THE UNIVERSITY OF CHICAGO

CAUSAL DISCOVERY AT SCALE FOR BIOLOGICAL MECHANISM INFERENCE

A DISSERTATION SUBMITTED TO
THE FACULTY OF THE DIVISION OF THE PHYSICAL SCIENCES
IN CANDIDACY FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE

BY
ASHKA SHAH

CHICAGO, ILLINOIS
AUGUST 2026

For my family

We are all in the gutter, but some of us are looking at the stars.

– Oscar Wilde

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I want to thank my advisor, Professor Rick Stevens, for trusting me and giving me the freedom to pursue anything I wanted. And for your patience and grace with me throughout my PhD. Thank you to my committee members Professors Ian Foster and Bryon Aragam for your encouragement, advice and enthusiasm for my work. Thank you to my amazing collaborators, all of whom I also consider friends: Austin Clyde, Valerie Hayot-Sasson, Nathaniel Hudson, and Adela DePavia. Our collaborations showed me how fun research can be. Thank you to the incredibly smart and dedicated biologists I have worked with over the years, and especially those at Argonne National Laboratory; you reminded me why I love interdisciplinary science so much, and you humbled me with the very real, tangential challenges of experimental biology which we computational people often forget!

I would be remiss if I did not mention how this has been an incredibly long and difficult process for me. Just as I was starting to feel confident in my research ability, I had a traumatic health issue that completely changed my life. I wasn't sure if it made sense for me to continue my PhD, if it was all worth it, or if I even had the physical or mental capacity to do research anymore. I am so grateful to have found the resolve to continue through my community in Chicago. Thank you to the friends who literally took care of me at my bedside, supported me at my lowest, and showed up for me in every way possible. Thank you for the girls' nights, long walks on the Lakeshore trail, and equally long venting sessions. Thank you especially to Jasmine Lu for your friendship and steadfast support. Thank you to my brother, Arnav, for paving the way, for being constantly and annoyingly curious about science. And finally thank you to my mother and father who gave up so much for me to do this and who always believed that I could do anything.

ABSTRACT

Mechanism discovery is a central objective in science. In biology, mechanisms (cascades of molecular events that lead to observable phenotypes) provide insight into cellular programming, explain disease processes, and identify targets for therapeutics. Despite advances in high-throughput experimentation that enable genome-scale measurement and the success of deep learning models in prediction, data-driven discovery of biological mechanisms remains elusive. Recently, causal discovery has emerged as a method for learning cause and effect relationships, represented by directed acyclic graphs (DAGs), from data. Given the success of networks for representing mechanistic knowledge in biology (e.g., gene regulatory networks), causal discovery algorithms are promising approaches for mechanism discovery in biology. In this thesis, I examine the current state of causal discovery and its application to biological mechanism inference. I address challenges including (1) scale, enabling inference over genome-wide systems with limited samples; (2) structural priors, incorporating biological knowledge or correlative structure to guide inference; and (3) robustness to real-world biological data characterized by heterogeneous environments, perturbations, and limited ground truth.

In our first contribution, we introduce a hybrid constraint- and score-based causal discovery algorithm, named SP-GIES, that first estimates a skeleton from observational data and then learns the DAG from interventional data over a constrained search space. We see an improvement in convergence of greedy algorithms, in both accuracy and runtime, by first estimating a structural prior. Moreover, we analyze SP-GIES for downstream optimal experimental design for choosing maximally informative perturbational experiments.

In the second contribution, we propose a distributed framework for causal discovery using our novel causal graph partition on a structural prior we call a superstructure. We prove under certain assumptions that causal discovery can be decomposed into independent subproblems by leveraging a closely related maximal ancestral graph class, which is defined by

projections of DAGs onto subsets of nodes. We scale causal discovery to 10,000 variables with our distributed framework, achieving genome-scale recovery of synthetic biological networks.

Finally, these contributions are applied to a low-sample, high-dimensional perturbational dataset to uncover cellular processes involved in response to radiation from human transcriptomics data. We compare a VAE-based causal discovery algorithm DAG-GNN, a gene regulatory network inference algorithm GENIE3, and the standard bioinformatics approach of differential expression analysis to identify important genes in response to radiation. We use bioinformatics techniques to map these genes to known pathways, and show that graph algorithms achieve higher enrichment of radiation pathways. Additionally, we find that the presence of a strong upstream perturbation induces different environments from which we can learn invariant mechanisms; in the causal graphs, these invariant edges corroborate evidence of a bifurcation of reactive oxidative stress and cell death pathways in response to radiation.

These contributions demonstrate the impact of scaling causal discovery in genome-wide studies to infer regulatory response networks and identify important genes for downstream pathway analysis. More broadly, this work aims to enable causal, testable insights from high-dimensional biological data.

CHAPTER 1

INTRODUCTION

1.1 Motivation

Causality and experimentation are intrinsically linked. The causal statement *"Her headache went away after she took a pill"* evokes the question of the counterfactual *"What would have happened if she hadn't taken the pill?"*. While the alternative universe in which she did not take the pill is impossible to realize, it can be approximated with an experiment where the next time she has a headache she does not take a pill. The result of the experiment provides evidence that it was the pill that caused her headache relief. At the population level, we can conduct a randomized control trial to infer the efficacy of the pill in treating headaches to determine if, in general, taking a pill causes headaches to go away. Underlying the causal statement about headaches is a *biological mechanism*. In the case of the widely used drug Ibuprofen, the small molecule binds to proteins COX1 and COX2 which stops the conversion of proinflammatory molecules; this leads to a reduction in pain, fever and inflammation. To validate a hypothesized mechanism we conduct molecular level perturbation experiments; for example, we can measure the abundance of proinflammatory molecules before and after introducing Ibuprofen molecules to validate the described mechanism. Biological mechanisms that connect genes, proteins and biochemical processes provide insight into cellular programming. As such, the discovery of novel mechanisms, or causal relationships, from experimental data is a central objective in biology.

Statisticians and economists have been interested in causality as a way to predict future events from past trends. The works of Judea Pearl, Peter Spirtes, Clark Glymour and others laid the ground work for a mathematical formulation of causation through functional relationships, conditional probability distributions, and graphs. The Ibuprofen mechanism described earlier can be described graphically as Ibuprofen $\mapsto$ COX1/COX2 $\mapsto$

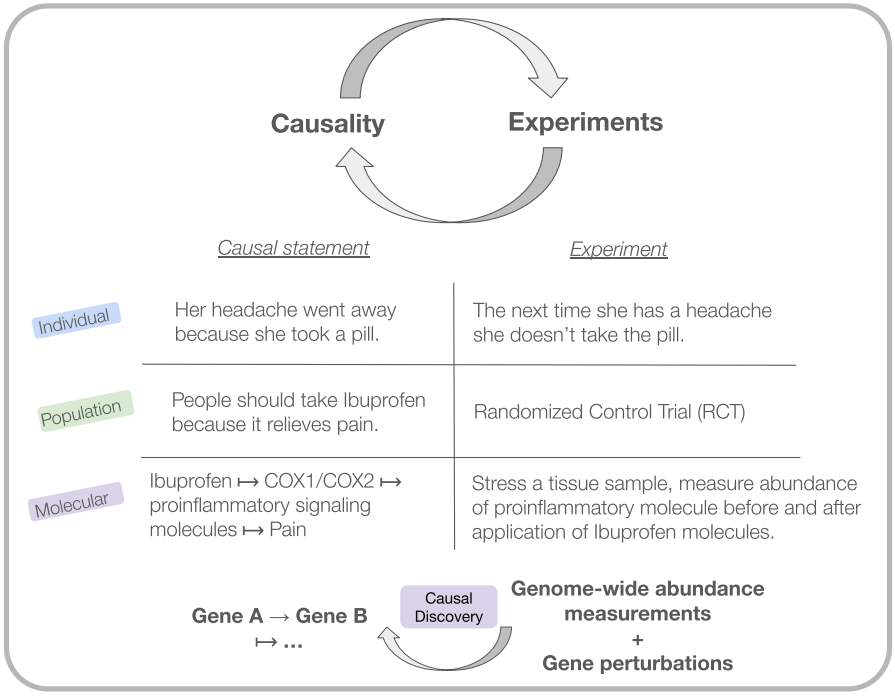


Figure 1.1: Causality and experimentation are intrinsically linked. Every causal statement can be validated or invalidated with an experiment, and every experiment encodes a causal statement. In this thesis, we use causal discovery algorithms to infer causal statements as graphs from genome-wide abundance measurements and perturbations.

proinflammatory signaling molecules $\mapsto$ pain. Inferring a graphical or functional model of a mechanism from data as a causal graph is called *causal structure learning* or *causal discovery*. Critically, a causal graph represents observational (steady-state) and interventional (perturbed/experimental) distributions. Algorithms for causal discovery were first proposed in the early 1990s, with constraint-based algorithms like the PC algorithm, while many new approaches continue to be developed today. Causal discovery algorithms, coupled with large-scale experimental data collection, have enabled data-driven mechanism discovery with applications in the social and physical sciences. In this thesis, we focus on the application of causal discovery to biological mechanism inference from large-scale gene expression data. These gene-to-gene causal relationships are represented by gene regulatory networks (GRNs) which, in humans, are only partially characterized across diverse environmental contexts, disease states, and mutational backgrounds. A complete mapping of gene regulation would represent a major milestone in biology by providing a blueprint of cellular decision-making that could inform novel therapeutics, cellular reprogramming, and personalized medicine.

1.2 Overview of Thesis

In this thesis, we aim to design a framework for generating new causal hypotheses about gene regulation. We tackle challenges associated with applying causal discovery to gene regulatory network inference on the human genome scale, with approximately 20,000 variables; namely, addressing limited scalability of causal discovery with the incorporation of structural priors. In Chapter 2 we provide background on causal graphs, causal discovery algorithms, and gene regulatory networks. We introduce directed acyclic graphs (DAGs) and Bayesian networks, and formulate the causal structure learning problem that will be used throughout the thesis. We provide context as to why gene regulatory network inference can be viewed as a causal graph, and we describe where assumptions for learning may fall short.

In Chapter 3 we introduce our first contribution, **SP-GIES**, which is a hybrid causal

structure learner that jointly learns from observational and interventional data to estimate causal graphs. We explore different experimental design strategies for selecting optimal interventions using SP-GIES.

Chapter 4 introduces our second contribution a **Causal Graph Partition** – a theoretically sound framework for scaling up structure learning using divide-and-conquer distributed parallelism. A causal partition leverages a structural prior, called a superstructure, to partition the graph space.

Chapter 5 is our final contribution and a case study in applying causal discovery to gene expression data collected from chronic exposure to low-dose radiation in human cells. We used experimental data collected over weeks of radiation exposure. We incorporate a protein-protein interaction network as a structural prior and leverage the causal graph partition to learn causal graphs over the entire genome. We leverage upstream environmental perturbations to infer downstream response mechanisms.

CHAPTER 2

BACKGROUND

In this section, we provide some preliminary background knowledge and definitions. We start with basic definitions of graphs and build towards a definition of a causal graph which we will refer to throughout the dissertation. We then describe gene regulatory networks and discuss the types of data used to infer them. Finally, we provide background on experimental design protocols based on known or partially known causal graphs.

2.1 Directed Acyclic Graphs

A graph $\mathcal{G} = (V, E)$ is defined on $p = |V|$ vertices with $m = |E|$ edges. A directed edge between vertices $u, v \in V$ in defined as $(u, v) \in E$ or $u \to v \in E$. A directed acyclic graph (DAG) is a graph $\mathcal{G}$ with directed edges such that no path $u \to ... \to u$ exists for any $u \in V$ – such a path is known as a ***cycle***. Because of this, every DAG has a linear ordering of its vertices, known as a ***topological ordering*** such that for every directed edge (u, v) from vertex u to vertex v, u comes before v in the ordering. The ***skeleton*** of a DAG $\mathcal{G}$, is the undirected graph obtained by replacing each directed edge in $\mathcal{G}$ with a undirected edge. A vertex u is called a ***collider*** if the directed edges are such that $w \to u \leftarrow v$. If there is no edge between w and v, then $w \to u \leftarrow v$ is also called a ***v-structure***. We will use DAGs as the graphical representation for causal events (edges) between random variables (vertices). The topological ordering of vertices reflects our intuition that causality means events can only affect the future, they never affect the past, and therefore there are no causal loops, or cycles in the graph.

2.2 Bayesian Networks

A Bayesian Network (BN) is a graphical model that represents the joint probability distribution of a collection of random variables $\{X_1, X_2...X_n\}$. A BN ($\mathcal{G}$, $\mathcal{P}$) is defined by a DAG $\mathcal{G}$, with nodes corresponding to the random variables, and a probability distribution $\mathcal{P}(X_1, X_2...X_n)$ that factorizes according to $\mathcal{G}$ – meaning that $\mathcal{P}(X_1, X_2...X_n) = \prod_{i=1}^{n} P(X_i|\mathbf{PA}(X_i))$. $\mathbf{PA}(X_i)$ is the parent set of the random variable X_i in $\mathcal{G}$ (e.g, an edge $u \to v$ means that $u \in \mathbf{PA}(v)$). Because we can decompose the joint probability distribution into local conditional probability distributions as functions of parent sets of $\mathcal{G}$, BNs can compactly model real-world systems.

The decomposition of the joint probability distribution based on $\mathcal{G}$ relies on the fact that each random variable is conditionally independent of its non-descendants given its parents in a BN. This is represented in the DAG $\mathcal{G}$, and is formalized by a property called d-separation.

Definition 2.2.1 (d-separation)**.** *If $\mathcal{G}$ is a DAG in which X, Y and Z are disjoint sets of vertices, then X and Y are d-connected by Z in G if and only if there exists an undirected path U between some vertex in X and some vertex in Y such that for every collider C on U, either C or a descendent of C is in Z, and no non-collider on U is in Z. X and Y are d-separated by Z in $\mathcal{G}$ if and only if they are not d-connected by Z in G. We often write d-separation between X and Y by Z as $X \perp_d Y|Z$*

The ***Markov property*** states that a probability distribution $\mathcal{P}$ is Markov with respect to a DAG $\mathcal{G}$ if d-separation in $\mathcal{G}$ implies conditional independence in $\mathcal{P}$. Two DAGs are Markov equivalent if they encode the same set of conditional independence relations. It is known that two graphs are Markov equivalent if and only if they have the same skeleton and v-structures [Koller and Friedman, 2009]. The ***Markov Equivalence Class (MEC)*** of a DAG $\mathcal{G}$ is the set of DAGs that share conditional independence relationships as $\mathcal{G}$. The Markov property allows us to draw conclusions about $\mathcal{P}$ from $\mathcal{G}$: $X \perp_d Y|Z \implies X \perp\!\!\!\perp_{\mathcal{P}} Y|Z$. In the opposite

direction, the ***faithfulness property*** states that $\mathcal{P}$ is faithful with respect to a DAG $\mathcal{G}$ if conditional independence in $\mathcal{P}$ implies d-separation in $\mathcal{G}$: $X \perp\!\!\!\perp_{\mathcal{P}} Y|Z \implies X \perp_d Y|Z$.

2.2.1 Structure learning

Structure learning is the task of learning the DAG $\mathcal{G}$ from n samples of data $\mathcal{D} = (X_1 ... X_p)$ where each $X_i \in \mathbb{R}^n$. Structure learning is an NP-hard problem due to the exponential increase in potential graphs as p increases [Peters et al., 2017]. The sampled data X is assumed to be independently and identically distributed (i.i.d.) and representative of the joint probability distribution $\mathcal{P}$. Structure learning methods generally fall into four categories: constraint-based, score-based, hybrid, and continuous optimization methods. We will discuss each method here, but for a complete understanding of the space of methods see review papers: Kitson et al. [2023], Drton and Maathuis [2017].

Constraint-based In constraint-based structure learning, we rely on conditional independence tests to determine the presence and absence of edges in $\mathcal{G}$. In order to learn $\mathcal{G}$, we assume that the faithfulness property holds – we can use the conditional independencies in $\mathcal{P}$ to infer d-separations in $\mathcal{G}$. If variables X_i are assumed to follow a Gaussian distribution, the conditional independence between variables reduces to a zero partial correlation. In the general non-parametric case, conditional independence testing is impossible without assumptions [Shah and Peters, 2020]. Given a reliable, albeit heuristic, conditional independence test we can learn the skeleton of the DAG $\mathcal{G}$ with the following: there is no edge between X_i and $X_j \iff \exists S \in \mathbf{X} \backslash \{X_i, X_j\}$ s.t. $X_i \perp\!\!\!\perp X_j|S$. Additionally, we can orient some edges according to the conditional independence relationships of v-structures. For any $X_i - X_j - X_k$ in the skeleton, where there is no edge between X_i and X_j, we can orient $X_i \rightarrow X_j \leftarrow X_k \iff X_i \not\perp\!\!\!\perp X_k|X_j$. This is a specific case of Meek rules for orienting edges; for a full description of these rules see Meek [2013]. A naive strategy for reconstructing the

skeleton is for every pair X_i, X_j test all subsets S to determine conditional independence and remove the edge between X_i and X_j if we find a satisfying S. This strategy is highly inefficient since for $\binom{p}{2}$ possible pairs there are 2^{p-2} possible subsets S. The PC algorithm [Spirtes et al., 2000c] provides a more efficient framework for running conditional independence tests of pairs of variables in *level sets* (increasing conditional set size $|S| = 0, 1, 2..$) and only testing pairs that are still adjacent in the current graph using the local Markov property.

Score-based In score-based structure learning, we test different DAGs for their ability to fit the data $\mathcal{D}$. The idea is to assign a score $\mathcal{S}(\mathcal{D}, \mathcal{G})$ to each graph $\mathcal{G}$ and search over the space of DAGs to find the DAG with the highest score:

$$\hat{\mathcal{G}} = \underset{\mathcal{G} \in \text{ DAG space}}{\text{argmax}} \quad \mathcal{S}(\mathcal{D}, \mathcal{G}) \tag{2.1}$$

. There are many options for the scoring function $\mathcal{S}$, see Kitson et al. [2023] for an overview of all score functions. One example is the Bayesian Information Criterion (BIC) which is based on the maximum likelihood estimate $\hat{\theta}$ for θ, the parameters of the BN (estimates of the conditional probabilities in the factorized joint distribution): $\mathcal{S}(\mathcal{D}, \mathcal{G}) = \log \mathcal{P}(\mathcal{D}|\hat{\theta}, \mathcal{G}) - \frac{\lambda}{2} \log n$. The first term corresponds to the log likelihood of the graph given the dataset and the second term is a regularizer to favor sparse structures, where λ is the number of free model parameters. For linear Gaussian data, $\lambda = 1$. To maximize the score function, we have to traverse the search space of all DAGs which grows super-exponentially with p. It is intractable to compute the score over all graphs, and instead greedy search algorithms are used to optimize the score function. In the case of linear Gaussian joint distributions, the MEC shares the same score, and so it's better to search over the space of MECs Koller and Friedman [2009]. The greedy equivalence search (GES) algorithm traverses the space of MECs, assuming linear Gaussian data, and outputs a CPDAG corresponding to the optimal MEC [Chickering,

2002a]. Another score-based structure learner is the hill-climb (HC) algorithm [Bouckaert, 1994].

Hybrid Hybrid structure learning methods aim to combine the advantages of both constraint- and score-based structure learning methods to improve the efficiency and accuracy of learning. Hybrid methods first constrain the search space with a constraint-based method by estimating the skeleton of the DAG, and then greedily optimize the subspace defined by the skeleton using a score-based method. Examples of hybrid methods include MMHC [Tsamardinos et al., 2006] and ARGES [Nandy et al., 2018].

2.3 Causal graphs

Here, we define causal graphs and how they differ from BNs; the definitions and notation in this section mostly comes from Peters et al. [2017]. A causal graph is often loosely stated as a BN where directed edges encode causal mechanisms. Concretely, this means the DAG $\mathcal{G}$ can also be represented by a structural causal model (SCM).

Definition 2.3.1 (Structural causal models)**.** *A structural causal model (SCM)* $\mathcal{C} := (\boldsymbol{S}, P_{\boldsymbol{N}})$ *is a collection* $\boldsymbol{S}$ *of* p *structural assignments such that*

$$X_i := f(\boldsymbol{PA}_i, N_i) \text{ for } i = 1, ..., p \tag{2.2}$$

where $\boldsymbol{PA}_i \subseteq \{X_1, ..., X_p\} \backslash \{X_i\}$ *are the parents of* X_i *and a joint distribution* $P_{\boldsymbol{N}}$ *of independent noise terms – that is* $P_{\boldsymbol{N}}$ *is a product distribution.*

An SCM implies a causal graph $\mathcal{G}$, with one vertex for each X_i, and directed edges from each parent $\mathbf{Pa}_i$ to X_i. We assume there are no cyclic assignments in $\mathcal{C}$ and therefore $\mathcal{G}$ is a DAG. Note that in many cases cyclic assignments are representations of equilibrium processes which can be "unrolled" by adding time-delineated structural assignments, however,

the acyclic assumption fundamentally prohibits SCMs from modeling stochastic processes. The parents $\mathbf{PA}_i$ are also the ***direct causes*** of X_i. An SCM also implies a probability distribution over variables $\mathbf{X} = (X_1, ..., X_p)$ which we name $P_{\mathbf{X}}$. Unlike BNs, SCMs are the key framework for causal reasoning and causal learning. In a BN, directed edges represent statistical relationships which are inferred from observational distributions and represented by conditional probabilities : $P(X_i|\mathbf{PA}_i)$. SCMs entail observational distributions and ***interventional distributions*** because of the functional relationships between X_i and it's direct causes $\mathbf{PA}_i$;the structural assignments in $\mathcal{C}$ also tell us how X_i changes when $\mathbf{PA}_i$ changes. Formally an interventional distribution from an SCM $\mathcal{C}$ is

Definition 2.3.2 (interventional distribution)**.** *Consider an SCM $\mathcal{C} := (\mathbf{S}, P_{\mathbf{N}})$ and it's entailed distribution $P_{\mathbf{X}}$. We replace at least one structural assignment to obtain a new SCM $\widetilde{\mathcal{C}}$. If we replace the assignment of X_j by $X_k := \widetilde{f}(\widetilde{\boldsymbol{PA}}_j, \widetilde{N}_j)$ then we call the entailed distribution of the new SCM an interventional distribution. The variables whose structural assignment we have replaced have been **intervened** on. The interventional distribution is denoted as $\widetilde{P}_{\mathbf{X}} = P(\mathbf{X}|do(X_k := \widetilde{f}(\widetilde{\boldsymbol{PA}}_j, \widetilde{N}_j))$.*

We will think of interventional distributions in two ways. The first type of intervention is if $\widetilde{f}(\widetilde{\mathbf{PA}}_j, \widetilde{N}_j)$ equals a real value a, which we call a perfect or "hard" intervention, and denote by $\text{do}(X_k := a)$. The second type of intervention is if the parent set remains the same $\widetilde{\mathbf{PA}}_j = \mathbf{PA}_j$, but the functional relationships between X_j and $\mathbf{PA}_j$ change. We call this an imperfect or "soft" intervention. The $\text{do}(\cdot)$ operator comes from Pearl's definition of interventions and counterfactuals [Pearl, 2009].

A counterfactual statement follows a structure similar to *what would have happened, given what actually did happen.* We will not discuss or define counterfactual statements here because unlike interventional distributions, counterfactual distributions cannot be fully represented by causal graphs; that is $\mathcal{G}$ and the corresponding $P_{\mathbf{X}}$. Moreover, for the purposes of discovering new causal relationships in gene regulatory networks it is not necessary to

infer counterfactual distributions. Counterfactual distributions have many biomedical implications such as understanding patient responses to environmental factors or treatment in the absence of randomized control trials.

2.3.1 Causal structure learning

The learning procedure for inferring the structure of Bayesian networks overlaps with learning the causal structure; after all, they are both represented by DAGs. However, a Bayesian network is still valid even if the directionality of some arrows is inconsistent with the true data-generating process. A classic example from Peters et al. [2017] is that of the relationship between altitude A and temperature T. We know that the underlying process/causal structure is $A \to T$. A Bayesian Network inferred from samples (a, t) that outputs a structure $T \to A$ and a corresponding probability distribution $P(A, T) = P(A|T)P(T)$ may correctly encode the conditional probabilities in the sampled data; however, we know this is the incorrect causal structure because when the altitude changes the temperature changes and not the other way around. This is confirmed when we sample at different altitudes and find that $P(T|A)$ holds across locations rather than $P(A|T)$; in other words, only one of these mechanisms is invariant.

The challenge of causal structure learning is determining whether we can learn these invariant causal mechanisms given the data we sampled. In order for the causal structure to be ***identifiable*** we must make certain assumptions about the data we sample. There are specific structural assignments for which the causal structure is identifiable with only observational data, e.g., linear non-Gaussian additive noise models, linear Gaussian models with equal error variances, nonlinear Gaussian additive noise model. These each have corresponding ***consistent*** structure learning algorithms, meaning they are provably correct in learning causal DAG $\mathcal{G}$. When we have access to interventional data, more of the causal structure becomes identifiable without structural assumptions. Consequently, many adap-

tations of structure learners in Section 2.2.1 have been adapted to incorporate data from interventional data e.g, GIES is an interventional adaptation of the GES algorithm [Hauser and Bühlmann, 2012]. Further, many score-based continuous optimization structure learners were built based on leveraging the structural assignments in the SCM.

Continuous optimization The combinatorial optimization in Eq 2.1 was relaxed into a continuous optimization in Zheng et al. [2018] with the NOTEARS structure learning algorithm. Here, causal structure learning is rephrased as an optimization over real-valued matrices $W \in \mathbb{R}^{d\times d}$ rather than over discrete DAGs $\mathcal{G}$. The translation between graphs and SCM is central to their approach. Each X_i is modeled as a linear function of it's parents and independent noise terms as in Eq. 2.2. In vector form this is $\mathbf{X} = W^T\mathbf{X} + N$. A smooth loss function is defined as a least squares function plus an $\ell 1$-regularizer $F(W) = \frac{1}{2n}||\mathbf{X}-\mathbf{X}W||_F$. Critically, they derived a smooth characterization of DAGs: $W \in \mathbb{R}^{d\times d}$ is a DAG if and only if $h(W) = \text{tr}(e^W \circ e^W) - d = 0$ where $\circ$ is the Hadamard product. Now the optimization becomes:

$$\begin{aligned} &\min_{W\in\mathbb{R}^{d\times d}} F(W) \\ &\text{subject to } h(W) = 0 \end{aligned} \tag{2.3}$$

Because the constraint is differentiable with respect to W, we can use gradient-based optimization while ensuring the learned structure is a DAG. Many new algorithms were proposed in addition to NOTEARS that included nonlinear adaptations (NOTEARS-MLP from Zheng et al. [2020]), graph neural networks (DAG-GNN from Yu et al. [2019]) and variational inference (GraN-DAG from Lachapelle et al. [2019]).

Scaling All structure algorithms suffer from scaling issues either because of the size of the DAG search space, the number of conditional independence tests needed, or the memory footprint required to store an adjacency matrix. As a result there are minimal applications

of structure learning to biological networks at scale. A notable exception is the adaptation of GES to fGES (Fast Greedy Equivalence Search) which allowed learning structures over 51,000 variables from temporal brain image data with a total runtime of 14 hours [Ramsey et al., 2017]. Scaling in fGES is achieved by cacheing and parallelizing the computation of the score function. The learned structure, however, was not evaluated for accuracy and strong sparsity was assumed in order to estimate the final graph. Scaling is a key challenge we address in this dissertation because we are interested in learning genome-scaled causal graphs (this is thousands of genes), and so we will return to different scaling strategies in Chapters 3 and 4. We also note that scaling for causal structure learning in biology is often limited by the availability of interventional data at scale and not just by algorithmic complexity.

2.4 Gene regulatory networks

We now turn to gene regulatory network (GRN) inference — an application of causal structure learning. In this section, we provide an overview of GRNs, experimental data from which we can infer GRNs, and existing network knowledge bases for GRNs.

Gene regulation is the process by which genes, pieces of DNA, modulate other genes. Gene regulation controls gene expression, or the amount of RNA transcribed for a particular gene — this is proxy for the abundance of the protein coded by the gene. If a gene up-regulates another gene, this means the expression of the downstream gene goes up, the abundance of the coded protein goes up, and a specific function in the cell is carried out. In contrast, if a gene down-regulates another gene this implies the suppression of a specific function in the cell. Changes to gene expression are often triggered by external signals received by a cell — e.g., environmental stress. For example, HSP70 is a gene that is upregulated when cells experience environmental stress such as elevated temperatures, oxidative stress, or toxic exposure. The HSP70 protein helps protect other proteins from denaturation and assists in

refolding damaged proteins [Lindquist et al., 1988]. The orchestration of gene regulation is highly complex, with certain 'hub' genes (we call these transcription factors) regulating hundreds or thousands of other genes. Further, gene regulation is context-specific — meaning regulation of certain genes is not necessarily present across organisms, cell types, tissue types, or environmental conditions. GRNs provide us with a systematic model for this complicated system. In a GRN, genes are represented as nodes in a graph, and regulation between genes is represented by directed edges. The human genome has ∼20,000 genes, so in theory the human GRN would represent the regulatory landscape between all these genes, however, little of the full human gene regulatory network is mapped with high confidence. GRNs that are context-aware (cell-type, tissue-type, disease-state or environment) are still vastly underexplored.

How was the currently known human GRN mapped? Directed edges between transcription factors to target genes are experimentally validated using different types of experiments including ChIP-sequencing, RNA-sequencing, CRISPR and Perturb-seq experiments. Here, we briefly describe each of these technologies.

1. ***ChIP-seq*** is an experiment that determines how proteins (transcription factors) interact with DNA (target genes). If a transcription factor binds to the promotor region of a gene, this is a good indication that there is a regulatory relationship between the transcription factor and target gene.

2. ***RNA-seq*** is an experiment that measures the copies of messenger RNA after transcription from DNA to RNA; this is also known as gene expression. RNA-seq allows for genome-wide gene expression measurements — allowing us to discover more of the regulatory search space. There are several types of RNA-seq experiments; these include bulk (averaged over a population of cells), single-cell, spatial and temporal measurements.

3. ***CRISPR and Perturb-seq*** are gene perturbation experiments. CRISPR is the tech-

nology that allows for specific genes to be cut out (knocked out), suppressed (knocked down) or activated using the Cas9 enzyme. Perturb-seq is a combination of CRSIPR and single-cell RNA-seq. Perturb-seq measures the effects of genetic perturbations of specific genes with CRISPR on the gene expression of the entire genome.

ENCODE is a large-scale database that contains all the experimentally validated regulatory elements in the human genome from assays such as ChIP-seq and RNA-seq to characterize transcription factor binding and gene regulation across diverse cell types and conditions [Luo et al., 2020]. TRRUST is a database of human and mouse transcriptional regulatory interactions based on literature curation and text-mining from biomedical papers [Han et al., 2018]. These serve as the primary resources for curating ground truth networks for evaluating GRN inference methods.

GRN inference methods predict a network from data, and are broadly categorized into associative and regression based algorithms. Associative methods compute pairwise scores between genes and use a threshold to determine the presence of an undirected edge between genes. Examples of associative algorithms include ARACNE [Lee and Kim, 2019], CLR [Faith et al., 2007], and WGCNA [Langfelder and Horvath, 2008]. Regression methods answer the question for each target gene's expression, which other genes' expression values are the most predictive? These methods use an ensemble of models, one per target gene, and a machine learning model such as Lasso, RandomForest, or XGBoost to fit and generate feature importance scores for each candidate upstream gene. A threshold is selected to filter the top k most predictive upstream genes for each downstream gene. Examples of regression methods include GENIE3 [Marbach et al., 2012], GRNBoost2 [Moerman et al., 2019], and TIGRESS [Haury et al., 2012]. These methods have been applied extensively to observational gene expression data; however, they generally lack the ability to jointly model observational and interventional data, limiting their capacity to recover causal regulatory relationships.

GRN inference can also be thought of as a causal structure learning problem. We are

given observational and perturbation gene expression data through RNA-seq, ChIP-seq or CRISPR experiments. The expression of genes by other genes is causal in nature, since a gene must first be expressed and translated into proteins in order for the up-regulation or down-regulation of other genes to occur. We can neatly assign genes to random variables or nodes and regulatory events as directed edges between nodes in the DAG. Since the experimental technologies are "genome-wide", we observe all genes and satisfy causal sufficiency – at least over the gene space. Perturb-seq experiments give us large scale interventional data meaning we have more information to separate the true DAG from the MEC.

Seemingly, there are many reasons to abstract GRN inference to causal structure learning and consequently there has been an explosion of causal structure learning papers that state GRN inference as a use case: Dai et al. [2024], Brouillard et al. [2020], Squires et al. [2020], Nazaret et al. [2023], Lopez et al. [2022] to name a few. However, there are several ways in which GRN inference fails to meet the strict assumptions needed for causal structure learning.

- **Feedback loops**: It is well-known that GRNs contain feedback loops. For example, genes often self-regulate. In the case of a negative feedback loop, the expression of gene X is modulated by its own expression value; once expression reaches a certain threshold, the gene will negatively self-regulate so that expression does not exceed a threshold – this is also known as homeostasis. This clearly violates the acyclic assumption; it seems unreasonable to represent GRNs as DAGs in the first place. To ameliorate this issue, we can use time as an axis to "unroll" cycles in a GRN [Assaad et al., 2022, Spirtes and Zhang, 2016], however, in order to infer a DAG we would then need time-series data which is expensive to acquire. Many works also relax the acyclic constraint to learn graphs with cycles [Mooij et al., 2011, Strobl, 2019].

