Intervention Efficient Algorithms for Approximate Learning of Causal Graphs

Raghavendra Addanki RADDANKI@CS.UMASS.EDU
Andrew McGregor MCGREGOR@CS.UMASS.EDU
Cameron Musco CMUSCO@CS.UMASS.EDU
University of Massachusetts Amherst, MA 01003, USA

Editors: Vitaly Feldman, Katrina Ligett and Sivan Sabato

Abstract

We study the problem of learning the causal relationships between a set of observed variables in the presence of latents, while minimizing the cost of interventions on the observed variables. We assume access to an undirected graph G on the observed variables whose edges represent either all direct causal relationships or, less restrictively, a superset of causal relationships (identified, e.g., via conditional independence tests or a domain expert). Our goal is to recover the directions of all causal or ancestral relations in G, via a minimum cost set of interventions.

It is known that constructing an exact minimum cost intervention set for an arbitrary graph G is NP-hard. We further argue that, conditioned on the hardness of approximate graph coloring, no polynomial time algorithm can achieve an approximation factor better than $\Theta(\log n)$, where n is the number of observed variables in G. To overcome this limitation, we introduce a bi-criteria approximation goal that lets us recover the directions of all but ϵn^2 edges in G, for some specified error parameter $\epsilon > 0$. Under this relaxed goal, we give polynomial time algorithms that achieve intervention cost within a small constant factor of the optimal. Our algorithms combine work on efficient intervention design and the design of low-cost *separating set systems*, with ideas from the literature on graph property testing.

1. Introduction

Discovering causal relationships is one of the fundamental problems of causality (Pearl, 2009). In this paper, we study the problem of *learning a causal graph* where we seek to identify all the causal relations between variables in our system (nodes of the graph). It has been shown that, under certain assumptions, observational data alone lets us recover the existence of a causal relationship between, but not the direction of all relationships. To recover the direction, we use the notion of an intervention (or an experiment) described in Pearl's Structural Causal Models (SCM) framework (Pearl, 2009).

An intervention requires us to fix a subset of variables to each value in their domain, inducing a new distribution on the free variables. For example, we may intervene to require that some patients in a study follow a certain diet and others do not. As performing interventions is costly, a widely studied goal is to find a minimum set of interventions for learning the causal graph (Shanmugam et al., 2015). This goal however does not address the fact that interventions may have different costs. For example, interventions that fix a higher number of variables will be more costly. Additionally, there may be different intervention costs associated with different variables. For example, in a medical study, intervening on certain variables might be impractical or unethical. Hyttinen et al. (2013) address the need for such cost models and give results for the special case of learning the

directions of *complete graphs* when the cost of an intervention is equal to the number of variables contained in the intervention. Generalizing this notion, we study a *linear cost model* where the cost of an intervention on a set of variables is the sum of (possibly non-uniform) costs for each variable in the set. This model was first introduced in Kocaoglu et al. (2017a) and has received recent attention (Lindgren et al., 2018; Addanki et al., 2020).

Significant prior work on efficient intervention design assumes *causal sufficiency*, i.e., there are no unobserved (latent) variables in the system. In this setting, there is an exact characterization of the interventions required to learn the causal graph, using the notion of *separating set systems* (Shanmugam et al., 2015; Eberhardt, 2007). Recently, the problem of learning the causal graph with latents using a minimum number of interventions has received considerable attention with many known algorithms that depend on various properties of the underlying causal graph (Kocaoglu et al., 2017b; Addanki et al., 2020; Kocaoglu et al., 2019). However, the intervention sets used by these algorithms contain a large number of variables, often as large as $\Omega(n)$, where n is the number of observable variables. Thus, they are generally not efficient in the linear cost model. Some work has considered efficient intervention design in the linear cost model for recovering the ancestral graph containing all indirect causal relations (Addanki et al., 2020). Other algorithms such as IC* and FCI with running times exponential in the size of the graph, aim to learn the causal graph in the presence of latents using only observational data; however, they can only learn a part of the entire causal graph (Verma and Pearl, 1992; Spirtes et al., 2000).

1.1. Our Results

In order to address the shortcomings when there are latents, we consider two settings. In the first setting, we assume that we are given an undirected graph that contains all causal relations between observable variables, but must identify their directions. This undirected graph may be obtained, e.g., by running algorithms that identify conditional dependencies and consulting a domain expert to identify causal links. In the second setting, we study a relaxation where we are given a supergraph H of G containing all causal edges and other additional edges which need not be causal. The second setting is less restrictive, modeling the case where we can ask a domain expert or use observational data to identify a *superset* of possible causal relations.

From H we seek to recover edges of the ancestral graph[1] of G, a directed graph containing all causal *path* relations between the observable variables. Depending on the method by which H is obtained, it may have special properties that can be leveraged for efficient intervention design. For example, if we use FCI/IC* (Spirtes et al., 2000) to recover a partial ancestral graph from observational data, the remaining undirected edges form a chordal graph (Zhang, 2008a). Past work has also considered the worst case when H is the complete graph (Addanki et al., 2020). In this work, we do not assume anything about how H is obtained and thus give results holding for general graphs.

In both settings, we show a connection to separating set systems. Specifically, to solve the recovery problems it is necessary and sufficient to use a set of interventions corresponding to a separating set system when we are given the undirected causal graph G and a *strongly* separating set system when we are given the supergraph H. A separating set system is one in which each pair of nodes connected by an edge is separated by at least one intervention – one variable is intervened on

1. We note that *ancestral graph* defined here and in Kocaoglu et al. (2017b); Addanki et al. (2020) is slightly different from the widely used notion from Richardson and Spirtes (2002).

and the other is free. A strongly separating set system requires that every connected edge (u, v) is separated by two interventions – there exists a intervention including u but not v and an intervention that includes v but not u.

Unfortunately, finding a minimum cost (strongly) separating set system for an arbitrary graph G is NP-hard (Lindgren et al., 2018; Hyttinen et al., 2013). We give simple algorithms that achieve $O(\log n)$ approximation and further argue that, conditioned on the hardness of approximate graph coloring, no polynomial time algorithm can achieve $o(\log n)$ approximation, where n is the number of observed variables.

To overcome this limitation, we introduce a *bi-criteria approximation* goal that lets us recover all but ϵn^2 edges in the causal or ancestral graph, where $\epsilon > 0$ is a specified error parameter. For this goal, it suffices to use a relaxed notion of a set system, which we show can be found efficiently using ideas from the graph property testing literature (Goldreich et al., 1998).

In the setting where we are given the causal edges in G and must recover their directions, we give a polynomial time algorithm that finds a set of interventions from which we can recover all but ϵn^2 edges with cost at most ~ 2 times the optimal cost for learning the full graph. Similarly, in the setting of ancestral graph recovery, we show how to recover all but ϵn^2 edges with intervention cost at most ~ 4 times the optimal cost for recovering all edges.

Our result significantly extends the applicability of a previous result (Addanki et al., 2020) that gave a 2-approximation to the minimum cost strongly separating set system assuming the worst case when the supergraph H is a complete graph. That algorithm does not translate to an approximation guarantee better than $\Omega(\log n)$ for general graphs.

Finally, for the special case when G is a hyperfinite graph (Hassidim et al., 2009) with maximum degree Δ, we give algorithms (See Appendix E) that obtain approximation guarantees as above, and recover all but $\epsilon n \Delta$ edges of G.

1.2. Other Related Work

There is significant precedent for assumptions on background knowledge in the literature. For example, (Hyttinen et al., 2013) and references therein, study intervention design in the same model: a skeleton of possible edges in the causal graph is given via background knowledge, which may come e.g., from domain experts or previous experimental results. Assuming causal sufficiency (no latents), most work focuses on recovering causal relationships based on just observational data. Examples include algorithms like *IC* (Pearl, 2009) and *PC* (Spirtes et al., 2000), which have been widely studied (Hauser and Bühlmann, 2014; Hoyer et al., 2009; Heinze-Deml et al., 2018; Loh and Bühlmann, 2014; Shimizu et al., 2006). It is well-known that to disambiguate a causal graph from an equivalence class of possible causal structures, interventional, rather than just observational data is required (Hauser and Bühlmann, 2012; Eberhardt and Scheines, 2007; Eberhardt, 2007). There is a growing body of recent work devoted to minimizing the number of interventions (Shanmugam et al., 2015; Kocaoglu et al., 2017b, 2019) and costs of intervention (Lindgren et al., 2018; Kocaoglu et al., 2017a). Since causal sufficiency is often too strong an assumption (Bareinboim and Pearl, 2016), many algorithms avoiding the causal sufficiency assumption, such as IC* (Verma and Pearl, 1992) and FCI (Spirtes et al., 2000), and using just observational data have been developed. There is a growing interest in optimal intervention design in this setting (Silva et al., 2006; Hyttinen et al., 2013; Parviainen and Koivisto, 2011; Kocaoglu et al., 2017b, 2019).

2. Preliminaries

Causal Graph Model. Following the SCM framework (Pearl, 2009), we represent a set of random variables by $V \cup L$ where V contains the endogenous (observed) variables that can be measured and L contains the exogenous (latent) variables that cannot be measured. We define a directed causal graph $\mathcal{G} = \mathcal{G}(V \cup L, \mathcal{E})$ on these variables where an edge corresponds to a causal relation between the corresponding variables: a directed edge (v_i, v_j) indicates that v_i causes v_j.

We assume that all causal relations belong to one of two categories : (i) $E \subseteq V \times V$ containing direct causal relations between the observed variables and (ii) $E_L \subseteq L \times V$ containing relations from latents to observable variables. Thus, the full edge set of our causal graph is $\mathcal{E} = E \cup E_L$. We also assume that every latent $l \in L$ influences exactly two observed variables, i.e., $(l, u), (l, v) \in E_L$ and no other edges are incident on l. This *semi-Markovian* assumption is widely used in prior work (Kocaoglu et al., 2017b; Shpitser and Pearl, 2006) (see Appendix A for a more detailed discussion). Let $G(V, E)$ denote the subgraph of $\mathcal{G}$ restricted to observable variables, referred to as the observable graph.

Similar to Kocaoglu et al. (2017b); Addanki et al. (2020), we define *ancestral graph* of G over observable variables V, denoted by $\mathrm{Anc}(G)$ as follows : $(v_i, v_j) \in \mathrm{Anc}(G)$ iff there is a directed path from v_i to v_j in G (equivalently in $\mathcal{G}$ due to the semi-Markovian assumption). Throughout we denote $n = |V|$.

Intervention Sets. Our primary goal is to recover either G or $\mathrm{Anc}(G)$ via interventions on the observable variables. We assume the ability to perform an *intervention* on a set of variables $S \subseteq V$ which fixes $S = s$ for each s in the domain of S. We then perform a conditional independence test answering for all v_i, v_j *"Is v_i independent of v_j in the interventional distribution* $\mathrm{do}(S = s)$*?"* and denote it using $v_i \perp\!\!\!\perp v_j \mid \mathrm{do}(S)$. Here $\mathrm{do}(S = s)$ uses Pearl's do-notation to denote the interventional distribution when the variables in S are fixed to s.

An *intervention set* is a collection of subsets $\mathcal{S} = \{S_1, \ldots, S_m\}$ that we intervene on in order to recover edges of the observable or ancestral graph. It will also be useful to associate a matrix $L \in \{0, 1\}^{n \times m}$ with the collection where the ith column is the characteristic vector of set S_i, i.e., row entry corresponding to node in S_i is 1 iff it is present in S_i. We can also think of L as a collection of $n = |V|$ length-m binary vectors that indicate which of the m intervention sets $S_1, \ldots, S_m$ each variable v_i belongs to.

As is standard, we assume that $\mathcal{G}$ satisfies the *causal Markov condition* and assume *faithfulness* (Spirtes et al., 2000), both in the observational and interventional distributions following (Hauser and Bühlmann, 2014). This ensures that conditional independence tests lead to the discovery of true causal relations rather than spurious associations.

Cost Model and Approximate Learning. In our cost model, each node $u \in V$ has a cost $C(u) \in [1, W]$ for some $W \geq 1$ and the cost of intervention on a set $S \subseteq V$ has the linear form $C(S) = \sum_{u \in S} C(u)$. That is, interventions that involve a larger number of, or more costly nodes, are more expensive. Our goal is to find an intervention set $\mathcal{S}$ minimizing $C(\mathcal{S}) = \sum_{S \in \mathcal{S}} \sum_{u \in S} C(u)$, subject to a constraint m on the number of interventions used. This *min cost intervention design* problem was first introduced in Kocaoglu et al. (2017a).

Letting $L \in \{0, 1\}^{n \times m}$ be the matrix associated with an intervention set $\mathcal{S}$, the cost $C(\mathcal{S})$ can be written as $C(L) = \sum_{j=1}^{n} C(v_j) \cdot \|L(j)\|_1$, where $\|L(j)\|_1$ is the *weight* of L's j^{th} row, i.e., the number of 1's in that row or the number of interventions in which v_j is involved.

We study two variants of causal graph recovery, in which we seek to recover the observable graph G or the ancestral graph $\mathrm{Anc}(G)$. We say that an intervention set $\mathcal{S}$ is α-optimal for a given recovery task if $C(\mathcal{S}) \leq \alpha \cdot C(\mathcal{S}^*)$, where $\mathcal{S}^*$ is the minimum cost intervention set needed for that task. For both recovery tasks we consider a natural approximate learning guarantee:

Definition 1 (ϵ-Approximate Learning) *An algorithm ϵ-approximately learns $G(V, E)$ (analogously, $\mathrm{Anc}(G)$) if it identifies the directions of a subset $\widetilde{E} \subseteq E$ of edges with $|E \setminus \widetilde{E}| \leq \epsilon n^2$.*

Generally, we will seek an intervention set $\mathcal{S}$ that lets us ϵ-approximately learn G or $\mathrm{Anc}(G)$, and which has cost bounded in terms of $\mathcal{S}^*$, the minimum cost intervention set needed to *fully* learn the graph. In this sense, our algorithms are bicriteria approximations.

Independent Sets. Our intervention set algorithms will be based on finding large independent sets of variables, that can be included in the same intervention sets, following the approach of Lindgren et al. (2018). Given $G(V, E)$, a subset of vertices $Z \subseteq V$ forms an independent set if there are no edges between any vertices in Z, i.e., $E[Z] = \emptyset$ where $E[Z]$ is set of edges in the sub-graph induced by Z. Given a cost function $C : V \to \mathbb{R}^+$, an independent set Z is a maximum cost independent set (MIS) if $C(Z) = \sum_{u \in Z} C(u)$ is maximized over all independent sets in G. Since finding MIS is hard (Cormen et al., 2009), we will use the following two notions of a MIS, with the first often referred to as simply NEAR-MIS, in our approximate learning algorithms :

Definition 2 ((γ, ϵ)-NEAR-MIS) *A set of nodes $S \subseteq V$ is a (γ, ϵ)-near-MIS in $G = (V, E)$ if $C(S) \geq (1 - \gamma)C(T)$ and $|E[S]| \leq \epsilon n^2$ where T is a maximum cost independent set (MIS) in G.*

Definition 3 ((ρ, γ, ϵ)-Independent-Set) *A set of nodes $S \subseteq V$ is a (ρ, γ, ϵ)-independent-set in $G = (V, E)$ if $C(S) \geq \rho(1 - \gamma) \cdot C(V)$ and $|E[S]| \leq \epsilon n^2$.*

3. Separating Set Systems

Our work focuses on two important classes of intervention sets which we show in Sections 4 and 5 are necessary and sufficient for recovering G and $\mathrm{Anc}(G)$ in our setting. Missing details from this section are collected in Appendix B.

