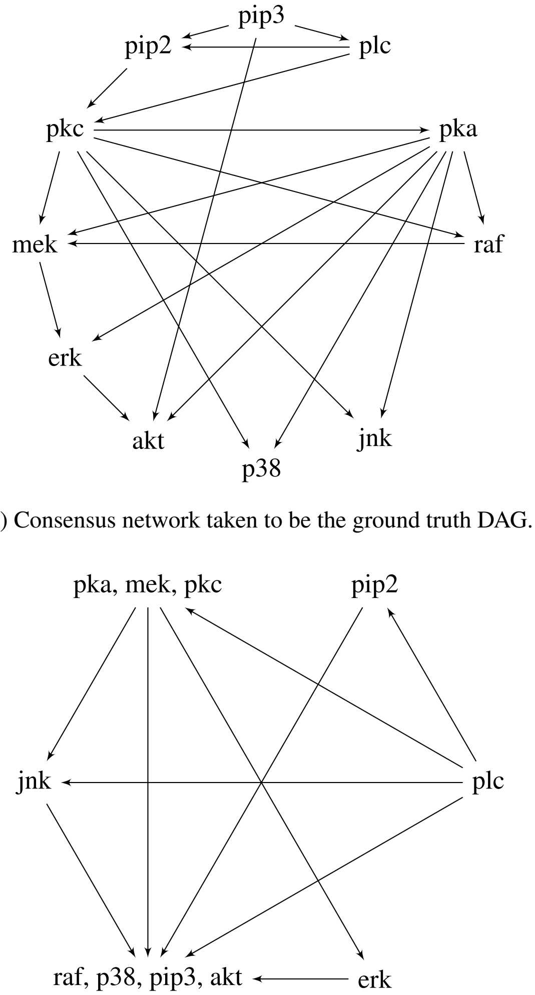
Table 1: Evaluation of coarsenings on data of Sachs et al. [2005] across different reference settings.

Reference	Parts	Edges	F-score	ARI
pkc	6	10	0.53	-0.04
pka	7	13	0.57	0.10
p38	2	1	1.00	-0.05
pip3	6	10	0.71	-0.11
plc8	7	13	0.55	-0.12

4.2 FLOW CYTOMETRY DATA APPLICATION

We now apply `RePaRe` to a real dataset from Sachs et al. [2005] that contains measurements of protein-signaling networks in the human immune system. We use a discretized, preprocessed version of the data, obtained from the benchmarking repository https://github.com/cmu-phil/example-causal-datasets. It has 7466 observations across six different perturbations.

The supposed ground truth DAG that used for evaluation is shown in Figure 2a, and an example of a coarsened DAG learned by `RePaRe` is shown in Figure 2b. Lacking observational data, there is no clear reference data for the test in `IKS-REFINE`. Table 1 shows results across five different possibilities for treating one of the perturbations as the observational reference data—note that the coarsening shown in Figure 2b corresponds to the row of reference 'pip3' in the table. The results are reasonably consistent across different references (e.g., 6–7 parts and 10–13 edges in 4 out of 5 graphs), but they are consistently bad: the average ARI is around 0, indicating chance-level partitioning; and the average F-score is around 0.6, indicating mediocre edge recovery.



(a) Consensus network taken to be the ground truth DAG.

(b) Learned coarsening based on available interventions.

Figure 2: Application to the data of Sachs et al. [2005].

5 DISCUSSION

Because this is a work-in-progress, we focus the discussion on understanding current shortcomings and hinting at future work.

- **Exploration of Alternative Oracle-replacements**: In

Section 3.2, we focus on specific tests, which perform poorly in Section 4—this is likely due to the low statistical power of the Kolmogorov-Smirnov (KS) test, particularly for nodes that are farther downstream from the intervention. It would be beneficial to consider a broader range of statistical tests or even score-based methods. This is the most interesting direction to continue for practically-oriented abstraction research.