- **Latent confounders**: Despite genome-wide technologies' ability to collect data over the full genome, there is always the presence of unobserved environmental and pre-

or post-transcriptional confounding variables. For example, DNA methylation is a biological process that adds methyl groups (CH3) to DNA; when this happens in the promoter region of a gene, this effectively suppresses the transcription of that gene. Since this process of methylation depends on factors outside of gene expression (local DNA context, chromatin accessibility, radiation exposure, stress, and many others), measuring the full transcriptome does not deconfound the methylation landscape. FCI [Zhang, 2008b], GPS [Claassen and Bucur, 2022], and DCD [Bhattacharya et al., 2021] are examples of constraint, score, and continuous optimization algorithms, respectively, that handle latent confounders.

- **Faithfulness**: In order for identifiability of a causal graph given data, we need the faithfulness assumption. Both feedback loops and the presence of latent confounders inherently break faithfulness. Even if these two issues are mitigated, faithfulness is not guaranteed in gene expression data. For example, gene regulation is often non-linear and we usually describe the dynamics of regulation using Hill functions. This means that gene expression activates at a specific threshold of another gene, and then saturates at another threshold. If a transcription factor needs to cross a threshold to activate a target gene, below that threshold the target may look independent of the regulator, even though a causal edge does exist. Some works address this issue, showing that identifiability can still be achieved with weaker faithfulness assumptions [Lin and Zhang, 2020, Zhang and Spirtes, 2008, Ng et al., 2021].

Finally, there are practical reasons why causal discovery is difficult to apply to GRN inference. These are issues of statistical efficiency with a large number of nodes (p) and a low sample size (n). One of the objectives of this thesis is to demonstrate how causal discovery can still apply to GRN inference; mainly by tackling practical issues like scalability. We show that despite theoretical limitations, the causal framework is useful for GRN inference and, in particular, for identifying experimental targets to uncover more of the human GRN.

CHAPTER 3

INCORPORATING STRUCTURAL PRIORS FOR BOOSTING GREEDY CAUSAL DISCOVERY

Causal discovery of genome-scale networks is important for identifying pathways from genes to observable traits – e.g. differences in cell function, disease, drug resistance and others. Causal learners based on graphical models rely on interventional samples to orient edges in the network. However, these models have not been shown to scale up the size of the genome, which is on the order of 10^3-10^4 genes. We introduce a new learner, SP-GIES, that jointly learns from interventional and observational datasets and achieves almost 4x speedup against an existing learner for 1,000 node networks. SP-GIES achieves an AUC-PR score of 0.91 on 1,000 node networks, and scales up to 2,000 node networks – this is 4x larger than existing works. We also show how SP-GIES improves downstream optimal experimental design strategies for selecting interventional experiments to perform on the system. This is an important step forward in realizing causal discovery at scale via autonomous experimental design.

3.1 Introduction

A biological network describes causal interactions between variables in a biological system. These variables can be genes, transcription factors, proteins and metabolites. Interactions between these variables, along with external environmental factors, maps the genome of a species (or genotype) to an observable trait (or phenotype) (See Fig. 3.1). Predicting phenotypes from genotypes is one of the core challenges in systems biology Pigliucci [2010], Ritchie et al. [2015], Lewis et al. [2012]. Genotype-phenotype models are useful for predicting cell function, disease, drug resistance, and many other problems related to biology and public health. In this paper we focus on biological network recovery of the first layer in

the interaction hierarchy (the gene regulatory network) – which we name a "genome-scale network" since the number of variables corresponds to the number of genes in a genome. Recovery of genome-scale networks is important for understanding drivers for phenotypic changes and for identifying new drug targets. Moreover, a better understanding of genome-scale networks enables more accurate downstream phenotype prediction models including those that use machine learning (See latent topics in Fig. 3.1).

Biological networks are inferred from experimental data. Recovery of biological networks is a reverse engineering problem: given experimental data, what is the network that gives rise to the data? Existing methods for network recovery fall into two primary categories: pairwise information theoretic methods, and graphical model methods. The advantages of the former are that they are embarrassingly parallel and have been shown to be effective on large-scale biological datasets Faith et al. [2007], Lachmann et al. [2016], Margolin et al. [2006]. The main disadvantage of these methods is that they can only learn from one data distribution – e.g. they cannot incorporate new experimental data after an intervention has been applied to a system. The second category of network recovery methods are graphical models, often called structure learners. The advantages of these are that they have an immediate causal interpretation (the direction of an edge implies cause and effect) and as a result a suite of methods have been built to jointly learn from observational and interventional data distributions Hauser and Bühlmann [2012], Wang et al. [2017], Tong and Koller [2001] – we call these "joint learners". This ability to jointly learn from different distributions takes a statistical model (e.g. a Bayesian Network) and converts it into a causal model (e.g. a Causal Bayesian Network). The disadvantages of graphical models are that they do not scale well with the number of nodes in the graph (the graph search space is combinatorial), and joint learners do not have parallel implementations. This means that joint structure learners have not been evaluated on datasets at scale – at most we observed evaluations on 500 node networks. Given that the genomes of most species are on the order of 10^3-10^4 genes, there

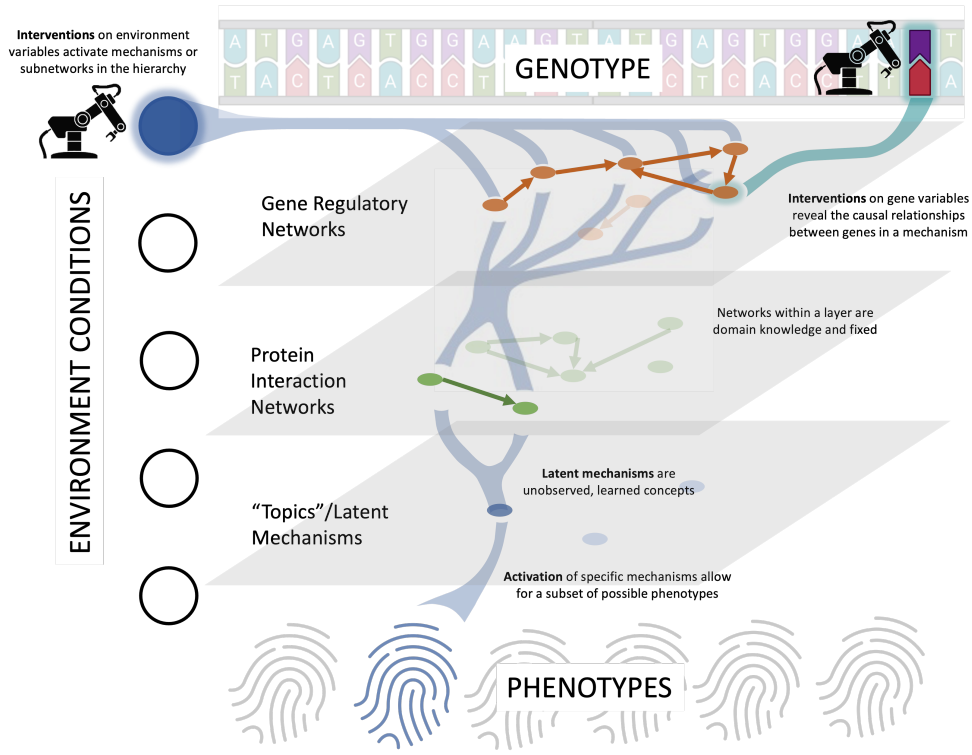


Figure 3.1: The hierarchy of biological networks is partially known. Levels in this hierarchy can be represented as causal networks. Interventions activate mechanisms of action and reveal causal relationships within a mechanism. The space of possible interventions is large, motivating the need for autonomous design loops like AI driven experimental design and robotic laboratories. In theory, recovery of all networks allows for full genotype to phenotype mapping. However, in practice it is impossible to fully capture the data distribution and control for confounding variables. This motivates representing mechanisms of action as topic latent variables that can be learned via topic modeling.

is a need to scale up these joint structure learners to larger networks. In this paper we introduce a new hybrid structure learner, named SP-GIES, that leverages the advantages of both categories of methods.

A causal model is preferable to a statistical model because directed causal edges in a network allow us to identify pathways, or mechanisms of action, from genotype to phenotype. In applications of biology and medicine, researchers seek to model both functional outcomes and the mechanisms that lead to outcomes so that these can be understood and manipulated through therapeutics. Given this priority, there is a demand for causal models

in this application space. This motivates the need to incorporate data from interventions on the system. We estimate the space of possible interventions in the environment and gene spaces to be 10^4-10^5 depending on the species under investigation. A brute force sampling of this space of interventions is experimentally expensive and wasteful. Instead, there is a need to design feedback loops that choose advantageous interventions based on domain knowledge and learned knowledge. Work in optimal experimental design (OED) uses expected gain to choose interventions on variables in Bayesian Networks to drive causal discovery [Tong and Koller, 2001, Hauser and Bühlmann, 2014, Agrawal et al., 2019, Ghassami et al., 2018]. This paradigm allows us to reason about how new inferred knowledge can connect to our existing domain knowledge, and provide a path forward for integrating AI into science domains.

The contributions of this work are as follows:

1. The implementation of a new hybrid structure learning method named Skeleton-Primed Greedy Interventional Equivalence Search[1] (SP-GIES), based on the existing method GIES Hauser and Bühlmann [2012], that jointly learns from observational and interventional data. SP-GIES achieves better network recovery accuracy and a faster time to solution on large-scale networks compared to existing methods.

2. The application of SP-GIES to an optimal experimental design feedback loop that chooses optimal interventions on genes.

3. An analysis of the complexity of these methods and discussion of future directions, including unified libraries of causal algorithms, in order to achieve genome-scale network recovery for genotype to phenotype mapping with interventions.

1. Code is available at `https://github.com/shahashka/SP-GIES`

3.2 Preliminaries

3.2.1 Causal Bayesian Networks

Let $G = (V, E)$ be a directed acyclic graph defined by a set of vertices V and directed edges E. The vertices of the graph represent random variables $X_1 ... X_p$. Under the Markov Assumption for Bayesian Networks, each variable X_i is conditionally independent of its non-descendants given its parents. The joint distribution of a Bayesian network factorizes as $P(\mathbf{X}) = \prod_{i=1}^{p} P(X_i|Pa(X_i))$ Spirtes et al. [2000a]. The following definitions describe important properties of Bayesian Networks that are relevant for structure learning.

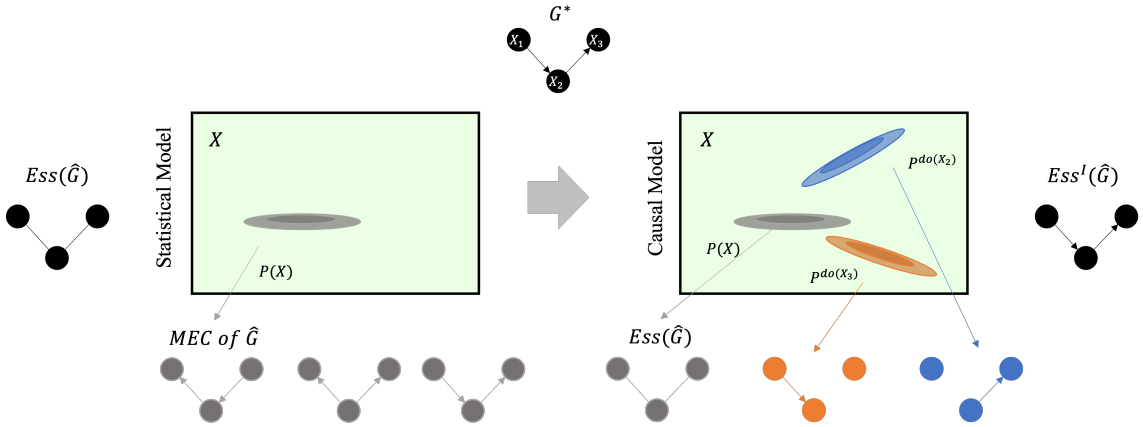


Figure 3.2: How to convert from purely statistical model (such as a Bayesian Network) to causal model (such as a Causal Bayesian Network) with interventional data. By modeling interventional distributions, the interventional essential graph $Ess^I(G)$ is closer to the true underlying causal graph (G^*) compared to the observational essential graph $Ess(G)$. This figure is adapted from Schölkopf et al. [2021]

Definition 3.2.1 (Markov Equivalence Class (MEC))**.** *The Markov Equivalence class (MEC) of a Bayesian Network G consists of all directed acyclic graphs (DAGs) that share the same conditional independence relationships.*

Definition 3.2.2 (Essential Graph (Ess))**.** *An essential graph Ess(G) is a partially oriented graph that uniquely represents the MEC of a DAG. Directed arrows only exist on edges*

consistent across the equivalence class and all other edges are left undirected Andersson et al. [1997].

A Causal Bayesian Network is a Bayesian Network where edges represent cause and effect relationships Pearl [1995a]. For example, if a directed edge exists from X_i to X_j, this means that no unobserved confounding variables are responsible for their correlation and X_i is a direct cause of X_j.

3.2.2 Structure Learning with Observational Data

Constructing the graph structure from a set of instances (sampled values for each node X_i) is called structure learning. The following assumptions are made for learning a graph $\hat{G}$ from data.

1. **Causal sufficiency** - All random variables are observed, i.e. there are no hidden variables

2. **Causal Markov Assumption** - The data is generated from an underlying Bayesian Network (G^*, θ^*) over a set of random variables X

3. **Faithfulness Assumption** - The distribution P^* over X induced by (G^*, θ^*) satisfies no independencies beyond those implied by the structure of G^*

We are given a data set $D = \{X_1...X_p\}$ of N samples from P^* – this data is assumed to be independent and identically distributed (i.i.d.). The task is to learn a model $\hat{M} = (\hat{G}, \hat{\theta})$ that defines a distribution $\hat{P}$ that best fits true distribution P^* Koller and Friedman [2009].

All graphs in the same MEC give rise to the same observational distribution. Therefore the best we can do with only observational data is recover the true graph's MEC. Several graphical model based methods learn the structure from observational data only. These are split into constraint-based and score-based methods. There are also pairwise information theoretic methods that learn from only observational data. We discuss these in Section 3.3.

3.2.3 Structure Learning with Interventional Data

In addition to structure learning on observational data, we want to incorporate interventional datasets into our learning procedures. Intervening on a set of random variables $I \subseteq X$ removes incoming edges to the random variables I, and sets the joint distribution to a new interventional distribution $P^{do(X_i=x)}$. Here, we are setting node X_i to a value x. In this paper we consider "hard" interventions, where a node is set to a constant value, because this mimics the types of interventions we are able to perform in biology applications – e.g. gene knockout experiments. This is in contrast to a "soft" intervention which samples $X_i = x$ from a distribution.

Definition 3.2.3 (Interventional Distribution)**.** *The joint distribution after a hard intervention is:*

$$P^{do(X_i=x)} = \begin{cases} \prod_{j \neq i}^{p} P(X_j | Pa(X_j) & if X_i = x \\ 0 & otherwise \end{cases}$$

Jointly learning on observational and interventional data allows us to correctly isolate causal relationships and orient edges in the graph. An estimated $Ess(G)$ from observational data can be further refined into an I-essential graph ($Ess^I(G)$). This is a partially oriented DAG that represents an interventional Markov Equivalence class (I-MEC) (See Fig. 3.2). There are several existing methods that jointly estimate structure given observational and interventional data. See Vowels et al. [2021] for a full list of these structure learners.

3.2.4 Optimal Experimental Design

We have access to new interventional data by sampling the system, but we wish to prioritize the interventional experiments that will recover the true causal graph the fastest. This is done via maximization of an expected utility function, over the set of potential interventions:

$\hat{I} = \arg\max_I \mathbb{E}_{G|D}[U]$. Here, U is a utility function like mutual information or number of oriented edges.

Our goal is to recover the underlying Causal Bayesian Network given a mix of interventional and observational samples, and some prior information about the causal edges in the graph. We cast this as a Bayesian Inference problem over the space of DAGs. We have a prior $P(G)$ which encodes any prior structural knowledge about the underlying DAG. Applying Bayes Theorem gives us the posterior distribution $P(G|D) \propto P(D|G)P(G)$. The likelihood is $P(D|G) = \int_\theta P(D, \theta|G)d\theta = \int_\theta P(D|\theta, G)P(\theta|G)d\theta$, where we have marginalized out the parameters of the graph Tong and Koller [2001]. D is a mix of observational and interventional data. The posterior distribution is sampled via the joint structure learners described in the previous section. After sampling new data from the chosen optimal intervention, the new data is concatenated to the existing data and the posterior is re-sampled using the joint structure learners.

3.3 Related Work

Table 3.1 and Table 3.2 provide a summary of the related work in biological network recovery and optimal experimental design respectively.

3.3.1 Structure Learning With Interventional Data for Biology Network Recovery

Pairwise information theoretic learners use calculated metrics to estimate the dependence of two variables. A threshold is applied to the metric to obtain a network representation of the system. Pinna et al. [2010] use a deviation matrix (difference between intervened samples and steady state samples) to estimate an initial network. This algorithm won the DREAM4 network recovery challenge and was the state-of-the-art for gene regulatory network recon-

Table 3.1: Properties of all the learners covered in this paper. Methods are split row-wise by the nature of the learner. Data type refers to observational and/or interventional data capability of the learners. Evaluation dataset lists the benchmarks used for evaluation in each paper. Finally, the table also lists the maximum number of nodes the algorithm was evaluated on and the worst case complexity of the algorithm as reported in the corresponding papers. p is the number of nodes, n is the number of samples. k is the maximal degree of any node in the graph. Note that for joint learners the average case complexity is not exponential, however exact calculations depend on clique size and network topology.

Category	**Method**	**Data Type**		**Evaluation Dataset**			**Scaling**	
		Observ.	Interv.	Random	DREAM	Large-Scale	Max Nodes	Runtime
Pairwise Info. Theoretic	CLR [Faith et al., 2007]	✓	✗	✗	✗	✓	4,345	$O(p^3n^3)$
	ARACNE-AP [Margolin et al., 2006]	✓	✗	✗	✗	✓	1,331	$O(p^3 + p^2n^2)$
	Pinna et al. [2010]	✗	✓	✓	✓	✗	100	$O(p^3 + np)$
Graphical Models	PC [Spirtes et al., 2000c]	✓	✗	✓	✓	✓	5,361	$O(p^{k+2})$
	GIES [Hauser and Bühlmann, 2012]	✓	✓	✓	✓	✗	500	$O(2^p)$
	IGSP [Wang et al., 2017]	✓	✓	✗	✗	✗	24	$O(2^p)$
	MCMC Mallows [Rau et al., 2013]	✓	✓		✓	✗	10	$O(2^p)$
Hybrid	SP-GIES (this paper)	✓	✓	✓	✓	✓	2,000	$O(2^p)$

Table 3.2: Properties of the OED methods covered in the paper. OED criteria refers to the utility function used for selection of the optimal intervention. Note that the complexity listed here corresponds only to choosing the next intervention or sets of interventions. Each algorithm also has the cost of sampling from the posterior distribution which is equivalent to the complexity of the learners in Table 3.1. $O(p^k)$ refers to any polynomial in p.

Method	**OED Criteria**		**Evaluation Dataset**			**Scaling**	
	Info. Theory	Edge Orient.	Random	DREAM	Large-Scale	Max Nodes	Runtime
Ness et al. [2017]	✗	✓	✗	✓	✗	17	$O(\|E\| \cdot p!)$
Hauser and Bühlmann [2014]	✗	✓	✗	✗	✗	40	$O(p^k)$
Tong and Koller [2001]	✓	✗	✗	✗	✗	12	$O(pn)$
ABCD Agrawal et al. [2019]	✓	✗	✓	✓	✗	10	$O(p^2n)$
BED Ghassami et al. [2018]	✓	✗	✓	✓	✗	100	$O(p^k)$

struction at the time. Faith et al. [2007] introduce CLR (context likelihood of relatedness) which calculates the pairwise B-spline mutual information between genes in the E. coli K-12 gene regulatory network. CLR is implemented in MATLAB, with a fast parallel kernel for calculating the pairwise mutual information. Margolin et al. [2006] developed ARACNE (Algorithm for the Reconstruction of Accurate Cellular Networks) which is a similar pairwise mutual information based network recovery algorithm. ARACNE additionally employs a DPI (data processing inequality) algorithm which removes indirect interactions. ARACNE was more recently upgraded to ARACNE-AP Lachmann et al. [2016], which is implemented

in Java to handle adaptive partitioning for calculating the mutual information – this achieved a 200x computational performance improvement.

Graphical model based learners reveal a much finer structure than pairwise methods because they are based on Bayesian Networks. The PC algorithm is a constraint-based structure learner that only handles observational data. PC learns a graph by performing conditional independence testing on pairs of nodes conditioned on other nodes [Spirtes et al., 2000c]. Since its introduction, many parallel implementations and optimizations of the PC algorithm have been added (Le et al. [2016], Madsen et al. [2017]) including one that uses GPUs (Zarebavani et al. [2019]). Existing structure learners that jointly learn from observational and interventional data are either score-based or hybrid score/constraint-based methods. Rau et al. [2013] propose an algorithm called MCMC Mallows based on Markov Chain Monte Carlo sampling over the space of potential Gaussian DAGs to find the best scoring DAG. MCMC Mallows was evaluated on the DREAM4 networks and scored better than Pinna et al. [2010] on two of the networks. Hauser and Bühlmann [2012] proposed Greedy Interventional Equivalence Search (GIES) which is a greedy score-based approach that is an extension of the original GES learner. The worst case complexity of GIES is exponential in p (the number of nodes in the graph), however the authors note and show empirically that GIES is more efficient than this worst case. GIES was evaluated on DREAM4 and achieved scores in the top third of all participants. Intervention Greedy Sparsest Permutation (IGSP, Wang et al. [2017]) is a hybrid score/constraint, and a non-parametric extension of the GSP learner. IGSP associates a DAG to every permutation of random variables and greedily updates the DAG by transposing elements of the permutation. IGSP performed comparably to GIES on protein signaling data Sachs et al. [2005] and better than GIES on a real gene expression dataset with 24 genes. The worst case complexity of IGSP is also exponential in p. IGSP and GIES can be implemented with polynomial complexity by restricting the maximum degree of each node in the graph, however, for large scale networks with hubs (nodes

with high connectivity) the maximum degree may be difficult to determine. We show the empirical scaling of these joint learners against our proposed learner, SP-GIES, in Section 3.4.

3.3.2 Optimal Experimental Design for Biology Network Recovery

One way to choose the optimal intervention is to choose the set of interventions that maximizes the mutual information or leads to the greatest decrease in entropy. Tong and Koller [2001] sample a set of orderings from the distribution over graphs and parameters. They then use this set of orderings to compute entropy terms and select the intervention with the lowest expected posterior loss. Agrawal et al. [2019] extend the work of Tong and Koller [2001] with the ABCD strategy – a greedy implementation of batched experimental design. They use a utility function based on the expected entropy decrease of an intervention. This requires calculating expectations over the graph and parameter spaces, however, they are able to make a tractable algorithm by using bootstrapping and weighted importance sampling.

Another way to choose the optimal intervention is to choose the intervention that leads to the maximal number of oriented edges. Ness et al. [2017] use optimal experimental design to recover protein signaling networks Sachs et al. [2005]. They use a utility function based on the expected information gain of an intervention given the observational MEC and other interventions in the batch. This algorithm, however, has factorial dependence on batch size. Ghassami et al. [2018] use the expected number of oriented edges of an essential graph as the utility function. The essential graph of the ground truth network is first estimated using a constraint based method like the PC algorithm. Hauser and Bühlmann [2014] similarly propose a utility function based on the number of oriented edges of a skeleton graph.

3.4 Method

Here we describe our hybrid algorithm Skeleton-Primed Greedy Interventional Equivalence Search (SP-GIES). A skeleton is an undirected graph with edges that correspond to edges in a DAG. The algorithm is a simple sequential use of an observational learner to estimate a skeleton, and the joint graphical model structure learner GIES to orient edges. The two step algorithm is as follows:

1. Use ARACNE, CLR or PC to generate a skeleton with only observational samples.
2. Use the output of (1) to restrict the possible edge set for the GIES learner. Jointly learn from observational and interventional data using GIES.

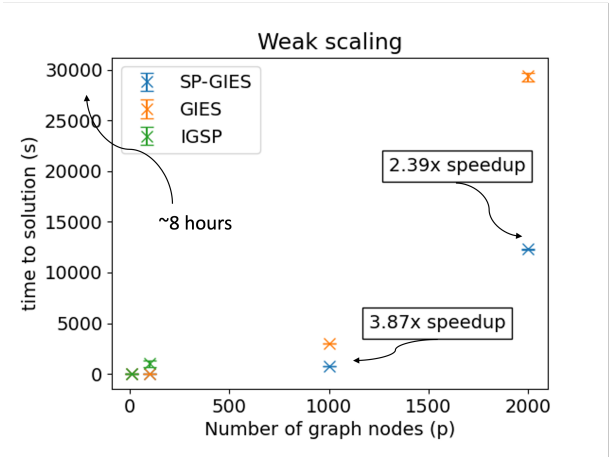


Figure 3.3: A weak scaling study comparing the SP-GIES, GIES and IGSP algorithms. The number of samples is fixed to n=1,000. Data points are averaged over 3 runs. The study was performed on a 2.8 Ghz AMD EPYC Milan 7543P 32 core CPU and a NVIDIA A100 GPU. The IGSP 1,000 and 2,000 node runs exceeded our quota and are not plotted here.

If the PC algorithm is used then the input to (2) is an Ess(G) which is represented by a partially oriented DAG or PDAG. We chose GIES for (2) since it has an open source implementation that is part of the widely used `pcalg` library in R. For scaling studies, we chose to use the R implementation of PC algorithm. This is because CLR and ARACNE are implemented in MATLAB and Java respectively. Since GIES only has an R implementation, we chose to use the R implementation of PC for our scaling studies.

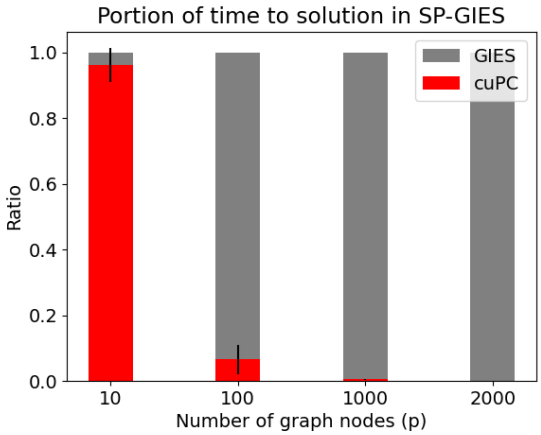


Figure 3.4: Fraction of runtime for each step of the SP-GIES algorithm. Step (1) is cuPC and Step (2) is GIES. We see that as the graph size increases, the bottleneck is the GIES step.

While (1) is easily parallelizable on multiple processors or deployable to a GPU, the bottleneck of the computation is in the GIES step (see Fig. 3.4). Still in Fig. 3.3, we see that SP-GIES is able to reach a faster time to solution than GIES. To understand why this is, we investigated the GIES algorithm. GIES is a greedy score-based structure learner. The score function is the Bayesian Information Criterion which is based on the maximum likelihood estimate of the graph given the current data. Starting from an empty essential graph, in the forward phase of the algorithm an edge is added to $Ess(G)$ such that the new essential graph has the maximal score. The algorithm iterates through all possible edges until it finds the corresponding essential graph with the highest score. The backward phase is similar, except an edge is removed instead of added. By restricting the set of possible edges to the GIES algorithm by first estimating the skeleton, we narrow the size of the search space for both phases – this is what results in the speedup. Moreover, since we use a fast GPU implementation of the PC algorithm (Zarebavani et al. [2019]), the overhead of estimating the skeleton first is minimal.

Table 3.1 shows that all joint learners have a worst case exponential complexity – this is attributed to the combinatorial search space over graphs. However, the average empirical complexity is related to the topology of the network i.e, the size of the largest clique. Since

the average complexity is difficult to calculate, we show in Fig. 3.3 that SP-GIES reaches a faster time to solution as the size of the network increases. We ran a scaling study for SP-GIES using a similar study as Hauser and Bühlmann [2012] except we were able to scale up to p=2,000 nodes versus the p=500 nodes in the GIES paper. IGSP was unable to complete for the 1,000 and 2,000-node networks within our allotted quota of 72 hours. We see that SP-GIES is able to reach a faster time to solution than both IGSP and GIES because of the restriction of the edge set discussed previously.

SP-GIES is a heuristic interventional version of algorithms like ARGES Nandy et al. [2018], which restricts the search space by providing an initial guess for observational datasets. There is an accuracy trade off in SP-GIES since restricting the search space can result in convergence to local maxima. As such, SP-GIES does not provide the consistency guarantees of algorithms like ARGES and IGSP. In Section 3.6, we show empirically that SP-GIES performs well against existing learners on benchmark datasets.

3.5 Datasets

We tested SP-GIES against existing learners on datasets at various scales. Random networks were generated using the `networkx` Python library. Observational data was generated assuming a linear Gaussian model. We generated 100 observational samples and one interventional sample per node. The DREAM4 insilico challenge provides observational and interventional gene knockout data of gene regulatory networks synthetically generated from stochastic ODEs. The dataset contains 10 observational samples and one interventional sample per node. The ground truth networks are subnetworks of the larger E.coli gene regulatory network. The RegulonDB dataset was provided by Faith et al. [2007]. The network is the current estimated E.coli K12 gene regulatory network and contains 1146 nodes, 3179 edges and 524 real experimental samples. Of the 524 experimental samples, 35 are samples are taken from gene knockout experiments. The RegulonDB dataset also contains environ-

Table 3.3: Evaluation of learners on random networks of size 10. Three different types of random networks were generated: Erdös Renyi Erdős et al. [1960], Scale free Barabási and Bonabeau [2003], and Small world Watts and Strogatz [1998]. p and k refer to parameters used to generate the graphs, not the number of nodes and degree as used previously. Results are averaged over 30 random graphs, each learned with 100 data samples. The superscript on the algorithms refers to the data type used (O for observational, OI for mixed observational and interventional). For SP-GIES, ARACNE, CLR, and then PC were used for skeleton estimation for Erdös Renyi, Scale Free and Small World respectively since these were the best performers.

Algorithm	**Erdös–Rényi** ($p = 0.5$)			**Scale-Free** ($k = 2$)			**Small-World** ($p = 0.5, k = 2$)		
	SHD	SID	AUC-PR	SHD	SID	AUC-PR	SHD	SID	AUC-PR
PC^{O}	22.77	70.73	0.35	11.27	61.00	0.62	8.23	36.33	0.59
CLR^{O}	22.67	71.43	0.34	17.27	68.30	0.35	10.83	51.17	0.43
ARACNE-AP^{O}	21.83	79.00	0.36	16.63	76.70	0.37	10.10	52.80	0.43
GES^{O}	23.27	53.34	0.47	**11.20**	**28.57**	**0.69**	10.23	**17.27**	0.67
$GIES^{OI}$	23.93	**39.23**	0.59	14.13	36.43	0.68	21.30	17.70	0.55
$Pinna^{OI}$	22.67	57.60	0.26	18.00	56.93	0.28	13.97	29.73	0.13
$IGSP^{OI}$	**20.00**	67.27	0.38	15.00	53.23	0.41	6.73	22.43	0.55
SP-GIES^{OI}	24.00	39.87	0.59	11.34	48.53	0.64	**5.90**	26.53	**0.68**
NULL	21.00	66.83	**0.61**	16.00	61.30	0.58	10.00	30.00	0.55

mental interventions. Existing learners currently only model gene variables, therefore any environment interventional samples were treated as observational data. Future work includes explicitly modeling environmental condition variables.

Each learner was evaluated on three metrics: Structural Hamming Distance (SHD), Structural Interventional Distance (SID) and the AUC-PR. The SHD is the L1 error between the generated DAG and the ground truth DAG. The SID counts the incorrect interventional distributions as defined by Eq. 3.2.3. The AUC-PR is the area under the precision-recall curve. All three of these metrics are common evaluations for causal discovery. However, we note that SHD and AUC-PR are biased towards empty graphs. Suppose a structure learner A learns a graph G_A with no edges and with $SHD_A = |E|$. Another learner B learns a graph G_B with $SHD_B > |E|$, but correctly learns some of the causal edges in the true graph. According to the SHD, G_A is better and we may posit that A is the better learner for the data. However, an argument can be made for G_B because it captures more causal relationships than G_A. This means that a lower SHD may not always indicate a model

Table 3.4: Evaluation of learners on the DREAM4 networks of size 10. The dataset contains 11 observational samples and 10 interventional samples. Interventional samples are gene knockout experiments. For SP-GIES, ARACNE was used for skeleton estimation, except for Network 4 which used CLR. Note that for Network 3, we used adaptive learning in the GIES subroutine (adaptive="triples") to achieve the best performance for the SP-GIES learner. We did not see significant improvement with adaptive learning for the other networks.

Algorithm	**Net 1**			**Net 2**			**Net 3**			**Net 4**			**Net 5**		
	SHD	SID	AUC-PR	SHD	SID	AUC-PR	SHD	SID	AUC-PR	SHD	SID	AUC-PR	SHD	SID	AUC-PR
PC^{O}	14	39	0.31	12	56	0.43	16	64	0.26	14	45	0.16	11	57	**0.59**
CLR^{O}	13	36	0.45	**10**	56	0.53	14	62	0.36	14	47	0.19	15	62	0.33
ARACNE-AP^{O}	**11**	28	0.56	12	56	0.45	13	61	0.43	15	45	0.06	12	57	0.46
GES^{O}	13	29	0.49	15	52	0.30	17	64	0.15	19	48	0.10	11	51	0.14
$GIES^{OI}$	13	31	0.51	12	49	0.51	16	**40**	0.49	15	**26**	0.34	9	39	0.37
$Pinna^{OI}$	**11**	**22**	0.49	11	56	0.08	14	61	0.57	**12**	42	0.37	11	38	0.47
$IGSP^{OI}$	13	27	0.07	13	54	0.08	16	65	0.07	15	38	0.06	13	48	0.19
SP-$GIES^{OI}$	**11**	23	**0.63**	**10**	**34**	**0.60**	**12**	47	**0.63**	**12**	35	0.33	**6**	**30**	0.45
NULL	12	27	0.57	11	53	0.58	14	61	0.57	**12**	36	**0.56**	11	46	0.56

that best fits the data distribution. A similar argument can be made for the AUC-PR. The drawback of the SID is the computational complexity is quadratic in the number of nodes for sparse networks, and cubic in the number of nodes for dense networks. This is shown empirically up to 50 nodes by Peters and Bühlmann [2015].