Definition 4 (Separating Set System) *For any undirected graph $G(V, E)$, a collection of subsets $\mathcal{S} = \{S_1, \cdots, S_m\}$ of V is a separating set system if every edge $(u, v) \in E$ is separated, i.e., there exists a subset S_i with $u \in S_i$ and $v \notin S_i$ or with $v \in S_i$ and $u \notin S_i$.*

Definition 5 (Strongly Separating Set System) *For any undirected graph $G(V, E)$, a collection of subsets $\mathcal{S} = \{S_1, \cdots, S_m\}$ of V is a strongly separating set system if every edge $(u, v) \in E$ is strongly separated, i.e., there exist two subsets S_i and S_j such that $u \in S_i \setminus S_j$ and $v \in S_j \setminus S_i$.*

For a separating set system, each pair of nodes connected in G must simply have different assigned row vectors in the matrix $L \in \{0, 1\}^{n \times m}$ corresponding to $\mathcal{S}$ (i.e., the rows of L form a valid coloring of G). For a strongly separating set system, the rows must not only be distinct, but one cannot have support which is a subset of the other's. We say that such rows are *non-dominating*: there are distinct $i, j \in [m]$ such that $L(u, i) = L(v, j) = 0$ and $L(u, j) = L(v, i) = 1$. We observe that every *strongly separating set system* must satisfy the non-dominating property (as also observed

in Lemma A.9 from Addanki et al. (2020)). When $\mathcal{S}$ is a (strongly) separating set system for G we call its associated matrix L a *(strongly) separating matrix* for G.

Finding an exact minimum cost (strongly) separating set system is NP-Hard (Lindgren et al., 2018; Hyttinen et al., 2013) and thus we focus on approximation algorithms. We say the $\mathcal{S}$ is an α-optimal (strongly) separating set system if $C(\mathcal{S}) \leq \alpha \cdot C(\mathcal{S}^*)$, where $\mathcal{S}^*$ is the minimum cost (strongly) separating set system. Equivalently, for matrices $C(L) \leq \alpha \cdot C(L^*)$ where L, L^* correspond to $\mathcal{S}, \mathcal{S}^*$ respectively.

Unfortunately, even when approximation is allowed, finding a low-cost set system for an arbitrary graph G is still hard. In particular, we prove a conditional lower bound based on the hardness of approximation for 3-coloring. Achieving a coloring for 3-colorable graphs that uses sub-polynomial colors in polynomial time is a longstanding open problem (Wigderson, 1983; Blum and Karger, 1997; Karger et al., 1994), with the current best known algorithm (Arora and Chlamtac, 2006) achieving an approximation factor $O(n^{0.2111})$. Thus Theorem 6 shows the hardness of finding near optimal separating set systems, barring a major breakthrough on this classical problem.

Theorem 6 *Assuming 3-colorable graphs cannot be colored with sub-polynomial colors in polynomial time, there is no polynomial time algorithm for finding an $o(\log n)$-optimal (strongly) separating set system for an arbitrary graph G with n nodes when $m = \beta \log n$ for some constant $\beta > 2$.*

Proof We give a proof by contradiction for the case of separating set system A similar proof can be extended to strongly separating set systems. Suppose G is a 3-colorable graph containing n nodes with *unit* costs for every node. We argue that if there is an a $o(\log n)$-optimal algorithm for separating set system then, we can use it to obtain an algorithm for 3-coloring of G using $n^{o(1)}$ colors, thereby giving a contradiction.

First, we observe that the cost of an optimal separating system on G when $m = \beta \log n$ is at most n, as each color class forms an independent set in G and every node in the color class can be assigned a vector of weight at most 1. Let $\mathcal{A}(G)$ denote the separating set system output by an α-optimal algorithm where $\alpha = o(\log n)$. We outline an algorithm that takes as input $\mathcal{A}(G)$ and returns a $n^{o(1)}$-coloring of G.

We have $C(\mathcal{A}(G)) \leq \alpha C(\mathcal{S}^*)$ where $\mathcal{S}^*$ is an optimal separating set system for G. Letting L be the separating matrix associated with $\mathcal{A}(G)$, we thus have

$$C(\mathcal{A}(G)) = \sum_{j=1}^{n} \|L(j)\|_1 \leq \alpha C(\mathcal{S}^*) \leq \alpha n.$$

Using an averaging argument, we have that in $\mathcal{A}(G)$, there are at most $\frac{n}{4}$ nodes (denoted by $V \setminus D^{(1)}$) with weight $\|L(j)\|_1$ more than 4α. Consider the remaining $\frac{3n}{4}$ nodes given by $D^{(1)}$. Let $D^{(1)}_j$ denote the nodes that have been assigned weight j by $\mathcal{A}(G)$. For each of the at most $\binom{m}{j}$ vectors with weight j that are feasible, we create a new color and color each node in $D^{(1)}_j$ using these new colors based on the weight j vectors assigned to the node in $\mathcal{A}(G)$. We repeat this procedure for every weight j in $D^{(1)}$. As the maximum weight of a node in $D^{(1)}$ is 4α, the total number of colors

that we use to color all the nodes of $D^{(1)}$ is

$$\sum_{j=0}^{4\alpha}\binom{m}{j} \leq \sum_{j=0}^{4\alpha}\frac{m^j}{j!} = \sum_{j=0}^{4\alpha}\frac{(4\alpha)^j}{j!}\left(\frac{m}{4\alpha}\right)^j \leq \mathrm{e}^{4\alpha}\left(\frac{m}{4\alpha}\right)^{4\alpha} \leq 2^{4\alpha\log e+4\alpha\log\frac{m}{4\alpha}}$$
$$< 2^{4\alpha\log e+\sqrt{4m\alpha}}$$
$$< 2^{o(\log n)+\sqrt{\log n\cdot o(\log n)}}$$
$$< n^{o(1)},$$

where the first strict inequality used the fact that $\log\frac{m}{4\alpha} \leq \sqrt{\frac{m}{4\alpha}}$ for $\frac{m}{4\alpha} > \frac{\beta\log n}{o(\log n)} > 32$.

After coloring the nodes of $D^{(1)}$, we remove these nodes from G and run α-optimal algorithm $\mathcal{A}$ on the remaining nodes $V \setminus D^{(1)}$. Observing that a sub-graph of a 3-colorable graph is also 3-colorable, we have that the set of nodes obtained by running $\mathcal{A}$ on $V \setminus D^{(1)}$ that have weight at most 4α (denoted by $D^{(2)}$) also require at most $n^{o(1)}$ colors. As $|D^{(i)}| \geq \frac{3|V\setminus D^{(i-1)}|}{4}$ for all $i \in \{1, 2, \cdots, \log n\}$, in at most $\log n$ recursive calls to $\mathcal{A}$, we will fully color G using at most $n^{o(1)}\log n = n^{o(1)}$ colors. Hence, we have obtained a $n^{o(1)}$-coloring of G using an α-optimal algorithm when $\alpha = o(\log n)$. ■

Remark. The results of Theorem 6 can be extended to any m. When $m = o(\log n)$, in our hardness example that uses 3 colors, any valid separating set system using m interventions would lead to a coloring of the graph using at most $2^m = n^{o(1)}$ colors, i.e., a sub-polynomial number of colors. Thus, even finding a valid separating matrix in this scenario is hard, under our assumed hardness of 3-coloring.

We shall now proceed to discuss a $O(\log n)$ approximation algorithm for finding (strongly) separating set systems. It is easy to check that for a strongly separating set system, every node must appear in at least one intervention (because of non-dominating property), and so the set system has cost as least $\sum_{v\in V} C(v)$. At the same time, with $m \geq 2\log n$, we can always find a strongly separating set system where each node appears in $\log n$ interventions. In particular, we assign each node to a unique vector with weight $\log n$. Such an assignment is non-dominating and since $\binom{2\log n}{\log n} \geq n$, is feasible. It achieves cost $C(\mathcal{S}) = \log n \cdot \sum_{v\in V} C(v)$, giving a simple $\log n$-approximation for the minimum cost strongly separating set system problem. For a separating set system, a simple $O(\log n)$-approximation is also achievable by first computing an approximate minimum weight vertex cover and assigning all nodes in its complementary independent set the weight 0 vector i.e., assigning them to no interventions. We give a sketch of the arguments involved in proving the approximation ratio of the above algorithm and defer the full details to Appendix B.

A $2\log n$-Approximation Algorithm. Find a 2-approximate weighted vertex cover X in G using the classic algorithm from Williamson and Shmoys (2011). In L, assign zero vector to all nodes of $V \setminus X$; assign every node in X with a unique vector of weight $\log n$ and return L.

We observe that all the nodes that are part of maximum cost independent set (complement of minimum weighted vertex cover) are assigned a weight 0 vector by optimal separating system for G. Therefore, the cost of optimal separating set system is at least the cost of minimum cost vertex cover in G. As every node is assigned a vector of weight $\log n$ and the cost of vertex cover is at most twice the cost of the minimum weighted vertex cover, we have $C(L) \leq 2\log n \cdot C(L^*)$.

By Theorem 6, it is hard to improve on the above $O(\log n)$ approximation factor (up to constants). Therefore, we focus on finding relaxed separating set systems in which some variables are not separated. We will see that these set systems still suffice for approximately learning G and $\mathrm{Anc}(G)$ under the notion of Definition 1.

Definition 7 (ϵ-(Strongly) Separating Set System) *For any undirected graph $G(V, E)$, a collection of subsets $\mathcal{S} = \{S_1, \cdots, S_m\}$ of V is an ϵ-separating set system if, letting $L \in \{0,1\}^{n \times m}$ be the matrix corresponding to $\mathcal{S}$, $|\{(v_i, v_j) \in E : L(i) = L(j)\}| < \epsilon n^2$. It is strongly separating if $|\{(v_i, v_j) \in E : L(i), L(j)$ are not non-dominating$\}| < \epsilon n^2$.*

For ϵ-strongly separating set systems, when the number of interventions is large, specifically $m \geq 1/\epsilon$, a simple approach is to partition the nodes into $1/\epsilon$ groups of size $\epsilon \cdot n$. We then assign the same weight 1 vector to nodes in the same group and different weight 1 vectors to nodes in different groups. For ϵ-separating set system, we first find an approximate minimum vertex cover, and then apply the above partitioning. In Appendix B, we show that we get within a 2 factor of the optimal (strongly) separating set system. Therefore, for the remainder of this paper we assume $m < 1/\epsilon$. While m is an input parameter, smaller m corresponds to fewer interventions and this is the more interesting regime.

4. Observable Graph Recovery

We start by considering the setting where we are given all edges in the observable graph G (i.e., all direct causal relations between observable variables) e.g., by a domain expert, and wish to identify the direction of these edges. It is known that, assuming causal sufficiency (no latents), a separating set system is necessary and sufficient to learn G (Eberhardt, 2007). In Appendix C we show that this is also the case in the *presence of latents* when we are given the edges in G but not their directions. We also show that an ϵ-separating set system is sufficient to approximately learn G in this setting:

Claim 8 *Under the assumptions of Section 2, if $\mathcal{S} = \{S_1, S_2, \cdots S_m\}$ is an ϵ-separating set system for G, $\mathcal{S}$ suffices to ϵ-approximately learn G.*

In particular, if $\mathcal{S}$ is an ϵ-separating set system, we can learn all edges in G that are separated by $\mathcal{S}$ up to ϵn^2 edges which are not separated. Given Claim 8, our goal becomes to find an ϵ-separating matrix L_ϵ for G satisfying for some small approximation factor α, $C(L_\epsilon) \leq \alpha \cdot C(L^*)$ where L^* is the minimum cost separating matrix for G. Missing technical details of this section are collected in Appendix C.

We follow the approach of Lindgren et al. (2018), observing that every node in an independent set of G can be assigned the same vector in a valid separating matrix. They show that if we greedily peel off maximum independent sets from G and assign them the lowest remaining weight vector in $\{0,1\}^m$ not already assigned as a row in L, we will find a 2-approximate separating matrix. Their work focuses on chordal graphs where an MIS can be found efficiently in each step. However for general graphs G, finding an MIS (even approximately) is hard (see Appendix A). Thus, in Algorithm 1, we modify the greedy approach and in each iteration we find a *near* independent set with cost at least as large as the true MIS in G (Def. 2). Each such set has few internal edges, this leads to few non-separating assignments between edges of G in L_ϵ. Let ϵ be parameter that bounds the number of non-separating edges, and δ is the failure probability parameter of our Algorithm 1.

All the error parameters are scaled appropriately (See Appendix C for more details) when we pass them along in a procedure call to NEAR-MIS (line 5 in Algorithm 1).

Algorithm 1 ϵ-SEPARATING MATRIX(G, m, ϵ, δ)

Input : Graph $G = (V, E)$, cost function $C : V \to \mathbb{R}^+$, m, error ϵ, and failure probability δ.
Output : ϵ-Separating Matrix $L_\epsilon \in \{0,1\}^{n \times m}$.
Mark all vectors in $\{0,1\}^m$ as available.
while $|V| > 0$ **do**
    $S \leftarrow$ NEAR-MIS $(G, \epsilon^2, \epsilon\delta)$
    $\forall\, v_j \in S$, Set $L_\epsilon(j)$ to smallest weight vector available from $\{0,1\}^m$ and mark it unavailable.
    Update G by $E \leftarrow E \setminus E[S]$ and $V \leftarrow V \setminus S$.
end while
return L_ϵ

Observe that any subset of fewer than ϵn nodes has at most $\epsilon^2 n^2$ internal edges and so the NEAR-MIS $(G, \epsilon^2, \epsilon\delta)$ routine employed in Algorithm 1 always returns at least ϵn nodes. Thus the algorithm terminates in $1/\epsilon$ iterations. Across all $1/\epsilon$ NEAR-MIS's there are at most $\epsilon^2 n^2 \cdot 1/\epsilon = \epsilon n^2$ edges with endpoints assigned the same vector in L_ϵ, ensuring that L_ϵ is indeed ϵ-separating for G.

In Algorithm 2, we implement the NEAR-MIS routine by using the notion of a (ρ, γ, ϵ)-Independent-Set (Definition 3). We find a value of ρ that achieves close to the MIS cost via a search over decreasing powers of $(1+\gamma)$.