- **Broader Notions of Coarsening**: While we present the most general notion of coarsening and describe marginal coarsening, there are likely other coarsening strategies that could be relevant. Future research could explore application-specific coarsening techniques that may yield different insights or improve the applicability of our framework in various contexts.
- **Relation to the Search Space of Causal Discovery**: It would be valuable to investigate how our findings relate to the search space of classic causal discovery algorithms [cf. Linusson et al., 2023]). Algorithm 1 already returns a DAG over the full dataset, so exploring its computational complexity or its relation to the space of CPDAGs could lead to insights for (non-abstraction) causal discovery. Additionally, our practical algorithm used interventional data and hence could be related to identifiability results about interventional Markov equivalence classes [Yang et al., 2018].

Acknowledgements The work by AM was supported by Novo Nordisk Foundation Grant NNF20OC0062897. PM received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 883818).

REFERENCES

S. Beckers. Equivalent causal models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6202–6209, 2021.

S. Beckers and J. Y. Halpern. Abstracting causal models. In *The Thirty-Third AAAI Conference on Artificial Intelligence*, pages 2678–2685, 2019. doi: 10.1609/aaai.v33i01.33012678. arXiv:1812.03789 [cs.AI].

S. Beckers, F. Eberhardt, and J. Y. Halpern. Approximate causal abstractions. In *Uncertainty in artificial intelligence*, pages 606–615. PMLR, 2020.

D. Deligeorgaki, A. Markham, P. Misra, and L. Solus. Combinatorial and algebraic perspectives on the marginal independence structure of Bayesian networks. *Algebraic Statistics*, 14(2):233–286, 2023. doi: 10.2140/astat.2023.14.233.

J. L. Gamella, A. Taeb, C. Heinze-Deml, and P. Bühlmann. Characterization and greedy learning of Gaussian structural causal models under unknown interventions. *arXiv preprint arXiv:2211.14897*, 2022.

M. Grosse-Wentrup, A. Kumar, A. Meunier, and M. Zimmer. Neuro-cognitive multilevel causal modeling: A framework that bridges the explanatory gap between neuronal activity and cognition. *PLOS Computational Biology*, 20 (12):e1012674, 2024.

J. Gu and Q. Zhou. Learning big Gaussian Bayesian networks: Partition, estimation and fusion. *Journal of machine learning research*, 21(158):1–31, 2020.

J. L. Hodges. The significance probability of the Smirnov two-sample test. *Arkiv för matematik*, 3(5):469–486, 1958.

L. Hubert and P. Arabie. Comparing partitions. *Journal of classification*, 2:193–218, 1985.

S. Linusson, P. Restadh, and L. Solus. Greedy causal discovery is geometric. *SIAM Journal on Discrete Mathematics*, 37(1):233–252, 2023.

C. D. Manning. *An introduction to information retrieval*. Cambridge University Press, 2009.

R. Massidda, S. Magliacane, and D. Bacciu. Learning causal abstractions of linear structural causal models. *UAI*, 2024.

M. Meyniel. Une condition suffisante d'existence d'un circuit hamiltonien dans un graphe oriente. *Journal of Combinatorial Theory, Series B*, 14(2):137–147, 1973.

J. Otsuka and H. Saigo. On the equivalence of causal models: A category-theoretic approach. In *Conference on Causal Learning and Reasoning*, pages 634–646. PMLR, 2022.

J. Pearl. *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. Morgan kaufmann, 1988.

P. K. Rubenstein, S. Weichwald, S. Bongers, J. M. Mooij, D. Janzing, M. Grosse-Wentrup, and B. Schölkopf. Causal consistency of structural equation models. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2017.

J. Runge. Conditional independence testing based on a nearest-neighbor estimator of conditional mutual information. In *International Conference on Artificial Intelligence and Statistics*, pages 938–947. PMLR, 2018.

K. Sachs, O. Perez, D. Pe'er, D. A. Lauffenburger, and G. P. Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529, 2005.

R. P. Stanley. *Enumerative Combinatorics, Volume 1*. Cambridge studies in advanced mathematics, second edition, 2011.

S. Tikka, J. Helske, and J. Karvanen. Clustering and structural robustness in causal diagrams. *Journal of Machine Learning Research*, 24(195):1–32, 2023.

P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, et al. Scipy 1.0: fundamental algorithms for scientific computing in Python. *Nature methods*, 17(3):261–272, 2020.

K. Yang, A. Katcoff, and C. Uhler. Characterizing and learning equivalence classes of causal dags under interventions. In *International Conference on Machine Learning*, pages 5541–5550. PMLR, 2018.