3.6 Results

Results of our evaluation on random networks, DREAM4 networks, and genome-scale networks are shown in Table 3.3, Table 3.4, and Table 3.5 respectively. The observational learners we evaluated are PC, ARACNE, and CLR. The interventional learners we evaluated are GIES, IGSP, Pinna and SP-GIES. We did not include MCMC Mallows in our evaluation because of the scalability challenges associated with Monte Carlo sampling. See Rau et al. [2013] for an evaluation of MCMC Mallows on small DREAM4 and random datasets. As a reference, we also include the scores for a network without any dependencies (no edges), named NULL. For CLR and ARACNE, a threshold value must be chosen to filter edges. For CLR with the random and DREAM4 networks, we chose values so that approximately the top 10% of edges were retained. For CLR with the RegulonDB dataset, we followed

Table 3.5: Evaluation of learners on large-scale networks. RegulonDB has 1146 nodes and 3179 edges. Small-World has 1000 nodes and 1000 edges. Pinna et al. [2010] requires one interventional sample per gene, since the RegulonDB dataset does not provide this we did not evaluate Pinna on this dataset. GES^{O} required setting the maximum degree to 10 in order to achieve convergence. IGSP did not converge in 72 hours for these networks.

Algorithm	**RegulonDB**			**Small-World**		
	SHD	SID	AUC-PR	SHD	SID	AUC-PR
$ARACNE\text{-}AP^{O}$	3,752	25,979	0.04	1,000	3,883	0.50
CLR^{O}	**3,095**	**18,069**	0.30	2,620	116,560	0.09
PC^{O}	3,963	23,908	0.01	467	2,447	0.82
GES^{O}	8,712	74,224	0.005	1,523	**879**	0.88
$GIES^{OI}$	8,355	80,580	0.006	1,623	1,097	0.84
$Pinna^{OI}$	–	–	–	16,577	4,334	0.002
SP-GIES^{OI}	3,154	22,114	0.10	**341**	975	**0.91**
NULL	3,179	18,294	**0.50**	1,000	3,883	0.50

the work of Faith et al. [2007] and chose the threshold that resulted in 60% precision. For ARACNE, we used the threshold associated with a p-value of 10^{-8} – this is the default used by Lachmann et al. [2016].

For random networks of size 10, the best performing learner depends on the network type. GES with no interventional data performs the best on scale free networks. SP-GIES performs best on small world networks. Both GIES and SP-GIES perform comparably for Erdös Renyi networks. For DREAM4, SP-GIES most frequently gets the best score across all metric types over all five networks. We see that the addition of interventional data improves the performance of algorithms that can handle both data types (namely GIES and SP-GIES).

For the RegulonDB network, the best scoring method is CLR. This is because the CLR method calculates a threshold value to prune edges so that the learner achieves 60% precision compared to the ground truth network – this makes it an unfair comparison since the ground truth network must be known a priori. We used CLR for the skeleton estimation of SP-GIES. An unexpected result is that SP-GIES appears to worsen the initial estimate of the CLR skeleton. To understand this, we zoomed in on a subnetwork of RegulonDB that contains

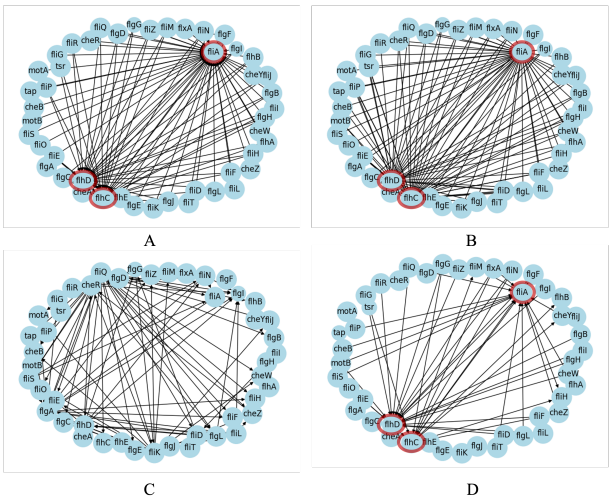


Figure 3.5: (A) The subnetwork of the ground truth network for the RegulonDB dataset, which contains 3 hubs highlighted in red. (B) The estimated undirected subnetwork given by CLR, which correctly identifies the 3 hubs. The threshold was chosen so that 60% of the ground truth network was recovered. (C) The subnetwork estimated by SP-GIES using the CLR skeleton. With real data, no hubs are detected. (D) The subnetwork estimated by SP-GIES using the CLR skeleton with synthetic data. The 3 hubs are detected, although the graph is more sparse than the CLR estimate and ground truth network.

three hubs (highly connected nodes). Fig. 3.5 shows that CLR correctly identifies 3 hubs in the subnetwork. However, SP-GIES removes many edges from this initial estimate and adds new edges elsewhere. As a result, no hubs are detected using SP-GIES. One possible reason is that the GIES learner uses the maximum likelihood estimate to score each candidate graph. This calculation assumes the data is sampled from a linear Gaussian distribution – which is not the case for real datasets like RegulonDB where nonlinear effects may be present. Since mutual information can capture nonlinearity, we do not see this issue reflected in CLR. To test this theory, we generated synthetic linear Gaussian data from the RegulonDB subnetwork topology. Fig. 3.5 shows that with synthetic data the SP-GIES can detect 3 hubs in the subnetwork. The nonlinearity of the dataset is only a partial explanation because the SP-GIES estimate with synthetic data is still further from the ground truth compared to the CLR estimate. Further analysis is needed to understand the data regimes and network topologies within the scope of joint learners like SP-GIES.

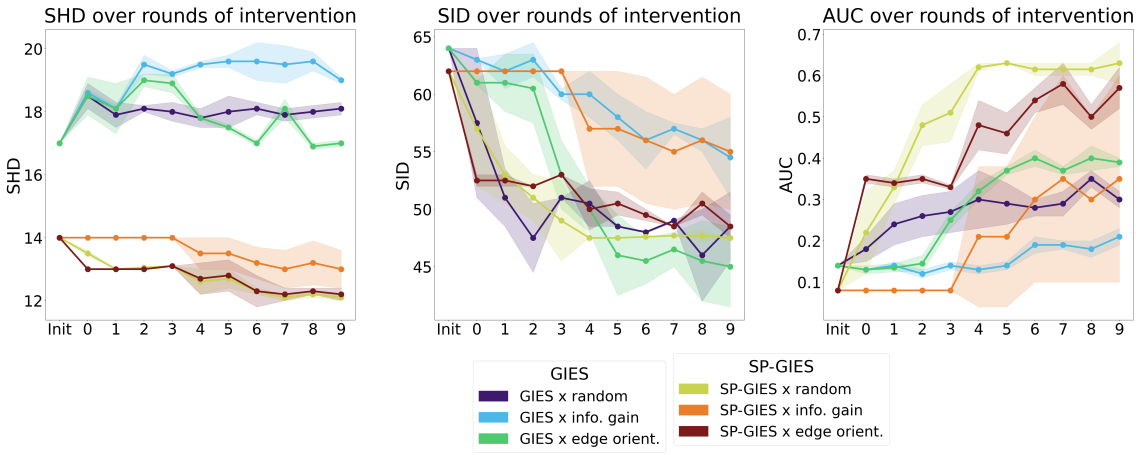


Figure 3.6: An evaluation of OED strategies for DREAM4 10 node network #3 over 10 rounds of intervention. At each round of intervention, a new interventional data sample is added to the current dataset corresponding to the chosen intervention. GIES and SP-GIES are joint learners. Metrics are averaged over 30 runs.

For a few of the networks we evaluated here, the NULL graph achieves the best score. Many real world networks, including gene regulatory networks, are sparse. As a result, the NULL learner often appears to perform very well since the true networks lie close to an empty network in combinatorial space. Interestingly, the NULL graph never scores best for the SID metric. This could indicate that causal network recovery methods should be evaluated with the SID rather than other metrics. However, more analysis is needed to understand the validity of this and is outside the scope of this paper.

To understand the performance gains of SP-GIES over GIES on optimal experimental design, we evaluated both algorithms with two different OED criteria on the DREAM4 insilico dataset. We limited this evaluation to small networks since the time complexity of OED includes posterior sampling and optimal gene selection for each round of intervention. We follow the framework of Agrawal et al. [2019], which includes bootstrap sampling of the posterior distribution. An example result for a DREAM4 insilico network is shown in Fig. 3.6. We observe three trends here. First, the edge orientation strategy achieves better performance than the information gain strategy. Second, the SP-GIES learner boosts the initial performance of the learner, however, it easily stagnates and additional interventional

samples do not continue to improve the algorithm at the same rate as the GIES based strategies. Still the SP-GIES x OED runs achieve better performance than GIES x OED, and there is some improvement in accuracy with the addition of interventional data. We speculate that stagnation may happen with the SP-GIES joint learner because by fixing the skeleton, we also restrict the search space of the learner. This may result in the GIES step of the algorithm getting stuck in a local minima. This motivates incorporating some level of uncertainty into the skeleton, or randomly including sets of edges outside the skeleton into the search space – we leave this to future work. Third, we see that the random strategy is comparable to the OED strategies. In other words, prioritizing certain experiments over others does not necessarily result in a better estimate of the causal network. One possible explanation is that the network topology may be such that intervening on certain nodes is not advantageous compared to others.

Another reason for the poor performance of the OED strategies is that they rely on a good model (causal graph) to estimate the posterior distribution. Without a good model, the calculation of the utility function will be inaccurate and the wrong experiments may be prioritized. Although SP-GIES provides us with a better estimate of the causal graph compared to GIES, it may not be sufficient for OED. This is analogous to what we see in active learning; when the model performance is poor, a random labeling of new samples is effective for model improvement.

3.7 Discussion

Understanding genome-scale networks is important for building a causal mapping from genotype to phenotype. However, recovery of genome-scale networks from interventional datasets has not yet been realized because of the computational complexity of structure learners that jointly learn from mixed data. Existing methods like GIES, IGSP, MCMC Mallows and others are only evaluated on small networks up to 500 nodes and exhibit worst case exponential

complexity. This makes subsequent optimal experimental design techniques computationally intractable since the complexity is then multiplied by the computation of choosing the optimal intervention. This is a huge issue in realizing causal discovery at scale via autonomous design loops.

In this paper, we note that structure learners built for observational datasets have parallel implementations on multiple processors and on GPUs. Our SP-GIES learner first estimates a skeleton using these parallel implementations. This restricts the edge set for the subsequent joint structure learner and improves the time to solution. However, SP-GIES is still bottlenecked by the scaling of the joint learner GIES – for example running network sizes of 10^4 nodes did not complete in 24 hours. To realize recovery of genome-scale networks on the order of 10^3-10^4, and to sample an interventional space on the order of 10^4-10^5, we must have breakthroughs in distributed parallel implementations of joint learners. Joint structure learners currently do not have parallel implementations because they are score-based learners (or hybrid constraint/score-based learners). At each step of the algorithm the candidate graph is scored, typically using a function of the likelihood $P(D|G)$ over the entire graph. As we saw in Section 3.2, the joint distribution of a DAG is a product of conditional probabilities which, for each node, depends on the parent nodes. As a result, the calculation of the likelihood is sequential and typically optimization over this space is done greedily. It is not intuitive how the graph may be partitioned into subgraphs for individual compute kernels. Constraint based methods, on the other hand, use pairwise conditional independence testing to resolve edges. A naive approach to scaling is to generate a separate compute thread per calculation, however, one can group variables into blocks to further minimize computations and boost performance. This type of performance enhancement has not yet been realized by score-based learners, and this presents a roadblock for existing joint learners.

The performance of SP-GIES on the RegulonDB dataset – which is the most biologically relevant dataset – suggests that future work should also incorporate nonlinearity into joint

learners. One approach is to modify the GIES score function. Since closed form solutions for the maximum likelihood estimate only exist for a small set of functions, an alternative is to use a nonparametric learner like IGSP. However, IGSP does not scale to the size of the RegulonDB dataset. Moreover, the IGSP learner in general appears to perform poorly on random and DREAM4 networks. We note that the use case for this algorithm is considerably different than the datasets used here. IGSP was designed for cases where there is sufficient interventional data in order for hypothesis testing to be successful. For the datasets used here, only a handful of interventional samples are available for each node. Structure learning for nonlinear datasets has been studied in works such as Gretton et al. [2009], Yu et al. [2019], Zheng et al. [2018]. Recent works use neural networks by recasting the combinatorial optimization into a continuous optimization. However, these methods suffer in low data regimes. In order to realize network recovery at scale for real biological datasets, there is a need to incorporate strategies from nonlinear structure learners into joint learners.

In evaluating existing methods and designing the SP-GIES method, we found that learners and OED strategies are implemented in a varied set of languages including Java, MATLAB, R, Python and C. This is due to the diverse nature of backgrounds interested in causal discovery of biological networks including from biology, statistics, computer science, machine learning, and representation learning. There has been a recent interest in unifying tools for causal discovery – for example `CausalDiscoveryToolbox` Kalainathan and Goudet [2019] , `Pgmpy` Ankan and Panda [2015] and `CausalDAG` Chandler Squires [2018] are Python libraries that provide varying support for graphical and/or causal modeling. However, in the case of `CausalDiscoveryToolbox`, the library is a Python wrapper for R code, which in turn is a wrapper for C code. For `Pgmpy`, no joint interventional learners are implemented. `CausalDAG` provides the best overall framework for causal discovery with interventions, although many of the implementations are still incomplete. To realize better parallel implementations of causal algorithms and OED strategies, we should encourage collaborations with the supercomputing

community. Therefore, there is a need for exclusive Python or C implementations of these models and algorithms since these languages are popular in the supercomputing community.

3.8 Conclusion

We present SP-GIES, a joint structure learner that leverages parallel observational learners to estimate a skeleton and initialize the GIES joint learner. We provide a systematic evaluation of SP-GIES against existing methods for datasets at various scales and with various metrics. We see that SP-GIES is able to provide better network recovery accuracy for large scale networks up to 2,000 nodes and on biological networks up to 1,146 nodes. This scale of network recovery for joint learners has not been achieved to our knowledge. SP-GIES reaches a faster time to solution than existing method GIES and provides up to 3.87x speedup. We also show that SP-GIES improves subsequent OED strategies. Future work includes a distributed parallel implementation of SP-GIES, and incorporation of SP-GIES into an autonomous experimentation design loop.

CHAPTER 4

SCALING CAUSAL DISCOVERY TO THE "GENOME-SCALE" WITH DISTRIBUTED PARALLELISM

The aim in many sciences is to understand the mechanisms that underlie the observed distribution of variables, starting from a set of initial hypotheses. Causal discovery allows us to infer mechanisms as sets of cause and effect relationships in a generalized way—without necessarily tailoring to a specific domain. Causal discovery algorithms search over a structured hypothesis space, defined by the set of Directed Acyclic Graphs (DAG), to find the graph that best explains the data. For high-dimensional problems, however, this search becomes intractable and scalable algorithms for causal discovery are needed to bridge the gap. In this paper, we define a novel causal graph partition that allows for divide-and-conquer causal discovery with theoretical guarantees under the Maximal Ancestral Graph (MAG) class. We leverage the idea of a superstructure—a set of learned or existing candidate hypotheses—to partition the search space. We prove under certain assumptions that learning with a causal graph partition always yields the Markov Equivalence Class of the true causal graph. We show our algorithm achieves comparable accuracy and a faster time to solution for biologically-tuned synthetic networks and networks up to 10^4 variables. This makes our method applicable to gene regulatory network inference and other domains with high-dimensional structured hypothesis spaces. Code is available at `https://github.com/shahashka/causal_discovery_via_partitioning`.

4.1 Introduction

Causal discovery aims to find meaningful causal relationships using large-scale observational data. Causal relationships are often represented as a graph, where nodes are random variables and directed edges are cause-effect relationships between random variables [Spirtes

et al., 2000c]. Causal graphs have high expressive power as they allow us to investigate complex relationships between many variables simultaneously—making them relevant for many problems in science, economics, and decision systems [Pearl, 1995b].

Exploring the graph search space to find the causal graph is an NP-hard problem. Causal discovery algorithms have benefited from some performance enhancements and parallel strategies [Ramsey, 2015, Laborda et al., 2023, Lee and Kim, 2019]. Recent work explores a distributed divide-and-conquer version of causal discovery by partitioning variables into subsets, locally estimating graphs, and merging graphs to resolve a causal graph. Existing divide-and-conquer methods do not provide theoretical guarantees for consistency; meaning in the infinite data limit they do not necessarily find the Markov Equivalence Class of the true causal graph. Existing algorithms also rely on an extra learning step to merge graphs which can be computationally expensive. Finally, these algorithms ignore the violations to causal assumptions when learning on subsets of variables [Spirtes et al., 2000c, Eberhardt, 2017].

To address these limitations in literature, we propose a ***causal partition***. A causal partition is a graph partition of the hypothesis space, defined by a superstructure, into overlapping variable sets. A causal partition allows for merging locally estimated graphs *without* an additional learning step. We can efficiently create a causal partition from any disjoint partition. This means that a causal partition can be an extension to any graph partitioning algorithm.

We are interested in causal discovery for high-dimensional scientific problems; in particular, biological network inference. Biological networks are organized into hierarchical scale-free sub-modules [Albert, 2005, Wuchty et al., 2006, Ravasz, 2009]. The causal partition allows us to leverage the inherent, interpretable communities in these networks for scaling.

Our contributions are as follows: **(A)** We define a novel ***causal partition*** which leverages a superstructure and extends any disjoint partition. **(B)** We prove, under certain

assumptions, that learning with a causal partition is consistent without an additional learning procedure. **(C)** We show the efficacy of our algorithm on synthetic biologically-tuned networks up to 10,000 nodes.

4.2 Related Work

Causal discovery algorithms are categorized into two types: (i) Constraint-based algorithms use conditional independence tests to determine dependence between nodes [Spirtes et al., 2000c,b], and (ii) Score-based algorithms greedily optimize a score function over the space of potential graphs [Chickering, 2002b, Hauser and Bühlmann, 2012]. To address the intractable search space for causal discovery, many "hybrid" methods first constrain the search space with a constraint-based method and then greedily optimize the subspace using a score-based method [Tsamardinos et al., 2006, Nandy et al., 2018]. Perrier et al. [2008] formalize this approach by defining the superstructure $G = (V, E)$ where for a true causal graph $G^* = (V, E^*)$, $E^* \subseteq E$. The superstructure can be found using a constraint-based method like the PC algorithm, which is sound and complete. The superstructure can also be informed by domain knowledge e.g., for gene regulatory networks genes that are functionally related likely constrain underlying regulatory relationships [Cera et al., 2019]. Incorporating prior knowledge into causal discovery allows us to infer which hypotheses or known relationships are best supported by data.

Another approach to scaling causal discovery algorithms is the divide-and-conquer approach. In this approach, random variables are partitioned into subsets. Causal discovery is run on each subset in parallel before a final merge to resolve a graph over the full variable set. Huang and Zhou [2022] and Gu and Zhou [2020] use hierarchical clustering of the data to obtain a disjoint partition of variables. Similarly, Li et al. [2014] partition the node set using the Girvan-Newman community detection algorithm. Similar to our work, Zeng and Poh [2004] use an overlapping partition, however, they do not provide any theoretical guarantees

for learning. Tan et al. [2022] use an ancestral partition to restrict candidate parents for exact causal discovery using dynamic programming. Laborda et al. [2023] employ ring-based distributed parallelism. Our work differs from these because we use a superstructure G to partition nodes into overlapping subsets using a novel causal graph partition with theoretical guarantees. The causal partition avoids any additional learning step to combine subsets. We show that a causal partition can be an extension to any disjoint partition, allowing us to learn effectively on graphs of varying topologies. Finally, Zhang et al. [2024] also leverage a superstructure to partition the variable set and define a causal partition with similar properties to ours. However, the variable set can only be partitioned into two subsets using an optimal edge cut, meaning the scaling potential of this algorithm is limited. Our work has no constraints on the number of subsets, and as a result we can scale up to 10,000 variables.

4.3 Background

4.3.1 Causal Discovery

Causal discovery considers a set of data sampled from the joint distribution of random variables $\mathbf{X} \triangleq (X_1, \ldots, X_p)$ where p is the number of random variables in the system. Each random variable $X_i \in \mathbb{R}^n$ is defined as a real-valued column vector where each value is an individual observation for random variable X_i. We assume these relationships can be represented by a *Directed Acyclic Graph* (DAG). This DAG is a tuple $G^* = (V, E^*)$ where V is the node (or vertex) set made up of p nodes corresponding to the random variables, and $E^* \subset V \times V$ is the set of directed edges between nodes. For each directed edge $(X_i, X_j) \in E^*$, we refer to the source node of the edge (X_i) as the "cause" and the target node of the edge (X_j) as the "effect". The joint distribution of random variables is given by a probability density function that factorizes as:

Table 4.1: Table of Relevant Notation

Symbol	Description
G^*	Underlying true causal graph represented by a DAG.
H^*	CPDAG representing MEC of G^*.
G	Superstructure.
$\mathbf{X}$	The complete observed data matrix (of dimensionality $n \times p$).
$X_i \in \mathbf{X}$	Observational data for the i^{th} random variable; also used to denote nodes in graphical models.
(X_i, X_j)	Directed edge from random variables (nodes) X_i to X_j.
$\mathscr{A}$	Consistent causal learner that outputs an PAG on subsets S.
$\{S_1, \ldots, S_N\}$	Partition over node set V, where $S \subset V$.
$\partial_{\text{out}}(S)$	The outer vertex boundary of a set of nodes S.
$X_i \sim_{G'} X_j$	Nodes X_i and X_j are adjacent in some graph G'.

$$P(X_1 ... X_p) = \prod_i^p P\left(X_i | Pa^{G^*}(X_i)\right) \tag{4.1}$$

Where $Pa^{G^*}(X_i)$ is the set of parents of node i in G^*. Nodes that are *d-separated* in G^* imply a conditional independence in P. Let $X, Y \in V$ and $Z \subseteq V / \{X, Y\}$. If Z d-separates X from Y in DAG G^*, then the random variables X and Y are conditionally independent given Z. We assume access to only observational data. In this setting, causal discovery algorithms only estimate a graph within the *Markov Equivalence Class* (MEC) of G^*. The MEC of a causal graph G consists of the set of DAGs that share the same conditional independence relationships and therefore *d-separation* criteria. A *Completed Partially Directed Acyclic Graph* (CPDAG) is the graph class that represents the MEC of a DAG. In this paper we denote the MEC of the true DAG G^* as the CPDAG H^*. In particular H^* has the same adjacencies and unshielded colliders (triples with the following structure $i \to j \leftarrow k$ where i and k are not adjacent) as G^* [Zhang, 2008a]. As a helpful reference, we include relevant definitions in Table 4.1 .

4.3.2 Graph Classes for Latent Variables

While the causal graph can be represented by a DAG, we consider alternative graphical representations that consider latent (unobserved) variables. Namely, we consider two well-studied graph classes: *(i) Maximal Ancestral Graphs* (MAGs) and *(ii) Partial Ancestral Graphs* (PAG) [Richardson and Spirtes, 2003, Zhang, 2008a].

Definition 4.3.1 (mixed graph, MAG)**.** *A mixed graph G consists of a set of nodes V and a set of directed edges $E \subset V \times V$ and a set of bi-directed edges $B \subset V \times V$. If $(X_i, X_j) \in E$ we say there is a directed edge between X_i and X_j and we write $X_i \to X_j$. If $\{X_i, X_j\} \in B$ we say there is a bi-directed edge and write $X_i \leftrightarrow X_j$. A mixed graph is called a maximal ancestral graph (MAG) if it contains no almost directed cycles and there is no inducing path between non-adjacent nodes.*

An almost directed cycle is a cycle that contains both directed and bi-directed edges. An inducing path is defined as follows:

Definition 4.3.2 (Inducing path)**.** *Given $L \subset V$, an inducing path relative to L between vertices u and v is a path $\Pi = \{u, q_1, \ldots, q_k, v\}$ such that every non-endpoint node in $\Pi \cap \{V \setminus L\}$ is a collider on Π and an ancestor of at least one of u or v.*

Some examples of inducing paths are illustrated in Figure 4.1. The idea of *d-separation* in DAGs can be extended to *m-separation* in mixed graphs [Zhang, 2008a]. The graph class that characterizes the Markov Equivalence Class of MAGs, governed by *m-separation*, is the partial ancestral graph [Richardson and Spirtes, 2003].

Definition 4.3.3 (partial mixed graph, PAG)**.** *A partial mixed graph can contain four kinds of edges: $\to, \circ\!\!-\!\!\circ, -$, and $\circ\!\!\to$ and therefore has three kinds of end marks for edges: arrowhead (>), tail (-) and circle ($\circ$).*[1] *Let $[M]$ be the Markov equivalence class of an arbitrary MAG*

1. Additionally, we will use $*$ as a "wild card" end mark. For example $u *\!\!\to v$ means that the end mark at u can be any of three outlined in the Defn. 4.3.3.

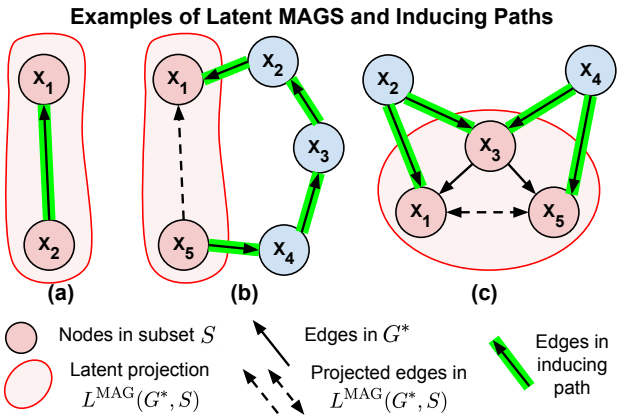


Figure 4.1: Examples of latent MAGS $L^{\mathrm{MAG}}(G^*, S)$. Inducing paths Π relative to $V \setminus S$ are highlighted in green. (a) For $x_1, x_2 \in S$, any edge (x_1, x_2) in G^* is an inducing path relative to $V \setminus S$ between x_1 and x_2. (b) Π is an inducing path relative to $V \setminus S$ between x_1 and x_5 because all non-endpoint nodes on the path are in $V \setminus S$. (c) Π is an inducing path relative to $V \setminus S$ between x_1 and x_5 because every non-endpoint is either in $V \setminus S$ (nodes x_2, x_4), or is in S *and* is a collider on the path *and* is an ancestor of at least one of x_1 or x_5 (node x_3).

M. The partial ancestral graph (PAG) for [M], PAG[M], is a partial mixed graph such that ***(i)*** *PAG[M] has the same adjacencies as M (and any member of [M]) does;* ***(ii)*** *A mark of arrowhead is in PAG[M] if and only if it is shared by all MAGs in [M]; and* ***(iii)*** *A mark of tail is in PAG[M] if and only if it is shared by all MAGs in [M].*

We will prove, that under certain assumptions, we can reconstruct the CPDAG representing the MEC (H^*) of the true DAG (G^*) from PAGs estimated on subsets of variables.

4.3.3 Causal Discovery on Subsets of Variables

We now describe the problem setup for learning over subsets of variables. Column-wise subsets of $\mathbf{X}$ are marked with a subscript: e.g., for a subset of nodes S, the corresponding subset of data is $\mathbf{X}_S = \{X_i^n\}_{i \in S}$. The presence of latent variables outside the subset S complicates our learning procedure. We must use MAGs rather than DAGs to represent graphs estimated on subsets of variables to ensure consistency of our algorithm. To this end we define a latent projection, as used by Zhang [2008a], of the true graph G^* onto a subset of nodes S. An example is shown in Fig. 4.1.

Definition 4.3.4 (Latent MAG)**.** *Let G be a DAG with variables V and $S \subset V$, where V*

contains no selection variables.[2] *The latent MAG* $L^{MAG}(G, S)$ *is the MAG that contains all nodes in* S *and satisfies:*

1. $u, v \in S$ *and* $u \to v \in G \Rightarrow u \to v \in L^{MAG}(G, S)$
2. *(projected edge)* $\in L^{MAG}(G, S)$ *if there is an inducing path between* u *and* v *relative to* $V \backslash S$ *in* G^**. The edge is directed* $u \to v$ *if* u *is an ancestor to* v *in* G^**. The edge is directed* $v \to u$ *if* v *is an ancestor to* u *in* G^**. Otherwise the edge is bi-directed* $u \leftrightarrow v$*.*

Latent projections are well-studied objects in the causal discovery literature, see [Verma and Pearl, 2022, Faller et al., 2023, Richardson et al., 2023, Zhang, 2008a] for further definitions. A ground-truth DAG G^* induces a *latent MAG* $L^{\text{MAG}}(G^*, S)$ on a subset S. The Markov equivalence class of this MAG is denoted $[L^{\text{MAG}}(G^*, S)]$.

Next, we assume that the structure learner employed on each subset is a complete and consistent PAG learner, even in the presence of confounder variables. Algorithms known to satisfy these assumptions include the seminal FCI algorithm [Zhang, 2008b].

Assumption 1. *We have a consistent structure learning algorithm* $\mathscr{A}$ *that operates on data matrix* X_S *for a subset of random variables* $S \subseteq V$*. When the distribution* P *satisfies faithfulness, then in the infinite data limit*

$$\mathscr{A}(X_S) = PAG[L^{MAG}(G^*, S)]$$

In particular, by definition of the latent MAG and latent PAG operators, Assumption 1 implies the output of $\mathscr{A}$ satisfies several properties.

Lemma 1. *Given* $\mathscr{A}$ *satisfying Assumption 1,*

2. There is no selection bias in our setting, since data is sampled from the full vertex set V which retains causal sufficiency.

1. *For any $x_i, x_j \in S$, the output $\mathscr{A}(X_S)$ has an edge between x_i and x_j if and only if there is an inducing path in G^* relative to $V \setminus S$ between them.*

2. *For any triple $x_i, x_j, x_k \in S$ that form an unshielded collider in G^* as $x_i \to x_j \leftarrow x_k$, the output $\mathscr{A}(X_S)$ will have an edge between x_i and x_j as well as x_j and x_k, and both of these edges will have an arrowhead at x_j.*

3. *For any $u, v \in S$ such that $u \sim_{G^*} v$, if $u \sim_{\mathscr{A}(S)} v$ with an arrowhead at v in $\mathscr{A}(X_S)$, then $u \to v$ in G^*.*

The proofs for Lemma 1 are deferred to Appendix A.2. These properties, at a high level, allow us to determine the alignment of the adjacencies and the unshielded colliders in locally estimated graphs $\mathscr{A}(X_S)$ to the underlying DAG G^*. These will be important for resolving the CPDAG H^* using locally estimated graphs.

4.3.4 Defining a Causal Partition

Here, we outline the properties of our novel causal partition, which admits a divide-and-conquer algorithm to estimate H^* a CPDAG corresponding to G^* by learning over subsets. Since learning on the entire variable set with $\mathscr{A}(X_V)$ can be computationally intractable, we use an initial structure over the entire variable set to help partition V into subsets. We first assume access to an initial superstructure G.

Assumption 2. *We have access to superstructure $G = (V, E)$, an undirected graph, that constrains the true graph G^*. This means all edges in G^* are in G, but not all edges in G are necessarily in G^*.*[3]

Consider some overlapping partition $\{S_1, \ldots, S_N\}$ of V, and the output $\{\mathscr{A}(X_{S_i})\}_{i=1}^N$.

3. This assumption is not required to prove identifiability of H^*, rather it allows us to define the causal partition when the superstructure is not fully connected, and therefore, when we can exploit the communities in the superstructure to enable scaling.

Using Assumption 1, we show that given a partition with a particular structure defined below, one can recover H^* from $\{\mathscr{A}(X_{S_i})\}_{i=1}^N$.

Definition 4.3.5 (Causal Partition)**.** *We say an overlapping partition $\{S_1, \ldots, S_N\}$ is **causal** with respect to superstructure G and ground-truth DAG G^* if, given any learner $\mathscr{A}$ satisfying Assumption 1, all of the following hold:*

(i) The partition is edge-covering with respect to the superstructure G.

(ii) For any vertices u, v such that $u \not\sim_{G^*} v$ and $u \sim_G v$, there exists some subset S_i such that $u, v \in S_i$ and $\mathscr{A}(X_{S_i})$ does not contain an edge between u and v.

(iii) For any unshielded collider $u \to v \leftarrow w$ in G^*, there exists some subset S_i such that $\{u, v, w\} \subseteq S_i$.

In particular, property (ii) in Definition 4.3.5 is crucial to the divide-and-conquer strategy proposed in this work, as it allows the algorithm to identify and discard projected edges learned on a subset S_i (as in Defn 4.3.4) by comparing the output $\mathscr{A}(X_{S_i})$ to results on other subsets. In Section 4.5.1, we show that given a superstructure satisfying Assumption 2, a simple and computationally tractable procedure yields a causal partition satisfying all above properties.