In Algorithm 3 we show how to obtain a (ρ, γ, ϵ)-Independent-Set (denoted by S) whenever the cost of MIS in G is at least $\rho \cdot C(V)$. So, $C(S) \geq \rho C(V) - \rho\gamma C(V)$ and we might lose a cost of at most $\gamma\rho C(V)$ compared to the MIS cost. Therefore, we add $\epsilon \cdot n$ nodes of highest cost (denoted by $S_{\epsilon/2}$) to S and argue that by setting $\gamma = O(\epsilon/W)$, $S \cup S_{\epsilon/2}$ has a cost at least the cost of MIS, i.e., $S \cup S_{\epsilon/2}$ is a $(0, \epsilon)$-NEAR-MIS.

Algorithm 2 NEAR-MIS

Input : Graph $G(V, E)$, cost function $C : V \to \mathbb{R}^+$, error ϵ, and failure probability δ.
Output : Set of nodes that is a $(0, \epsilon)$-NEAR-MIS in G.
Initialize $\rho = 1$, and let T be the set of $\sqrt{\epsilon}n$ nodes in G with the highest cost.
while $\rho \geq \sqrt{\epsilon}$ **do**
    $S \leftarrow$ INDEPENDENT-SET$(G, \rho, \epsilon/8W, \epsilon, \delta')$ where $\delta' = \epsilon\delta/4W\log(1/\epsilon)$
    Let $S_{\epsilon/2}$ denote the highest cost $\epsilon \cdot n/2$ nodes in $V \setminus S$.
    if $C(S \cup S_{\epsilon/2}) \geq C(T)$ **and** $|E[S \cup S_{\epsilon/2}]| \leq \epsilon n^2$ **then**
        return $S \cup S_{\epsilon/2}$
    end if
    $\rho = \rho/(1+\gamma)$
end while
return T

4.1. (ρ, γ, ϵ) – INDEPENDENT-SET

In this section, we introduce several new ideas and build upon the results for finding a $(\rho, 0, \epsilon)$-Independent-Set which has been used to obtain independent set property testers for graphs with unit vertex costs (Goldreich et al., 1998). First, we describe an overview of the general approach.

Unit Cost Setting. Suppose S is a fixed MIS in G with $|S| \geq \rho \cdot n$ and $U \subset S$. Let $\Gamma(u)$ represent the set of nodes that are neighbors of node u in G. Let

$$\Gamma(U) = \bigcup_{u \in U} \Gamma(u) \text{ and } \overline{\Gamma}(U) = V \setminus \Gamma(U).$$

Here, $\overline{\Gamma}(U)$ denotes the set of nodes with no edges to any node of U. We claim that $S \subseteq \overline{\Gamma}(U)$. First, we observe that $S \subseteq \overline{\Gamma}(S)$ as S is an independent set so no node in S is a neighbor of another node in S (i.e., all nodes in S are in $\overline{\Gamma}(S)$). Then, we use the fact $\overline{\Gamma}(S) \subseteq \overline{\Gamma}(U)$ since $U \subseteq S$ to conclude $S \subseteq \overline{\Gamma}(U)$. Further, Goldreich et al. (1998) proves that, if U is sampled randomly from S, taking the lowest degree $\rho \cdot n$ nodes in the induced subgraph on $\overline{\Gamma}(U)$ will with high probability yield a $(0, \epsilon)$-NEAR-MIS for G. Intuitively, the nodes in $\overline{\Gamma}(U)$ have no connections to U and thus are unlikely to have many connections to S.

To find a U that is fully contained in S, we can sample a small set of nodes in G; since we have $|S| \geq \rho \cdot n$ the sample will contain with good probability a representative proportion of nodes in S. We can then brute force search over all subsets of this sampled set until we hit U which is entirely contained in S and for which our procedure on $\overline{\Gamma}(U)$ returns a $(\rho, 0, \epsilon)$-Independent-Set, i.e., a NEAR-MIS.

General Cost Setting. In the general cost setting, when S is a high cost MIS, may not contain a large number of nodes, making it more difficult to identify via sampling. To handle this, we partition the nodes based on their costs in powers of $(1 + \gamma)$ into $k = O(\gamma^{-1} \log W)$ (where W is the maximum cost of a node in V) partitions $V_1, \ldots, V_k$.

A *good partition* is one that contains a large fraction of nodes in S: at least $\gamma\rho|V_i|$. Focusing on these partitions suffices to recover an approximation to S. Intuitively, all bad partitions have few nodes in S and thus ignoring nodes in them will not significantly affect the MIS cost.

Definition 9 ((γ, ρ)-good partition) *Let S be an independent set in G with cost $\geq \rho C(V)$. Then $F_{(\gamma,\rho)} = \{i \mid |V_i \cap S| \geq \gamma\rho|V_i|\}$ is the set of* good partitions *of V with respect to S.*

Claim 10 *Suppose S is an independent set in G with cost $C(S) \geq \rho C(V)$, then, there exists an independent set $S' \subseteq S$ such that $C(S') \geq \rho(1 - 2\gamma)C(V)$ and $S' \cap V_i = S \cap V_i$ for all $i \in F_{(\gamma,\rho)}$.*

While we do not a priori know the set of good partitions, if we sample a small number t of nodes uniformly from each partition, with good probability, for each good partition we will sample $\gamma\rho t/2$ nodes in S. We search over all possible subsets of partitions and in one iteration of our search, we have all the good partitions denoted by $\{V_1, V_2 \cdots V_\tau\}$. Now, for such a collection of good partitions, we search over all possible subsets $\mathcal{U} = U_1 \cup U_2 \cdots \cup U_\tau$ where $|U_i| = \gamma\rho t/2$ and in at least one instance have all U_i in good partitions fully contained in S. Let

$$Z(\mathcal{U}) := \bigcup_{i=1}^{\tau} V_i \setminus \bigcup_{i=1}^{\tau} \Gamma(U_i)$$

be the nodes in every *good* partition V_i with no connections to any of the nodes in U_i. Analogous to unit cost case, we sort the nodes in a *good* partition V_i by their degree in the induced subgraph on $Z(\mathcal{U})$. We select low degree nodes from each partition until the sum of the total degrees of the nodes selected is $\epsilon n^2/k$. We output union of all such nodes iff it is a $(\rho, 3\gamma, \epsilon)$-independent set. One key difference is that while including nodes from $Z(\mathcal{U})$, we do not include the nodes in the sorted order until sum of degrees is ϵn^2. Instead, we process each *good* partition and include the nodes from each partition separately. Later, we will argue that by doing so we have made sure that the cost contribution of a particular partition is accounted for accurately.

Algorithm 3 (ρ, γ, ϵ) INDEPENDENT-SET

Input : Graph $G = (V, E)$, cost function $C : V \to \mathbb{R}^+$, parameters ρ, γ, ϵ and δ
Output : $(\rho, 3\gamma, \epsilon)$ independent set in G if one exists.
For $i = 1, \ldots, k$, define $V_i = \{v \in V \mid (1+\gamma)^{i-1} \leq C(v) < (1+\gamma)^i\}$ where $k = \gamma^{-1} \log W$
Sample $t = O(\frac{k \log(k/\epsilon\delta)}{\epsilon\gamma\rho})$ nodes $\widetilde{V}_i$ in each partition V_i.
for every collection of partitions $\{V_1, V_2, \cdots V_\tau\} \subseteq \{V_1, V_2, \cdots V_k\}$ **do**
    for $\mathcal{U} = U_1 \cup U_2 \cup \cdots \cup U_\tau$ such that $U_i \subseteq \widetilde{V}_i$ with size $\gamma\rho t/2$ for all $i \in [\tau]$ **do**
        Let $Z(\mathcal{U}) := \bigcup_{i=1}^{\tau} V_i \setminus \bigcup_{i=1}^{\tau} \Gamma(U_i)$.
        for $i = 1 \ldots \tau$ **do**
            Sort nodes in $Z(\mathcal{U}) \cap V_i$ in increasing order of degree in the induced graph on $Z(\mathcal{U})$.
            Let $\widehat{Z}_i(\mathcal{U}) \subseteq Z(\mathcal{U}) \cap V_i$ be set of nodes obtained by considering the nodes in the sorted order until the total degree is $\epsilon n^2/k$.
        end for
        Let $\widehat{Z}(\mathcal{U}) = \bigcup_{i=1}^{\tau} \widehat{Z}_i(\mathcal{U})$.
        return $\widehat{Z}(\mathcal{U})$ if $C(\widehat{Z}(\mathcal{U})) \geq \rho(1-3\gamma)C(V)$.
    end for
end for

By construction, our output, denoted by $\widehat{Z}(\mathcal{U})$ will have at most ϵn^2 internal edges. Thus, the challenge lies in analyzing its cost. We argue that in at least one iteration, all chosen U_i for good partitions will not only lie within the MIS S, but their union will accurately represent connectivity to S. Specifically, any vertex $v \in Z(\mathcal{U})$, i.e., with no edges to U_i for all $i \in F_{(\gamma,\rho)}$, should have few edges to S. We formalize this notion using the definition of ϵ_2-IS representative subset below.

Definition 11 (ϵ_2-IS representative subset) *$R \subseteq \bigcup_{i \in F_{(\gamma,\rho)}} (S \cap V_i)$ is an ϵ_2-IS representative subset of S if for all but $\epsilon_2 n$ nodes of* good *partitions i.e., $\bigcup_{i \in F_{(\gamma,\rho)}} V_i$, we have the following property:*

$$\textit{Suppose } v \in \bigcup_{i \in F_{(\gamma,\rho)}} V_i : \quad \textit{if } \Gamma(v) \cap R = \emptyset \textit{ then } |\Gamma(v) \cap S| \leq \epsilon_2 n.$$

We show that there is a *ϵ_2-IS representative subset* containing at least $\gamma\rho t/2$ nodes from each good partition among our sampled nodes $\bigcup_{i=1}^{k} \widetilde{V}_i$. Setting $\epsilon_2 = \epsilon/2k$ we have:

Lemma 12 *If $t = O(\frac{k \log(k/\epsilon\delta)}{\epsilon\gamma\rho})$ nodes are uniformly sampled from each partition V_i to give $\widetilde{V}_i$, with probability $1-\delta$, there exists an $\epsilon/2k$-IS representative subset R such that, for every $i \in F_{(\gamma,\rho)}$, $|\widetilde{V}_i \cap R| = \gamma\rho t/2$.*

Lemma 12 implies that in at least one iteration, our guess $\mathcal{U}$ restricted to the *good partitions* is in fact an $\epsilon/2k$-IS representative subset. Thus, nearly all nodes in $Z(\mathcal{U})$ lying in good partitions have at most $\epsilon n/2k$ edges to S.

In the graph induced by nodes of $Z(\mathcal{U})$, with edge set $E[Z(\mathcal{U})]$, consider the degree incident on nodes of $S \cap V_i$ for each partition V_i. As there are at most n nodes in V_i, from Defn. 11, we have the total degree incident on $S \cap V_i$ is at most $\epsilon n^2/k$. Thus, including the nodes with lowest degrees in $\widehat{Z}_i(\mathcal{U})$ until the total degree is $\epsilon n^2/k$ will yield a set of nodes at least as large as $S \cap V_i$. Since all nodes in V_i have cost within a $1\pm\gamma$ factor of each other, we will have $C(\widehat{Z}_i(\mathcal{U})) \geq (1-\gamma)\cdot C(S \cap V_i)$. As the cost of S in the bad partitions is small, using Claim 10, we have $\widehat{Z}(\mathcal{U}) = \bigcup_{i=1}^{\tau} \widehat{Z}_i(\mathcal{U})$ is a $(\rho, O(\gamma), \epsilon)$-independent set.

4.2. Approximation Guarantee

Overall, Algorithm 3 implements a $(\rho, \gamma, \epsilon)-$INDEPENDENT-SET as required by Algorithm 2 to compute a NEAR-MIS in each iteration of Algorithm 1. It just remains to show that, by greedily peeling off NEAR-MIS from G iteratively, Algorithm 1 achieves a good approximation guarantee for ϵ-Approximate Learning G. To do this, we use the analysis of a previous work from Lindgren et al. (2018). In their work, an exact MIS is computed at each step, since their graph is chordal so the MIS problem is polynomial time solvable (Lindgren et al., 2018). However, the analysis extends to the case when the set returned has cost that is at least the cost of MIS (in our case a NEAR-MIS), allowing us to achieve near 2-factor approximation, as achieved in (Lindgren et al., 2018). Our final result is:

Theorem 13 *For any $m \geq \eta \log 1/\epsilon$ for some constant η, with probability $\geq 1-\delta$, Algorithm 1 returns L_ϵ with $C(L_\epsilon) \leq (2 + \exp(-\Omega(m))) \cdot C(L^*)$, where L^* is the min-cost separating matrix for G. Moreover L_ϵ ϵ-separates G. Algorithm 1 has a running time $O(n^2 f(W, \epsilon, \delta))$ where $f(W, \epsilon, \delta) = O\left(\frac{W}{\epsilon^2} \log \frac{1}{\epsilon} \exp\left(O\left(\frac{W^2 \log^2 W}{\epsilon^6} \log \frac{W}{\epsilon} \log \frac{W \log W \log 1/\epsilon}{\epsilon\delta}\right)\right)\right)$.*

5. Ancestral Graph Recovery

In Section 4, we assumed knowledge of the edges in the observable graph G and sought to identify their directions. In this section, we relax the assumption, assuming we are given any undirected supergraph H of G i.e., it includes all edges of G and may also include edges which do not represent causal edges. When given such a graph H, we cannot recover G itself and therefore, we seek to recover all directed edges of the ancestral graph $\mathrm{Anc}(G)$ appearing in H (i.e., the set of intersecting edges), which we denote by $\mathrm{Anc}(G) \cap H$. This problem strictly generalizes that of Section 4, as when $H = G$ we have $\mathrm{Anc}(G) \cap H = G$. Missing details of this section are collected in Appendix D.

First, we show that to recover $\mathrm{Anc}(G) \cap H$, a strongly separating system (Def 4) for H is both necessary and sufficient. Furthermore, an ϵ-strongly separating system suffices for approximate learning. We formalize this using the following lemma:

Lemma 14 *Under the assumptions of Section 2, if $\mathcal{S} = \{S_1, S_2, \cdots S_m\}$ is an ϵ-strongly separating set system for H , $\mathcal{S}$ suffices to ϵ-approximately learn $\mathrm{Anc}(G) \cap H$.*

Given Lemma 14, our goal becomes to find an ϵ-strongly separating matrix for H, L_ϵ with cost within an α factor of the optimal strongly separating matrix for H, for some small α.

To do so, our algorithm builds on the separating set system algorithm of Section 4. We first run Algorithm 1 to obtain an ϵ-separating matrix L_ϵ^S and construct $S_1, S_2, \cdots S_{1/\epsilon}$ where each set S_i contains all nodes assigned the same vector in L_ϵ^S – i.e., S_i corresponds to the NEAR-MIS computed at step i of Algorithm 1.