4.4 Guarantees in the Infinite Data Limit

Now we prove that given any causal partition $\{S_1, \ldots, S_N\}$ with respect to DAG G^* and superstructure G, one can recover H^* a CPDAG corresponding to G^*. Our main theorem states that Algorithm 1 recovers H^* from local output $\{\mathscr{A}(X_{S_i})\}_{i=1}^N$.

Theorem 1. *Given superstructure G satisfying Assumption 2, a learner $\mathscr{A}$ satisfying Assumption 1, and $\{S_1, \ldots, S_N\}$ a causal partition with respect to G and G^*, let H^* denote*

the output of Algorithm 1

$$H^* = \texttt{Screen}(G, \{\mathscr{A}(X_{S_i})\}_{i=1}^N).$$

Then H^* satisfies the following properties: ***(i)*** $\forall u, v \in V$, $u \sim_{H^*} v$ *if and only if* $u \sim_{G^*} v$*;* ***(ii)*** *For any unshielded collider $u \to v \leftarrow w$ in H^*, it holds that $u \to v \leftarrow w$ in G^*; and* ***(iii)*** *For any unshielded collider $u \to v \leftarrow w$ in G^*, $u \sim_{H^*} v$ and $v \sim_{H^*} w$ and both edges have an arrowhead at v in H^*.*

Property (i) in Theorem 1 states that H^* contains the same adjacencies as G^*. Properties (ii) and (iii) combine to imply that an unshielded collider $u \to v \leftarrow w$ appears oriented in H^* if and only if that unshielded collider exists in G^*. These combined properties ensure that H^* is the CPDAG that represents the MEC of G^*.

The proof of Theorem 1, included in Appendix A.2, relies on the fact that by definition of a causal partition, for any u, v not adjacent in G^*, there must be a subset S_i such that $u, v \in S_i$ and the local output $\mathscr{A}(S_i)$ does not contain an edge between u and v. This allows us to "screen" projected edges from true edges as edges that are not consistent across all locally estimated graphs.

We note that `Screen` is computationally lightweight. The dominant cost is $O(N \cdot m' \cdot d)$, for N the number of partitions, m' the total number of learned edges, and d the maximum degree in the learned graph. Of note, $m' \leq p^2$ for p the number of random variables, and in real-world applications learned graphs tend to be sparse so typical instances have $m' \ll p^2$ [Barabási, 2013].

We highlight that because we only assume access to observational data, we can only recover cause-effect relationships contained in the Markov equivalence class of G^*, and therefore our guarantees relate to learning H^* a CPDAG representing the MEC of G^*. In Section 4.8, we discuss potential extensions of our framework for settings where both interventional and observational data are available, which might allow one to further reduce the size of the

learned equivalence class.

Algorithm 1: `Screen`$(G, \{H_i\}_{i=1}^N)$

Input: a superstructure G, a set of PAGS $\{H_i = (S_i, E_i)\}_{i=1}^N$

Result: $H^* = (V, E^*)$ a CPDAG

Initialize $V = \cup_{i=1}^N S_i$; $E_{\text{candidates}} \leftarrow \cup_{i=1}^N E_i$; $E^* \leftarrow \emptyset$;

`// Discard edges not in superstructure.`

$E_{\text{candidates}} \leftarrow E_{\text{candidates}} \cap \{u \ast\!\!-\!\!\ast\, v \mid u \sim_G v\}$;

foreach u, v *such that* $\{u \ast\!\!-\!\!\ast\, v\} \in E_{candidates}$ **do**

  if $\forall i$ *s.t.* $S_i \supseteq \{u, v\}$, $u \sim_{\mathcal{A}(S_i)} v$ **then**

    `// If an edge between` u `and` v `appears in the output on all subsets, add undirected edge to output graph.`

    $E^* \leftarrow E^* \cup \{u - v\}$;

`// Orient unshielded colliders`

foreach $i \in [N]$ **do**

  foreach *Unshielded* $u \ast\!\!\rightarrow v \leftarrow\!\!\ast\, w$ *in* H_i **do**

    if $u - v$ *and* $v - w$ *in* E^* **then**

      `discard` $\leftarrow \{u - v, v - w\}$;

      `orient` $\leftarrow \{u \rightarrow v, v \leftarrow w\}$;

      $E^* \leftarrow \{E^* \setminus$ `discard`$\} \cup$ `orient`;

return $H^* = (V, E^*)$

4.5 A Practical Algorithm for Causal Discovery with a Causal Partition

Here, we describe a practical procedure for causal discovery motivated by the idealized results studied in Section 4.4. We discuss how partitions satisfying Defn. 4.3.5 can be efficiently constructed, and detail a full end-to-end algorithm for causal discovery.

4.5.1 Efficient Creation of a Causal Partition

The causal partition structure, described in Defn. 4.3.5, is crucial to the guarantees of Theorem 1 in the infinite data limit. While the first property of a causal partition—edge coverage with respect to superstructure G—is easy to ensure, it is not obvious how to satisfy properties (ii) and (iii) without knowledge of the ground truth G^*. Here we present a simple and intuitive method for constructing causal partitions. This construction is efficient and adapts to arbitrary superstructure topologies.

Given a graph $G = (V, E)$ and $S \subseteq V$, let $\partial_{\text{out}}(S)$ denote the outer vertex boundary of set S in G:

$$\partial_{\text{out}}(S) \equiv \{v \in V(G) \setminus S : \exists u \in S \text{ such that } v \sim_G u\}$$

where $v \sim_G u$ if any of $(u, v), (v, u)$ or $\{u, v\} \in E$.

Given any initial vertex-covering partition of the superstructure G, we consider the overlapping partition formed by expanding subsets via the addition of vertices from the outer boundary.

Definition 4.5.1. *Let $\{S_1, \ldots, S_N\}$ be a vertex-covering partition of graph G. The causal expansion of $\{S_1, \ldots, S_N\}$ with respect to G is defined as $\{S'_1, \ldots, S'_N\}$ with subsets $S'_i = S_i \cup \partial_{out}(S_i)$.*

As the name suggests, we show that a causal expansion satisfies the properties of a causal partition. The proof is deferred to Appendix A.2.

Lemma 2. *Given G a superstructure satisfying Assumption 2, $\{S_1, \ldots, S_N\}$ a vertex-covering partition of G. Then the causal expansion $\{S'_1, \ldots, S'_N\}$ is a causal partition with respect to G and G^*.*

This simple construction, illustrated in Fig. 4.2, offers several advantages. Firstly, this method can be run on any vertex-covering initial partition $\{S_1, \ldots, S_N\}$. Graph partitioning

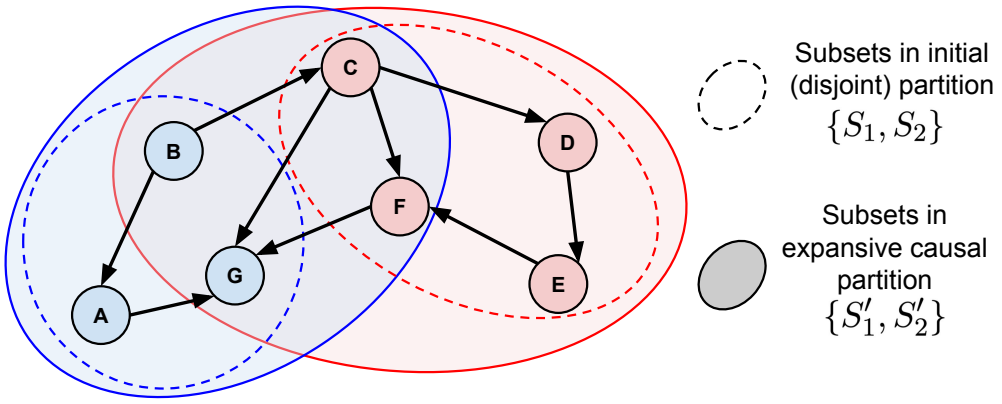


Figure 4.2: Expansive causal partition $\{S'_1, S'_2\}$ made from initial disjoint partition $\{S_1, S_2\}$.

algorithms form an extensive field [Girvan and Newman, 2002, Clauset et al., 2004, Schaeffer, 2007, Malliaros and Vazirgiannis, 2013, Harenberg et al., 2014], and depending on the topology of G different partitioning may be more appropriate to a specific superstructure. The causal expansion allows a user to first partition the superstructure G using whatever method is most appropriate to the application, and then easily derive a corresponding causal partition.

The causal expansion is computationally efficient, both to construct and in its incorporation into the full causal discovery procedure, described in Algorithm 2. Given an initial partition $\{S_1, \ldots, S_N\}$, constructing its causal expansion takes time linear in the size of the superstructure G. In section A.5, we discuss how connectivity properties of the initial partition $\{S_1, \ldots, S_N\}$ dictate the size of the largest subset.

Algorithm 2: `causal_discovery`(V, X, G)

Input: a set of variables V, a matrix of observations X, superstructure G

Result: $G_{\text{out}} = (V, E)$ a CPDAG

1 $\{D_1, \ldots, D_N\} \leftarrow$ `disjoint_partition`(G);

`/* construct causal expansion */`

2 $S_i \leftarrow D_i \cup \partial_{\text{out}}(D_i) (\forall 1 \leq i \leq N)$;

3 $\{G_{S_i} = \mathscr{A}(X_{S_i})\}_{i=1}^{N}$;

4 **return** $G_{\text{out}} \leftarrow$ `Screen`$(G, \{G_{S_i}\})$;

Now, we describe our divide-and-conquer causal discovery algorithm with an expansive causal partition as described in Section 4.5.1. Algorithm 2 requires a set of variables V, a data matrix X and a superstructure G. In Section 4.6.3 we also study the case where G is derived from data using the PC algorithm. Any causal learner can be plugged into $\mathscr{A}$, but for consistent learning we require that the assumptions for $\mathscr{A}$ allow for causal insufficiency (confounders may be present) and causal faithfulness. Any graph partitioning algorithm can be plugged into `disjoint_partition`. A complexity analysis of divide-and-conquer with an expansive causal partition is shown in section A.4. In the next sections we show the use of this practical algorithm on random and biologically-tuned networks with synthetic data.

4.6 Empirical Results on Random Networks

In this section we describe experiments for evaluating Algorithm 2 on synthetic random networks with finite data. For causal discovery on subsets (i.e., $\mathscr{A}$) we evaluate with four different algorithms: (1) Peters-Clark (PC) [Spirtes et al., 2000c], (2) Greedy Equivalence Search (GES) [Hauser and Bühlmann, 2012], (3) Really Fast Causal Inference (RFCI) [Colombo et al., 2012], and (4) DAGMA [Bello et al., 2022]. Note that only RFCI is a PAG learner that satisfies Assumption 1. The other algorithms are DAG learners that assume causal sufficiency; still we include them in this evaluation because (a) they are popular causal discovery benchmarks, and (b) even with the violation to causal sufficiency, we observe good performance with the causal partition. For details on how the DAG subgraphs are merged see section A.6. For `disjoint_partition` in Algorithm 2 we use greedy modularity based community detection [Clauset et al., 2004]. We benchmark our algorithm with another divide-and-conquer method *PEF* [Gu and Zhou, 2020].

For evaluation, we use the following metrics: (1) *True Positive Rate* (TPR) of correct edges in the estimated graph, $\hat{G}$, compared to the edges in G^*; and (2) *Structural Hamming Distance* (SHD), which is the number of incorrect edges. An incorrect edge is any edge in

G^* that is missing in $\hat{G}$ or any edge in $\hat{G}$ that is not in G^*. Additionally for larger networks we include (3) *False Positive Rate* (FPR); and (4) *Time*.

Default parameters: We use the following parameters by default unless stated otherwise. The graph topology is constructed by generating two scale-free networks using the Barabási-Albert generative model [Barabási and Bonabeau, 2003]; both graphs have $p = 50$ nodes each, with one graph being constructed with a $m = 1$ edge per node and the second graph being constructed with $m = 2$ edges per node (edge connections are established via preferential attachment as per the generative model). We use $n = 100,000$ samples from the joint, multivariate Gaussian distribution (details on the DAG and data generating process are in Appendix A.6.1). The fraction of additional edges in a perfect superstructure G is 10% of the edges in G^* (all edges in G are undirected). For causal discovery on subsets we set $\mathscr{A}$ to PC, GES, RFCI or DAGMA. Finally, for `disjoint_partition` in Algorithm 2 we use greedy modularity [Clauset et al., 2004] from the `networkx` Python library. The parameter settings for $\mathscr{A}$ and each partitioning algorithm are detailed in Appendix A.6.2 and A.6.3.

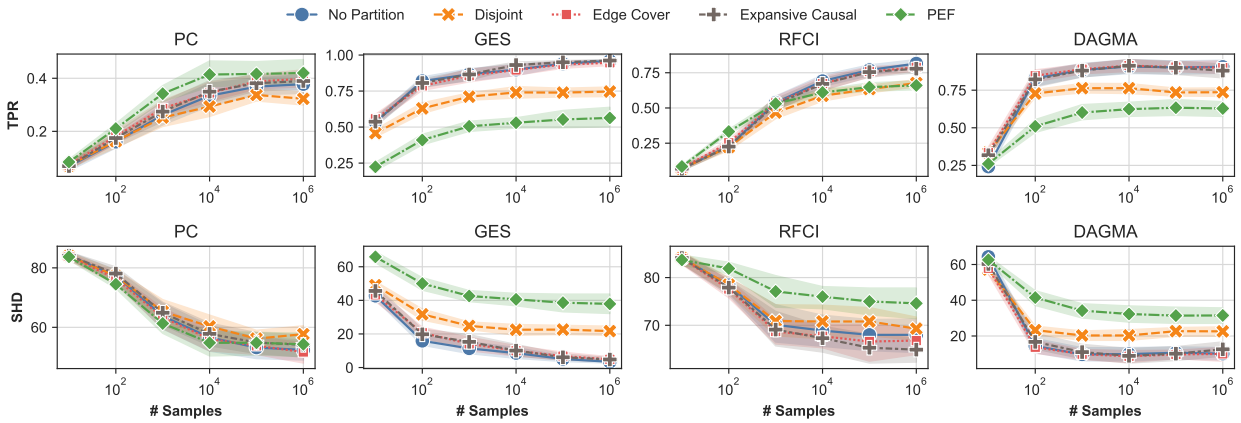


Figure 4.3: Experiment increasing the number of samples $\boldsymbol{n}$. Error bars are 95% confidence intervals.

4.6.1 *Number of samples*

In this experiment, we test the consistency of Algorithm 2 with increasing samples n. We use a perfect superstructure and add a fraction 10% extra extraneous edges to G that are not in G^*. Results are shown in Fig. 4.3. As the sample size increases, we see the convergence of *No Partition* with the MEC of G^*. This empirically supports our theoretical result that Algorithm 2 is consistent in the infinite data limit. Interestingly, even when the $\mathscr{A}$ does not permit latent variables (as in PC, GES, DAGMA), we still see convergence of *No Partition* with *Expansive Causal*. We also show results for an *Edge Cover* partition; this partition only accounts for edge coverage of G (*(i)* in Defn 4.3.5). We see the *Edge Cover* partition performs comparably to the *Expansive Causal* partition. This implies that of the properties of a causal partition described in Defn. 4.3.5, edge coverage appears to be the most important. With the exception of the PC algorithm, we also outperform the benchmark *PEF*.

4.6.2 *Density of superstructure* G

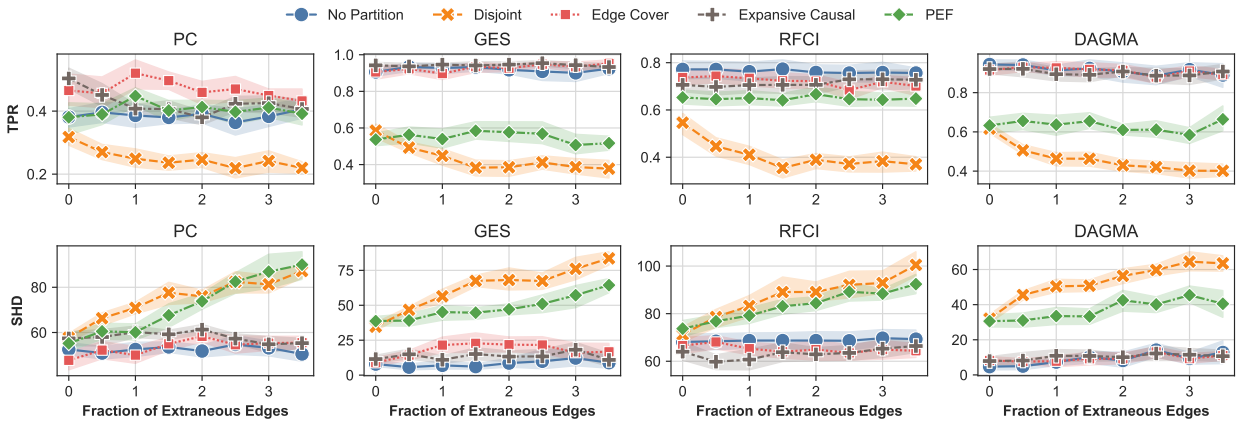


Figure 4.4: Experiment increasing fraction of extraneous edges in a perfect superstructure.

This experiment assumes a perfect superstructure G. We increase the fraction of extraneous edges in G and not in G^*. In Fig. 4.4, we see comparable learning of *Edge Cover*, *Expansive Causal*, and *No Partition*. This means that although G^* is increasingly obscured

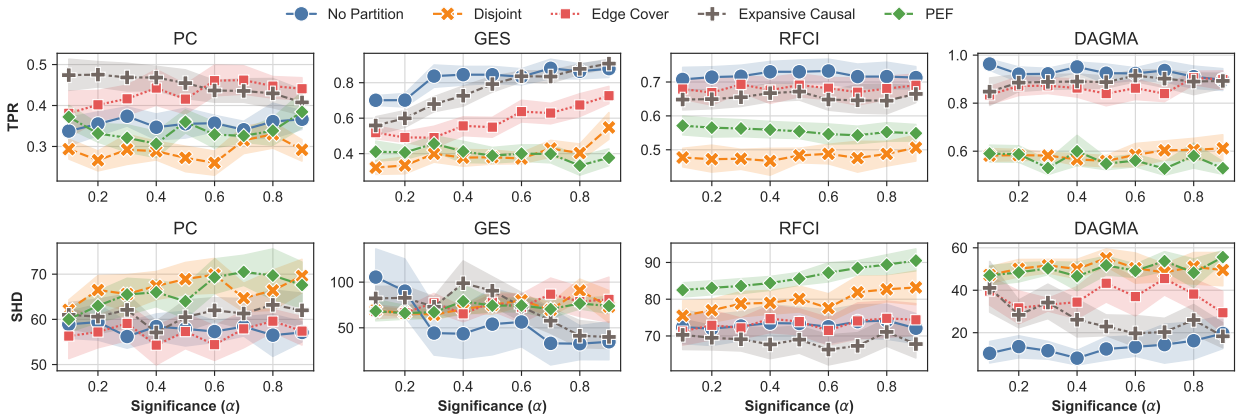
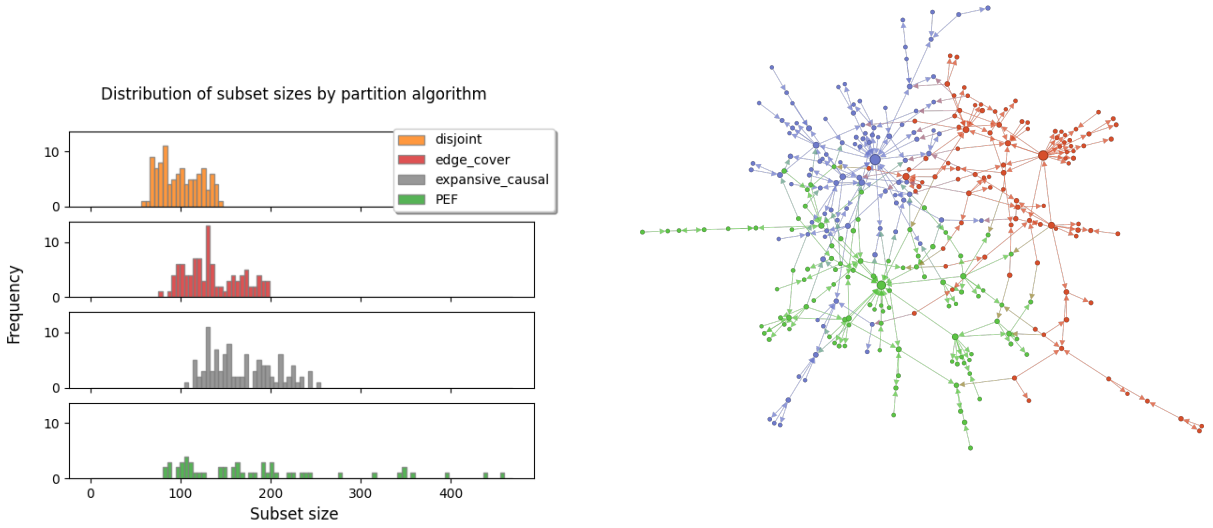


Figure 4.5: Increase in density of the imperfect superstructure by increasing the significant level $\boldsymbol{\alpha}$ of the PC algorithm.

by G, and even though partitioning is done on G, we can still estimate close to the MEC H^*.

4.6.3 Imperfect superstructure G

In this experiment we use the PC algorithm to estimate the superstructure G. Since the superstructure now relies on the data, it is imperfect and does not include all edges in G^*. We vary the "perfection" of the superstructure by increasing the the significance level α of the PC algorithm. A larger α means a denser superstructure and a structure that is more likely to include more edges in G^*. Results are shown in Fig. 4.5. We turn off the superstructure screening step, as in `Screen`, for this experiment. For the GES and DAGMA causal learning algorithms, *Expansive Causal* outperforms *Edge Cover* slightly – unlike in previous experiments. Still, the edge coverage property of the causal expansion accounts for most of the improvement in accuracy compared to a disjoint partition. The causal partition may provide additional benefits to learning when the superstructure G is imperfect.



(a) Subset Sizes

(b) Subset of the topology

Figure 4.6: Topological structure of 10,000 node graphs defined by case **(A)**. This network has 10,000 nodes. The nodes are divided into 100-node subsets, of which there are 100. To generate the edges in the network, we first generate a Barabasi-Albert scale-free graph on each 100-node subset, and then randomly add edges connecting these subsets together. **(a)** Distribution of subset sizes for each partitioning algorithm. **(b)** Example communities extracted from the 10,000 node network. These three communities account for 3% of the total nodes and 2.81% of the total edges. The size of the node is proportional to its degree.

4.6.4 Number of Nodes

In this experiment we highlight the scalability of our algorithm by evaluating on networks with 10,000 nodes. We test two network structure cases here: **(A)** one hundred communities each with 100 nodes with a Barabasi-Albert scale-free topology; this is identical to the preceding experiments, except with more communities, and **(B)** hierarchical scale-free graphs which are characterized by highly connected hub nodes that are preferentially attached to other hubs. This is similar to gene regulatory networks [Yu and Gerstein, 2006], but these structures are more sparse than typical biological networks. For both networks we obtain 10,000 samples from the multivariate Gaussian distribution. Unlike the preceding experiments, in this experiment we show results for when the sample size is equal to the number of variables: $n = p$.

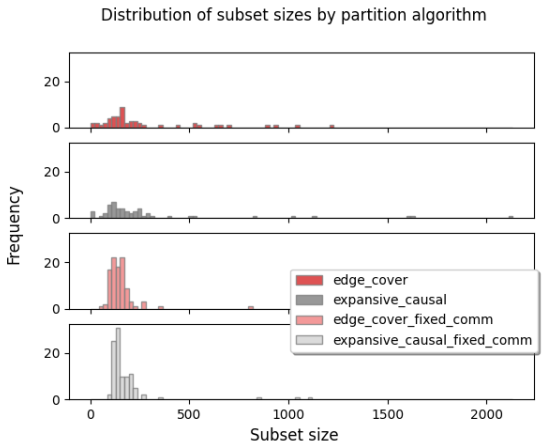
In Table 4.2 we show results for **(A)** across each divide-and-conquer method and differ-

Table 4.2: Average time and accuracy results on 10,000 node graphs with $\sim$ 10,000 edges comprised of 100 scale free networks each of size 100 (averaged over over 5 networks). The number of samples n is 10,000. Dash (-) indicates the algorithm did not complete in 24 hours. This experiment was run with 2x AMD EPYC 7713 CPU @2GHz with a total of 128 cores and 256 GB of RAM.

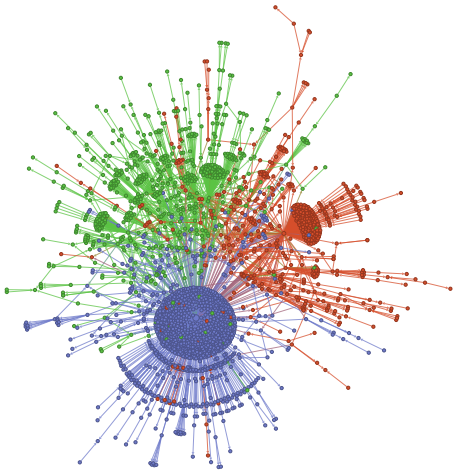
Algorithm	**Partition**	SHD ↓	TPR ↑	FPR ↓	Time (min) ↓
$\mathscr{A}$ = GES	No Partition	**857.2 ± 35.6**	0.930 ± 0.002	**9.0e-6**	11.695 ± 0.230
	Disjoint	4382.8 ± 201.8	0.650 ± 0.011	1.0e-5	**0.119 ± 0.005**
	Edge Cover	1306.4 ± 35.4	0.895 ± 0.002	1.3e-5	0.148 ± 0.007
	Expansive Causal	1550.6 ± 54.3	**0.947 ± 0.003**	1.5e-5	0.163 ± 0.009
	PEF	3775.8 ± 119.3	0.697 ± 0.010	3.7e-5	103.767 ± 3.067
$\mathscr{A}$ = PC	No Partition	–	–	–	–
	Disjoint	8519.2 ± 227.0	0.322 ± 0.007	**4.9e-5**	**4.891 ± 8.716**
	Edge Cover	**7050.8 ± 195.0**	0.443 ± 0.004	6.3e-5	10.343 ± 11.363
	Expansive Causal	8683.0 ± 401.0	**0.504 ± 0.014**	7.4e-5	450.744 ± 187.366
	PEF	–	–	–	–
$\mathscr{A}$ = RFCI	No Partition	–	–	–	–
	Disjoint	10294.0 ± 191.5	0.491 ± 0.013	6.5e-5	**0.715 ± 0.311**
	Edge Cover	**9683.4 ± 288.4**	**0.691 ± 0.005**	8.9e-5	5.430 ± 4.210
	Expansive Causal	9924.5 ± 113.8	0.647 ± 0.003	9.0e-5	107.788 ± 11.204
	PEF	–	–	–	–
$\mathscr{A}$ = DAGMA	No Partition	–	–	–	–
	Disjoint	3722.6 ± 274.7	0.694 ± 0.015	**1.0e-6**	**1.172 ± 0.107**
	Edge Cover	**925.6 ± 240.8**	**0.925 ± 0.019**	2.0e-6	2.502 ± 0.234
	Expansive Causal	1828.0 ± 36.5	0.858 ± 0.006	3.0e-6	3.593 ± 0.607
	PEF	4135.6 ± 84.6	0.667 ± 0.005	3.1e-5	122.405 ± 5.736

ent causal discovery algorithms $\mathscr{A}$. Time to solution for the divide-and-conquer methods (*Disjoint*, *Expansive Causal*, *Edge Cover*, and *PEF*) includes partitioning into subsets. The time to solution is bounded by the compute time of the largest subset (see Appendix A.4). The distribution of subset sizes shown in Fig. 4.6a. As a result, many algorithms did not complete in 24 hours with *No Partition* (PC, RFCI, DAGMA) and *PEF* (PC, RFCI). For PC, RFCI and DAGMA our *Edge Cover* algorithm generally outperforms other methods in SHD and TPR while incurring a small cost in compute time compared to the fastest algorithm *Disjoint*. This further supports the claim that *Edge Cover* may be sufficient for many problems. For GES, however, we see the increased benefit of the *Expansive Causal* partition compared to *Edge Cover* for the TPR. Since *No Partition* runs in a few minutes on this network, it may be unclear why partitioning is needed. In the next example we will see

how the compute times increase dramatically when the network has a much more complex community structure.



(a) Subset Sizes



(b) Subset of the topology

Figure 4.7: Topological structure of 10,000 node graphs defined by case **(B)**: a 10,000 node hierarchical scale-free network . **(a)** Distribution of subset sizes for each partitioning algorithm. **(b)** Example communities extracted from the 10,000 node network with the *Disjoint* partition. These three communities account for 25% of the total nodes and 20.5% of the total edges. Each community has a set of hub nodes that connected to a large number of other nodes (as seen by the large clusters of nodes in the visualization).

Table 4.3: Average time and accuracy results on 10,000 node hierarchical scale-free graphs with $\sim$ 10,000 edges (averaged over 5 networks). The number of samples n is 10,000. We show results only for $\mathscr{A}$ = GES. This experiment was run with an AMD Zen 3 (Milan) 7543P CPU @2.8 GHz with 64 cores and 512 GB of RAM.

Partition	SHD ↓	TPR ↑	FPR ↓	Time (hrs.) ↓
No Partition	**333.0 ± 36.8**	**0.976 ± 0.003**	**3.0e-6**	25.46 ± 17.24
Edge Cover	1214.6 ± 40.6	0.913 ± 0.004	9.0e-6	1.84 ± 0.02
Expansive Causal	987.2 ± 27.1	0.928 ± 0.002	7.0e-6	11.96 ± 5.72
Edge Cover 100 Comms	3506.4 ± 563.9	0.752 ± 0.040	1.4e-5	**0.04 ± 0.02**
Expansive Causal 100 Comms	2510.8 ± 72.3	0.821 ± 0.002	8.0e-6	1.97 ± 0.28

In Table 4.3 we show results for **(B)** with $\mathscr{A}$ = GES. The subsets of this network are larger and more dense compared to **(A)** (see Fig. 4.6b compared to Fig. 4.7b); the GES algorithm takes significantly longer on this network topology. Our *Expansive Causal* achieves a faster time to solution compared to *No Partition* while maintaining the second highest accuracy

score. Compared to *No Partition*, *Expansive Causal* provides 2.13x speedup and *Edge Cover* provides 13.8x speedup. Note that *PEF* did not converge in 72 hours.[4]

In *Expansive Causal 100 Comms* and *Edge Cover 100 Comms* we set the number of subsets to one hundred and ensure the size of the largest subset is smaller for each partitioning algorithm (Fig. 4.7a). We see significant speedup (12.9x for *Expansive Causal 100 Comms* and 606x for *Edge Cover 100 Comms*) compared to *No Partition*. However, this does come at a cost to accuracy as seen in Table 4.3. We present an initial study of the subset size, speedup, and accuracy trade off in section A.7. *No Partition* achieves the best accuracy, particularly with respect to SHD and FPR, however now the compute time is close to 25 hours, motivating the need for partitioning. Finally, we note that given $n = p$ for these experiments it is probable that the limitations we see for *Expansive Causal* are due to the statistical problems that arise in this data setting. We leave understanding the sample inefficiency of partitioning algorithms to future work.

We conclude that our methods *Expansive Causal* and *Edge Cover* provide a faster time to solution on large graphs, are relatively robust to dense and imperfect superstructures, and provide comparable accuracy compared to *No Partition*.

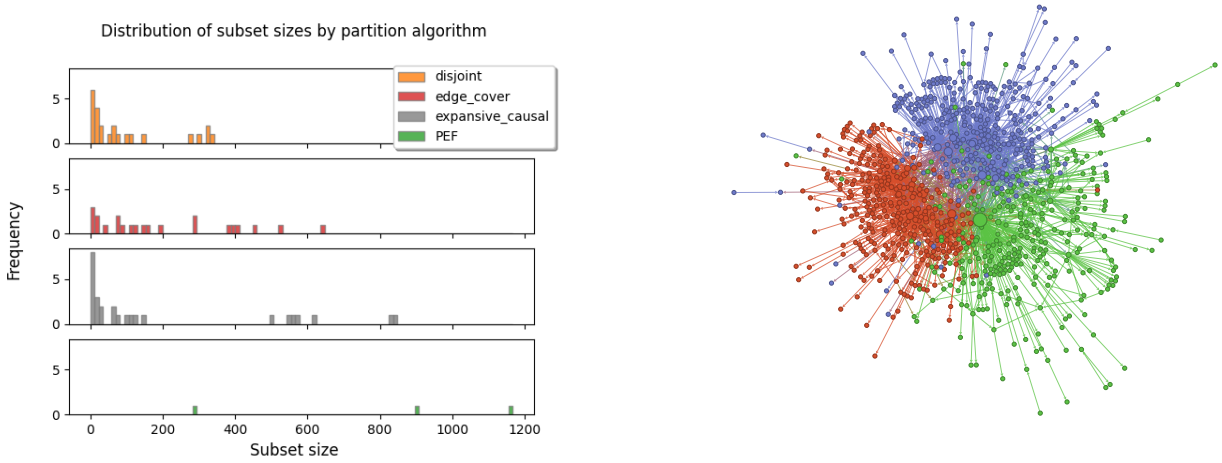
4.7 Empirical Results on Synthetically Tuned E.coli Networks

This section contains results for biological networks. We use the topologies of *E. coli* biological networks due to their availability and popularity. To better benchmark the algorithms, we leverage a *proximity-based* topology generative model from the literature proposed by Hufbauer et al. [2020]. The model was designed with the goal of generating structures with the following properties: *(i)* small-world *(ii)* exponential degree distribution (i.e., scale-free), and *(iii)* presence of inherit community structures. Coincidentally, these properties are also

4. PEF does not leverage an initial superstructure to partition, as a result the time for partitioning is longer than our proposed methods which use an artificial superstructure.

relevant for real-world biological networks [Barabasi and Oltvai, 2004, Koutrouli et al., 2020], thus we take advantage of this generative method. We seed this tuning algorithm with the known *E. coli* regulatory network reconstructed from experimental data in Fang et al. [2017] to generate synthetic networks with *E. coli*-like topology. See Fig. 4.8b for a visualization of the highly connected hub nodes of an example tuned network. We impose a random causal ordering on the topology and generate data from the DAG using the multivariate Gaussian distribution described in Appendix A.6.1.