We form a new graph by contracting all nodes in each S_i into a single *super node* and denote the resulting at most $1/\epsilon$ vertices by V_S. In Addanki et al. (2020), the authors give a 2-approximation algorithm for finding a strongly separating matrix on a set of nodes, provided the graph on these nodes is complete. As H is an arbitrary super graph of G, the contracted graph on V_S is also arbitrary. However we simply assume the worst case, and run the Algorithm of Addanki et al. (2020) on it to produce L_ϵ^{SS}, which strongly separates the complete graph on V_S. It is easy to show that as a consequence, L_ϵ^{SS} ϵ-strongly separates H.

Algorithm 4 ANCESTRAL GRAPH(H, m, ϵ, δ)

1: $L_\epsilon^S :=$ ϵ-SEPARATING MATRIX(H, m, ϵ, δ).
2: Construct $S_1, S_2, \cdots S_{1/\epsilon}$ where each set S_i contains nodes assigned the same vectors in L_ϵ^S.
3: Construct a set of nodes V_S by representing S_i as a single node w_i and $C(w_i) = \sum_{u \in S_i} C(u)$.
4: $L_\epsilon^{SS} :=$ SSMATRIX(V_S, m) from Addanki et al. (2020).
5: **return** L_ϵ^{SS}

To prove the approximation bound, we extend the result of Addanki et al. (2020), showing that their algorithm actually achieves a cost at most 2 times the cost of a *separating matrix* for the complete graph on V_S which satisfies two additional restrictions: (1) it does not assign the all zeros vector to any node and (2) it assigns the same number of weight one vectors as the optimal strongly separating matrix. Further, we show via a similar analysis to Theorem 13 that this cost on V_S is bounded by 2 times the cost of the optimal strongly separating matrix on the contracted graph over V_S. Combining these bounds yields the final 4 approximation guarantee of Theorem 15.

Theorem 15 *Let $m \geq \eta \log 1/\epsilon$ for some constant η and L_ϵ^{SS} be matrix returned by Algorithm 4. Then with probability $\geq 1 - \delta$, L_ϵ^{SS} is an ϵ-strongly separating matrix for H and $C(L_\epsilon^{SS}) \leq (4 + \exp(-\Omega(m))) \cdot C(L^*)$ where L^* is the min-cost strongly separating matrix for H. Algorithm 4 runs in time $O(n^2 f(W, \epsilon, \delta))$ where*

$$f(W, \epsilon, \delta) = O\left(\frac{W}{\epsilon^2} \log \frac{1}{\epsilon} \exp\left(O\left(\frac{W^2 \log^2 W}{\epsilon^6} \log \frac{W}{\epsilon} \log \frac{W \log W \log 1/\epsilon}{\epsilon\delta}\right)\right)\right).$$

6. Open Questions

We highlight that in both the settings, although we consider the presence of latents in the system, in this paper, we provide results for learning causal relations among only the observable variables. Identification of latents is an important goal and has been well-studied (Kocaoglu et al., 2017b, 2019; Addanki et al., 2020) when the objective is to minimize the *number of interventions*. However, in Addanki et al. (2020), for the *linear cost model*, the authors argue that there is no good cost lower

bound known, even for recovering the observable (rather than ancestral) graph in the presence of latents. This makes the development of algorithms with approximation guarantees in terms of the optimum cost difficult. We view addressing this difficulty as a major open question.

Our results on bounded degree graphs make an additional assumption that the graph is hyperfinite, which gives more structure but still captures many graph families. It is an interesting open question if we can extend them to general sparse graphs. This setting is challenging since even finding a Near-MIS with $\epsilon \cdot |E|$ edges is still open and likely to be hard (Ron, 2010). We conjecture that if $|E| = O(n)$, a constant approximation for our objectives is not possible (assuming standard complexity theoretic conjectures) if we must separate all but $\epsilon \cdot |E|$ many edges.

It would also be very interesting to extend our work to the setting where we seek to identify a specific subset of edges of the causal graph, or where certain edges are 'more important' than others. We hope that our work is a first step in this direction, introducing the idea of partial recovery to overcome hardness results that rule out non-trivial approximation bounds for full graph recovery in the linear cost model.

Acknowledgements

The work was partially supported by NSF grants CCF-1934846, CCF-1908849 and CCF-1637536.

References

Raghavendra Addanki, Shiva Prasad Kasiviswanathan, Andrew McGregor, and Cameron Musco. Efficient intervention design for causal discovery with latents. *International Conference on Machine Learning*, 2020.

Noga Alon, Paul Seymour, and Robin Thomas. A separator theorem for graphs with an excluded minor and its applications. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 293–299, 1990.

Sanjeev Arora and Eden Chlamtac. New approximation guarantee for chromatic number. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 215–224, 2006.

Rémi Bardenet and Odalric-Ambrym Maillard. Concentration inequalities for sampling without replacement. *Bernoulli*, 21(3):1361–1385, Aug 2015. ISSN 1350-7265. doi: 10.3150/14-bej605. URL http://dx.doi.org/10.3150/14-BEJ605.

Elias Bareinboim and Judea Pearl. Causal inference and the data-fusion problem. *Proceedings of the National Academy of Sciences*, 113(27):7345–7352, 2016.

Avrim Blum and David Karger. An O($n^{3/14}$)-coloring algorithm for 3-colorable graphs. *Information processing letters*, 61(1):49–53, 1997.

Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT press, 2009.

Artur Czumaj, Asaf Shapira, and Christian Sohler. Testing hereditary properties of nonexpanding bounded-degree graphs. *SIAM Journal on Computing*, 38(6):2499–2510, 2009.

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

Frederick Eberhardt and Richard Scheines. Interventions and causal inference. *Philosophy of Science*, 74(5):981–995, 2007.

Uriel Feige, Shafi Goldwasser, Laszlo Lovász, Shmuel Safra, and Mario Szegedy. Interactive proofs and the hardness of approximating cliques. *Journal of the ACM (JACM)*, 43(2):268–292, 1996.

Andras Frank. Some polynomial algorithms for certain graphs and hypergraphs. In *Proceedings of the 5th British Combinatorial Conference*, 1975.

Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *Journal of the ACM (JACM)*, 45(4):653–750, 1998.

Avinatan Hassidim, Jonathan A Kelner, Huy N Nguyen, and Krzysztof Onak. Local graph partitions for approximation and testing. In *IEEE Symposium on Foundations of Computer Science*, pages 22–31, 2009.

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.

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

Christina Heinze-Deml, Marloes H Maathuis, and Nicolai Meinshausen. Causal structure learning. *Annual Review of Statistics and Its Application*, 5:371–391, 2018.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. In *The Collected Works of Wassily Hoeffding*, pages 409–426. Springer, 1994.

Patrik O Hoyer, Dominik Janzing, Joris M Mooij, Jonas Peters, and Bernhard Schölkopf. Nonlinear causal discovery with additive noise models. In *Advances in neural information processing systems*, pages 689–696, 2009.

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

David Karger, Rajeev Motwani, and Madhu Sudan. Approximate graph coloring by semidefinite programming. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 2–13. IEEE, 1994.

Murat Kocaoglu, Alex Dimakis, and Sriram Vishwanath. Cost-optimal learning of causal graphs. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1875–1884. JMLR. org, 2017a.

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, 2017b.

Murat Kocaoglu, Amin Jaber, Karthikeyan Shanmugam, and Elias Bareinboim. Characterization and learning of causal graphs with latent variables from soft interventions. In *Advances in Neural Information Processing Systems*, pages 14346–14356, 2019.

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.

Po-Ling Loh and Peter Bühlmann. High-dimensional learning of linear causal networks via inverse covariance estimation. *The Journal of Machine Learning Research*, 15(1):3065–3105, 2014.

Pekka Parviainen and Mikko Koivisto. Ancestor relations in the presence of unobserved variables. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 581–596. Springer, 2011.

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

Thomas Richardson and Peter Spirtes. Ancestral graph markov models. *Ann. Statist.*, 30(4):962–1030, 08 2002. doi: 10.1214/aos/1031689015.

Dana Ron. Algorithmic and analysis techniques in property testing. 2010.

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.

Shohei Shimizu, Patrik O Hoyer, Aapo Hyvärinen, and Antti Kerminen. A linear non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7(Oct):2003–2030, 2006.

Ilya Shpitser and Judea Pearl. Identification of joint interventional distributions in recursive semi-markovian causal models. In *Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts, USA*, pages 1219–1226, 2006.

Ricardo Silva, Richard Scheine, Clark Glymour, and Peter Spirtes. Learning the structure of linear latent variable models. *Journal of Machine Learning Research*, 7(Feb):191–246, 2006.

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

Jin Tian and Ilya Shpitser. On the identification of causal effects. 2003.

Thomas Verma and Judea Pearl. An algorithm for deciding if a set of observed independencies has a causal explanation. In *Uncertainty in artificial intelligence*, pages 323–330. Elsevier, 1992.

Avi Wigderson. Improving the performance guarantee for approximate graph coloring. *Journal of the ACM (JACM)*, 30(4):729–735, 1983.

David P Williamson and David B Shmoys. *The design of approximation algorithms*. Cambridge university press, 2011.

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 A. Discussions

A.1. Semi-Markovian Assumption

Our assumption that each latent only affects two observable variables is commonly known as the semi-Markovian condition and is standard in the literature, e.g., see Tian and Shpitser (2003); Kocaoglu et al. (2017b). In fact, using (pairwise) conditional independence tests, it is impossible to discover latent variables that affect more than two observables, even with unlimited interventions. Consider observables x, y, z and a latent l_{xyz} that is a parent of all them. If we test whether x, y, and z are all pairwise independent and they all turn out to be false, we can't distinguish the cases where a single latent l_{xyz} or three separate latents l_{xy}, l_{yz} and l_{xz} are causing this non-independence. Thus, we cannot remove the assumption without changing our intervention model or making more restrictive assumptions. As an example, Silva et al. (2006) considers the case when latents affect more than two observables, however, they make very strong assumptions – that there are no edges between observables, and each observable has only one latent parent.

A.2. Hardness of Independent Set

For the linear cost model, the problem of learning a causal graph was introduced in Kocaoglu et al. (2017a). It was shown recently that the problem of obtaining an optimum cost set of interventions is NP-hard (Lindgren et al., 2018). Under causal sufficiency (no latents), it is well known that the undirected graph (also called Essential Graph (Zhang, 2008b; Lindgren et al., 2018)) recovered after running the IC^* algorithm is chordal. Further, an intervention set which is a separating set system (Def. 4) for the Essential Graph of G is both necessary and sufficient (Eberhardt, 2007; Shanmugam et al., 2015) for learning the causal graph.

The authors of Lindgren et al. (2018) give a greedy algorithm to construct a 2-approximation to the optimal cost separating set system of the essential graph. Their algorithms requires at each step finding a maximum independent set in G and peeling it off the graph, and is the basis for our approach in Section 4. Since G is *chordal* , there is an algorithm for finding an exact maximum independent set in polynomial time (Frank, 1975). However, without the assumption of causal sufficiency, we cannot directly extend their algorithm, since finding a maximum independent set in a general graph G is NP-hard (Cormen et al., 2009). Moreover, finding an approximate independent set within a factor of n^ϵ for any $\epsilon > 0$ in polynomial time is also not possible unless $NP \subseteq BPP$ (Feige et al., 1996).

Appendix B. Missing Details From Section 3

B.1. $2 \log n$ Approximation Algorithm for Separating Set System

In this section, we show that the algorithm presented in section 3 obtains a $2 \log n$-optimal separating set system for a given graph G. To do so, we first make the following two simple claims. Let $\mathcal{S}^* = \{S_1, S_2, \cdots S_m\}$ be the minimum cost separating set system for G and I denote the maximum cost independent set in G.

Definition 16 *(Vertex Cover). A set of nodes S is a vertex cover for the graph $G(V, E)$, if for every edge $(u, v) \in E$, we have $\{u, v\} \cap S \neq \phi$.*

Claim 17 *The set of vertices in $V \setminus I$ forms a minimum weighted vertex cover for G.*

Proof Suppose X denote a minimum weighted vertex cover in G, then, $V \setminus X$ is an independent set in G. We have $C(X) = C(V) - C(V \setminus X) \geq C(V) - C(I)$ as I is maximum cost independent set. Observe that the vertex cover given by $X := V \setminus I$ satisfies the above equation with equality. Hence, the claim. ■

Claim 18 $C(\mathcal{S}^*) \geq C(V \setminus I)$.

Proof Let L^* denote optimal separating matrix corresponding to $\mathcal{S}^*$. We can rewrite $C(\mathcal{S}^*)$ in terms of $C(L^*) = \sum_{j=1}^{n} C(v_j) \|L(j)\|_1$. It is easy to observe that every node in an independent set of G can be assigned the same vector in a separating matrix. So, nodes with weight zero in L^* are from an independent set (say I_{L^*}) in G. As weight of the vectors assigned to remaining nodes in L^* is at least 1, we have $C(L^*) \geq C(V) - C(I_{L^*}) \geq C(V) - C(I)$, using the definition of I. ■

Combining Claims 17 and 18, we can observe that a good approximation for weighted vertex cover will result in a good approximation for separating set system. There is a well known 2-approximation algorithm for weighted vertex cover problem using linear programming that runs in polynomial time (Page 10, Theorem 1.6 Williamson and Shmoys (2011)).

Lemma 19 *If $m \geq 2 \log n$, then, there is an algorithm that returns a separating set system that is $2 \log n$-optimal.*

Proof Let X denote the minimum weighted vertex cover which is a 2-approximation obtained using the well known linear programming relaxation (Williamson and Shmoys, 2011). In our algorithm, we assign every node in X with a unique vector of weight $\log n$. This is feasible because the set of nodes in $V \setminus X$ form an independent set, and $\binom{m}{\log n} \geq \binom{2 \log n}{\log n} \geq n$. Combining Claims 17 and 18, we have

$$C(L) = \log n \, C(X) \leq 2 \log n \, C(V \setminus I) \leq 2 \log n \, C(\mathcal{S}^*).$$

■

B.2. Algorithms for ϵ-(Strongly) Separating Set System when $m \geq 1/\epsilon$

ϵ-Separating Set System. For ϵ-separating set system on $G(V, E)$, we first find a 2-approximate minimum weighted vertex cover X using the well-known linear programming based algorithm from Williamson and Shmoys (2011) (Refer Page 10, Theorem 1.6 in Williamson and Shmoys (2011)). We then partition the nodes of X randomly into $1/\epsilon$ groups of expected size $\epsilon \cdot n$. We then assign the same weight 1 vector to nodes in the same group and different weight 1 vectors to nodes in different groups. This is possible since $m \geq 1/\epsilon$. It is easy to see that the total number of edges that are not separated on expectation is $\epsilon|E| \leq \epsilon n^2$. For the remaining nodes in $V \setminus X$ that form an independent set, we assign the zero vector. Therefore, total cost of ϵ-separating set system is given by $C(X)$. From Claim 17, we have $C(X) \leq 2C(V \setminus I)$ where I is maximum weighted independent set in G. Using Claim 18, we have $C(X) \leq 2C(\mathcal{S}^*)$ where $\mathcal{S}^*$ is optimal separating set system for G. Therefore, we get within a 2 factor of the optimal separating set system.

ϵ-Strongly Separating Set System. For ϵ-strongly separating set system on $H(V, E)$, we partition the nodes randomly into $1/\epsilon$ groups of expected size $\epsilon \cdot n$. We then assign the same weight 1 vector to

nodes in the same group and different weight 1 vectors to nodes in different groups. This is possible since $m \geq 1/\epsilon$. It is easy to see that the total number of edges that are not strongly separated on expectation is $\epsilon|E| \leq \epsilon n^2$. As every vector assigned to a node in a valid strongly separating matrix should have weight at least 1, this results in an ϵ-strongly separating matrix, and the corresponding set system with optimal cost.

Appendix C. Missing Details From Section 4

In this section, we refer to the conditional independence test described in section 2 as CI-test.

Claim 20 *Suppose a set on interventions $\mathcal{S} = \{S_1, S_2, \cdots S_m\}$ is used for learning the edges of an undirected causal graph G. Then, under the assumptions of section 2, $\mathcal{S}$ is a separating set system for G.*

Proof First, we show that when $\mathcal{S}$ is a separating set system for G, we can recover the directions of G. Consider an edge $(v_i, v_j) \in G$ and let $S_k \in \mathcal{S}$ be such that $v_i \in S_k$ and $v_j \notin S_k$. As $\mathcal{S}$ is a separating set system, we know that such a set S_k exists for every edge in G. Consider the CI-test between v_i and v_j in the interventional distribution $\mathrm{do}(S_k)$. If the test returns that $v_i \perp\!\!\!\perp v_j \mid \mathrm{do}(S_k)$, then, we infer $v_i \to v_j$, otherwise we infer that $v_i \leftarrow v_j$. When we intervene on v_i obtained by $\mathrm{do}(S_k)$, the latent edges affecting v_i and all other incoming edges to v_i are removed. As we know that there is a causal edge between the two variables, if the independence test returns true, it must mean that there is no incoming edge into v_i from v_j.

In Eberhardt (2007), it was shown that a separating set system is necessary for learning the directions among the observable variables assuming causal sufficiency. As we are trying to recover G using interventions, such a condition will also hold for our case that is a generalization when not assuming causal sufficiency. Hence, the claim. ■

Claim 21 *(Claim 8 restated) Under the assumptions of Section 2, if $\mathcal{S} = \{S_1, S_2, \cdots S_m\}$ is an ϵ-separating set system for G , $\mathcal{S}$ suffices to ϵ-approximately learn G.*

Proof Given $\mathcal{S}$ denotes an ϵ-separating set system for $G(V, E)$. So, there are at most ϵn^2 edges $(u, v) \in E$ such that for all $i \in [m]$, either $\{u, v\} \cap S_i = \phi$ or $\{u, v\} \cap S_i = \{u, v\}$. For every such edge, any intervention on a set in $\mathcal{S}$, say S_i cannot recover the direction from a CI-test $u \perp\!\!\!\perp v \mid \mathrm{do}(S_i)$? because for both the cases $u \leftarrow v$ or $u \to v$, the CI-test returns that they are dependent. For the remaining edges $(u, v) \in E$, in the intervention S_j where $\{u, v\} \cap S_j = \{u\}$, we can recover the direction using the CI-test : $u \to v$ if $u \not\perp\!\!\!\perp v \mid \mathrm{do}(S_j)$ and $u \leftarrow v$ otherwise. From Def. 1, we have that $\mathcal{S}$ ϵ-approximately learns G. ■

Claim 22 *(Claim 10 restated) Suppose S is an independent set in G with cost $C(S) \geq \rho C(V)$, then, there exists an independent set $S' \subseteq S$ such that $C(S') \geq \rho(1 - 2\gamma)C(V)$ and $S' \cap V_i = S \cap V_i$ for all $i \in F_{(\gamma,\rho)}$.*

Proof Construct S' using (γ, ρ)-good partitions of V. For every $i \in F_{(\gamma,\rho)}$, include $S \cap V_i$ in S'. Therefore, we have

$$\begin{aligned}
C(S') &= C(S) - \sum_{i \notin F_{(\gamma,\rho)}} C(S \cap V_i) \\
&\geq \rho C(V) - \gamma\rho \sum_{i \notin F_{(\gamma,\rho)}} |V_i|(1+\gamma)^i \\
&\geq \rho C(V) - \gamma\rho(1+\gamma) \sum_{i \notin F_{(\gamma,\rho)}} |V_i|(1+\gamma)^{i-1} \\
&\geq \rho C(V) - \gamma\rho(1+\gamma) C(V) \\
&\geq \rho(1-2\gamma) C(V).
\end{aligned}$$

■

Lemma 23 *(Lemma 12 restated) If $t = O(\frac{k}{\epsilon\gamma\rho} \log \frac{4k}{\epsilon\delta})$ nodes are uniformly sampled from each partition V_i to give $\widetilde{V}_i$, with probability $1-\delta$, there exists an $\epsilon/2k$-IS representative subset R such that, for every $i \in F_{(\gamma,\rho)}$, $|\widetilde{V}_i \cap R| = \gamma\rho t/2$.*

Proof

Consider a good partition V_i for some $i \in F_{(\gamma,\rho)}$. So, $|V_i \cap S| \geq \gamma\rho|V_i|$. As $\widetilde{V}_i$ consists of t nodes that are uniformly sampled from V_i, using Hoeffding's inequality (Bardenet and Maillard (2015); Hoeffding (1994)), we know that $|\widetilde{V}_i \cap S| \geq \gamma\rho t/2$ with probability at least

$$1 - \exp(-\gamma\rho t/8) \geq 1 - \exp\left(-\frac{k}{\epsilon}\log\frac{4k}{\epsilon\delta}\right) \geq 1 - \frac{\delta}{2k}.$$

Applying union bound, we have for every $i \in F_{(\gamma,\rho)}$, $|\widetilde{V}_i \cap S| \geq \gamma\rho t/2$ with probability at least

$$1 - k\frac{\delta}{2k} \geq 1 - \frac{\delta}{2}.$$

Consider the union of all subsets $U_j \subseteq \widetilde{V}_j \cap S$ of good partitions such that $|U_j| = \gamma\rho t/2$, i.e.,

$$R = \bigcup_{j=1 \mid j \in F_{(\gamma,\rho)}}^{k} U_j.$$

We claim that R is a $\epsilon/2k$-IS representative subset of V by arguing that if v has no neighbours in R, then, the degree to S is more than $\epsilon n/2k$ with *low* probability.

First, consider the case when $v \in S$, then $|\Gamma(v) \cap S| = 0$ and $\Gamma(v) \cap R = \phi$. Suppose $v \in V_j \setminus S$ for some $j \in F_{(\gamma,\rho)}$ and $|\Gamma(v) \cap S| \geq \epsilon n/2k$. If $\Gamma(v) \cap R = \phi$, then $\Gamma(v) \cap R \cap V_i = \phi$ for all $i \in F_{(\gamma,\rho)}$. As R is formed using the sampled nodes, we have that every node in R should be from

$V_i \setminus (\Gamma(v) \cap S \cap V_i)$ for the condition $\Gamma(v) \cap R = \phi$ to be satisfied. As every element in R is chosen uniformly at random from the respective good partitions independently, we have :

$$\begin{aligned}
\Pr_{\forall i, U_i \sim V_i}[\forall i : \Gamma(v) \cap R \cap V_i = \phi \textbf{ and } |\Gamma(v) \cap S| > \epsilon n/2k] &\leq \Pi_{i \in F_{(\gamma,\rho)}} \left(\frac{|V_i| - |\Gamma(v) \cap S \cap V_i|}{|V_i|} \right)^{|U_i|} \\
&\leq \exp\left(- \sum_i \frac{|U_i||\Gamma(v) \cap S \cap V_i|}{|V_i|} \right) \\
&\leq \exp\left(-\frac{\gamma\rho t}{n} \sum_i |\Gamma(v) \cap S \cap V_i| \right) \\
&\leq \exp\left(-\frac{\gamma\rho t}{n} \frac{\epsilon n}{2k} \right) \\
&\leq \epsilon\delta/2k.
\end{aligned}$$

Therefore, on expectation, there are at most $n \cdot \epsilon\delta/4k$ nodes such that the number of neighbours in S is more than $\epsilon n/2k$. Using Markov's inequality, with probability $1 - \delta/2$, we have that at most $\epsilon n/2k$ nodes have number of neighbours in S greater than $\epsilon n/2k$. Applying union bound, we have with probability $1 - \delta$ that R is a $\epsilon/2k$-IS representative subset. ■

Lemma 24 *Suppose S is an independent set in G with cost $C(S) \geq \rho C(V)$ for some $\rho > 0$ and $\widehat{Z}(\mathcal{U})$ denote the set found by Algorithm 3 such that $\mathcal{U}$ is a $\epsilon/2k$-IS representative subset. Then, with probability $1 - \delta$, we have*

$$C(\widehat{Z}(\mathcal{U})) \geq \rho(1 - 3\gamma)C(V).$$

Proof Consider $\widehat{Z}_i(\mathcal{U})$ for some $i \in F_{(\gamma,\rho)}$ and let $F := \bigcup_{i \in F_{(\gamma,\rho)}} V_i$. In Algorithm 3, we obtained $\widehat{Z}_i(\mathcal{U})$ by including nodes from $Z(\mathcal{U}) \cap V_i$ in the sorted order of degree such that the total degree of nodes in the induced graph $Z(\mathcal{U})$ is bounded by $\epsilon n^2/k$. First, when $\mathcal{U}$ is a $\epsilon/2k$-IS representative subset, we observe that

$$\mathcal{U} \subseteq S \cap F \subseteq F \setminus \Gamma(S \cap F) \subseteq Z(\mathcal{U}).$$

So, $S \cap V_i \subseteq Z(\mathcal{U}) \cap V_i$. From Lemma 12, we have, for every node in $Z(\mathcal{U}) \cap V_i$ except for $\epsilon/2k$ many, the maximum degree to $S \cap V_i$ is at most $\epsilon n/2k$, and the remaining nodes can have a maximum degree of n. Combining these statements, we have that the total degree incident on the nodes in $S \cap V_i$ from the nodes $Z(\mathcal{U}) \cap V_i$ is at most

$$\frac{\epsilon n}{2k} \cdot |Z(\mathcal{U}) \cap V_i| + n \cdot \frac{\epsilon n}{2k} \leq \frac{\epsilon n^2}{k}.$$

As we include nodes in $\widehat{Z}_i(\mathcal{U})$ until sum of degrees is $\epsilon n^2/k$, we have that the size of $\widehat{Z}_i(\mathcal{U}_i)$ will only be more than the size of $S \cap V_i$ and satisfies $|\widehat{Z}_i(\mathcal{U})| \geq |S \cap V_i|$. We know that every node in V_i has cost in the range $[(1+\gamma)^{i-1}, (1+\gamma)^i)$, therefore, we have

$$C(\widehat{Z}_i(\mathcal{U})) \geq \frac{1}{(1+\gamma)} C(S \cap V_i)$$

$$\sum_{i\in F_{(\gamma,\rho)}} C(\widehat{Z}_i(\mathcal{U})) \geq \frac{1}{(1+\gamma)} \sum_{i\in F_{(\gamma,\rho)}} C(S\cap V_i)$$

From Claim 22, we know

$$\begin{aligned}\sum_{i\in F_{(\gamma,\rho)}} C(\widehat{Z}_i(\mathcal{U})) &\geq \frac{1}{(1+\gamma)} C(S') \\ &\geq (1-\gamma)(1-2\gamma)\rho C(V) \\ C(\widehat{Z}(\mathcal{U})) &\geq \rho(1-3\gamma)C(V).\end{aligned}$$

■

Lemma 25 *Let G contain an independent set of cost $\rho C(V)$, then, Algorithm 3 returns a set of nodes $\widehat{Z}(\mathcal{U})$ such that $C(\widehat{Z}(\mathcal{U})) \geq \rho(1-3\gamma)C(V)$ and $|E[\widehat{Z}(\mathcal{U})]| \leq \epsilon n^2$ with probability $1-\delta$ and runs in time $O\left(n^2 \exp\left(O\left(\frac{k^2}{\epsilon}\log\frac{1}{\gamma\epsilon}\log\frac{k}{\epsilon\delta}\right)\right)\right)$.*

Proof As our Algorithm 3 selects nodes from each partition V_i such that the total degree of nodes in $\widehat{Z}_i(\mathcal{U})$ in the graph induced by $E[Z(\mathcal{U})]$ is at most $\epsilon n^2/k$. Therefore, total degree of nodes in $\widehat{Z}(\mathcal{U}) = \bigcup_{i\in F_{(\gamma,\rho)}} \widehat{Z}_i(\mathcal{U})$ is at most $k\cdot\epsilon n^2/k$. Hence, $|E[\widehat{Z}(\mathcal{U})]| \leq \epsilon n^2$. From Lemma 24, we have $C(\widehat{Z}(\mathcal{U})) \geq \rho(1-3\gamma)C(V)$.