A comparison of all algorithms is shown in Table 4.4 for $\mathscr{A}$ = GES. *Expansive Causal* provides 1.7x speedup compared to *No Partition*. While there is a significant speedup, we note the decrease in accuracy for all divide-and-conquer algorithms. Still compared to other methods based on partitioning shown here, using a causal partition accelerates causal discovery and provides the best trade off in accuracy. We expect that scaling up to larger gene set sizes (e.g., 10^4 genes for eukaryotic cells) will be severely expensive for methods without partitioning since these networks are more dense and complex than those evaluated in Section 4.6.4.



(a) Subset Sizes

(b) Subset of the topology

Figure 4.8: Topological structure of a synthetically-tuned *E. coli* network. **(a)** Distribution of subset sizes for each partitioning algorithm. **(b)** Example communities extracted from the 2,332 node network with the *Disjoint* partition. These three communities account for 40.7% of the total nodes and 40% of the total edges.

Table 4.4: Results for a synthetically-tuned E.coli network made up of 2,332 nodes and 5,691 edges. n=10,000 samples. We show results only for $\mathscr{A}$ = GES. This experiment was run with an Intel(R) Xeon(R) Gold 6242 CPU @ 2.80GHz with 64 cores and 192 GB of RAM.

Partition	SHD ↓	TPR ↑	FPR ↓	Time (hrs) ↓
No Partition	**805**	**0.859**	8.5e-5	11.8
PEF	1,766	0.692	8.3e-5	22.3
Disjoint	3,903	0.479	1.2e-4	23.9
Edge Cover	1,791	0.698	1.1e-4	7.1
Expansive Causal	1,717	0.701	**6.4e-5**	**6.9**

4.8 Conclusions & Future Directions

We propose a divide-and-conquer causal discovery algorithm based on a novel causal partition. Our algorithm leverages a superstructure—i.e., a known or inferred structured hypothesis space. We prove the consistency of our algorithm under assumptions for the causal learner and in the infinite data limit using the Maximal Ancestral Graph (MAG) class. Unlike existing works, our algorithm allows for the merging of locally estimated graphs *without* an additional learning step. Motivated by a complex scientific application space, we also show an example for gene regulatory network inference for a small organism (*E.coli*). This example shows the applicability of our work to real-world networks, but we leave evaluation of our method on larger organisms (e.g, eukaryotes) to future work.

One limitation of our work is the reliance on a perfect superstructure to create a causal partition. Although in Section 4.6.3 we empirically explore the impact that using imperfect superstructures generated by the PC algorithm has on the learned output, more experiments (including other methods of generating superstructures) are needed to fully characterize the impact of learning when the superstructure does not constrain the edge set of the true causal graph. Further, while the divide-and-conquer method developed in this work can substantially reduce the runtime of performing causal discovery on large variable sets, characterizing the trade-off between time-savings and sample complexity remains an area of open work. An

important open question for future research is how the partitioning described in this paper impacts the sample efficiency of causal discovery algorithms. As discussed in Section 4.4, another interesting direction for future research is extending the divide-and-conquer framework presented in this paper to the setting where both interventional and observational data are available. There exist causal discovery algorithms which can leverage a mixture of observational and interventional data (such as the $\mathcal{I}$-FCI and Ψ-FCI algorithms) [Kocaoglu et al., 2019, Jaber et al., 2020]. If one uses one of these algorithms to perform local learning on the variable subsets, then this might allow one to strengthen the guarantees presented in this work beyond only recovering cause-effect pairs in the MEC of G^*. Given these limitations and open questions, we believe that this work provides a meaningful contribution to causal discovery at scale and to knowledge discovery for domains with high-dimensional structured hypothesis spaces.

CHAPTER 5

INFERRING LOW-DOSE RADIATION RESPONSE MECHANISMS FROM HUMAN TRANSCRIPTOMICS DATA

Next-generation sequencing technologies, including RNA-sequencing, provide genome-wide measurements of gene expression and enable broad explorations of biomarkers and mechanisms underlying disease and treatment response. Bioinformatics tools for processing this data, such as differential expression analysis, are largely univariate, linear, and rely on predefined pathway knowledge annotations, which limits their ability to capture nonlinear and multivariate gene interactions. This paper explores the application of causal discovery to characterizing transcriptional responses to radiation as a function of dose rate in human cells. By jointly modeling radiation perturbations and gene expression, we learn directed gene networks that capture important regulatory relationships beyond correlation and exhibit significant enrichment of known radiation response pathways compared to baseline approaches. We find that inferred causal graphs reveal structured network features such as high in-degree housekeeping genes and high out-degree transcription factors. Further analysis suggests a hierarchical organization of stress response pathways and triggered cell death pathways. This work highlights the potential of causal discovery in healthcare settings with applications to understanding response mechanisms, identifying regulatory targets, and improving interpretation of complex genomic data. Code is available at `https://github.com/shahashka/lucid_cd`.

5.1 Introduction

RNA-sequencing experiments provide genome-wide insight into gene expression levels across conditions at high resolution; however, finding genes of interest from this vast data remains a difficult challenge [Trapnell et al., 2009, Love et al., 2014]. Differential expression analysis

is the most popular tool used to identify genes that statistically vary between perturbed and control samples as an initial step to identify important genes [Rosati et al., 2024]. Subsequent pathway enrichment onto knowledge bases is used to understand the molecular mechanisms underlying the conditions analyzed. Finally, biomarkers are found by linking genes to known disease pathways. Despite its popularity, differential expression analysis combined with pathway enrichment suffers from two main drawbacks: (1) genes are identified using linear models and univariate tests; this misses coordinated and nonlinear signals from genes that may individually be weak (2) its outputs do not directly imply any mechanistic understanding limiting discovery of novel mechanisms and pathways. To address these issues multivariate tests [Glazko and Emmert-Streib, 2009], supervised machine learning with feature ranking [Wenric and Shemirani, 2018], and graph learning algorithms [Marbach et al., 2012, Moerman et al., 2019] have been applied to identify important genes under perturbed conditions.

Recently causal discovery (algorithms for inferring cause and effect relationships represented by graphs) has emerged as a data-driven approach to mechanism discovery. Causal discovery often focuses on synthetic data because real world data violate the assumptions needed for identifiability of the true causal graph; these include the absence of confounding variables, acyclic relationships, and large sample sizes. There is a growing call for the application of these methods to more realistic and scientific settings [Brouillard et al., 2024]. To this end, benchmarking frameworks with large-scale transcriptomics and CRISPR perturbation datasets have been developed; however, they largely reveal that causal discovery algorithms fail to learn regulatory relationships between genes from single-cell RNA-sequencing datasets [Chevalley et al., 2025].

In this paper, we take an alternative approach by applying causal discovery to low-sample, perturbed-control transcriptomics studies for understanding radiation response. We jointly model radiation exposure, which is viewed as a perturbation upstream of gene regulation that affects many pathways, alongside the multivariate nonlinear distribution of gene ex-

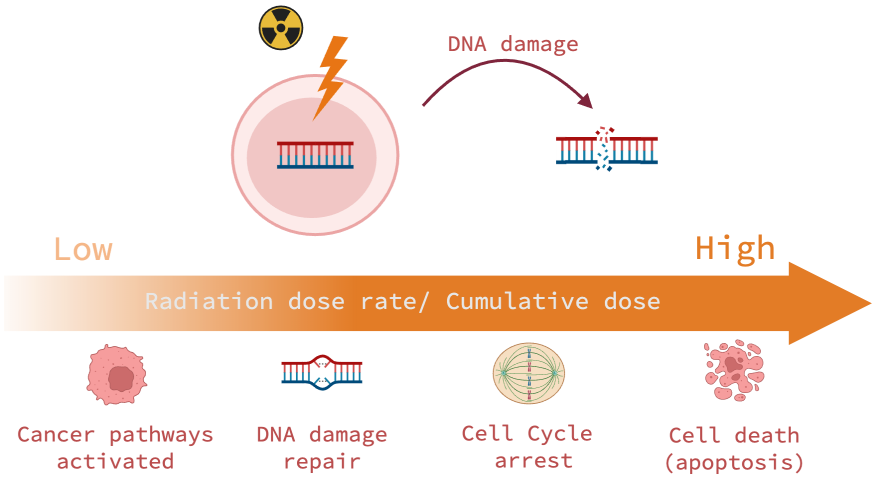


Figure 5.1: Exposure to ionizing radiation causes DNA breakage in cells. Cells activate pathways in order to repair the damage before the cell divides. When damage is excessive or occurs too rapidly, cells initiate programmed cell death (apoptosis).

pression. We apply this framework to RNA sequencing data sampled from RPE1 cell lines exposed to chronic low-dose radiation across increasing dose rates. Such prolonged exposure is associated with elevated risks of cancer and cardiovascular disease and is known to activate cellular stress-response pathways [Shimizu et al., 2010]. A complete understanding of the interplay between dose rate, cumulative dose, and cell type in the activation of low-dose radiation pathways remains an open question (Fig. 5.1) and has implications for people chronically exposed to low-dose radiation (e.g., space exploration, radon exposure) [Verheij and Bartelink, 2000]. To address this gap, we use causal discovery to uncover relationships linking radiation exposure to gene expression, and provide mechanistic insight into how dose rate affects cellular stress-response pathways.

Generalizable Insights about Machine Learning in the Context of Healthcare

This work applies causal discovery to infer important genes and to learn novel pathways from high-dimensional biological data with limited samples and structured perturbations, a common setting across healthcare (e.g., drug response, disease progression, and other treatment-driven processes). The limitations of traditional pipelines, such as differential

expression analysis, are ubiquitous in these settings: overly simplistic univariate and linear model assumptions, and a lack of discovery power by relying on incomplete knowledge bases. We demonstrate that causal discovery can be a complementary method that considers nonlinearity, multivariate interactions, and the distribution of the perturbation variable to identify driver variables and provide structure-based hypotheses of novel mechanisms.

5.2 Related Work

5.2.1 Finding Important Genes: Univariate and Multivariate Approaches

Differential expression analysis is the standard bioinformatics tool used to infer genes with expression levels that differ significantly between conditions (e.g., perturbed and control). This analysis tests the null hypothesis that the logarithmic fold change (LFC) between perturbed and control for a gene's expression is exactly zero, i.e., that the gene is not affected by the perturbation. The goal is to produce a list of genes passing multiple-test adjustment, ranked by p-value.

The most popular tool `DESeq2` [Love et al., 2014] employs this approach; each gene is modeled by a generalized linear model and negative binomial distribution. `DESEq2` models relationships between genes by assuming that genes of similar average expression have a similar dispersion; however, the statistical test for determining significant change between condition groups is univariate for each gene. Therefore, this method does not account for correlations and coordinated effects between groups of genes.

Multivariate tests including Hotelling's T^2 [Lu et al., 2005] and N-statistic [Klebanov et al., 2007] consider the joint distribution of expression levels. Glazko and Emmert-Streib [2009] evaluate univariate and multivariate tests and show that different tests find common but also complementing differentially expressed gene sets – each test projects on different aspects of the data. They propose use of both univariate and multivariate tests for the

analysis of biological data for increased power.

Wenric and Shemirani [2018] propose multivariate classifiers to jointly model effects across genes and classify samples into perturbed and control groups. They find that random forest classifiers outperform differential expression analysis in 9 of the 12 cancer datasets evaluated. They further demonstrate that differential expression analysis might miss important genes, and a supervised learning-based gene selection method can be used to supplement these shortcomings.

5.2.2 Inferring Novel Pathways with Graph Learning

Once important genes are identified pathway enrichment is used to identify the processes that underlie response to perturbations [Raudvere et al., 2019]. Pathway enrichment algorithms determine the significance of gene overlap to known pathways, recorded in knowledge bases like Gene Ontology [Consortium, 2019], KEGG [Kanehisa, 2002], Reactome [Milacic et al., 2024], and WikiPathways [Agrawal et al., 2024].

There is growing interest in developing algorithms that can identify novel pathways to guide the discovery of new mechanisms and biomarkers. Many gene regulatory network (GRN) inference algorithms have been proposed to solve this challenge. These methods include GENIE3 [Marbach et al., 2012] which uses an ensemble of random forest regression models to recover transcription factor (TF) to target gene directed relationships. GRNBoost2 [Moerman et al., 2019] takes a similar approach, but with gradient boosting which is much faster and more scalable. Methods like GENELink [Chen and Liu, 2022] and GRINDC [Feng et al., 2023] use deep learning to predict the presence or absence of an edge from embedded representations of single-cell RNA-sequencing datasets [Dixit et al., 2016]. Several of these algorithms have been benchmarked against BEELINE [Pratapa et al., 2020]. The authors show that GRN inference remains an open problem despite these recent advances.

Causal discovery methods infer directed acyclic graphs (DAGs) from data. These ap-

proaches benefit from theoretical guarantees of identifying the true causal graph under certain assumptions. A breakthrough in causal discovery was the definition of a continuous acyclicity constraint, allowing for gradient-based learning of DAGs [Zheng et al., 2018]. This includes deep learning approaches such as DAG-GNN [Yu et al., 2019] which employs a variational autoencoder to learn an adjacency matrix under the DAG constraint while capturing nonlinear relationships between genes. A recent benchmark paper [Chevalley et al., 2025] evaluates several causal discovery algorithms with single-cell RNA-seq and Perturb-seq datasets, using knowledge graphs for edge validation. Their findings underscored issues with scalability and suboptimal use of perturbational data, and report low precision and recall of ground truth regulatory edges.

In this paper, we use causal discovery to learn causal graphs over the joint distribution of gene expression *and a perturbation variable*. Therefore, we learn pathways specific to perturbation response. Our goal is not to reconstruct all known regulatory relationships, which previous works have shown is notoriously difficult, but rather to propose an alternative approach for identifying important genes under perturbed conditions. A key advantage of our approach is the inference of a directed graph over variables, enabling the generation of hypotheses about novel response mechanisms. To our knowledge this is the first use of causal discovery in this setting.

5.3 Data Collection

The bulk RNA-seq dataset is described by Jantre et al. [2026]. Here, we provide a brief overview of the procedure. RPE1 cells were grown in the lab and exposed to low-dose ionizing radiation from a Cesium-137 gamma ray source at $D = 5$ different dose rates measured in milliGray (mGy) per hour [1], including a control group with no radiation exposure. RNA-seq measurements were made for control and exposed (perturbed) samples after each week for

1. One Gray is equal to one joule of absorbed radiation energy per kilogram of matter

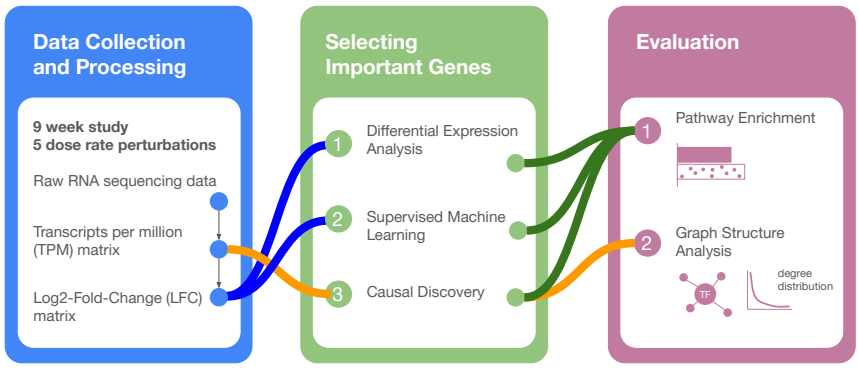


Figure 5.2: Our workflow comprises three main pipelines. Our contribution, shown in orange, is the use of causal discovery to identify important genes from transcriptomics data and downstream graph structural evaluation. We compare against standard pipelines for selecting important genes and subsequent pathway enrichment.

$T = 9$ weeks with 2 replicates per group for a total of $n = 2 \times (D+1) \times T = 108$ samples. Raw RNA-seq reads were preprocessed with alignment to the human reference genome and filtered for low quality reads. Additionally, transcripts per million (TPM) values were computed per sample using `TPMCalculator v0.0.3`, resulting in a data matrix $X_{\text{TPM}} \in \mathbb{R}^{n \times p}$, where $p = 15,694$ genes. We denote $x_{(t,d),g}$ as a value in the matrix corresponding to week t, dose rate d (including $d = 0$ controls) and gene g. We compute the log2-Fold-Change matrix $X_{\text{LFC}} \in \mathbb{R}^{m \times p}$, where $m = T \times D = 45$ exposed conditions. This matrix is computed element-wise for each gene: $X_{\text{LFC}_{(t,d),g}} = \log_2\left(\frac{x_{(t,d),g}}{x_{(t,0)g}}\right)$.

5.4 Methods

A visualization of our framework is shown in Fig. 5.2 which is comprised of three pipelines. In this section, we describe the second and third pipelines.

5.4.1 Selecting Important Genes

Differential Expression Analysis

Differential expression analysis was performed using `pyDESeq2 v0.4.9` with dataset X_{LFC} to determine which genes had statistically significant expression change in each of the 45 control-exposed pairs. Wald tests were performed and p-values were adjusted for multiple testing using the Benjamini–Hochberg false discovery rate (FDR). Genes with adjusted p-value < 0.05 and $|X_{\text{LFC}_{(t,d),g}}| > 1$ were considered differentially expressed. For analysis, we aggregate differentially expressed genes (DEGs) at each dose rate $d \in D$ across time: $G_d = \bigcup_{t \in T} \{g : |X_{\text{LFC}_{(t,d),g}}| > 1, p\text{-value} < 0.05\}$.

Supervised Machine Learning

`RandomForestRegression` models were trained to predict the `(week,dose_rate)` labels of each sample in X_{LFC}. Weeks were assigned ordinal values $T = 1, 2, 3, 4, 5, 6, 7, 8, 9$ and dose rates were assigned $D = 0.28, 0.38, 0.55, 6.66, 12.11$ (mGy/hr) scalar values according to the measured exposure. We perform `RepeatedKFold` cross validation with 5 folds and 5 repeated runs. Within each fold, data is normalized to between (0,1]. The top 1000 features in each trained model's `feature_importance_` vector were extracted, and we selected the final set of genes that were stable in 20 out of 25 folds. Note that this gene set is dose agnostic since we use a stratified training split of `(week, dose_rate)` labels to train the models.

Causal Discovery

We use `DAG-GNN` [Yu et al., 2019], a variational autoencoder parameterized by a graph neural network, to construct DAGs at each dose rate from X_{TPM} and an additional column vector with the cumulative radiation of each sample, computed as `dose_rate` $\times$ `week` $\times$ 168 hours/week. We split X_{TPM} row-wise by dose rate so that X_{TPM_d} represents the samples

Table 5.1: Dimensions of each X_{TPM_d} for each dose rate after pre-processing.

Dose Rate d (mGy/hr)	# Genes p	# Samples n
0.28	9262	36
0.38	9414	36
0.55	6274	36
6.66	8326	36
12.11	10603	36

collected at dose rate d. `DAG-GNN` training is time and memory intensive; to learn an adjacency matrix over $O(10,000)$ variables would be intractable. Therefore, we first filter out genes at each dose rate with p-value > 0.05 from the differential expression analysis, thereby removing genes that were not significantly expressed while still retaining genes with small LFC values. The resultant X_{TPM_d}'s at each dose rate after this preprocessing is shown in Table 5.1. We further partitioned the gene set according to a causal partition defined in Shah et al. [2025] using an initial undirected structure to obtain overlapping subsets of genes that retain consistency of causal discovery. For our work, we use the protein-protein interaction network in the STRING database as the initial undirected structure, and the maximum subset size in the partition of the structure was 2866 genes. A final DAG is obtained using the merge algorithm outlined in Shah et al. [2025] which is based on a union of the subgraphs. We estimate a DAG at each dose rate $d \in D$ corresponding to a subset of X_{TPM}. We bootstrap `DAG-GNN` over 10 resamples of the data with replacement to produce a distribution of DAGs at each dose rate. Further, we derive a consensus graph over the distribution of DAGs, where we take the union over edges that co-occur in 50% or more of the runs. To determine important genes at each dose rate, we take the nodes of the consensus graphs with at least one edge. In practice, many nodes end up being islands in the consensus graphs, which results in a gene set at each dose rate that is significantly smaller than the entire gene set $p = 15,694$. For a description of the partitioning algorithm, training procedure and selected hyperparameters see Appendix B.1.

5.4.2 Evaluation

Pathway Enrichment

After important genes are identified, pathway enrichment is used to understand the functional roles these genes play in the exposed condition. We use the Python interface for `gProfiler` for pathway enrichment analysis of genes identified from differential expression analysis, supervised machine learning with random forest regression models, and causal discovery. `gProfiler` performs functional profiling of gene lists using various sources of biological evidence (Gene Ontology terms, biological pathways, regulatory DNA elements, human disease gene annotations, and protein-protein interaction networks). `gProfiler` uses Fisher's one-tailed test as the p-value measuring the randomness of the intersection between the query gene set and the pathway term. The higher the $-\log_{10} p$-value the stronger the enrichment of that term is in the gene set. We provide a set of background genes (all genes in X_{TPM}) to ensure pathway enrichment p-values are comparable to each other across queries. We screen for pathways with `term_size` < 300 to filter out broad biological processes in the Gene Ontology that have thousands of genes.

Graph Structural Analysis

We validate the edge set and graph topologies of the DAG distributions inferred at each dose rate. We perform TF enrichment to determine if nodes with high out-degree (hub nodes) overlap with known TF genes which code for proteins that regulate hundreds of other genes. We use TRRUST [Han et al., 2018] and a ChIP-Seq network constructed with ENCODE [Davis et al., 2018] and Chip-Atlas [Zou et al., 2022] knowledge bases to identify human transcription factors and directed regulatory edges. Following the strategy of Chevalley et al. [2025] we use networks from a well studied cell line HepG2 because the RPE1 cell line has not been as comprehensively characterized (both cell lines are epithelial). We measure

the overlap of edges in the DAGs with known protein-protein interactions (STRING Mering et al. [2003] and CORUM Giurgiu et al. [2019]), and regulatory edges (TRRUST, ChIP-seq network). We report the precision and F1-scores of the consensus graphs at each dose rate.

5.5 Results

Differential expression analysis with `DESeq2` and causal discovery with `DAG-GNN` both identify gene sets at each dose rate. Gene set sizes and intersections are visualized as Venn diagrams in Fig. 5.3. Differential expression identifies larger sets in general, with the exception of dose rate 0.55 mGy/hr. Gene sets between methods have small overlap, the highest at dose rate 12.11 mGy/hr – these are also the largest gene sets for each method. Supervised machine learning with the `RandomForestRegression` models yielded 68 genes. These were inferred across all doses and have minimal overlap with causal graph and differential expression gene sets (Appendix Fig. B.3).

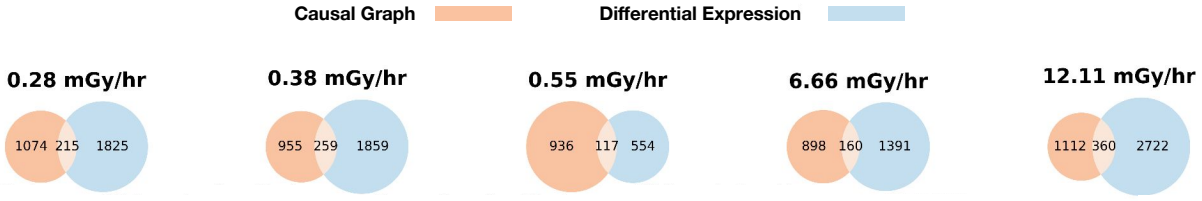


Figure 5.3: Venn diagrams showing gene overlaps between causal graphs and differential expression at each dose rate.

Pathway enrichment with `gProfiler` is shown in Fig. 5.4 for a set of manually curated radiation response specific pathways from the Gene Ontology, KEGG, WikiPathways and Reactome. In general, causal graph gene sets have higher enrichment across radiation response pathways, indicating that causal gene sets better identify important genes for radiation response with smaller gene sets. Supervised machine learning did not enrich any radiation pathways with the exception of `Genes and complexes involved in DNA repair pathways`. Certain pathways such as `Non-homologous end-joining` and `Cell`

`cycle arrest` were not enriched in any method. We do not see a significant dose-dependent enrichment signal; that is, higher dose rates do not correspond to specific pathways compared to lower dose rates. We additionally validate our enrichment results against random genes from the background set, and genes that have high linear correlation with the cumulative radiation variable (Appendix Fig. B.6).

TF enrichment for each causal graph is shown in Figure 5.5. We see that for the top $k = 10$ hub genes in the causal graph, up to 80% are verified as genes that code for transcription factors in the TRRUST database. This percentage decreases as k increases; however, enrichment is always above the baseline random assignment of transcription factors to genes (shown in dashed lines). TF enrichment compared to the ChIP-Seq network shows significantly less overlap; this may be because of the mismatch in cell types since there is no comprehensive network for RPE1 cell lines.

To rule out the possibility that causal discovery simply assigns highly variant genes as hub nodes, we compute the Spearman ranked correlation between node degree and variance in X_{TPM}; see Fig. 5.6. We find that the highest correlation from all dose rates is $\rho = 0.119$. This means there is a small tendency for higher-degree genes in the causal graph to have higher expression variance, but causal discovery is not just recapitulating high-variance genes and instead is finding biologically meaningful structure. This is evidenced by high out-degree nodes mapping to known TFs which are master regulators, and high in-degree nodes map to known housekeeping genes (Fig. 5.6) which are maintenance genes essential for cellular function. Housekeeping genes have also recently been shown to have high instability in stress conditions [Eisenberg and Levanon, 2013], which includes radiation exposure.

In Table 5.2, we show the amount of edge overlap between causal graph at each dose rate and different knowledge graphs, including undirected protein-protein interaction graphs or complexes (STRING, CORUM) and directed regulatory graphs (TRRUST, ChIP-Seq). We see minimal overlap, with a maximum true positive edge count of 143, and maximum F1

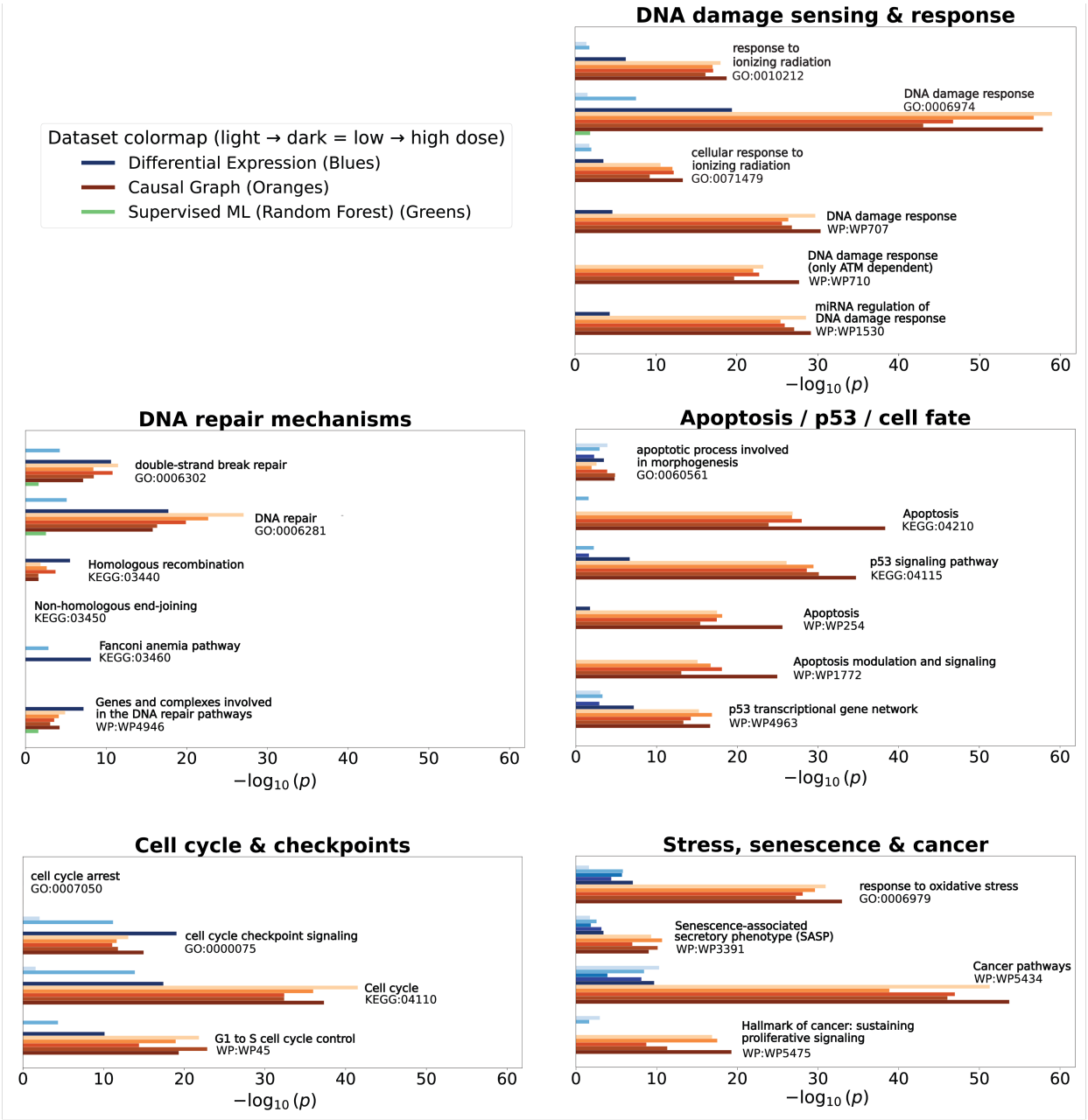


Figure 5.4: Pathway enrichment with `gProfiler` for manually curated radiation pathways across knowledge bases and categorized by radiation response type. In general, causal graph have the highest enrichment, followed by differential expression analysis. Supervised ML has very low enrichment across categories.

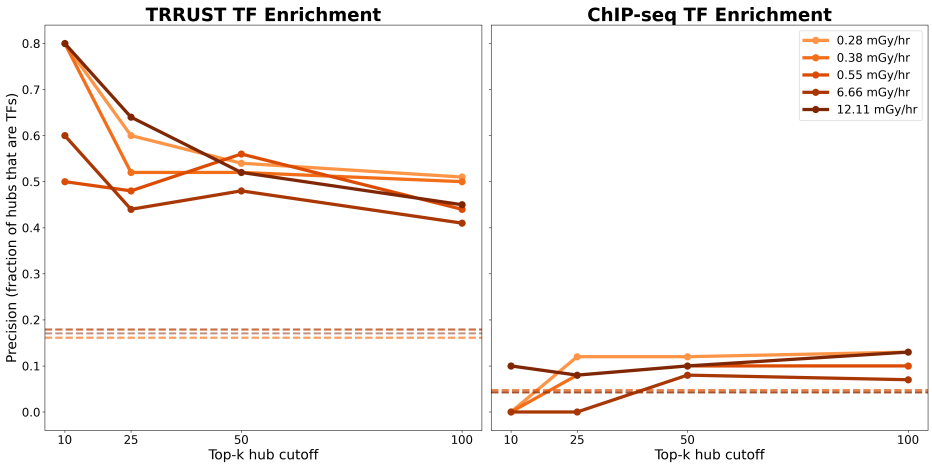


Figure 5.5: Percent of top-k out-degree nodes (hubs) that are known transcription factor genes for high out-degree nodes by dose rate for TRRUST and ChIP-seq knowledge bases. Dashed lines represent the baseline fraction of known transcription factors are in the graph's node set.

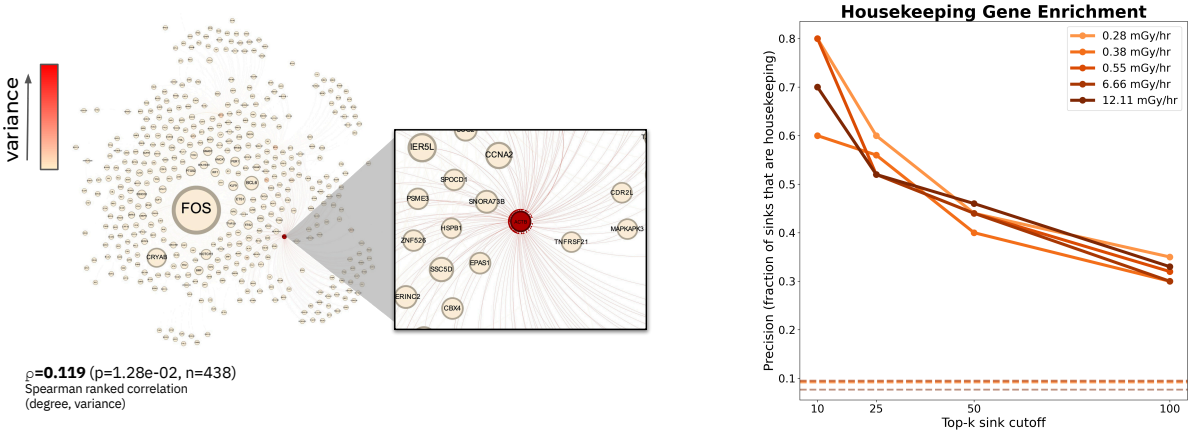


Figure 5.6: (**Left**) An example causal graph at dose rate 6.66 mGy/hr, with nodes colored by variance. Several highly connected sink nodes correspond to housekeeping genes; ACTB is shown in the zoomed-in panel. (**Right**) Percent of top-k in-degree nodes (sinks) that are housekeeping genes for each dose rate. Dashed lines represent the baseline fraction of housekeeping genes in the graph's gene set.