In Algorithm 3, we iterate over all subsets of the partitions $\{V_1, V_2, \cdots V_k\}$. Consider a subset $\{V_1, V_2, \cdots V_\tau\}$ and in each partition, we iterate over all subsets U_i of size $\gamma\rho t/2$. Therefore, total number of subsets $\mathcal{U}$ formed from the union of subsets in each partition $\cup_{i=1}^{\tau} U_i$ is given by $\binom{t}{\gamma\rho t/2}^{\tau}$. Using $t = O(\frac{k\log k/\epsilon\delta}{\rho\gamma\epsilon})$ and $\rho \geq \sqrt{\epsilon}$, we have that the total number of iterations is at most

$$\begin{aligned}2^k \cdot \binom{t}{\gamma\rho t/2}^k &\leq 2^k \cdot \left(\frac{2te}{\gamma\rho t}\right)^{\gamma\rho tk/2} \\ &\leq 2^k \cdot \left(\frac{6}{\gamma\rho}\right)^{\gamma\rho tk/2} \leq \exp\left(O\left(\frac{k^2}{\epsilon}\log\frac{1}{\gamma\epsilon}\log\frac{k}{\epsilon\delta}\right)\right).\end{aligned}$$

In each iteration, we can find $Z(\mathcal{U})$ in $O(|\mathcal{U}|n)$ time. After that, we calculate the degree of nodes in $Z(\mathcal{U})\cap V_i$ in the induced sub-graph $E[Z(\mathcal{U})]$ which requires $O(|Z(\mathcal{U})|^2) = O(n^2)$ running time. Hence, the claim. ■

Lemma 26 *Suppose S^* denotes MIS in $G(V,E)$. Algorithm 2 returns a set of nodes S such that $C(S) \geq C(S^*)$, $|S| \geq \sqrt{\epsilon}n$ and $|E[S]| \leq \epsilon n^2$ with probability $1-\delta$ and runs in time*

$$O\left(\frac{n^2 W}{\epsilon}\log\frac{1}{\epsilon}\exp\left(O\left(\frac{W^2\log^2 W}{\epsilon^3}\log\frac{W}{\epsilon}\log\frac{W\log W\log 1/\epsilon}{\epsilon\delta}\right)\right)\right).$$

Proof

Let T denote the set of $\sqrt{\epsilon}n$ nodes from V with highest cost. It is easy to observe that $C(T) \geq \sqrt{\epsilon}\, C(V)$. If $C(S^*) < C(T)$, then, Algorithm 2 outputs the set T. Therefore,

$$C(T) > C(S^*) \text{ and } |E[T]| \leq (\sqrt{\epsilon}n)^2 = \epsilon n^2.$$

Otherwise, in Algorithm 2, we search for MIS with cost $\rho C(V)$ using decreasing powers of $(1+\gamma)$ with the help of the parameter ρ when $\rho \geq \sqrt{\epsilon}$. If $C(S^*) \geq C(T)$, then, $|S^*| \geq |T| = \sqrt{\epsilon}n$ and for some $1 \leq j \leq \frac{1}{2\gamma}\log\frac{1}{\epsilon}$ and $\rho = \frac{1}{(1+\gamma)^j}$ (i.e., $\sqrt{\epsilon} \leq \rho \leq 1$) we have

$$\rho C(V) \leq C(S^*) \leq \rho(1+\gamma)C(V).$$

For this value of ρ, Algorithm 3 returns a set of nodes S such that $|E[S]| \leq \frac{\epsilon}{8W}n^2$. We observe that

$$\begin{aligned} C(S) &\geq \frac{1}{(1+\gamma)^j}(1-3\gamma)C(V) \\ &\geq \frac{1-3\gamma}{1+\gamma}\frac{C(V)}{(1+\gamma)^{j-1}} \geq (1-4\gamma)C(S^*). \end{aligned}$$

In our call to the Algorithm 3 from Algorithm NEAR-MIS, we set $\gamma = \frac{\epsilon}{8W}$.

$$\begin{aligned} C(S_{\epsilon/2}) &\geq \frac{\epsilon n}{2} \quad \text{(since, cost of a node is at least 1)} \\ \Rightarrow C(S \cup S_{\epsilon/2}) &\geq C(S^*) + \frac{\epsilon n}{2} - \frac{\epsilon}{2W}C(S^*) \\ &\geq C(S^*) + \frac{\epsilon n}{2} - \frac{\epsilon}{2W}n \cdot W \geq C(S^*). \end{aligned}$$

As every node in $S_{\epsilon/2}$ has degree at most n, we have

$$|E[S \cup S_{\epsilon/2}]| \leq \frac{\epsilon n^2}{8W} + \frac{\epsilon n^2}{2} \leq \epsilon n^2.$$

As $C(S \cup S_{\epsilon/2}) \geq C(T)$ where T contains the $\sqrt{\epsilon}n$ highest cost nodes, we have $|S \cup S_{\epsilon/2}| \geq \sqrt{\epsilon}n$. When $C(S^*) \geq C(T)$, we search for the correct value of ρ and for each guess, we call the routine Algorithm 3. In total, the number of calls that are made to Algorithm 3 is at most $\frac{1}{2\gamma}\log\frac{1}{\epsilon}$. However, in each call to Algorithm 3, we fail to output with probability δ'. As we set the failure probability to $\delta' = 2\gamma\delta/\log(1/\epsilon)$, overall the iterations, using union bound, the failure probability is at most $\delta' \cdot \frac{1}{2\gamma}\log\frac{1}{\epsilon} = \delta$.

From Lemma 25, Algorithm 3 runs in time $O\left(n^2 \exp\left(O\left(\frac{k^2}{\epsilon}\log\frac{1}{\gamma\epsilon}\log\frac{k}{\epsilon\delta'}\right)\right)\right)$. Substituting $k = \gamma^{-1}\log W, \delta', \gamma = \frac{\epsilon}{8W}$ and for a total of $\frac{1}{2\gamma}\log\frac{1}{\epsilon}$ calls to Algorithm 3, the running time of Algorithm 2 is

$$O\left(\frac{n^2 W}{\epsilon}\log\frac{1}{\epsilon}\exp\left(O\left(\frac{W^2\log^2 W}{\epsilon^3}\log\frac{W}{\epsilon}\log\frac{W\log W\log 1/\epsilon}{\epsilon\delta}\right)\right)\right).$$

■

From Lemma 26, we have that in each iteration, Algorithm 2 returns a set of nodes S that have a cost $C(S) \geq C(S^*)$ where S^* is the maximum independent set in G. In a previous work (Lindgren

et al., 2018), it was shown that by using maximum independent set in each iteration, we obtain a $(2 + \exp(-\Omega(m))$-optimal separating set system. Following the exact same analysis, gives us an approximation factor close to 2. We refer the reader to the analysis in Appendix F, and give the main statement of the Lemma below.

Lemma 27 *For any $m \geq \eta \log 1/\epsilon$ for some constant $\eta > 2$, with probability $\geq 1 - \delta$, Algorithm 1 returns L_ϵ with $C(L_\epsilon) \leq (2 + \exp(-\Omega(m))) \cdot C(L^*)$, where L^* is the min-cost separating matrix for G*

Scaling Parameters. In Algorithm 1, we pass a scaled value of ϵ by setting it to ϵ^2 when we call Algorithm 2, as this ensures that total number of edges returned over $\frac{1}{\epsilon}$ calls is at most ϵn^2. We also set the failure probability for each call as $\epsilon\delta$, to ensure that over $\frac{1}{\epsilon}$ calls, total failure probability using union bound is at most δ.

Theorem 28 *(Theorem 13 restated) For any $m \geq \eta \log 1/\epsilon$ for some constant η, with probability $\geq 1 - \delta$, Algorithm 1 returns L_ϵ with $C(L_\epsilon) \leq (2 + \exp(-\Omega(m))) \cdot C(L^*)$, where L^* is the min-cost separating matrix for G. Moreover L_ϵ ϵ-separates G. Algorithm 1 has a running time $O(n^2 f(W, \epsilon, \delta))$ where $f(W, \epsilon, \delta) = O\left(\frac{n^2 W}{\epsilon^2} \log \frac{1}{\epsilon} \exp\left(O\left(\frac{W^2 \log^2 W}{\epsilon^6} \log \frac{W}{\epsilon} \log \frac{W \log W \log 1/\epsilon}{\epsilon\delta}\right)\right)\right)$.*

Proof Using all the above scaled parameters, from Lemma 26, the running time of Algorithm 1 that internally calls Algorithm 2 for $\frac{1}{\epsilon}$ number of times, is given by

$$O\left(\frac{n^2 W}{\epsilon^2} \log \frac{1}{\epsilon} \exp\left(O\left(\frac{W^2 \log^2 W}{\epsilon^6} \log \frac{W}{\epsilon} \log \frac{W \log W \log 1/\epsilon}{\epsilon\delta}\right)\right)\right).$$

From Lemma 27, we have the approximation guarantee.

■

Remark. Observe that our running time is exponential in $1/\epsilon$ and therefore setting $\epsilon < 1/n^2$ to get a separating system with all edges separated requires exponential running time. As we have argued that finding such a set system with near optimal cost is hard conditioned on the hardness of approximate coloring (Theorem 6), it is thus also conditionally hard to improve our runtime to be polynomial in $1/\epsilon$. It is an interesting open question to study the parameterized hardness beyond polynomial factors with respect to ϵ.

By Theorem 13 with $m = O(\log(1/\epsilon))$ interventions we can ϵ-approximately learn any causal graph G. For learning the entire graph G, $m \geq \log \chi$ interventions are necessary, where χ is the chromatic number of G, since the rows of $L \in \{0, 1\}^{n \times m}$ must be a valid coloring of G (Lindgren et al., 2018).

Appendix D. Missing Details From Section 5

In this section, we say a pair of nodes (v_i, v_j) share an ancestral relation, if v_i has a directed path to v_j (v_i is an ancestor of v_j) or v_j has a directed path to v_i (v_j is an ancestor of v_i).

Lemma 29 *Suppose $\mathcal{S} = \{S_1, S_2, \cdots, S_m\}$ is a collection of subsets of V. If $\text{Anc}(G) \cap H$ is recovered from H using conditional independence tests by intervening on the sets $S_i \in \mathcal{S}$. Then, under the assumptions of section 2, $\mathcal{S}$ is a strongly separating set system on H.*

Proof First, we argue that to recover $\mathrm{Anc}(G) \cap H$ it is sufficient that $\mathcal{S}$ is a strongly separating set system on H. Suppose $(v_i, v_j) \in H$ and v_i, v_j share an ancestral relation i.e., either v_i is an ancestor of v_j or v_j is an ancestor of v_i. Therefore, $v_i \not\perp\!\!\!\perp v_j$ and $(v_i, v_j) \in \mathrm{Anc}(G)$. From Lemma 1 in Kocaoglu et al. (2017b), we know that, we can recover the ancestral relation between v_i and v_j using conditional independence tests (or CI-tests) on interventional distributions that strongly separate the two variables v_i and v_j. As $\mathcal{S}$ is a strongly separating set system for H, we can recover all ancestral relations in $\mathrm{Anc}(G) \cap H$.

Now, we show that a strongly separating set system on H is necessary. Here, we give a proof similar to Lemma A.1 from Addanki et al. (2020). Suppose $\mathcal{S}$ is not a strongly separating set system for H. If there exists a pair of nodes containing an ancestral relation, say $(v_i, v_j) \in H \cap \mathrm{Anc}(G)$ such that every set $S_k \in \mathcal{S}$ contains none of them, then, we cannot recover the ancestral relation between these two nodes as we are not intervening on either v_i or v_j and the results of an independence test $v_i \perp\!\!\!\perp v_j$ might result in a wrong inference, possibly due to the presence of a latent variable l_{ij} between them. Consider the case when only one of them is present in every set of $\mathcal{S}$. Let $\mathcal{S}$ be such that $\forall S_k \in \mathcal{S} : S_k \cap \{v_i, v_j\} = \{v_i\} \Rightarrow v_i \in S_k, v_j \notin S_k$. We choose our graph G to have two components $\{v_i, v_j\}$ and $V \setminus \{v_i, v_j\}$; and include the edge $v_j \to v_i$ in it. Observe that $v_i \not\perp\!\!\!\perp v_j$. Our algorithm will conclude from the CI-test $v_i \perp\!\!\!\perp v_j \mid \mathrm{do}(S_k)$? that v_i and v_j are independent. However, it is possible that $v_i \not\perp\!\!\!\perp v_j$ because of a latent l_{ij} between v_i and v_j, but when we do CI-test, we get $v_i \perp\!\!\!\perp v_j \mid \mathrm{do}(S_k)$ as intervening on v_i disconnects the $l_{ij} \to v_i$ edge. Therefore, our algorithm cannot distinguish the two cases $v_j \to v_i$ and $v_i \leftarrow l_{ij} \to v_j$ without intervening on v_j. For every $\mathcal{S}$ that is not a strongly separating set system on H, we can provide a graph G such that by intervening on sets in $\mathcal{S}$, we cannot recover $\mathrm{Anc}(G) \cap H$ from H correctly. ■

Lemma 30 *(Lemma 14 restated) Under the assumptions of Section 2, if $\mathcal{S} = \{S_1, S_2, \cdots S_m\}$ is an ϵ-strongly separating set system for H, $\mathcal{S}$ suffices to ϵ-approximately learn $\mathrm{Anc}(G) \cap H$.*

Proof Given $\mathcal{S}$ denotes an ϵ-strongly separating set system for H. So, there are at most ϵn^2 edges $(u, v) \in H$ such that for all $i \in [m]$, either $\{u, v\} \cap S_i = \phi$ or $\{u, v\} \cap S_i = \{u, v\}$ or $\{u, v\} \cap S_i = \{u\}$ (without loss of generality). For every such edge, any intervention on a set in $\mathcal{S}$, say S_i cannot recover the direction from a conditional independence test(CI-test) $u \perp\!\!\!\perp v \mid \mathrm{do}(S_i)$ because for both the cases $u \leftarrow v$ or $u \leftarrow l_{uv} \to v$, where l_{uv} is a latent, the CI-test returns that they are independent. Therefore, we cannot recover the ancestral relation (if one exists) between u, v that are not strongly separated in H. For edges $(u, v) \in H$ that are strongly separated using S_i and S_j, we can recover the ancestral relation using CI-tests $u \perp\!\!\!\perp v \mid \mathrm{do}(S_i)$ and $u \perp\!\!\!\perp v \mid \mathrm{do}(S_j)$. From Def. 1, we have that $\mathcal{S}$ ϵ-approximately learns G. ■

Algorithm SSMATRIX from Addanki et al. (2020) gives a 2-approximation guarantee for the output strongly separating matrix. However, we cannot directly extend the arguments as the guarantee holds when the input graph is complete. We get around this limitation, and show that Algorithm 4 achieves a close to 4-approximation, by relating the cost of ϵ-strongly separating matrix returned by SSMATRIX on the *supernode* set V_S, to the cost of 2-approximate ϵ-separating matrix that we find using Algorithm 1.

Let ALG_S denote the cost of the objective $\sum_{j=1}^{n} C(v_j) \|L_\epsilon^S(j)\|_1$ obtained by Algorithm 1 where L_ϵ^S is an ϵ-separating matrix; ALG_{SS} denote the cost of the objective $\sum_{j=1}^{n} C(v_j) \|L_\epsilon^{SS}(j)\|_1$ obtained by Algorithm 4 where L_ϵ^{SS} is an ϵ-strongly separating matrix. For the sake of analysis,

during assignment of vectors to nodes in L_ϵ^S, we assume that Algorithm 1 only allows vectors of weight at least 1. As SSMATRIX algorithm from Addanki et al. (2020) assigns vectors with weight atleast 1 (otherwise it will not be a valid strongly separating matrix), this assumption for ALG_S helps us in showing a relation between the costs of L_ϵ^{SS} and L_ϵ^S. As that is not sufficient to obtain the claimed guarantee, instead of assigning $\binom{m}{1}$ vectors of weight 1, we constraint it to a fixed number $r \leq \binom{m}{1}$. In Addanki et al. (2020), Algorithm SSMATRIX assigns vectors to L_ϵ^{SS} by guessing the exact number of weight 1 vectors in OPT_{SS}, the parameter r corresponds to this guess.