Table 5.2: Edge overlap between causal graphs and prior knowledge databases. TP = true positive number of edges found in the knowledge graph ; F1 = harmonic mean of precision and recall. For TRRUST and ChIP-seq, directed and reversed true positives are shown separately.

Dose	**Edges**	**TRRUST**			**STRING**		**CORUM**		**ChIP-seq**		
(mGy/hr)		TP_{dir}	TP_{rev}	F1	TP	F1	TP	F1	TP_{dir}	TP_{rev}	F1
0.28	2777	24	12	0.012	143	0.026	54	0.020	111	107	0.006
0.38	2610	17	7	0.009	130	0.025	47	0.019	112	80	0.006
0.55	1248	3	4	0.003	52	0.016	26	0.019	54	16	0.004
6.66	2263	16	8	0.010	113	0.031	34	0.019	88	82	0.006
12.11	2417	18	13	0.009	142	0.025	36	0.015	107	89	0.004

score of 0.031. This is similar to results from Chevalley et al. [2025], which used larger scale single-cell datasets for causal discovery.

The set of "invariant" causal graph nodes is the intersection of nodes across all dose rates and results in 438 genes. We performed pathway enrichment on this invariant gene set, which we hypothesized as encoding fundamental radiation response relationships. The top 10 enriched pathways are shown in Table 5.3. Note that unlike previous enrichment analyses, here we report the top pathways without explicit manual curation of radiation-specific pathways. Despite this, all top pathways correspond to well-known radiation response pathways [Sasaki et al., 2002, Feinendegen et al., 2004]. We include the top 10 pathways for differential expression which have some overlap to radiation response (specifically inflammation pathways and tumor necrosis) but have larger p-values in Appendix Table B.2.

We identify causal subgraphs over the invariant gene set at each dose rate and compute the top eigencentral genes (a measure of the node's influence in the subgraph) in Fig. 5.7. We find that top scoring genes correspond to transcription factors (FOS, TSC22D3, IRF1, EGR1, TEF, LHX2, EPAS1) and housekeeping genes (EEF1A1, ACTB), again validating the known structure regarding these genes. Notably, the highest dose-rate (12.11 mGy/hr) has different eigencentric genes suggesting a different biological structure compared to the lower dose rates. We additionally visualize a subset of edges in the subgraphs over the invariant gene set at each dose rate in Fig. 5.8 corresponding to "perfect" edges that co-occur across

Table 5.3: Top 10 Enriched Pathways for the Causal Invariant Gene Set ($n = 438$)

Pathway	$\mathbf{-\log_{10}(p)}$	**Term Size**	**Intersection Size**
Cell cycle `WP:WP179`	26.76	120	31
p53 signaling pathway `KEGG:04115`	26.48	74	26
Cell cycle `KEGG:04110`	25.94	157	33
miRNA regulation of DNA damage response `WP:WP1530`	23.34	73	24
signal transduction by p53 class mediator `GO:0072331`	23.02	164	31
Interleukin-4 and Interleukin-13 signaling `REAC:R-HSA-6785807`	22.70	111	27
DNA damage response `WP:WP707`	22.54	69	23
Integrated breast cancer pathway `WP:WP1984`	22.45	152	30
Cellular senescence `KEGG:04218`	22.43	155	30
Mitotic G1 phase and G1/S transition `REAC:R-HSA-453279`	21.65	148	29

100% of the bootstrap distribution. These correspond to stable edges identified by DAG-GNN and, although only a handful are true positives in knowledge bases, these edges could be candidates for novel relationships with biological significance in radiation response.

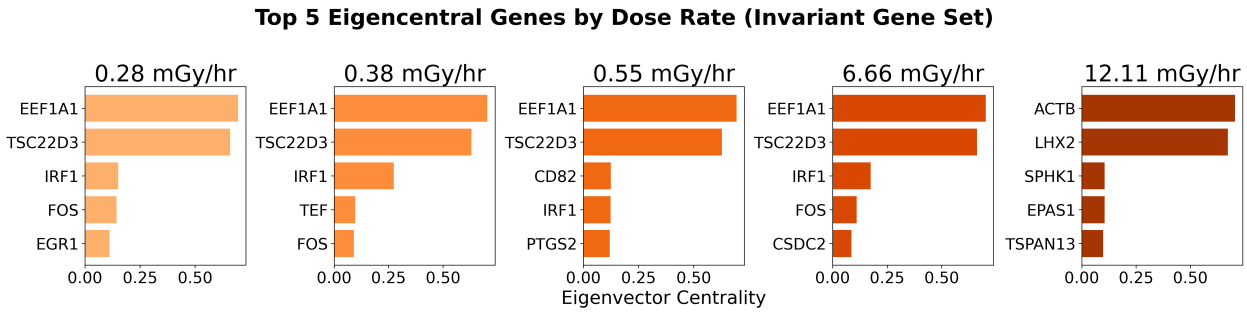


Figure 5.7: The top 5 identified eigencentric genes for each dose-rate causal subgraph induced by the invariant set of genes.

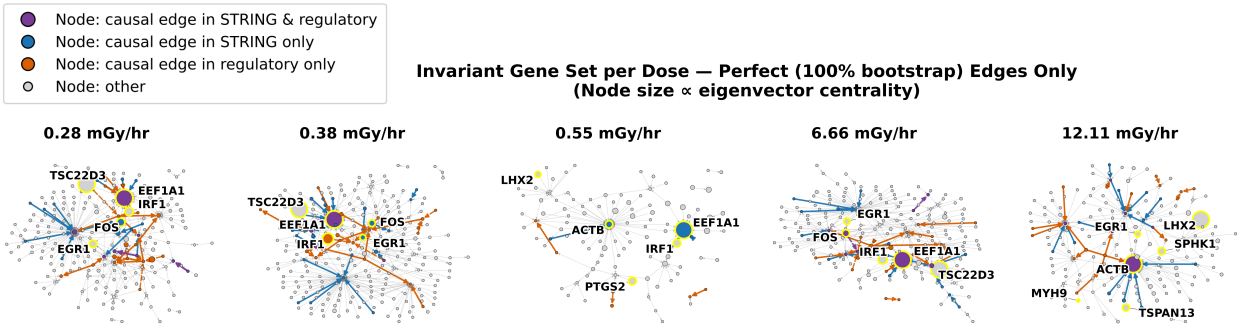


Figure 5.8: The invariant subgraphs at each dose rate with only the "perfect" edges that co-occur across 100% of the bootstrap distribution visualized. Annotated genes correspond to the top-5 eigencentric genes at each dose rate. Edges are highlighted according to true positives in knowledge bases where "regulatory" means either TRRUST or ChIP-seq edges.

Housekeeping genes had a high overlap in the causal invariant gene set (14.4%) compared to the differential expression invariant gene set (1.5%). In Fig. 5.9, we perform pathway enrichment on the ancestors compared to non-ancestors of housekeeping genes. We find that the enrichment profile between these two groups is distinct: the ancestor set overlaps with response to oxidative stress (ROS) pathways, and the non-ancestor set overlaps with membrane permeability pathways. This suggests a branching effect between stress (ROS) and apoptosis (membrane permeability is an indicator of triggered cell death). This branching

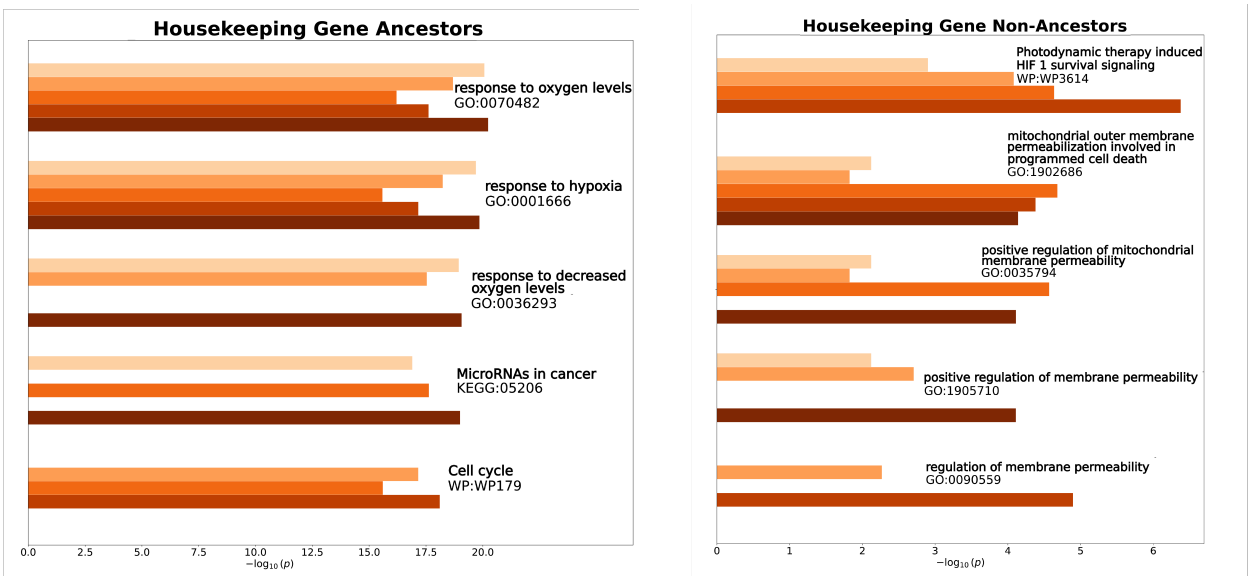


Figure 5.9: (**Left**) Enrichment of housekeeping genes in the parent sets of the causal subgraphs over the invariant gene set. (**Right**) Enrichment of housekeeping genes among the non-ancestors in the causal subgraphs over the invariant gene set.

effect is well documented in the literature [Fulda et al., 2010].

The results reported in this section use the consensus causal graphs over `DAG-GNN` bootstrap runs, which filter for edges that co-occur in more than 50% of the bootstrap runs. Analyses for each DAG in the bootstrap is in Appendix B.7. We additionally show a comparison to other non-causal graph algorithms in Appendix B.8. We see that these cheaper algorithms perform better at recovering exact edges in knowledge bases; however, `DAG-GNN` encodes higher-order, structured biological organization. The full impact and potential of this for novel mechanism discovery should be further explored with experimentation and rigorous evaluation.

5.6 Discussion

We present a previously unexplored application of causal discovery with transcriptomics data to identify important genes related to perturbation response. Typically differential

expression analysis is performed to identify important genes by comparing a control group to perturbed groups. These genes are then used for pathway enrichment to identify possible biomarkers of the perturbation condition. We demonstrate that differential expression analysis may potentially miss important genes related to perturbation response. By jointly modeling the perturbation variable and the multivariate gene expression distribution we show that causal discovery has a higher hit rate of finding perturbation response related gene sets, using radiation response as a use case. Furthermore, the learned DAG identifies structural signatures such as transcription factors and housekeeping genes and hub and sink nodes respectively. We find that causal subgraphs induced by a core set of invariant genes are enriched for perturbation-specific pathways and appear to encode branching between apoptosis and stress response pathways. This observation aligns with invariant causal prediction [Peters et al., 2016], which proposes that causal relationships exhibit stability across environments, here corresponding to different dose rates. To infer novel pathways with causal discovery, one can identify edges that co-occur across the entire bootstrap distribution corresponding to stable edges with potential biological significance. These novel pathways should then be validated with experimentation, for example, with CRISPR knockouts of upstream genes.

Applications of causal discovery have focused on exact recovery of causal edges; this is a notoriously difficult task when we consider the numerous ways in which most realistic datasets violate assumptions needed for identifiability. Research currently focuses on collecting large-scale perturbational datasets to improve identification of the true causal graph. However, there is a need for interpretable methods for biomarker and mechanism discovery in low-sample settings (e.g., small patient cohorts, rare diseases, or settings in which high throughput data is difficult to collect). In the presence of strong upstream perturbations (e.g., environmental stress or drug dosage), we propose that causal discovery should be considered as part of the bioinformatics toolbox as existing methods may be suboptimal in this

setting.

Limitations Our focus on radiation response may limit the generalizability of our findings. Future work will extend this analysis to additional transcriptomic datasets to evaluate the broader applicability of causal discovery for modeling perturbation response. We note that causal discovery incurs substantially higher computational cost than differential expression analysis and supervised machine learning (Appendix B.1); this limits large-scale statistical evaluation. As a result, our current analysis is primarily qualitative, relying on consensus graphs across bootstrap runs. A more rigorous quantification of these findings through statistical analysis over distributions of consensus graphs is an important direction for future work. Finally, we acknowledge that pathway enrichment is biased toward well-studied genes and can be prone to over-interpretation.

CHAPTER 6

DISCUSSION

Causal discovery algorithms are attractive for inferring biological mechanism from data because of their ability to incorporate observational and interventional data to learn directed relationships between biological entities; however, algorithms suffer from poor scaling and are difficult to apply to high-dimensional datasets. Consequently, many candidate driver genes and mechanisms for gene regulation may be missed. In Chapter 3, I develop a hybrid causal discovery algorithm to learn causal DAGs from observational and interventional data. I learned that restricting the search space with an initial structure over nodes greatly improves runtime and the accuracy of the recovered graph; this is in agreement with work such as Nandy et al. [2018]. We lack theoretical guarantees for learning the true causal graph using an initial structure which would greatly improve the reliability of this work. To this end, in Chapter 4, I extend the idea of "structure enabled scaling" by proposing a theoretically sound framework for distributed parallel causal discovery using a graph partition of the initial structure, called a superstructure. Surprisingly, we find that as long as the superstructure covers the underlying causal skeleton, we can always define a causal partition, decompose the learning problem, and recover the equivalence class of the true causal DAG. We show the efficacy of our algorithm in recovering hierarchical scale-free graphs with 10,000 nodes with synthetic linear Gaussian data distributions. We learned that the scalability of our algorithm strongly depends on the completeness and sparsity of the superstructure, and that in low sample settings (even when $n = p$) the accuracy of recovered graphs is stymied by the statistical problem of conditional independence. These two drawbacks are observed in Chapter 5 when we apply a causal partition to infer radiation response mechanisms in RNA-sequencing data of human cells. In this realistic data setting, learning the response mechanisms directly with causal discovery is prohibitively difficult. Instead, we take advantage of the upstream perturbation of the radiation dose rate and instead pivot to the problem

of finding important genes for radiation response. We map genes to mechanisms by relying on the plethora of known pathway knowledge stored in databases like STRING, ENCODE and TRRUST. We find that including radiation as a random variable in the learning process enhances recovery of radiation specific pathways, far exceeding standard practices in bioinformatics that use differential expression analysis. This opens entirely new applications of causal discovery as a feature selection algorithm for low-sample RNA-sequencing data. These contributions elucidate a path towards scalable data-driven mechanism discovery with the following significant results:

- **Structure enables scaling and accuracy.** We know that causal discovery is an NP-Hard problem, and in realistic data settings can also be ill-posed. Prior structural knowledge acts as an inductive bias in which (1) the search space collapses, (2) sample complexity drops, and (3) optimization becomes tractable.

- **Causal discovery is decomposable.** We find that, surprisingly, we can improve scaling by decomposing prior structural knowledge with a graph partition. Causal discovery is provably consistent under the Maximal Ancestral Graph class for inferring MECs from observational data with a causal graph partition. This enables causal discovery on high-performance computing platforms with a distributed algorithm; this is a transformative paradigm for scientific computing labs, which already rely on large compute platforms for multi-scale scientific modeling.

- **Causal discovery should not be limited to exact edge inference.** While the true potential of causal discovery lies in its ability to infer causality via directed edge inference, I also realized in the course of my work that the assumptions needed are exceptionally difficult to satisfy in biology. To this end, maybe it is time to consider re-purposing these algorithms to other problems? We demonstrate that causal discovery can be used as a feature selection algorithm in the presence of strong upstream

perturbations such as radiation. In order to make a generalizable claim to the success of causal discovery in finding important genes in low sample settings; however, we must show how this works in other settings with environment perturbations (e.g., drug or treatment response).

There is, as always, much room for improvement within each of these contributions to ensure their applicability and reliability to wider settings. Here, I reflect on the limitations and avenues for growth:

- **Superstructures**: We make the assumption of a perfect, sparse superstructure to enable scaling in Chapters 3 and 4. However, in practice prior structural knowledge is dense and noisy. We need to further characterize the impact of these flaws in recovering the MEC of the causal graph. Moreover, we need more reliable algorithms for inferring sparse superstructures from data. Beyond score-based causal discovery, incorporating structural priors to gradient-based [Vowels et al., 2021], permutation-based [Wang et al., 2017], and SAT solver-based [Triantafillou and Tsamardinos, 2015] causal discovery, would require non-graphical formulations of prior knowledge; this could be a promising area of research that would open up a wider range of causal discovery algorithms to high-dimensional datasets. Does structure still enable scaling and accuracy for these algorithms?

- **Implementation issues**: Due to the multidisciplinary history of causality (from philosophy to biology to machine learning) causal discovery algorithms are implemented in programming languages such as R, Python, MATLAB and Java, and even within machine learning community there are Tensorflow, JAX, and PyTorch implementations. This makes the barrier for entry for researchers and practitioners alike extremely high: which language, libraries or preexisting packages should we use? Even understanding the full model zoo of different algorithms is difficult, and often practitioners might select the incorrect algorithm for their application simply because it was more available

to them. This is a longstanding problem in the causality domain that we hope to address in the near future. With respect to distributed systems, it is integral that we use either Python or C++ which has infrastructure to support node-level and thread-level distributed programming. In Chapter 4, the Python implementation is only useful for thread-level parallelism and suffers from memory bottlenecks when the number of partitions is too large or when running multiple simulation settings within one job. Therefore, there are still many unanswered software engineering questions to address before reliably running our algorithm on high performance computing platforms.

- **Limited samples** $n << p$: For the majority of this thesis we assume access to infinite data, as this is the setting in which most causal discovery algorithms are provably consistent. We empirically explore low-sample settings in Chapter 5 and unsurprisingly we see that recovery of correct regulatory edges is minimal in this setting. We know that by leveraging a structural prior and partitioning causal discovery into subproblems we introduce inductive bias and improve the curse of dimensionality respectively. However, a theoretical characterization of structural priors and a causal graph partition in low sample settings would be extremely valuable and provide a broader impact. For example DAG-GNN, which is a novel formulation of a VAE, was trained in the low sample, overparameterized setting. Radhakrishnan et al. [2020] show that autoencoders exhibit memorization in overparameterized settings; the decoder learns a function, out of many possible solutions, with higher density around training samples. Can we apply this insight to the DAG-GNN/VAE setting? What can we decipher about the adjacency matrix A, which is part of the decoder? Does A exhibit specific graphical characteristics? Can we impose priors on A to encourage correct solutions?

- **Causal discovery and upstream environment perturbations**: In Chapter 5 we hypothesize that upstream environmental perturbations elucidate invariant mechanisms across dose rates; this connects strongly to the invariant causal prediction work

by Peters et al. [2016], who show that data collected across environments improves identifiability of certain causal relationships since these remain invariant to distribution shifts. We use DAG-GNN to learn an adjacency matrix A for dose rate separately, but would we see improved identifiability of invariant mechanisms if we trained across dose-rates? What can we prove about identifiability of invariant relationships from multi-environment data in low-sample settings?

- **Assumption violations**: As stated in Chapter 2, gene expression data may violate assumptions needed for many causal discovery algorithms, most significantly the acyclic assumption. There are several recent algorithms for causal discovery with cycles including Mooij et al. [2011], Forré and Mooij [2018], Rohbeck et al. [2024]; application of these algorithms within the causal graph partition framework would enable learning graphs with cycles at scale and is an important direction for future work. Despite this promise, there is some question as to whether RNA-sequencing data alone are sufficient to learn gene regulation. Instead, integrating complementary data modalities such as chromatin accessibility measured by ATAC-seq, transcription factor binding measured by ChIP-seq, and epigenetic state measured by DNA methylation sequencing or histone modification assays may provide additional mechanistic constraints needed to identify more reliable regulatory interactions.

In conclusion, this thesis develops methods to enable causal discovery at the genome-scale for inferring biological mechanisms represented by gene regulatory networks. My PhD work defines a theoretically sound and distributed framework based on structural priors and applies it to low-sample, high-dimensional, and perturbational transcriptomics data. This represents a wide class of problems encountered in modern genomics and epigenetics. Although significant challenges remain, this thesis establishes a foundation for inferring causal hypotheses from large-scale biological experiments and brings data-driven mechanism discovery closer to a practical reality.

REFERENCES

Emmanuel Abbe and Colin Sandon. Community detection in general stochastic block models: Fundamental limits and efficient algorithms for recovery. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 670–688. IEEE, 2015.

Ayushi Agrawal, Hasan Balcı, Kristina Hanspers, Susan L Coort, Marvin Martens, Denise N Slenter, Friederike Ehrhart, Daniela Digles, Andra Waagmeester, Isabel Wassink, et al. Wikipathways 2024: next generation pathway database. *Nucleic acids research*, 52(D1): D679–D689, 2024.

Raj Agrawal, Chandler Squires, Karren Yang, Karthikeyan Shanmugam, and Caroline Uhler. Abcd-strategy: Budgeted experimental design for targeted causal structure discovery. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3400–3409. PMLR, 2019.

Réka Albert. Scale-free networks in cell biology. *Journal of cell science*, 118(21):4947–4957, 2005.

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

Ankur Ankan and Abinash Panda. pgmpy: Probabilistic graphical models using python. In *Proceedings of the 14th Python in Science Conference (SCIPY 2015)*. Citeseer, 2015.

Charles K Assaad, Emilie Devijver, and Eric Gaussier. Survey and evaluation of causal discovery methods for time series. *Journal of Artificial Intelligence Research*, 73:767–819, 2022.

Albert-László Barabási. Network science. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1987):20120375, 2013.

Albert-László Barabási and Eric Bonabeau. Scale-free networks. *Scientific american*, 288 (5):60–69, 2003.

Albert-Laszlo Barabasi and Zoltan N Oltvai. Network biology: understanding the cell's functional organization. *Nature reviews genetics*, 5(2):101–113, 2004.

Kevin Bello, Bryon Aragam, and Pradeep Ravikumar. DAGMA: Learning DAGs via M-matrices and a Log-Determinant Acyclicity Characterization. In *Advances in Neural Information Processing Systems*, 2022.

Rohit Bhattacharya, Tushar Nagarajan, Daniel Malinsky, and Ilya Shpitser. Differentiable causal discovery under unmeasured confounding. In *International Conference on Artificial Intelligence and Statistics*, pages 2314–2322. PMLR, 2021.

Remco R Bouckaert. Properties of bayesian belief network learning algorithms. In *Uncertainty in Artificial Intelligence*, pages 102–109. Elsevier, 1994.

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

Philippe Brouillard, Chandler Squires, Jonas Wahl, Konrad P Kording, Karen Sachs, Alexandre Drouin, and Dhanya Sridhar. The landscape of causal discovery data: Grounding causal discovery in real-world applications. *arXiv preprint arXiv:2412.01953*, 2024.

Alanna Cera, Maria K Holganza, Ahmad Abu Hardan, Irvin Gamarra, Reem S Eldabagh, Megan Deschaine, Sarah Elkamhawy, Exequiel M Sisso, Jonathan J Foley IV, and James T Arnone. Functionally related genes cluster into genomic regions that coordinate transcription at a distance in saccharomyces cerevisiae. *Msphere*, 4(2):10–1128, 2019.

Chandler Squires. *`causaldag`: creation, manipulation, and learning of causal models*, 2018. URL `https://github.com/uhlerlab/causaldag`.

Guangyi Chen and Zhi-Ping Liu. Graph attention network for link prediction of gene regulations from single-cell rna-sequencing data. *Bioinformatics*, 38(19):4522–4529, 2022.

Mathieu Chevalley, Yusuf H Roohani, Arash Mehrjou, Jure Leskovec, and Patrick Schwab. A large-scale benchmark for network inference from single-cell perturbation data. *Communications Biology*, 8(1):412, 2025.

David Maxwell Chickering. Learning equivalence classes of bayesian-network structures. *Journal of machine learning research*, 2(Feb):445–498, 2002a.

David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of machine learning research*, 3(Nov):507–554, 2002b.

Tom Claassen and Ioan G Bucur. Greedy equivalence search in the presence of latent confounders. In *Uncertainty in Artificial Intelligence*, pages 443–452. Pmlr, 2022.

Aaron Clauset, Mark EJ Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical review E*, 70(6):066111, 2004.

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

Gene Ontology Consortium. The gene ontology resource: 20 years and still going strong. *Nucleic acids research*, 47(D1):D330–D338, 2019.

Anthony C Constantinou, Zhigao Guo, and Neville K Kitson. The impact of prior knowledge on causal structure learning. *Knowledge and Information Systems*, pages 1–50, 2023.

Haoyue Dai, Ignavier Ng, Gongxu Luo, Peter Spirtes, Petar Stojanov, and Kun Zhang. Gene regulatory network inference in the presence of dropouts: a causal view. *arXiv preprint arXiv:2403.15500*, 2024.

Carrie A Davis, Benjamin C Hitz, Cricket A Sloan, Esther T Chan, Jean M Davidson, Idan Gabdank, Jason A Hilton, Kriti Jain, Ulugbek K Baymuradov, Aditi K Narayanan, et al. The encyclopedia of dna elements (encode): data portal update. *Nucleic acids research*, 46(D1):D794–D801, 2018.

Atray Dixit, Oren Parnas, Biyu Li, Jenny Chen, Charles P Fulco, Livnat Jerby-Arnon, Nemanja D Marjanovic, Danielle Dionne, Tyler Burks, Raktima Raychowdhury, et al. Perturb-seq: dissecting molecular circuits with scalable single-cell rna profiling of pooled genetic screens. *cell*, 167(7):1853–1866, 2016.

Mathias Drton and Marloes H Maathuis. Structure learning in graphical modeling. *Annual Review of Statistics and Its Application*, 4(1):365–393, 2017.

Frederick Eberhardt. Introduction to the foundations of causal discovery. *International Journal of Data Science and Analytics*, 3:81–91, 2017.

Eli Eisenberg and Erez Y Levanon. Human housekeeping genes, revisited. *TRENDS in Genetics*, 29(10):569–574, 2013.

Paul Erdős, Alfréd Rényi, et al. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5(1):17–60, 1960.

Jeremiah J Faith, Boris Hayete, Joshua T Thaden, Ilaria Mogno, Jamey Wierzbowski, Guillaume Cottarel, Simon Kasif, James J Collins, and Timothy S Gardner. Large-scale mapping and validation of escherichia coli transcriptional regulation from a compendium of expression profiles. *PLoS biology*, 5(1):e8, 2007.

Philipp M Faller, Leena Chennuru Vankadara, Atalanti A Mastakouri, Francesco Locatello, and Dominik Janzing. Self-compatibility: Evaluating causal discovery without ground truth. *arXiv preprint arXiv:2307.09552*, 2023.

Xin Fang, Anand Sastry, Nathan Mih, Donghyuk Kim, Justin Tan, James T Yurkovich, Colton J Lloyd, Ye Gao, Laurence Yang, and Bernhard O Palsson. Global transcriptional regulatory network for escherichia coli robustly connects gene expression to transcription factor activities. *Proceedings of the National Academy of Sciences*, 114(38):10286–10291, 2017.

Ludwig E Feinendegen, Myron Pollycove, and Charles A Sondhaus. Responses to low doses of ionizing radiation in biological systems. *Nonlinearity in biology, toxicology, medicine*, 2 (3):15401420490507431, 2004.

Ke Feng, Hongyang Jiang, Chaoyi Yin, and Huiyan Sun. Gene regulatory network inference based on causal discovery integrating with graph neural network. *Quantitative Biology*, 11(4):434–450, 2023.

Patrick Forré and Joris M Mooij. Constraint-based causal discovery for non-linear structural causal models with cycles and latent confounders. *arXiv preprint arXiv:1807.03024*, 2018.

Simone Fulda, Adrienne M Gorman, Osamu Hori, and Afshin Samali. Cellular stress responses: cell survival and cell death. *International journal of cell biology*, 2010(1):214074, 2010.

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

Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.

Madalina Giurgiu, Julian Reinhard, Barbara Brauner, Irmtraud Dunger-Kaltenbach, Gisela Fobo, Goar Frishman, Corinna Montrone, and Andreas Ruepp. Corum: the comprehensive resource of mammalian protein complexes—2019. *Nucleic acids research*, 47(D1):D559–D563, 2019.

Galina V Glazko and Frank Emmert-Streib. Unite and conquer: univariate and multivariate approaches for finding differentially expressed gene sets. *Bioinformatics*, 25(18):2348–2354, 2009.

Arthur Gretton, Peter Spirtes, and Robert Tillman. Nonlinear directed acyclic structure learning with weakly additive noise models. *Advances in neural information processing systems*, 22, 2009.

Jiaying Gu and Qing Zhou. Learning big Gaussian Bayesian networks: Partition, estimation and fusion. *The Journal of Machine Learning Research*, 21(1):6340–6370, 2020.

Heonjong Han, Jae-Won Cho, Sangyoung Lee, Ayoung Yun, Hyojin Kim, Dasom Bae, Sunmo Yang, Chan Yeong Kim, Muyoung Lee, Eunbeen Kim, et al. Trrust v2: an expanded reference database of human and mouse transcriptional regulatory interactions. *Nucleic acids research*, 46(D1):D380–D386, 2018.

Steve Harenberg, Gonzalo Bello, La Gjeltema, Stephen Ranshous, Jitendra Harlalka, Ramona Seay, Kanchana Padmanabhan, and Nagiza Samatova. Community detection in large-scale networks: a survey and empirical evaluation. *Wiley Interdisciplinary Reviews: Computational Statistics*, 6(6):426–439, 2014.

Anne-Claire Haury, Fantine Mordelet, Paola Vera-Licona, and Jean-Philippe Vert. Tigress: trustful inference of gene regulation using stability selection. *BMC systems biology*, 6(1): 145, 2012.

Alain Hauser and Peter Bühlmann. Characterization and greedy learning of interventional markov equivalence classes of directed acyclic graphs. *The Journal of Machine Learning Research*, 13(1):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.

Jireh Huang and Qing Zhou. Partitioned hybrid learning of Bayesian network structures. *Machine Learning*, 111(5):1695–1738, 2022.

Emory Hufbauer, Nathaniel Hudson, and Hana Khamfroush. A proximity-based generative model for online social network topologies. In *2020 International Conference on Computing, Networking and Communications (ICNC)*, pages 648–653. IEEE, 2020.

Amin Jaber, Murat Kocaoglu, Karthikeyan Shanmugam, and Elias Bareinboim. Causal discovery from soft interventions with unknown targets: Characterization and learning. *Advances in neural information processing systems*, 33:9551–9561, 2020.

Sanket Jantre, Kriti Chopra, Guang Zhao, Clark Cucinell, Rebecca Weinberg, Sara Forrester, Thomas Brettin, Nathan M Urban, Xiaoning Qian, and Byung-Jun Yoon. Interpretable transcriptome-to-phenotype modeling of cell-painting nuclear morphology features from rna-seq under low-dose radiation exposure. *bioRxiv*, pages 2026–02, 2026.

Diviyan Kalainathan and Olivier Goudet. Causal discovery toolbox: Uncover causal relationships in python. *arXiv preprint arXiv:1903.02278*, 2019.

Minoru Kanehisa. The kegg database. In *'In silico'simulation of biological processes: Novartis Foundation Symposium 247*, volume 247, pages 91–103. Wiley Online Library, 2002.

Neville Kenneth Kitson, Anthony C Constantinou, Zhigao Guo, Yang Liu, and Kiattikun Chobtham. A survey of bayesian network structure learning. *Artificial Intelligence Review*, 56(8):8721–8814, 2023.

Lev Klebanov, Galina Glazko, Peter Salzman, Andrei Yakovlev, and Yuanhui Xiao. A multivariate extension of the gene set enrichment analysis. *Journal of bioinformatics and computational biology*, 5(05):1139–1153, 2007.

Murat Kocaoglu, Amin Jaber, Karthikeyan Shanmugam, and Elias Bareinboim. Characterization and learning of causal graphs with latent variables from soft interventions. *Advances in Neural Information Processing Systems*, 32, 2019.

Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. 2009.

Mikaela Koutrouli, Evangelos Karatzas, David Paez-Espino, and Georgios A Pavlopoulos. A guide to conquer the biological network era using graph theory. *Frontiers in bioengineering and biotechnology*, 8:34, 2020.

Jorge D Laborda, Pablo Torrijos, José M Puerta, and José A Gámez. A ring-based distributed algorithm for learning high-dimensional bayesian networks. In *European Conference on Symbolic and Quantitative Approaches with Uncertainty*, pages 123–135. Springer, 2023.

Sébastien Lachapelle, Philippe Brouillard, Tristan Deleu, and Simon Lacoste-Julien. Gradient-based neural dag learning. *arXiv preprint arXiv:1906.02226*, 2019.

Alexander Lachmann, Federico M Giorgi, Gonzalo Lopez, and Andrea Califano. Aracne-ap: gene network reverse engineering through adaptive partitioning inference of mutual information. *Bioinformatics*, 32(14):2233–2235, 2016.

Peter Langfelder and Steve Horvath. Wgcna: an r package for weighted correlation network analysis. *BMC bioinformatics*, 9(1):559, 2008.

Thuc Duy Le, Tao Hoang, Jiuyong Li, Lin Liu, Huawen Liu, and Shu Hu. A fast pc algorithm for high dimensional causal discovery with multi-core pcs. *IEEE/ACM transactions on computational biology and bioinformatics*, 16(5):1483–1495, 2016.

Sangmin Lee and Seoung Bum Kim. Parallel simulated annealing with a greedy algorithm for bayesian network structure learning. *IEEE Transactions on Knowledge and Data Engineering*, 32(6):1157–1166, 2019.

Nathan E Lewis, Harish Nagarajan, and Bernhard O Palsson. Constraining the metabolic genotype–phenotype relationship using a phylogeny of in silico methods. *Nature Reviews Microbiology*, 10(4):291–305, 2012.

Shuohao Li, Jun Zhang, Kuihua Huang, and Chenxu Gao. A graph partitioning approach for bayesian network structure learning. In *Proceedings of the 33rd Chinese Control Conference*, pages 2887–2892. IEEE, 2014.

Hanti Lin and Jiji Zhang. On learning causal structures from non-experimental data without any faithfulness assumption. In *Algorithmic Learning Theory*, pages 554–582. PMLR, 2020.

Suzan Lindquist, Elizabeth A Craig, et al. The heat-shock proteins. *Annual review of genetics*, 22(1):631–677, 1988.

Romain Lopez, Jan-Christian Hütter, Jonathan Pritchard, and Aviv Regev. Large-scale differentiable causal discovery of factor graphs. *Advances in Neural Information Processing Systems*, 35:19290–19303, 2022.

Michael I Love, Wolfgang Huber, and Simon Anders. Moderated estimation of fold change and dispersion for rna-seq data with deseq2. *Genome biology*, 15(12):550, 2014.

Yan Lu, Peng-Yuan Liu, Peng Xiao, and Hong-Wen Deng. Hotelling's t 2 multivariate profiling for detecting differential expression in microarrays. *Bioinformatics*, 21(14):3105–3113, 2005.

Yunhai Luo, Benjamin C Hitz, Idan Gabdank, Jason A Hilton, Meenakshi S Kagda, Bonita Lam, Zachary Myers, Paul Sud, Jennifer Jou, Khine Lin, et al. New developments on the encyclopedia of dna elements (encode) data portal. *Nucleic acids research*, 48(D1): D882–D889, 2020.

Anders L Madsen, Frank Jensen, Antonio Salmerón, Helge Langseth, and Thomas D Nielsen. A parallel algorithm for bayesian network structure learning from large data sets. *Knowledge-Based Systems*, 117:46–55, 2017.

Fragkiskos D Malliaros and Michalis Vazirgiannis. Clustering and community detection in directed networks: A survey. *Physics reports*, 533(4):95–142, 2013.

Daniel Marbach, James C Costello, Robert Küffner, Nicole M Vega, Robert J Prill, Diogo M Camacho, Kyle R Allison, Manolis Kellis, James J Collins, et al. Wisdom of crowds for robust gene network inference. *Nature methods*, 9(8):796–804, 2012.

Adam A Margolin, Ilya Nemenman, Katia Basso, Chris Wiggins, Gustavo Stolovitzky, Riccardo Dalla Favera, and Andrea Califano. Aracne: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. In *BMC bioinformatics*, volume 7, pages 1–15. BioMed Central, 2006.

Christopher Meek. Causal inference and causal explanation with background knowledge. *arXiv preprint arXiv:1302.4972*, 2013.

Christian von Mering, Martijn Huynen, Daniel Jaeggi, Steffen Schmidt, Peer Bork, and Berend Snel. String: a database of predicted functional associations between proteins. *Nucleic acids research*, 31(1):258–261, 2003.

Marija Milacic, Deidre Beavers, Patrick Conley, Chuqiao Gong, Marc Gillespie, Johannes Griss, Robin Haw, Bijay Jassal, Lisa Matthews, Bruce May, et al. The reactome pathway knowledgebase 2024. *Nucleic acids research*, 52(D1):D672–D678, 2024.

Thomas Moerman, Sara Aibar Santos, Carmen Bravo González-Blas, Jaak Simm, Yves Moreau, Jan Aerts, and Stein Aerts. Grnboost2 and arboreto: efficient and scalable inference of gene regulatory networks. *Bioinformatics*, 35(12):2159–2161, 2019.

Joris M Mooij, Dominik Janzing, Tom Heskes, and Bernhard Schölkopf. On causal discovery with cyclic additive noise models. *Advances in neural information processing systems*, 24, 2011.

Preetam Nandy, Alain Hauser, and Marloes H Maathuis. High-dimensional consistency in score-based and hybrid structure learning. *The Annals of Statistics*, 46(6A):3151–3183, 2018.

Achille Nazaret, Justin Hong, Elham Azizi, and David Blei. Stable differentiable causal discovery. *arXiv preprint arXiv:2311.10263*, 2023.

Robert Osazuwa Ness, Karen Sachs, Parag Mallick, and Olga Vitek. A bayesian active learning experimental design for inferring signaling networks. In *International Conference on Research in Computational Molecular Biology*, pages 134–156. Springer, 2017.

Ignavier Ng, Yujia Zheng, Jiji Zhang, and Kun Zhang. Reliable causal discovery with improved exact search and weaker assumptions. *Advances in Neural Information Processing Systems*, 34:20308–20320, 2021.

Judea Pearl. From bayesian networks to causal networks. In *Mathematical models for handling partial knowledge in artificial intelligence*, pages 157–182. Springer, 1995a.

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

Judea Pearl. *Causality*. Cambridge university press, 2009.

Eric Perrier, Seiya Imoto, and Satoru Miyano. Finding optimal Bayesian network given a super-structure. *Journal of Machine Learning Research*, 9(10), 2008.

Jonas Peters and Peter Bühlmann. Structural intervention distance for evaluating causal graphs. *Neural computation*, 27(3):771–799, 2015.

Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 78(5):947–1012, 2016.

Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT press, 2017.

Massimo Pigliucci. Genotype–phenotype mapping and the end of the 'genes as blueprint'metaphor. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 365(1540):557–566, 2010.

Andrea Pinna, Nicola Soranzo, and Alberto De La Fuente. From knockouts to networks: establishing direct cause-effect relationships through graph analysis. *PloS one*, 5(10): e12912, 2010.

Aditya Pratapa, Amogh P Jalihal, Jeffrey N Law, Aditya Bharadwaj, and andT M Murali. Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data. *Nature methods*, 17(2):147–154, 2020.

Adityanarayanan Radhakrishnan, Mikhail Belkin, and Caroline Uhler. Overparameterized neural networks implement associative memory. *Proceedings of the National Academy of Sciences*, 117(44):27162–27170, 2020.

Joseph Ramsey, Madelyn Glymour, Ruben Sanchez-Romero, and Clark Glymour. A million variables and more: the fast greedy equivalence search algorithm for learning high-dimensional graphical causal models, with an application to functional magnetic resonance images. *International journal of data science and analytics*, 3(2):121–129, 2017.

Joseph D Ramsey. Scaling up greedy causal search for continuous variables. *arXiv preprint arXiv:1507.07749*, 2015.

Andrea Rau, Florence Jaffrézic, and Grégory Nuel. Joint estimation of causal effects from observational and intervention gene expression data. *BMC systems biology*, 7(1):1–12, 2013.

Uku Raudvere, Liis Kolberg, Ivan Kuzmin, Tambet Arak, Priit Adler, Hedi Peterson, and Jaak Vilo. g: Profiler: a web server for functional enrichment analysis and conversions of gene lists (2019 update). *Nucleic acids research*, 47(W1):W191–W198, 2019.

Erzsébet Ravasz. Detecting hierarchical modularity in biological networks. *Computational Systems Biology*, pages 145–160, 2009.

Thomas S Richardson and Peter Spirtes. Causal inference via ancestral graph models. *Oxford Statistical Science Series*, pages 83–105, 2003.

Thomas S Richardson, Robin J Evans, James M Robins, and Ilya Shpitser. Nested markov properties for acyclic directed mixed graphs. *The Annals of Statistics*, 51(1):334–361, 2023.

Marylyn D Ritchie, Emily R Holzinger, Ruowang Li, Sarah A Pendergrass, and Dokyoon Kim. Methods of integrating data to uncover genotype–phenotype interactions. *Nature Reviews Genetics*, 16(2):85–97, 2015.

Martin Rohbeck, Brian Clarke, Katharina Mikulik, Alexandra Pettet, Oliver Stegle, and Kai Ueltzhöffer. Bicycle: Intervention-based causal discovery with cycles. In *Causal Learning and Reasoning*, pages 209–242. PMLR, 2024.

Diletta Rosati, Maria Palmieri, Giulia Brunelli, Andrea Morrione, Francesco Iannelli, Elisa Frullanti, and Antonio Giordano. Differential gene expression analysis pipelines and bioinformatic tools for the identification of specific biomarkers: A review. *Computational and structural biotechnology journal*, 23:1154–1168, 2024.

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

Masao S Sasaki, Yosuke Ejima, Akira Tachibana, Toshiko Yamada, Kanji Ishizaki, Takashi Shimizu, and Taisei Nomura. Dna damage response pathway in radioadaptive response. *Mutation Research/Fundamental and Molecular Mechanisms of Mutagenesis*, 504(1-2): 101–118, 2002.

Satu Elisa Schaeffer. Graph clustering. *Computer science review*, 1(1):27–64, 2007.

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Towards causal representation learning. *CoRR*, abs/2102.11107, 2021. URL `https://arxiv.org/abs/2102.11107`.

Ashka Shah, Adela Frances DePavia, Nathaniel C Hudson, Ian Foster, and Rick Stevens. Causal discovery over high-dimensional structured hypothesis spaces with causal graph partitioning. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL `https://openreview.net/forum?id=FecsgPCOHk`.

Rajen D Shah and Jonas Peters. The hardness of conditional independence testing and the generalised covariance measure. 2020.

Yukiko Shimizu, Kazunori Kodama, Nobuo Nishi, Fumiyoshi Kasagi, Akihiko Suyama, Midori Soda, Eric J Grant, Hiromi Sugiyama, Ritsu Sakata, Hiroko Moriwaki, et al. Radiation exposure and circulatory disease risk: Hiroshima and nagasaki atomic bomb survivor data, 1950-2003. *Bmj*, 340, 2010.

Justin R Smith. *The design and analysis of parallel algorithms*. Oxford University Press, 1993.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. MIT press, 2nd edition, 2000a.

Pater Spirtes, Clark Glymour, Richard Scheines, Stuart Kauffman, Valerio Aimale, and Frank Wimberly. Constructing bayesian network models of gene expression networks from microarray data. 2000b.

Peter Spirtes. An anytime algorithm for causal inference. In *International Workshop on Artificial Intelligence and Statistics*, pages 278–285. PMLR, 2001.

Peter Spirtes and Kun Zhang. Causal discovery and inference: concepts and recent methodological advances. In *Applied informatics*, volume 3, page 3. Springer, 2016.

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

Chandler Squires, Yuhao Wang, and Caroline Uhler. Permutation-based causal structure learning with unknown intervention targets. In *Conference on Uncertainty in Artificial Intelligence*, pages 1039–1048. PMLR, 2020.

Eric V Strobl. A constraint-based algorithm for causal discovery with cycles, latent variables and selection bias. *International Journal of Data Science and Analytics*, 8(1):33–56, 2019.

Xiangyuan Tan, Xiaoguang Gao, Zidong Wang, Hao Han, Xiaohan Liu, and Daqing Chen. Learning the structure of bayesian networks with ancestral and/or heuristic partition. *Information Sciences*, 584:719–751, 2022.

Simon Tong and Daphne Koller. Active learning for structure in bayesian networks. In *International joint conference on artificial intelligence*, volume 17, pages 863–869. Citeseer, 2001.

Cole Trapnell, Lior Pachter, and Steven L Salzberg. Tophat: discovering splice junctions with rna-seq. *Bioinformatics*, 25(9):1105–1111, 2009.

Sofia Triantafillou and Ioannis Tsamardinos. Constraint-based causal discovery from multiple interventions over overlapping variable sets. *The Journal of Machine Learning Research*, 16(1):2147–2205, 2015.

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

Marcel Verheij and Harry Bartelink. Radiation-induced apoptosis. *Cell and tissue research*, 301(1):133–142, 2000.

Thomas S Verma and Judea Pearl. Equivalence and synthesis of causal models. In *Probabilistic and causal inference: The works of Judea Pearl*, pages 221–236. 2022.

Matthew J Vowels, Necati Cihan Camgoz, and Richard Bowden. D'ya like dags? a survey on structure learning and causal discovery. *ACM Computing Surveys (CSUR)*, 2021.

Peng Wang, Zirui Zhou, and Anthony Man-Cho So. A nearly-linear time algorithm for exact community recovery in stochastic block model. In *International Conference on Machine Learning*, pages 10126–10135. PMLR, 2020.

Yuhao Wang, Liam Solus, Karren Yang, and Caroline Uhler. Permutation-based causal inference algorithms with interventions. *Advances in Neural Information Processing Systems*, 30, 2017.

Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world'networks. *nature*, 393(6684):440–442, 1998.

Stephane Wenric and Ruhollah Shemirani. Using supervised learning methods for gene selection in rna-seq case-control studies. *Frontiers in genetics*, 9:364621, 2018.

Stefan Wuchty, Erszébet Ravasz, and Albert-László Barabási. The architecture of biological networks. *Complex systems science in biomedicine*, pages 165–181, 2006.

Haiyuan Yu and Mark Gerstein. Genomic analysis of the hierarchical structure of regulatory networks. *Proceedings of the National Academy of Sciences*, 103(40):14724–14731, 2006.

Yue Yu, Jie Chen, Tian Gao, and Mo Yu. Dag-gnn: Dag structure learning with graph neural networks. In *International conference on machine learning*, pages 7154–7163. PMLR, 2019.

Behrooz Zarebavani, Foad Jafarinejad, Matin Hashemi, and Saber Salehkaleybar. cupc: Cuda-based parallel pc algorithm for causal structure learning on gpu. *IEEE Transactions on Parallel and Distributed Systems*, 31(3):530–542, 2019.

Yi-feng Zeng and Kim-leng Poh. Block learning bayesian network structure from data. In *Fourth International Conference on Hybrid Intelligent Systems (HIS'04)*, pages 14–19. IEEE, 2004.

Hao Zhang, Yixin Ren, Yewei Xia, Shuigeng Zhou, and Jihong Guan. Towards effective causal partitioning by edge cutting of adjoint graph. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Jiji Zhang. Causal reasoning with ancestral graphs. *Journal of Machine Learning Research*, 9(7), 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.

Jiji Zhang and Peter Spirtes. Detection of unfaithfulness and robust causal inference. *Minds and Machines*, 18(2):239–271, 2008.

Xun Zheng, Bryon Aragam, Pradeep K Ravikumar, and Eric P Xing. Dags with no tears: Continuous optimization for structure learning. *Advances in neural information processing systems*, 31, 2018.

Xun Zheng, Chen Dan, Bryon Aragam, Pradeep Ravikumar, and Eric Xing. Learning sparse nonparametric dags. In *International conference on artificial intelligence and statistics*, pages 3414–3425. Pmlr, 2020.

Zhaonan Zou, Tazro Ohta, Fumihito Miura, and Shinya Oki. Chip-atlas 2021 update: a data-mining suite for exploring epigenomic landscapes by fully integrating chip-seq, atac-seq and bisulfite-seq data. *Nucleic acids research*, 50(W1):W175–W182, 2022.

APPENDIX A

ADDENDUM FOR CHAPTER 4

A.1 Definitions

Definition A.1.1 (Collider on a path)**.** *Given a path $P = (X_1, \ldots, X_k)$ on a mixed graph G, a non-endpoint vertex X_i is a collider on path P if both edges adjacent to X_i on the path have a directed or bi-directed edge pointing to X_i. Examples include $X_{i-1} \to X_i, X_i \leftarrow X_{i+1}$ and $X_{i-1} \to X_i, X_i \leftrightarrow X_{i+1}$. A non-endpoint vertex which is not a collider is said to be a non-collider on that path.*

A.2 Deferred Proofs

A.2.1 Deferred Proofs from Section 4.3.3

Here we prove the properties in Lemma 1.

1. For any $x_i, x_j \in S$, the output $\mathscr{A}(X_S)$ has an edge between x_i and x_j if and only if there is an inducing path in G^* relative to $V \setminus S$ between them.

   Proof. We begin by noting that by definition, x_i and x_j are adjacent in $L^{MAG}(G^*, S)$ if and only if there is an inducing path in G^* relative to $V \setminus S$ between them [Zhang, 2008a]. Moreover, by definition the PAG $\mathscr{A}(X_S) = PAG[L^{MAG}(G^*, S)]$ has the same adjacencies as any member of $[L^{MAG}(G^*, S)]$, and therefore the same adjacencies as $L^{MAG}(G^*, S)$. Thus x_i and x_j are adjacent in $\mathscr{A}(X_S)$ if and only if there is an inducing path in G^* relative to $V \setminus S$ between them. □

2. For any triple $x_i, x_j, x_k \in S$ that form an unshielded collider in G^* as $x_i \to x_j \leftarrow x_k$,

the output $\mathscr{A}(X_S)$ will have an edge between x_i and x_j as well as x_j and x_k, and both of these edges will have an arrowhead at x_j.

Proof. We first note that for $\{x_i, x_j, x_k\} \subseteq S$, the edges $x_i \to x_j$ and $x_k \to x_j$ are inducing paths in G^* relative to $V \setminus S$ and thus the pairs x_i, x_j and x_k, x_j are adjacent in both $L^{MAG}(G^*, S)$ and $\mathscr{A}(X_S)$. To show that both edges will have an arrowhead at x_j in $\mathscr{A}(X_S)$, it thus remains to show the edges have arrowheads at x_k in every $[L^{MAG}(G^*, S)]$. By property (1) of Definition 4.3.4, both edges will have arrowheads at x_k in $L^{\text{MAG}}(G^*, S)$. Given $\{x_i, x_j x_k\}$ form an unshielded collider in G^*, all sets that d-separate x_i from x_k in G^* do not contain x_j. Thus given $\{x_i, x_j, x_k\} \subseteq S$, all sets that m-separate x_i form x_k in $L^{\text{MAG}}(G^*, S)$ do not contain x_j, and thus the collider is unshielded in $L^{\text{MAG}}(G^*, S)$[Zhang, 2008a].

By definition of the MEC of a MAG, every element in $[L^{MAG}(G^*, S)]$ has the same unshielded colliders, so every element in $[L^{\text{MAG}}(G^*, S)]$ has arrowheads at x_k[Zhang, 2008b]. Thus the PAG $\mathscr{A}(X_S) = PAG[L^{\text{MAG}}(G^*, S)]$ has arrowheads at x_k on both edges. □

3. For any $u, v \in S$ such that $u \sim_{G^*} v$, if $u \sim_{\mathscr{A}(S)} v$ with an arrowhead at v in $\mathscr{A}(X_S)$, then $u \to v$ in G^*.

Proof. Given $u \sim_{G^*} v$ for G^* a DAG, either $u \to v$ in G^* or $v \to u$ in G^*. Assume for the sake of contradiction that $v \to u$ in G^*. By the definition of $PAG[L^{\text{MAG}}(G^*, S)]$, given $u \sim_{\mathscr{A}(X_S)} v$ with an arrowhead at v in $\mathscr{A}(X_S)$, it holds that u and v are adjacent with an arrowhead at v for every element of $[L^{\text{MAG}}(G^*, S)]$ [Zhang, 2008a]. In particular, u and v are adjacent with an arrowhead at v in $L^{MAG}(G^*, S)$. By definition of the latent MAG, u and v are adjacent with an arrowhead at v in $L^{MAG}(G^*, S)$ implies that one of the following hold: (1) $u \to v$ in G^*, (2) $u \in \text{anc}_{G^*}(v)$ and there is

an inducing path in G^* between u and v relative to $V \setminus S$, or (3) there is some other inducing path between u and v but $u \notin \text{anc}_{G^*}(v)$ and $v \notin \text{anc}_{G^*}(u)$. If either (1) or (2) hold, then $v \to u$ in G^* would imply the existence of a cycle in G^*, contradicting the assumption that G^* is a DAG. Moreover (3) cannot hold, as given $u \sim_{G^*} v$ it must be that either $u \in \text{anc}_{G^*}(v)$ or $v \in \text{anc}_{G^*}(u)$. Thus in all three cases we arrive at a contradiction, and so we conclude that $v \not\to u$ in G^*, and thus that $u \to v$ in G^*. □

A.2.2 Deferred Proofs from Section 4.4

In this section, we consider superstructure G satisfying Assumption 2, a learner $\mathscr{A}$ satisfying Assumption 1, $\{S_1, \ldots, S_N\}$ a causal partition with respect to G and G^*, and H^* the output of Algorithm 1 on $G, \{\mathscr{A}(X_{S_i})\}_{i=1}^N$. We begin by proving property (i) in Theorem 1.

Lemma 3. *For any $\forall \in V$, $u \sim_{H^*} v$ if and only if $u \sim_{G^*} v$.*

Proof. Consider any $u, v \in V$ such that $u \sim_{G^*} v$. Because G satisfies Assumption 2, $u \sim_G v$. By the definition of a causal partition, $\{S_1, \ldots, S_N\}$ is edge-covering with respect to G and thus $\exists i \in [N]$ such that $u, v \in S_i$. Moreover, given $u, v \in S_i$, the edge between the two nodes in G^* is an inducing path with respect to $V \setminus S_i$ and so by statement (1) in Lemma 1, $u \sim_{\mathscr{A}(X_{S_i})} v$. Thus $u \sim_G v$ and $u \sim_{\mathscr{A}(X_{S_i})} v$ so $u \ast\!\!-\!\!\ast\, v \in E_{\text{candidates}}$. Moreover, for any subset $S_j \ni u, v$, the edge between u and v in G^* is an inducing path with respect to $V \setminus S_j$, so $u \ast\!\!-\!\!\ast\, v \in E_j$ for all j such that $u, v \in S_j$. Thus an edge between u and v will be added to E^*, so $u \sim_{H^*} v$.

Conversely, consider any $u, v, \in V$ such that $u \not\sim_{G^*} v$. If $u \not\sim_G v$, or $\nexists i \in [N]$ such that $u \sim_{\mathscr{A}(X_{S_i})} v$, then $E_{\text{candidates}}$ will not contain an edge between u and v and thus neither will E^*. If $u \sim_G v$ and $\exists i \in [N]$ such that $u \sim_{\mathscr{A}(X_{S_i})} v$, then $E_{\text{candidates}}$ will contain an edge between u and v. However because $\{S_1, \ldots, S_N\}$ is a causal partition, by property (ii) of Definition 4.3.5 there exists some $j \in [N]$ such that $u, v \in S_j$ and u and v do not have an

edge between them in E_i the edges of output $\mathscr{A}(S_j)$. Thus no edge between u and v will be added to E^*. We thus conclude that $u \sim_{H^*} v$ if and only if $u \sim_{G^*} v$. □

In order to prove property (ii) of Theorem 1, we will use the following lemma:

Lemma 4. *For all $u, v \in V$ such that $u \to v$ in H^*, it holds that $u \to v$ in G^*.*

Proof. If the output H^* contains directed edge $u \to v$, then Lemma 3 implies $u \sim_{G^*} v$ and the definition of Algorithm 1 implies $\exists i \in [N]$ such that $u \to v$ is part of an unshielded collider $u \ast\!\!\to v \leftarrow\!\!\ast\, w$ in $\mathscr{A}(X_{S_i})$. Given $u \sim_{G^*} v$, by statement (3) of Lemma 1 the fact that $u \sim_{\mathscr{A}(X_{S_i})} v$ and $\mathscr{A}(X_{S_i})$ contains an arrowhead at v implies that $u \to v$ in G^*. □

We now prove property (ii) of Theorem 1.

Lemma 5. *For any unshielded collider $u \to v \leftarrow w$ in H^*, it holds that $u \to v \leftarrow w$ in G^*.*

The proof follows directly from application of Lemma 4.

We conclude with the proof of property (iii):

Lemma 6. *For any unshielded collider $u \to v \leftarrow w$ in G^*, $u \sim_{H^*} v$ and $v \sim_{H^*} w$ and both edges have an arrowhead at v in H^*.*

Proof. Given any unshielded collider $u \to v \leftarrow w$ in G^*, Lemma 3 implies that $u \sim_{H^*} v$, $v \sim_{H^*} w$, and $u \not\sim_{H^*} w$. It thus remains to show that the v-structure edges are oriented correctly in H^*. By the definition of a causal partition, $\exists i$ such that $\{u, v, w\} \subseteq S_i$. Thus by statement (2) in Lemma 1, $u \ast\!\!\to v$ and $w \ast\!\!\to v$ in $\mathscr{A}(X_{S_i})$. Thus the condition in Line 5 is satisfied so both $u \to v$ and $w \to v$ will be added to E^*, and thus the edges are oriented correctly in H^*. □

A.2.3 Deferred Proofs from Section 4.5.1

Throughout this section, we assume superstructure G satisfies Assumption 2. Consider $\{S_1, \ldots, S_N\}$ be a vertex-covering partition of G and denote by $\{S'_1, \ldots, S'_N\}$ the causal expansion of $\{S_1, \ldots, S_N\}$ with respect to G.

In order to prove Lemma 2, we introduce several auxiliary lemmas. Proving that the causal expansion satisfies properties (i) and (iii) of Definition 4.3.5 is straightforward. These arguments are contained in Lemmas 7 and 8 respectively:

Lemma 7. *The overlapping partition* $\{S'_1, \ldots, S'_N\}$ *is edge-covering with respect to superstructure* G*.*

Proof of Lemma 7. Consider any u, v such that $u \sim_G v$. Because the original partition $\{S_1, \ldots, S_N\}$ is vertex-covering, $\exists i \in [N]$ such that $u \in S_i$. Moreover, $u \sim_G v$ so $v \in \text{neighbors}(u) \subseteq S_i \cup \partial_{\text{out}} S_i = S'_i$. □

Lemma 8. *Given any unshielded collider in* G^**,* $u \to v \leftarrow w$*, there exists* $i \in [N]$ *such that* $\{u, v, w\} \subseteq S'_i$*.*

Proof of Lemma 8. As the original partition $\{S_1, \ldots, S_N\}$ is vertex-covering, $\exists i \in [N]$ such that $v \in S_i$. Moreover as G satisfies Assumption 2, $u \sim_G v$ and $w \sim_G v$. Thus by definition of the expansive causal partition, $\{u, v, w\} \subseteq S'_i$. □

Proving that the causal expansion satisfies property (ii) of Definition 4.3.5 is more involved. We first establish the following helper lemma:

Lemma 9. *Consider any* $S \subseteq V$ *and any* $u, v \in S$ *such that* $u \not\sim_{G^*} v$ *in DAG* G^**. Then any path* $\Pi \subseteq S$ *such that length*$(\Pi) > 1$ *is not an inducing path between* u *and* v *in* G^* *relative to* $V \setminus S$*. Moreover, any path* $\Pi = (u, q_1, q_2, \ldots, q_{k-1}, q_k, v)$ *such that either* $\{u, q_1, q_2\} \subseteq S$ *or* $\{q_{k-1}, q_k, v\} \subseteq S$ *is not an inducing path between* u *and* v *in* G^* *relative to* $V \setminus S$*.*

Proof of Lemma 9. Both conditions on Π imply the existence of non-endpoints $q, q' \in S$ adjacent along path Π. By definition of an inducing path, q and q' must both therefore be colliders on Π. This implies that the edge between q and q' in path Π must have an arrowhead at both q and q' in G^*. However G^* is a DAG and cannot contain bi-directed edges, so q and q' cannot both be colliders on Π, and Π is therefore not an inducing path. □

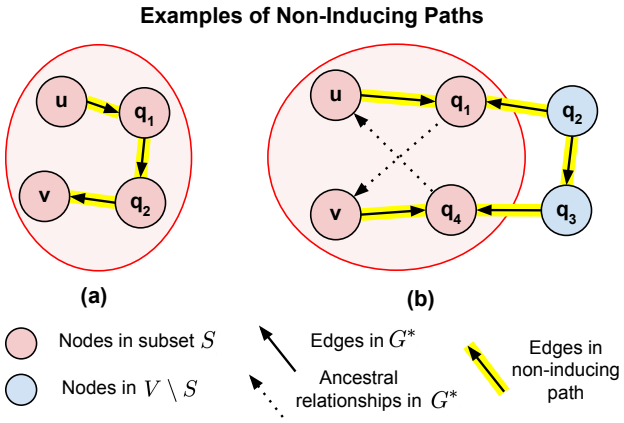


Figure A.1: Examples of non-inducing paths. The example in (a) illustrates the case described in Lemma 9. This path is not inducing because q_1, q_2 are non-endpoint paths in S, but they are not both colliders on the path. The example in (b) illustrates **Case 2** in the proof of Lemma 10. The definition of an inducing path requires that q_1 be an ancestor of u and q_4 be an ancestor of u, but this implies the existence of a cycle in G^* contains u, q_1, v, and q_4. Thus this path cannot exist.

We now use Lemma 9 to prove that the causal expansion satisfies property (ii) of Definition 4.3.5:

Lemma 10. *Given any $u \not\sim_{G^*} v$, there exists $i \in [N]$ such that $u, v \in S'_i$ and $u \not\sim_{\mathcal{A}(S'_i)} v$.*

Proof of Lemma 10. Consider some $u, v \in V$ such that $u \not\sim_{G^*} v$ and $u \not\sim_G v$. Recall that by Assumption 1, for any subset S'_i, $u \sim_{\mathcal{A}(S'_i)}$ if and only if there is an inducing path between u and v in G^* relative to $V \setminus S'_i$. Thus to prove Lemma 10, it suffices to show that $\exists i \in [N]$ such that no inducing path exists between u and v in G^* relative to $V \setminus S'_i$. By Lemma 9, any path Π in G^* of length greater than 1 such that $\Pi \subseteq S'_i$ cannot be such an inducing path. As $u \not\sim_{G^*} v$, all paths between u and v in G^* have length at least 1. Thus to prove Lemma 10, it suffices to show that $\exists i \in [N]$ such that no inducing path Π with $\Pi \cap \{V \setminus S'_i\} \neq \emptyset$ exists between u and v in G^* relative to $V \setminus S'_i$.

By Lemma 7, $\exists i \in [N]$ such that $u, v \in S'_i$. For any $u \in S \subseteq V$, denote by $\text{dist}_{G^*}(u, \partial_{\text{out}} S)$ the shortest-path distance from u to any node $v \in \partial_{\text{out}}(S)$. In other words, $\text{dist}_{G^*}(u, \partial_{\text{out}} S)$ is the minimum number of edges between a node u and any node $w \notin S$. Note that for any $u \in S$, $\text{dist}_{G^*}(u, \partial_{\text{out}} S) \geq 1$.

We consider four cases, parameterized by the distance from the endpoints u, v to $\partial_{\text{out}} S_i'$. Note that these four cases cover all possible positionings of u and v within S_i'. Thus to prove Lemma 10, we must show that each case implies the existence of some $S' \in \{S_1', \ldots, S_N'\}$, not necessarily equal to $S_i' i$, such that $u \not\sim_{\mathscr{A}(S')} v$.

Case 1. $\max\{\text{dist}_{G^*}(u, \partial_{\text{out}} S_i'), \text{dist}_{G^*}(v, \partial_{\text{out}} S_i')\} > 2$

Case 2. $\text{dist}_{G^*}(u, \partial_{\text{out}} S_i') = \text{dist}_{G^*}(v, \partial_{\text{out}} S_i') = 2$.

Case 3. $\text{dist}_{G^*}(u, \partial_{\text{out}} S_i') = 2$, and $\text{dist}_{G^*}(v, \partial_{\text{out}} S_i') = 1$.

Case 4. $\text{dist}_{G^*}(u, \partial_{\text{out}} S_i') = \text{dist}_{G^*}(v, \partial_{\text{out}} S_i') = 1$.

We now show that in each case, there exists some $S' \in \{S_1', \ldots, S_N'\}$ such that $u \not\sim_{\mathscr{A}(S')} v$.

Case 1. $\max\{\text{dist}_{G^*}(u, \partial_{\text{out}} S_i'), \text{dist}_{G^*}(v, \partial_{\text{out}} S_i')\} > 2$ implies that for any path Π between u, v, either $\Pi \subseteq S_i'$, or that Π contains a prefix $\{u, q_1, q_2\} \subseteq S_i'$, or that Π contains a suffix $\{q_{k-1}, q_k, v\} \subseteq S_i'$. In all of these cases, Lemma 9 implies that Π is not an inducing path between u and v in G^* relative to $V \setminus S_i'$. Thus $u \not\sim_{\mathscr{A}(S_i')} v$.

Case 2. $\text{dist}_{G^*}(u, \partial_{\text{out}} S) = \text{dist}_{G^*}(v, \partial_{\text{out}} S) = 2$ implies that for any path Π between u, v, either $\Pi \subseteq S_i'$ or that Π contains a prefix $\{u, q_1\} \subseteq S_i'$ and suffix $\{q_k, v\} \subseteq S_i'$. If $\Pi \subseteq S_i'$, then it is not an inducing path.

Consider the case when Π contains a prefix $\{u, q_1\} \subseteq S_i'$ and suffix $\{q_k, v\} \subseteq S_i'$ and assume for the sake of contradiction that Π is an inducing path between u and v in G^* relative to $V \setminus S_i'$. Both q_1 and q_k are non-endpoint vertices on $\Pi \cap S_i'$. They must therefore be colliders on Π as well as ancestors of at least one of u or v. Since q_1 be a collider on Π, it must be that $u \to q_1$ so $u \in \text{anc}_{G^*}(q_1)$, where

$$\text{anc}_{G^*}(x) \equiv \{z \in V : z \text{ is an ancestor of } x \text{ in } G^*\}.$$

Moreover, q_1 must be an ancestor of either u or v, and because $u \in \text{anc}_{G^*}(q_1)$ it cannot be that q_1 is an ancestor of u as this would imply the existence of a cycle in G^*. Thus it must be that $q_1 \in \text{anc}_{G^*}(v)$. However, we similarly conclude that as q_k be a collider on Π, it must be that $q_k \leftarrow v$so $v \in \text{anc}_{G^*}(q_k)$. Moreover q_k must be an ancestor of either u or v, and q_k cannot be an ancestor of v as G^* is acyclic, so $q_k \in \text{anc}_{G^*}(u)$.