Let OPT_{SS} and OPT_S denote optimum objective values associated with strongly separating and separating matrices for a graph H. Let $\mathrm{ALG}_{SS}(r)$ denote the cost $C(L_\epsilon^{SS})$ assuming first r columns are used for exactly r weight 1 vectors during the assignment in L_ϵ^{SS}, and the remaining $m - r$ columns are used for all the remaining vector assignments. Similarly, $\mathrm{ALG}_S(r)$, $\mathrm{OPT}_S(r)$ and $\mathrm{OPT}_{SS}(r)$ are defined.

Lemma 31 $\mathrm{OPT}_S(r) \leq \mathrm{OPT}_{SS}(r)$ *for any* $r \geq 0$.

Proof Observe that any strongly separating matrix for H is also a separating matrix for H. Now, consider a strongly separating matrix that achieves cost $\mathrm{OPT}_{SS}(r)$ using r weight 1 vectors, then, we have

$$\mathrm{OPT}_S(r) \leq \mathrm{OPT}_{SS}(r).$$

■

Lemma 32 $C(L_\epsilon^{SS}) \leq (4 + \gamma + \exp(-\Omega(m))) \cdot \mathrm{OPT}_{SS}$.

Proof First, we note that in *any* strongly separating matrix, for the *non-dominating* property to hold, the support of weight 1 vectors and the support of vectors of weight > 1 are column disjoint. Suppose a_1^* denote the number of columns of m that are used by OPT_{SS} for weight 1 vectors i.e, $\mathrm{OPT}_{SS}(a_1^*) = \mathrm{OPT}_{SS}$.

Following the exact proof of Lemma A.5 in Addanki et al. (2020) gives us the following guarantee about Algorithm 4

$$C(L_\epsilon^{SS}) \leq 2\,\mathrm{ALG}_S(a_1^*).$$

From Theorem 13 and Lemma 27 in Appendix F (or the analysis from Lindgren et al. (2018)) :

$$\mathrm{ALG}_S(a_1^*) \leq (2 + \exp(-\Omega(m)))\,\mathrm{OPT}_S(a_1^*).$$

From Lemma 31, we know $\mathrm{OPT}_S(a_1^*) \leq \mathrm{OPT}_{SS}(a_1^*)$. Therefore, we have

$$\begin{aligned}\mathrm{ALG}_S(a_1^*) &\leq (2 + \exp(-\Omega(m)))\,\mathrm{OPT}_{SS}(a_1^*) \\ &= (2 + \exp(-\Omega(m)))\,\mathrm{OPT}_{SS}\,.\end{aligned}$$

Hence, the lemma. ■

Theorem 33 *(Theorem 15 restated) Let $m \geq \eta \log 1/\epsilon$ for some constant η and L_ϵ^{SS} be matrix returned by Algorithm 4. Then with probability $\geq 1-\delta$, L_ϵ^{SS} is an ϵ-strongly separating matrix for H and $C(L_\epsilon^{SS}) \leq (4+\exp(-\Omega(m))) \cdot C(L^*)$ where L^* is the min-cost strongly separating matrix for H. Algorithm 4 runs in time $O(n^2 f(W,\epsilon,\delta))$ where*

$$f(W,\epsilon,\delta) = O\left(\frac{n^2 W}{\epsilon^2}\log\frac{1}{\epsilon}\exp\left(O\left(\frac{W^2\log^2 W}{\epsilon^6}\log\frac{W}{\epsilon}\log\frac{W\log W\log 1/\epsilon}{\epsilon\delta}\right)\right)\right).$$

Proof From Lemma 32, we have $C(L_\epsilon^{SS}) \leq (4+\exp(-\Omega(m)))C(L^*)$. The sets of nodes $S_1, S_2, \cdots S_{1/\epsilon}$ returned by Algorithm 4 are such that every set S_i contains at most $\epsilon^2 n^2$ edges with probability $1-\delta$. Therefore, in total at most $\frac{1}{\epsilon}\epsilon^2 n^2 \leq \epsilon n^2$ edges do not satisfy strongly separating property. As SSMATRIX has a running time of $O(|V_S|) = O(\frac{1}{\epsilon})$, and using the running time of Algorithm 1 from Theorem 28, our claim follows. ■

Appendix E. Hyperfinite Graphs : Better Guarantees

In this section, we show that when G has maximum degree Δ and satisfies hyperfinite property, we can obtain the same approximation guarantees, but the number of edges that are not (strongly) separated can be improved to $\epsilon \cdot n \cdot \Delta$. Informally, a hyperfinite graph can be partitoned into small connected components by removing $\epsilon \cdot n$ edges for every $\epsilon > 0$. Bounded degree hyperfinite graphs include the class of bounded-degree graphs with excluded minor (Alon et al., 1990), such as planar graphs, constant tree-width graphs, and also non-expanding graphs (Czumaj et al., 2009).

Definition 34 *A Graph $G(V,E)$ is (ϵ,k)-hyperfinite if there exists $E' \subseteq E$ and $|E' \setminus E| \leq \epsilon n$ such that every connected component in the induced subgraph of E' is of size at most k. A Graph G is said to be τ-hyperfinite, if there exists a function $\tau : \mathbb{R}^+ \to \mathbb{R}^+$ such that for every $\epsilon > 0$, G is $(\epsilon, \tau(\epsilon))$-hyperfinite.*

If a τ-hyperfinite graph G has maximum degree Δ, we give algorithms for (strongly) separating set systems on G that obtain the same approximation guarantees, but the number of edges that are not (strongly) separated at most $\epsilon \cdot n \cdot \Delta$. In order to obtain that, we extend the additive approximation algorithm of Hassidim et al. (2009) for finding the maximum independent set to the weighted graphs i.e., when the nodes have costs and return a NEAR-MIS instead of MIS. First, we define a very important partitioning of the nodes V possible in τ-hyperfinite graphs and give the lemma that describes the guarantees associated with finding the partitions.

Definition 35 (Hassidim et al. (2009)(ϵ,k) partitioning oracle $\mathcal{O}$) *For a given graph $G(V,E)$ and query q about $v \in V$, $\mathcal{O}$ returns the partition $P[v] \subseteq V$ containing v that satisfies :*

1. *for every node $v \in V$, $|P[v]| \leq k$ and $P[v]$ is connected*
2. *$|\{(u,w) \in E \mid P[u] \neq P[w]\}| \leq \epsilon \cdot n$ with probability $9/10$.*

Lemma 36 (Hassidim et al. (2009)) *If G is $(\epsilon,\tau(\epsilon))$-hyperfinite graph with maximum degree Δ, then, there is a $(\epsilon\cdot\Delta, \tau(\epsilon^3/54000))$ partition oracle that answers a given query q with probability $1-\delta$, using a running time $O(2^{\Delta^{O(\tau(\epsilon^3))}}/\delta \log 1/\delta)$.*

Using Lemma 36, we query every node to obtain the partitioning of V and formalize this in the following corollary.

Corollary 37 *If G is $(\epsilon, \tau(\epsilon))$-hyperfinite graph with maximum degree Δ, then, we can obtain a partitioning of the graph G, given by $V_1, V_2, \cdots$ such that with probability $1-\delta$ and a running time of $O(\frac{n}{\delta} \cdot 2^{\Delta^{O(\tau(\epsilon^3/\Delta^3))}} \log 1/\delta)$, we have :*

1. *For every $i, |V_i| \leq \tau(\epsilon^3/\Delta^3 54000)$ and V_i is connected*
2. *$|\{(u,w) \mid (u,w) \in E, u \in V_i, w \in V_j \text{ and } i \neq j\}| \leq \epsilon \cdot n$*

Given a τ-hyperfinite graph with maximum degree Δ, we describe an algorithm that returns a set of nodes that have at most $\epsilon n\Delta$ edges instead of ϵn^2 edges that we saw previously for general graphs G. To do so, we build upon the previous result from Hassidim et al. (2009) that returns a set of nodes R which is an additive ϵn approximation of MIS S^*, i.e., $|R| \geq |S^*| - \epsilon n$. To obtain this, the authors first use the partitioning obtained using Lemma 36 and find MIS in each partition separately. They show that ignoring the nodes that are incident on edges across the partitions obtained using Lemma 36 will only result in a loss of ϵn nodes. We observe that in Algorithm 5, by removing the $\epsilon \cdot n$ nodes (denoted by $\widehat{V}$) that are incident on the edges across partitions, and adding back ϵn nodes with highest cost, we will obtain a set of nodes with cost at least that of MIS while only adding $\epsilon n\Delta$ edges amongst the combined set of nodes.

Algorithm 5 NEAR-MIS in τ-Hyperfinite Graph G

Input : Graph $G = (V, E)$, cost function $C : V \to \mathbb{R}^+$, m, Δ, function $\tau(\cdot)$, error ϵ, failure probability δ.
Output : T that is a NEAR-MIS with at most $\epsilon \cdot n \cdot \Delta$ edges.
Let the set of partitions is $\{V_1, V_2 \cdots\}$ of $G(V, E)$ returned using Corollary 37 with parameters $\tau(\cdot)$, error $\epsilon/2$ and failure probability δ.
for each partition V_i **do**
    Calculate the maximum *cost* independent set T_i in V_i.
end for
$\widehat{E} \leftarrow \{(u,v) \mid (u,v) \in E$ and there exists i, j where $i \neq j, u \in V_i, v \in V_j\}$.
$\widehat{V} \leftarrow \{u \mid \exists v$ such that $(u,v) \in \widehat{E}\}$.
$T \leftarrow (\bigcup_{i=1} T_i) \setminus \widehat{V}$.
Let H denote $\epsilon \cdot n$ nodes of highest cost in $V \setminus T$.
return $T \cup H$.

Lemma 38 *In Algorithm 5, we have $|\widehat{V}| \leq \epsilon \cdot n$ and $C(T \cup H) \geq C(S^*)$.*

Proof From Corollary 37, we have $|\widehat{E}| \leq \epsilon \cdot n/2$. From the definition of $\widehat{V}$, we have

$$|\widehat{V}| \leq 2|\widehat{E}| \leq \epsilon \cdot n.$$

Suppose S^* is the maximum cost independent set in G. Now, consider all nodes in $\widehat{V}$. Similar to the above case, it is possible that $(u,w) \in \widehat{E}$ and $u \in T_i$, $w \in T_j$ for some $i \neq j$. Consider a node $u \in S$ that is isolated in $E' \subseteq E$, and included in some partition V_i. As T_i is maximum cost

independent set in V_i, we have $C(T_i) \geq C(S^* \cap V_i)$ where $S^* \cap V_i$ is an independent set induced by MIS S^* in the partition V_i. Combining it for all partitions, we have $C(\bigcup_i T_i) \geq C(S^*)$. As nodes in $\widehat{V}$, it is possible that including those that share an edge in $\bigcup_i T_i$ will result in the set of nodes not forming an independent set. However, the set $\bigcup_i T_i \setminus \widehat{V}$ formed by removing all the nodes that are incident with edges across the partitions, is an independent set. Since S^* is MIS, we have

$$C(\bigcup_i T_i \setminus \widehat{V}) \leq C(S^*).$$

As $|\widehat{V}|$ is at most $\epsilon \cdot n$, replacing them with H consisting of $\epsilon \cdot n$ highest cost nodes from T will only increase the cost. Therefore, we have

$$\Rightarrow C(T \cup H) = C\left(\left(\bigcup_i T_i \setminus \widehat{V}\right) \cup H\right) \geq C(\bigcup_i T_i) \geq C(S^*).$$

■

Theorem 39 *Algorithm 5 returns a set $T \subseteq V$ of nodes such that $C(T) \geq C(S^*)$ where S^* is the maximum cost independent set; $|E[T]| \leq \epsilon \cdot n \cdot \Delta$ and uses a running time $O(\frac{n}{\delta} \cdot 2^{\Delta^{O(\tau(\epsilon^3/\Delta^3))}} \log 1/\delta + n\Delta)$ with probability $1 - \delta$*

Proof From Lemma 38, we have $C(T) \geq C(S^*)$ and the nodes in H include $\epsilon \cdot n$ nodes that are added (line 10 in Algorithm 2) have at most $\epsilon \cdot n \cdot \Delta$ edges among themselves. Therefore, $|E[T]| \leq \epsilon \cdot n \cdot \Delta$. Using Corollary 37, we have that it takes $O(\frac{n}{\delta} \cdot 2^{\Delta^{O(\tau(\epsilon^3/\Delta^3))}} \log 1/\delta)$ time to find the partitions. After finding the partitions, we find maximum cost independent set in each of the at most n partitions each of size $O(\tau(\epsilon^3/\Delta^3))$, which takes a running time of

$$O(\text{finding maximum cost independent set in each partition}) = O(n \cdot 2^{O(\tau(\epsilon^3/\Delta^3))}).$$

Combining the running times for both these steps, along with $O(n\Delta)$, the time to find $\widehat{E}$, we have the running time as claimed. ■

We can use Algorithm 5 to obtain NEAR-MIS in each iteration of Algorithm 1; from Lemma 27 and Theorem 39, we have the following proposition about *separating set system* for G.

Proposition 40 *Let $G(V, E)$ be a Δ-degree bounded τ-hyperfinite graph. For any $m \geq \eta \log 1/\epsilon$ for some constant η, with probability $\geq 1 - \delta$, there is an algorithm that returns L_ϵ with $C(L_\epsilon) \leq (2 + \exp(-\Omega(m))) \cdot C(L^*)$, where L^* is the min-cost separating matrix for G and has a running time $O\left(\frac{n^3}{\delta} \cdot 2^{\Delta^{O(\tau(\epsilon^3/n^3\Delta^3))}} \log \frac{n}{\delta}\right)$. Moreover using L_ϵ, the number of edges that are not separated in G is at most $\epsilon \cdot n \cdot \Delta$.*

Proof In every iteration, we identify a set of nodes that has the cost at least the cost of MIS. Therefore, total number of iterations possible is at most n. Scaling the error parameter by setting $\epsilon' = \epsilon/n$ and $\delta' = \delta/n$ for each iteration, we have that Algorithm 1 returns L_ϵ such that the number

of edges that are not separated is $n \cdot (\epsilon' \cdot n \cdot \Delta) = \epsilon \cdot n \cdot \Delta$. Using Lemma 38, we have that the total running time of our algorithm is

$$O(n \cdot \frac{n}{\delta'} \cdot 2^{\Delta^{O(\tau(\epsilon'^3/\Delta^3))}} \log 1/\delta') = O\left(\frac{n^3}{\delta} \cdot 2^{\Delta^{O(\tau(\epsilon^3/n^3\Delta^3))}} \log \frac{n}{\delta}\right).$$

■

We can obtain a similar result for *strongly separating set system* for G using Algorithm 4 and give the following proposition.

Proposition 41 *Let $G(V, E)$ be a Δ-degree bounded τ-hyperfinite graph. For any $m \geq \eta \log 1/\epsilon$ for some constant η, with probability $\geq 1 - \delta$, there is an algorithm that returns L_ϵ with $C(L_\epsilon) \leq (4 + \exp(-\Omega(m))) \cdot C(L^*)$, where L^* is the min-cost* strongly *separating matrix for G and has a running time $O\left(\frac{n^3}{\delta} \cdot 2^{\Delta^{O(\tau(\epsilon^3/n^3\Delta^3))}} \log \frac{n}{\delta}\right)$. Moreover using L_ϵ, the number of edges that are not* strongly *separated in G is at most $\epsilon \cdot n \cdot \Delta$.*

Proof In Algorithm 4, we first find L_ϵ^S, a separating matrix obtained using Proposition 40 that does not separate $\epsilon \cdot n \cdot \Delta$ edges of G. Next, we find *super nodes* using the NEAR-MIS's returned and assign it vectors appropriately to form strongly separating matrix L_ϵ^{SS} on super nodes. Using Theorem 33, we have the claimed approximation guarantee. The running time follows from Proposition 40. ■

Appendix F. Additional Details for the analysis of 2-approximation result for ϵ-Separating Set Systems

In this section, we present already known results from Lindgren et al. (2018) filling in the details in the analysis of our Algorithm 1 for the sake of completion.
Let $\mathcal{I}$ denote the set of all independent sets in G. For some $\mathcal{A} \subseteq \mathcal{I}$, we have

$$J(\mathcal{A}) = \sum_{v \in \bigcup_{S \in \mathcal{A}}} C(v)$$

that is, it takes a set of independent sets and returns the sum of the cost of the vertices in their union. We observe that J is submodular, monotone, and non-negative (Lindgren et al., 2018).

Let S_0 denote the set of nodes that are assigned weight 0 vector after the first iteration of Algorithm 1. We set $V = V \setminus S_0$ for the remainder of this section and handle the cost contribution of nodes in S_0 separately in the analysis of approximation ratio.

Lemma 42 *Given a submodular, monotone, and non-negative function J over a ground set V and a cardinality constraint k. Let Algorithm 1 return S_{greedy} a collection of at most Ck (for some constant $C > 0$) sets that are $(0, \epsilon)$-*NEAR-MIS*, then,*

$$J(S_{greedy}) \geq (1 - e^{-C}) \max_{\mathcal{S} \subseteq \mathcal{I}, |\mathcal{S}| \leq k} J(\mathcal{S}).$$

Proof We have that in ith iteration of Algorithm 1, we pick a set of nodes S_i with cost at least the cost of MIS T_i in G, i.e., S_i is a $(0,\epsilon)$-NEAR-MIS in G and satisfies $C(S_i) \geq C(T_i)$. Let $\mathcal{S}^*$ be the collection of independent sets such that $J(\mathcal{S}^*) = \max_{\mathcal{S}\subseteq\mathcal{I},|\mathcal{S}|\leq k} J(\mathcal{S})$. Let $\bigcup_{j\leq i} S_j$ be denoted by $S_{1:i}$. We claim using induction that

$$J(\mathcal{S}^*) - J(S_{1:i}) \leq \left(1 - \frac{1}{k}\right)^i J(\mathcal{S}^*).$$

Consider ith iteration when Algorithm 1 picks S_i. Using submodularity, we have

$$J(\mathcal{S}^*) - J(S_{1:i-1}) \leq \sum_{B\in\mathcal{S}^*\setminus S_{1:i-1}} J(S_{1:i-1}\cup B).$$

Therefore, there exists one set $B \in \mathcal{S}^* \setminus S_{1:i-1}$, with cost at least $\frac{\sum_{B\in\mathcal{S}^*\setminus S_{1:i-1}} J(S_{1:i-1}\cup B)}{k}$. As argued in Lemma 26, we are picking a set S_i with cost $C(S_i) \geq C(T_i)$ where T_i is MIS in the ith iteration, we have :

$$C(S_i) \geq C(T_i) \geq \frac{\sum_{B\in\mathcal{S}^*\setminus S_{1:i-1}} J(S_{1:i-1}\cup B)}{k} \geq \frac{J(\mathcal{S}^*) - J(S_{1:i-1})}{k}$$
$$J(S_i) = C(S_i) \geq \frac{J(\mathcal{S}^*) - J(S_{1:i-1})}{k}.$$

For $i = 1$, our claim follows from the above statement, i.e., $C(S_1) = J(S_1) \geq \frac{J(\mathcal{S}^*)}{k}$. Assuming that our claim holds until iteration $i-1$ for some $i \geq 2$, we have after the ith iteration : $J(\mathcal{S}^*) - J(S_{1:i}) = J(\mathcal{S}^*) - J(S_{1:i-1}) - J(S_i)$. This is true because $J(S_{1:i}) = J(S_{1:i-1}) + J(S_i)$ as S_i is greedily chosen by picking a set containing nodes that are not previously selected. Therefore,

$$\begin{aligned} J(\mathcal{S}^*) - J(S_{1:i}) &= J(\mathcal{S}^*) - J(S_{1:i-1}) - J(S_i) \\ &\leq J(\mathcal{S}^*) - J(S_{1:i-1}) - \frac{J(\mathcal{S}^*) - J(S_{1:i-1})}{k} \\ &\leq (J(\mathcal{S}^*) - J(S_{1:i-1}))\left(1 - \frac{1}{k}\right) \\ &\leq \left(1 - \frac{1}{k}\right)^i J(\mathcal{S}^*). \end{aligned}$$

Setting $i = C\cdot k$, we have

$$J(\mathcal{S}^*) - J(S_{\text{greedy}}) \leq \left(1 - \frac{1}{k}\right)^{Ck} J(\mathcal{S}^*) \leq e^{-C}\cdot J(\mathcal{S}^*)$$
$$\Rightarrow J(S_{\text{greedy}}) \geq (1 - e^{-C}) J(\mathcal{S}^*).$$

■

Now, we define two types of submodular optimization problem, called the submodular chain problem and the supermodular chain problem that will be useful later.

Definition 43 *Given integers $k_1, k_2, \ldots, k_m$ and a submodular, monotone, and non-negative function J, over a ground set V, the submodular chain problem is to find sets $A_1, A_2, \ldots, A_m \subseteq 2^{[V]}$ such that $|A_i| \leq k_i$ that maximizes*

$$\sum_{i=1}^{m} J(A_1 \cup A_2, \cup \cdots \cup A_i).$$

Lemma 44 *Let $A_1^*, A_2^*, \ldots, A_m^*$ be the optimal solution to the submodular chain problem. Suppose that for all $1 \leq p \leq m/2 - 1$ we have that $\sum_{i=1}^{2p} k_i \geq \tau \sum_{i=1}^{p} k_i$. Also assume that $J(A_1 \cup A_2 \cup \cdots \cup A_m) = J(V)$. Then the greedy algorithm 1 for the submodular chain problem returns set $A_1, A_2, \ldots, A_m$ such that*

$$\sum_{i=1}^{m} J(A_1 \cup A_2 \cdots A_i) \geq J(V) + 2(1 - e^{-\tau}) \sum_{i=1}^{m/2-1} J(A_1^* \cup A_2^* \cup \cdots A_i^*).$$

Proof Given $\sum_{i=1}^{2p} k_i \geq \tau \sum_{i=1}^{p} k_i$. From Lemma 42, we have

$$J(A_1 \cup A_2 \cup \cdots A_{2p}) \geq (1 - e^{-\tau}) J(A_1^* \cup A_2^* \cup \cdots \cup A_p^*).$$

$$\sum_{i=1}^{m/2-1} J(A_1 \cup A_2 \cdots A_{2i}) \geq (1 - e^{-\tau}) \sum_{i=1}^{m/2-1} J(A_1^* \cup A_2^* \cup \cdots \cup A_i^*).$$

Now, we use the monotonicity property of the submodular function J to get

$$\begin{aligned}\sum_{i=1}^{m} J(A_1 \cup A_2 \cdots A_i) &= J(A_1 \cup A_2 \cdots \cup A_m) + \sum_{i=1}^{m/2-1} J(A_1 \cup A_2 \cup \cdots \cup A_{2i}) + J(A_1 \cup A_2 \cup \cdots \cup A_{2i+1}) \\ &\geq J(V) + 2 \sum_{i=1}^{m/2-1} J(A_1 \cup A_2 \cup \cdots \cup A_{2i}).\end{aligned}$$

Hence, the lemma. ■

Definition 45 *Given integers $k_1, k_2, \ldots, k_m$ and a submodular, monotone, and non-negative function F, over a ground set V, the supermodular chain problem is to find sets $A_1, A_2, \ldots, A_m \subseteq 2^{[V]}$ such that $|A_i| \leq k_i$ that minimizes*

$$\sum_{i=1}^{m} J(V) - J(A_1 \cup A_2 \cup \cdots \cup A_i).$$

For the greedy algorithm 1, we give the following claim for the supermodular chain problem.

Lemma 46 *Let $A_1^*, A_2^*, \ldots, A_m^*$ be the optimal solution to the supermodular chain problem. Suppose that for all $1 \leq p \leq m/2 - 1$ we have that $\sum_{i=1}^{2p} k_i \geq \tau \sum_{i=1}^{p} k_i$. Also assume that $J(A_1 \cup A_2 \cup \cdots \cup A_m) = J(V)$. Then the greedy algorithm 1 for the supermodular chain problem returns set $A_1, A_2, \ldots, A_m$ such that*

$$\sum_{i=1}^{m} J(V) - J(A_1 \cup A_2 \cup \cdots \cup A_i) \leq e^{-\tau} m \cdot J(V) + 2 \sum_{i=1}^{m} J(A_1^* \cup A_2^* \cup \cdots A_i^*).$$

Proof From Lemma 44, we have

$$\begin{aligned}(m+1)J(V) - \sum_{i=1}^{m} J(A_1 \cup A_2 \cup \cdots \cup A_i) &\leq mJ(V) - 2(1-e^{-\tau}) \sum_{i=1}^{m/2-1} J(A_1^* \cup A_2^* \cup \cdots A_i^*) \\ &\leq e^{-\tau} mJ(V) + mJ(V) - 2 \sum_{i=1}^{m/2-1} J(A_1^* \cup A_2^* \cup \cdots A_i^*) \\ &\leq e^{-\tau} mJ(V) + 2 \sum_{i=1}^{m/2-1} J(V) - J(A_1^* \cup A_2^* \cup \cdots A_i^*).\end{aligned}$$

Now, we use the monotonicity property of J to get

$$e^{-\tau} mJ(V) + 2 \sum_{i=1}^{m/2-1} J(V) - J(A_1^* \cup A_2^* \cup \cdots A_i^*) \leq e^{-\tau} mJ(V) + 2 \sum_{i=1}^{m} J(V) - J(A_1^* \cup A_2^* \cup \cdots A_i^*).$$

Finally, we have

$$\begin{aligned}\sum_{i=1}^{m} J(V) - J(A_1 \cup A_2 \cup \cdots \cup A_i) &\leq (m+1)J(V) - \sum_{i=1}^{m} J(A_1 \cup A_2 \cup \cdots \cup A_i) \\ &\leq e^{-\tau} m \cdot J(V) + 2 \sum_{i=1}^{m} J(V) - J(A_1^* \cup A_2^* \cup \cdots A_i^*).\end{aligned}$$

■

Lemma 47 *Suppose S^+ denote optimal separating set system that uses an additional color of weight 1 and uses weight 0 vector to color A_0. Let S^* denote optimal separating system. Then, we have $C(S^+) - C(S^*) \leq 0$.*

Proof We give a proof similar to Lemma 22 in Lindgren et al. (2018). Let the set of nodes A_0 selected in the first iteration of greedy Algorithm 1 and assigned weight 0 vector be denoted by S_0^+. Similarly, the set of nodes that are colored with weight 0 vector in S^* be denoted by S_0^*. As S^+ denotes optimal solution on $V \setminus S_0^+$, we can assume that a solution that uses a weight 1 color for nodes in $S_0^* \setminus S_0^+$ is only going to be worse. Therefore, we have :

$$\begin{aligned}C(S^+) - C(S^*) &\leq \sum_{v \in S_0^* \setminus S_0^+} C(v) - \sum_{v \in S_0^+ \setminus S_0^*} C(v) \\ &\leq \sum_{v \in S_0^* \setminus S_0^+} C(v) + \sum_{v \in S_0^* \cap S_0^+} C(v) - \sum_{v \in S_0^+ \setminus S_0^*} C(v) - \sum_{v \in S_0^* \cap S_0^+} C(v) \\ &\leq \sum_{v \in S_0^*} C(v) - \sum_{v \in S_0^+} C(v) \leq 0.\end{aligned}$$

■

Let L_ϵ denote the ϵ-separating matrix returned by Algorithm 1, and let $L_\epsilon = \{A_0, A_1, A_2, \cdots A_m\}$ where we abuse the previous notation and denote A_i to represent the set of all nodes (instead of a collection of subsets of V) that have weight i assigned by L_ϵ.

$$C(L_\epsilon) = \sum_{i=1}^{m} J(V) - J(A_1 \cup A_2 \cup \cdots A_i),$$

where $|L_i| \leq \binom{m}{i}$. We observe that this cost representation corresponds to the supermodular chain problem discussed above.

Assuming $m \geq \eta \log 1/\epsilon$ for $\eta > 2$, we have that, the greedy Algorithm 1 only uses vectors of weight at most $\log 1/\epsilon$ i.e., $m/2$. In Lemma 44, each of the values $k_1, k_2 \cdots k_i \cdots k_p$ represent number of weight i vectors available and from Lemma 21 in Lindgren et al. (2018), we have that $\tau = \Omega(m)$, for every p in the range.

Lemma 48 *(Lemma 27 restated) For any $m \geq \eta \log 1/\epsilon$ for some constant $\eta > 2$, with probability $\geq 1-\delta$, Algorithm 1 returns L_ϵ with $C(L_\epsilon) \leq (2+\exp(-\Omega(m)))\cdot C(L^*)$, where L^* is the min-cost separating matrix for G.*

Proof Using the definition of S^+ from Lemma 47, we argue that $C(S^+) \geq J(V)$ as every node in V is assigned a weight 1 vector. From Lemma 46, we have

$$\begin{aligned}
\mathrm{C}(L_\epsilon) &= \sum_{i=1}^{m/2} J(V) - J(A_1 \cup A_2 \cup \cdots A_i) \\
&\leq e^{-\tau} m \cdot J(V) + 2\sum_{i=1}^{m} J(V) - J(A_1^* \cup A_2^* \cup \cdots A_i^*) \\
&\leq e^{-\tau} m \cdot C(S^+) + 2\sum_{i=1}^{m} J(V) - J(A_1^* \cup A_2^* \cup \cdots A_i^*) \\
&\leq e^{-\tau} m \cdot C(S^+) + 2C(S^*).
\end{aligned}$$

From Lemma 47, we have

$$\leq (2 + \exp(-\Omega(m))) \cdot C(S^*) = (2 + \exp(-\Omega(m))) \cdot C(L^*).$$

■