However we have thus concluded that $u \in \text{anc}_{G^*}(q_1)$, $q_1 \in \text{anc}_{G^*}(v)$, $v \in \text{anc}_{G^*}(q_k)$, and $q_k \in \text{anc}_{G^*}(u)$. This implies the existence of a cycle in G^*, and thus cannot occur. Thus we conclude that no such path Π can be an inducing path between u and v in G^* relative to $V \setminus S_i'$. Thus $u \not\sim_{\mathscr{A}(S_i')} v$.

Case 3. Recall that by definition of the expansive causal partition, $S_i' = S_i \cup \partial_{\text{out}}(S_i)$ for original vertex-covering partition $\{S_1, \ldots, S_N\}$, where the outer boundary $\partial_{\text{out}}(S_i)$ is defined by the edges in superstructure G. Given $\text{dist}_{G^*}(v, \partial_{\text{out}} S_i') = 1$, $\exists z \notin S_i'$ such that $v \sim_{G^*} z$. Moreover, as G satisfies Assumption 2, this implies $v \sim_G z$. Thus by definition of the expansive causal partition it must be that $v \in S_i' \setminus S_i$. As the original partition $\{S_1, \ldots, S_N\}$ is vertex-covering, this implies $\exists j \in [N] \setminus \{i\}$ such that $v \in S_j$. Moreover, as $u \sim_G v$, this implies $u, v \in S_j'$ and that in S_j', $\text{dist}(v, \partial_{\text{out}}(S_j')) \geq 2$ and $\text{dist}(u, \partial_{\text{out}}(S_j')) \geq 1$. If $\text{dist}(v, \partial_{\text{out}}(S_j')) > 2$ or $\text{dist}(u, \partial_{\text{out}}(S_j')) > 1$, then either **Case 1** or **Case 2** respectively imply that $u \not\sim_{\mathscr{A}(S_j')} v$, which would conclude the proof. It thus remains to consider the case where $\text{dist}(v, \partial_{\text{out}}(S_j')) = 2$ and $\text{dist}(u, \partial_{\text{out}}(S_j')) = 1$.

We thus have the following setup: by assumption of **Case 3**, $\text{dist}_{G^*}(u, \partial_{\text{out}} S_i') = 2$ and $\text{dist}_{G^*}(v, \partial_{\text{out}} S_i') = 1$. Then by the above arguments, we have shown $j \neq i$ such that $\text{dist}_{G^*}(v, \partial_{\text{out}} S_j') = 2$ and $\text{dist}_{G^*}(u, \partial_{\text{out}} S_j') = 1$. Assume by way of contradiction that $u \sim_{\mathscr{A}(S_i'} v$ and $u \sim_{\mathscr{A}(S_j'} v$. Thus by Assumption 1, there must exist Π_i and inducing path between u and v with respect to $V \setminus S_i'$ and Π_j an inducing path between u and v with respect to $V \setminus S_j'$.

As $\text{dist}_{G^*}(u, \partial_{\text{out}} S_i') = 2$, Π_i must contain a prefix $\{u, q_i\} \subseteq \Pi_i \cap S_i'$ where $q_i \neq v$. By

definition of an inducing path q_i must be a collider on Π_i in G^*, so $u \in \text{anc}_{G^*}(q_i)$, and q_i must be an ancestor of either v or u. As G^* is acyclic and $u \in \text{anc}_{G^*}(q_i)$, q_i cannot be an ancestor of u and must therefore be an ancestor of v: $q_i \in \text{anc}_{G^*}(v)$.

Similarly, as $\text{dist}_{G^*}(v, \partial_{\text{out}} S'_j) = 2$, Π_j must contain a suffix $\{q_j, v\} \subseteq \Pi_j \cap S'_j$ such that $q_j \neq u$. Moreover by an analogous argument to the above, $v \in \text{anc}_{G^*}(q_j)$ and $q_j \in \text{anc}_{G^*} u$.

We have therefore concluded the following: $\exists q_i, q_j \in V$ such that $u \in \text{anc}_{G^*}(q_i)$, $q_i \in \text{anc}_{G^*}(v)$, $v \in \text{anc}_{G^*}(q_j)$, and $q_j \in \text{anc}_{G^*}(u)$. However this implies the existence of a cycle in G^*, which contradicts the assumption that G^* is a DAG. Thus it cannot hold that both $u \sim_{\mathscr{A}(S'_i)} v$ and $u \sim_{\mathscr{A}(S'_j)} v$, so we conclude $\exists S' \in \{S'_1, \ldots, S'_N\}$ such that $u \not\sim_{\mathscr{A}(S')} v$.

Case 4. Given $\text{dist}_{G^*}(u, \partial_{\text{out}} S'_i) = \text{dist}_{G^*}(v, \partial_{\text{out}} S'_i) = 1$, $\exists z \notin S'_i$ such that $u \sim_{G^*} z$. As superstructure G satisfies Assumption 2 this implies $u \sim_G z$ and thus by definition of the expansive causal partition, implies $u \in S'_i \setminus S_i$. As the original partition was vertex-covering, this implies $\exists j \neq i$ such that $u \in S_j$. Thus by definition of the expansive causal partition, $u \in S'_j$ and $\text{dist}_{G^*}(u, \partial_{\text{out}} S'_j) \geq 2$. Moreover as $u \sim_G v$, $v \in S'_j$ as well.

If $\text{dist}_{G^*}(u, \partial_{\text{out}} S'_j) > 2$, then **Case 1** implies $u \not\sim_{\mathscr{A}(S'_j)} v$. If $\text{dist}_{G^*}(u, \partial_{\text{out}} S'_j) = 2$ and $\text{dist}_{G^*}(v, \partial_{\text{out}} S'_j) = 2$, then **Case 2** implies $u \not\sim_{\mathscr{A}(S'_j)} v$. If $\text{dist}_{G^*}(u, \partial_{\text{out}} S'_j) = 2$ and $\text{dist}_{G^*}(v, \partial_{\text{out}} S'_j) = 1$, then the argument in **Case 3** implies the existence of $k \neq j$ such that either $u \not\sim_{\mathscr{A}(S'_j)} v$ or $u \not\sim_{\mathscr{A}(S'_k)} v$.

We have thus concluded in each case that $\exists S' \in \{S'_1, \ldots, S'_N\}$ such that $u \not\sim_{\mathscr{A}(S')} v$, and so the statement of Lemma 10 holds. □

Lemma 2 follows directly from Lemmas 7, 8, and 10.

A.3 Finite Sample Effects

While the theoretical results in Section 4.4 only apply to the infinite data regime, in this section we discuss heuristics for addressing the effects of learning with finite samples and

Algorithm 3: `Screen_Finite_Data`$(G, \{H_i\}_{i=1}^N, X)$

Input: a superstructure G, a set of PAGS $\{H_i = (S_i, E_i)\}_{i=1}^N$, a matrix of observations X.
Result: $H^* = (V, E^*)$ a PAG

Initialize $V = \cup_{i=1}^N S_i$; $E_{\text{candidates}} \leftarrow \cup_{i=1}^N E_i$; $E^* \leftarrow \emptyset$
foreach u, v *such that* $\{u \ast\!\!-\!\!\ast\, v\} \in E_{candidates}$ **do**
    `// If an edge between` u `and` v `appears in the learned output on all subsets containing` u `and` v`, add edge to output graph.`
    if $\forall i$ *s.t.* $S_i \supseteq \{u, v\}$, $u \sim_{\mathcal{A}(S_i)} v$ **then**
        `// If edge appears oriented in output, add oriented edge to` E^*`.`
        if $\exists i$ *such that* $E_i \ni \{u \ast\!\!\rightarrow v\}$ **then**
            $E^* \leftarrow E^* \cup \{u \rightarrow v\}$;
        else
            $E^* \leftarrow E^* \cup \{u \circ\!\!-\!\!\circ v\}$;
$H^* \leftarrow (V, E^*)$
while H^* *contains cycle* $\mathcal{C}$ **do**
    $H^* \leftarrow$ `score_and_discard`$(H^*, \mathcal{C}, \{S_1, \ldots, S_N\}, X)$;
return $H^* = (V, E^*)$;

describe a practical algorithm for real-world causal discovery problems. In the finite data setting, there are two key ways that finite samples cause divergence from the idealized assumptions studied in Section 4.4: (1) the superstructure may be imperfect and (2) the result of learning over a local subset may not be a latent projection and therefore the merged graph may contain cycles. We describe our finite sample screening procedure in Algorithm 3. In `score_and_discard`, we resolve cycles by discarding the edge corresponding to the lowest score, where the score is related to the log-likelihood of the data with and without each edge in the cycle.

Imperfect Superstructure : In real-world causal discovery applications, one may wish to learn a superstructure G from data [Constantinou et al., 2023]. Several algorithms for learning a superstructure from data exist; many, including the PC algorithm, are more easily parallelized than greedy score-based learners and thus can be run on the global variable set in reasonable time [Zarebavani et al., 2019, Le et al., 2016] . However, when the superstructure G is learned from data, it may be imperfect, i.e. there may exist edges in G^* which are not

in G. If the superstructure is missing a large fraction of the ground-truth edges, the step in `Screen`, which discards edges not in the superstructure may significantly reduce the rate of true positive edges returned by the algorithm, with the effect growing more severe with more imperfect superstructures. Thus in the finite sample limit, if working with a superstructure which is suspected to be highly imperfect, one option is to simply omit the step in `Screen`, which discards edges not in the superstructure. In Section 4.6.3, we examine the impact of learning imperfect superstructures from data, and show while imperfect superstructures do impact learning significantly, the expansive causal partition is most effective out of all partition schemes.

Potential cycles: When the result of learning over a subset is not a latent projection, the algorithm presented in Section 4.4 may fail to return a DAG. In particular, even if the output $\mathscr{A}(X_{S_i})$ is a DAG on every subset S_i, the output of `Screen` may contain cycles. However, it is possible to localize these cycles; if the output $\mathscr{A}(X_{S_i})$ is a DAG on every subset S_i, then any cycle in the output of $\texttt{Screen}(G, \{\mathscr{A}(X_{S_i})\}_{i=1}^N)$ will have some edge (u, v) such that one of the two endpoints lies in the overlap of partition $\{S_1, \ldots, S_N\}$, i.e. $\exists i \neq j$ such that $\{u, v\} \cap \{S_i \cap S_j\} \neq \emptyset$.

Using this observation about the location of all cycles in the output of `Screen`, adopt the following procedure. If the output of `Screen` contains a cycle, we find all edges in that cycle which intersect with the overlap of partition $\{S_1, \ldots, S_N\}$. We then rank these edges using a scoring function and discard the lowest-ranked edge. While a variety of edge scoring functions may be deployed for this step, in this work we assess edges using the log-likelihood induced by the linear structural equation

$$X_j = \sum_{i=1}^{p} W_{ij}^{(G)} X_i + \varepsilon_j \tag{A.1}$$

where $W_{ij}^{(G)}$ denotes the weighted adjacency matrix of a DAG G and $\varepsilon_j \sim \mathcal{N}(0, \sigma_j^2)$ denotes

Algorithm 4: `score_and_discard`

Input: a graph G, $\mathcal{C}$ a list of edges comprising a cycle in G, $\{S_i\}_{i=1}^N$ a partition of the nodes of G, a matrix of observations X

Result: a modified copy of G which does not contain cycle $\mathcal{C}$

$\hat{V} \leftarrow \bigcup_{i,j=1}^{N} \{S_i \cap S_j\}$; `// overlapping nodes`

$\hat{E} \leftarrow \{\}$; `// overlapping edges`

foreach $(u, v) \in \mathcal{C}$ **do**

  if $u \in \hat{V}$ *or* $v \in \hat{V}$ **then** $\hat{E} \leftarrow \hat{E} \cup \{(u, v)\}$;

$\hat{e} \leftarrow \arg\min_{(u,v)\in\hat{E}}$ `loglikelihood_score`(u, v, G, X);

G.removeEdge($\hat{e}$);

return G;

additive Gaussian noise. Then the joint distribution of $(X_1 \dots X_p)$ is a multivariate Gaussian distribution $\mathcal{N}(0, \Sigma)$ where $\Sigma = WW^T$. The log-likelihood under this model is

$$l(W, \Sigma) = \sum_{j=1}^{p} \left[-\frac{n}{2} \log(\sigma_j)^2 - \frac{1}{2\sigma_j^2} ||X_j - \mathbf{X}W_j||^2 \right] \tag{A.2}$$

In order to score an edge (i, j), we compare the log-likelihood at the least squares estimates (LSE) of the regression coefficients ($\hat{W}_{ij}$) in Eq. A.1 of two different DAGs: $G^{i,j}$ which contains edge (i, j), and $G^{0,j}$ in which we remove edge (i, j) so that i is no longer a parent of j. Edge (i, j) is then scored by how much including i as a parent of j increases the log-likelihood of X_j under the linear structural equation. The likelihood based score is outlined in Algorithm 5. The full procedure for cycle resolution is outlined in Algorithm 4.

In the case when the detected cycle has length two, i.e. there exist edges (i, j) and (j, i), we adopt the methodology of Gu and Zhou [2020] and use the risk inflation criterion (RIC) to determine whether to discard one or both of the edges forming the cycle. In this setting we compare three models: $G^{i,j}$ in which i is a parent of j, $G^{j,i}$ in which j is a parent of i, and G^0 in which neither edge appears. We then compute the RIC score for each model, which balances the log-likelihood with a sparsity-promoting term penalizing the total edges in the graph. If the model G^0 out-performs both $G^{i,j}$ and $G^{j,i}$, then both edges are removed

Algorithm 5: `loglikelihood_score`(i, j, G, X)

Input: a node i, a node j, a graph G, a matrix of observations X
Result: a score based on the likelihood of graph given the data in the presence and absence of edge (i, j)

`// least squares estimates of Eq. A.1`
$\hat{W}^{(G^{i,j})} \leftarrow \text{LSE}(X_j, G^{i,j})$
$\hat{W}^{(G^{0,j})} \leftarrow \text{LSE}(X_j, G^{0,j})$
$\Sigma \leftarrow cov(X)$ `// covariance matrix of X`
`// log-likelihoods from Eq. A.2`
$l_{ij} \leftarrow l(\hat{W}^{(G^{i,j})}, \Sigma)$
$l_0 \leftarrow l(\hat{W}^{(G^{0,j})}, \Sigma)$
score $\leftarrow l_{ij} - l_0$
return score

from the graph. If at least one of the models $G^{i,j}$, $G^{j,i}$ out-performs G^0, then the better-performing edge is retained and the other edge is discarded. For further details on using the RIC score to assess edges, we direct readers to Gu and Zhou [2020].

A.4 Computational Complexity of the Divide-and-Conquer Method

The motivation for the divide-and-conquer method developed in this work is that existing causal discovery algorithms' computational complexity grows prohibitively large as the number of variables in the dataset increases. The computational cost of the divide-and-conquer method includes the expense of producing the initial partition $\{D_i\}_{i=1}^N$, constructing the causal expansion $\{D_i \cup \partial_{\text{out}}(D_i)\}_{i=1}^N$, running some causal discovery algorithm on each of the subsets $D_i \cup \partial_{\text{out}}(D_i)$, and merging the results using Algorithm 1.

For most reasonable choices of partitioning algorithms, the dominant cost in the divide-and-conquer procedure will be running causal discovery on each subset in the expansive causal partition. A key aspect of the divide-and-conquer method's speedup is that the problem of running causal discovery on each subset of the expansive causal partition is

embarrassingly parallelizable. For any causal discovery algorithm, let $\mathcal{F}(\cdot)$ describe the worst-case runtime as a function of the number of random variables in the input, and let p denote the number of random variables in the global dataset. Running causal discovery on N subsets in parallel on a machine with N processors reduces the dominant cost from $\mathcal{F}(p)$ to $\max_{i\in[N]} \mathcal{F}(|D_i| + |\partial_{\text{out}}(D_i)|)$. In settings where $P < N$ processors are used, the wall-clock time for performing the parallelized causal discovery on subsets using P processors is given by Brent's law [Smith, 1993] [1] and is

$$O\left(\max_{i\in[N]} \mathcal{F}(|D_i| + |\partial_{\text{out}}(D_i)|) + \frac{1}{P}\sum_{i=1}^{N} \mathcal{F}(|D_i| + |\partial_{\text{out}}(D_i)|)\right).$$

For all causal discovery algorithms satisfying Assumption 1 known to the authors at the time of publication, the computational complexity $\mathcal{F}(\cdot)$ is dramatically super-linear such that $\sum_{i=1}^{N} \mathcal{F}(|D_i| + |\partial_{\text{out}}(D_i)|) \ll \mathcal{F}(p)$, as shown below for the FCI algorithm.

As an example, we describe the computational complexity of the full divide-and-conquer procedure in a simplified setting. Consider a variable set V of cardinality p, and assume the superstructure G reflects strong community structure. Specifically, consider the stochastic block model setting: each variable in V belongs to one of two equally-sized hidden communities and that for any two variables $u, v \in V$, G contains an edge between u and v with probability q_{in} if u and v belong to the same community, and probability q_{out} if they belong to different communities. We consider a regime where these communities can be efficiently recovered with high probability: assume

$$q_{\text{in}} = \alpha/p, \quad q_{\text{out}} = \beta/p$$

for $\alpha, \beta > 0$ and $\sqrt{\alpha} - \sqrt{\beta}$ sufficiently large. This regime is well-studied, and in this setting

1. The first term in the expression is the dominant cost in time if the problem is run on N processes. The second term is the time if the problem is run in serial divided by the actual number of processes available P

known clustering algorithms exactly recover the underlying partition with high probability in nearly-linear time $O(p\log^2(p))$ (see e.g. Abbe and Sandon [2015], Wang et al. [2020]). Employing these clustering algorithms, with high probability we recover an initial partition D_1, D_2 such that $|D_1| = |D_2| = p/2$. Moreover, in expectation $|\partial_{out}(D_1)| = |\partial_{out}(D_2)| = p\beta/4$, so conditioned on exact recovery the size of the clusters in the causal expansion are $(1/2 + \beta/4)p$ in expectation.

We consider running causal discovery using the FCI algorithm, which satisfies Assumption 1. While the FCI algorithm does not perform *every* pairwise conditional independence test, in the worst case its runtime still scales exponentially with the size of the input variable set, e.g. $\mathcal{F}(p) = c^p$ for $c \in (1, 2)$ [Spirtes, 2001]. When run in parallel on two processors, the wall-clock time will be exponential in the size of the subsets of the expansive causal partition: conditioned on exact recovery, for any fixed $\delta \in (0, 1-\beta/4)$, $|D_i \cup \partial_{out}(D_i)| \leq (1/2+\beta/4+\delta)p$ with high probability. This corresponds to reducing runtime from c^p to $c^{(1/2+\beta/4+\delta)p}$. The complexity of merging the results of the causal discovery output using Algorithm 1 is a function of the number of learned edges and maximum learned degree. In the SBM setting, with high probability its runtime is $O(p(p+q)p^3) = O(\alpha(\alpha+\beta)p^2)$.

In total, the time to run the divide-and-conquer procedure when parallelizing the causal discovery step is

$$O\left(p\log^2(p) + c^{(1/2+\beta/4+\delta)p} + \alpha(\alpha+\beta)p^2\right)$$

with high probability. In the large variable regime, the exponential term $c^{(1/2+\beta/4+\delta)p}$ dominates the runtime of the divide-and-conquer procedure. We compare this with the runtime of the FCI algorithm on the global variable set, which is $O(c^p)$. For small values of β, the divide-and-conquer method reduces runtime compared with the runtime of FCI on the global variable set from c^p to $\approx \sqrt{c^p}$, which represents considerable computational savings in the regimes of interest when p is large. This small-β regime corresponds to the setting where few intra-cluster edges are present.

A.5 Controlling Maximum Subset Size

A key factor in accuracy-timing trade-offs is controlling the size of the largest subset in the partition. Here we observe that the largest subset produced by the causal expansion in Section 4.3.4 is governed by specific connectivity properties of the initial partition on which it is built. In particular, for graphs with strong community structure, if the initial partition is strongly correlated with community structure, then the resultant subsets in the causal expansion will not be much larger than any of the subsets in the input.

For the causal expansion defined in Section 4.3.4, the maximum size of any subset is controlled by the sizes of the subsets in the input partition and their corresponding vertex expansion values. For any set S such that $|S| \leq |V|/2$, the vertex expansion of S in graph G is defined as

$$h(S) \equiv \frac{\partial_{\text{out}}(S)}{|S|}.$$

If the input expansion $\{S_1, \ldots, S_N\}$ satisfies $|S_i| \leq |V|/2$ for all $i \in [N]$, then the size of subsets $\{S'_1, \ldots, S'_N\}$ in the causal expansion is controlled as

$$\max_{i \in [N]} |S'_i| \leq \max_{j \in [N]} (1 + h(S_j))|S_j|.$$

In particular, if the superstructure G has strong community structure and the initial partition $\{S_1, \ldots, S_N\}$ is constructed appropriately, then the subsets of the causal expansion will not be dramatically larger than those in the initial partition. See Appendix A.7.

A.6 Experimental Setup

A.6.1 DAG generation

Ground truth DAGs, G^*, are synthetically created using a Barabasi-Albert scale-free model [Barabási and Bonabeau, 2003]. A random topological ordering is imposed on the nodes so

that the graph is acyclic. Data is generated assuming a Gaussian noise model: $(X_1, ..., X_p)^T = ((X_1, ..., X_p)W)^T + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \sigma_p^2)$. W is an upper-triangular matrix of edge weights where $w_{ij} \neq 0$ if and only if $i \to j$ is an edge in G^*. The variance σ^2 is uniformly sampled from $(0, 1]$. Each column vector X_i represents the data distribution for a variable corresponding to a node i in G^*. For our experiments we create graphs (p=50) with two communities, where each community has a Barabasi-Albert scale-free topology and communities are connected using preferential attachment. Any cycles created by this are removed to ensure the graph is a DAG.

A.6.2 Partitioning Algorithm Parameter settings

In this section we describe the parameter settings for the partitioning algorithms: *Disjoint*, *Edge Cover*, *Expansive Causal* and *PEF*. The *Disjoint* partition is `networkx.greedy_modularity`. For networks with 100 communities of size 100 nodes, we set the `best_n` and `cutoff` both to 100. For all other experiments we do not set these parameters. These parameters are also used for the *Edge Cover* and *Expansive Causal* partitions, and there are no additional parameters that need to be set here. For *PEF*, the minimum size of each community is set in order to select the optimal partitioning. For networks with 100 communities of size 100 nodes, this is set to 1% of the size of the node set. For all other experiments this is set to 5% of the size of the node set.

A.6.3 Causal Discovery Algorithm Parameter settings

In this section we describe all the parameter settings for the causal discovery algorithm $\mathscr{A}$ used for local learning. The four algorithms we chose are GES, PC, RFCI, and DAGMA. For GES, PC, and RFCI we use the R implementations in the `pcalg` library. For PC and RFCI the significance level α was set to 0.001. For GES, `fixedGaps` which controls which edges are added in the greedy search is set to all the edges not in the superstructure, `maxDegree`

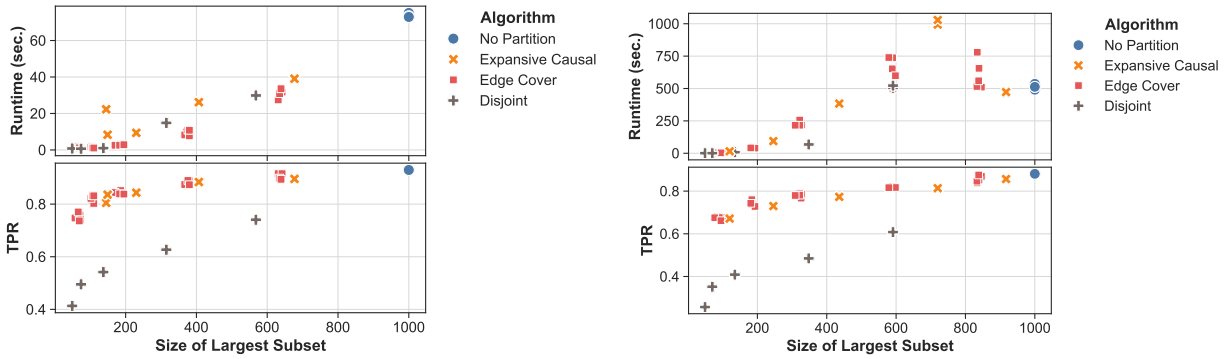


Figure A.2: Left: Accuracy and time trade-off for 1,000 node hierarchical scale-free graphs with 1,000 samples. **Right:** Accuracy and time trade-off for 1,000 graph with ten communities of size 100 with scale-free topology with 1,000 samples. We see that certain subset sizes take unexpectedly long for GES learner.

is not limited, and `adaptive` is set to "`triples`". For DAGMA, we used the `LinearModel` with L2 loss. During optimization `lambda1` was set to 0.02, while all other parameters were set to the default. These parameters were consistent across all experiments.

A.6.4 Details on merging DAG subgraphs

Causal discovery algorithms GES, PC and DAGMA do not satisfy Assumption 1 and instead output a DAG (for the PC algorithm we randomly choose a graph in the MEC). Despite this, we still employ a screen operation when merging subgraphs in a similar manner to Algorithm 1. However, given there are no consistency guarantees over the observed subset of nodes for these algorithms, it is possible that after merging the resultant directed graph contains cycles. Therefore for these algorithms we employ the algorithm discussed in Appendix A.3 which accounts for the presence of cycles. For these learners Algorithm 3 now takes in a set of DAGs instead of PAGs and returns a DAG.

A.7 Time and Accuracy trade offs

The computational bottleneck for divide-and-conquer algorithms is the size of the largest subset: $\max_i |S_i|$ for a partition $\{S_1 \dots S_N\}$. This is because we expect causal discovery algorithms to converge to an estimated graph faster for smaller variables sets (the graph space defined by a smaller variable set is smaller). However, we observe that the convergence of GES appears to be a function of *both* size of the subset, and the topology of G^*. Fig. A.2 (left) shows the time to solution and TPR as the size of the biggest subset increases. For this study, we use a 1,000 node hierarchical scale-free graph. This is equivalent to the types of graphs in Section 4.6.4. To control the size of the subsets we fix the number of communities and resolution for the greedy modularity disjoint partition – we sweep through five different disjoint partitions, increasing size of the largest subset. Here, we see expected scaling behavior – as the size of the largest subset increases so does the time solution. The largest time to solution is for the non-partitioned method on the entire 1,000 node graph. This means that partitioning the graph always enables some scaling. The *Expansive Causal* and *Edge Cover* partitions are extensions of each *Disjoint* partition. We observe that for our proposed partition methods (Expansive Causal Partition and Edge Cover) we do not increase the size of the largest subset significantly (this aligns with notes in Appendix E), and we benefit from a significant boost in accuracy. In Fig. A.2 (right) we run the same study but with a 1,000 node graph with 10 communities, each with size of 100 and scale-free topology. This is equivalent to the types of graphs in Section 4.6.1 through Section 4.6.3, but with more communities. Here, we observe good scaling when the size of the largest partition is small and close to the size of the natural communities. However beyond this, the time to solution increases to be even larger than the non-partitioned method. This suggest that certain 'bad' subsets incur a longer convergence time for the GES causal discovery algorithm. We hypothesize this is related to violation of causal sufficiency of these subsets – subsets that contain more confounders (unobserved common causes) outside may result in sub-optimal

convergence in the GES learner. Note that this result is not due to our causal partition or the divide-and-conquer methodology, but rather because of the use of the GES learner in this setting. Still, since we can control the size of the subsets with the disjoint partition we can still achieve accuracy and time benefits with GES as shown in our empirical results.

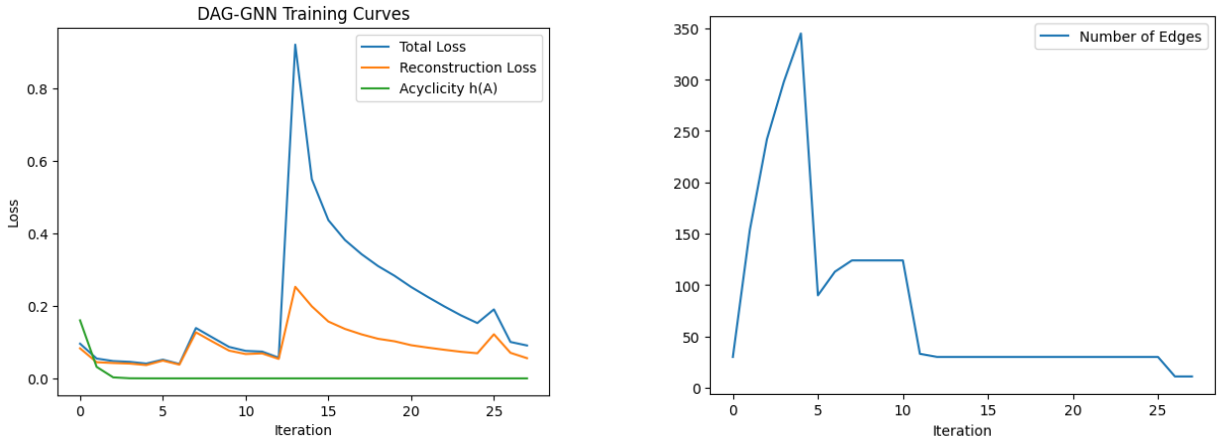
APPENDIX B

ADDENDUM FOR CHAPTER 5

B.1 `DAG-GNN` training at Scale

`DAG-GNN` is trained using a nested optimization loop where an inner loop updates the neural network parameters and adjacency matrix to minimize the evidence lower bound (ELBO) loss, while an outer loop enforces the DAG constraint using an augmented Lagrangian. A trainable adjacency matrix is represented by $A \in \mathbb{R}^{p \times p}$. The encoder is $z = (I - A^T)x$ for input $x \in \mathbb{R}^{n \times p}$, where I is the identity matrix. The decoder is $(I - A^T)^{-1}z$. We set the embedding dimension to 64. We trained `DAG-GNN` for 300 epochs using Adam with a learning rate of 3e-3 (`gamma`=1.0, `lr_decay`= 200). We use n for the batch size, because we do not have sufficient samples for stable mini-batching. The maximum number of iterations for the Lagrangian optimization was set to 100. The threshold for the adjacency matrix A is set to 0.3, which is the default value.

`DAG-GNN` training is time and memory intensive. To learn an adjacency matrix over $p = 15,694$ would be intractable. We partitioned the gene set according to a causal partition [Shah et al., 2025] of the protein-protein interaction network from the STRING database, then merged the graphs by taking the union of edges. Training times depend on the gene subset size, which vary by dose rate. The average training time for `DAG-GNN` on one subset of data across bootstraps was 23.3 min. Training curves for one example subset are shown in Fig. B.1. Note that loss spikes occur when the augmented Lagrangian penalty ρ is increased in the outer loop, abruptly strengthening the acyclicity constraint and perturbing the learned graph. This is expected behavior as the model rebalances reconstruction accuracy with enforcing a valid DAG structure. Models were trained on a NVIDIA Tesla V100-SXM2-32GB GPU.



(a) **(b)**

Figure B.1: Training curves for DAG-GNN of one subset of genes in the graph partition. (a) Loss curves and acyclicity constraint $h(A)$ which converges to 0 to ensure a DAG. (b) The number of edges in the learned adjacency matrix at each iteration using the default threshold 0.3

B.2 Linear Regression Baseline

We were concerned that causal gene sets might just be identifying genes that have strong correlative structure with the radiation vector, especially given a small sample size. We constructed a baseline algorithm based on linear regression of (week, dose_rate) labels onto each gene. Each model's R^2 measures how much variance in that gene's expression is explained by dose rate and week jointly. A gene with a high multiple correlation R^2 is one whose expression varies strongly as a linear function of dose rate and/or time. This captures genes that respond to radiation in a dose- or time-dependent manner — but purely through univariate linear association with the experimental conditions, without any causal graph structure or model-based feature importance. It serves as a "correlative baseline" against which the causal graph gene selections are compared. Similar to the supervised machine learning framework, we choose the 1000 genes at each fold with the top R^2 scores. Then we select genes that are stable in more than 50% of folds. This results in 481 genes and is visualized as a "Correlation" baseline in Fig. B.2. We also perform pathway enrichment of this gene set and report the top 10 pathways in Table B.1. We see enrichment of radiation

pathways as well, but a much higher enrichment for the invariant causal gene set (Table 5.3 which also has a smaller gene set size.

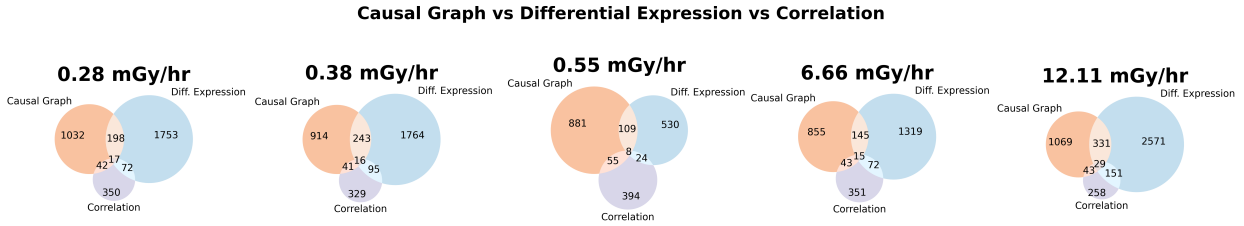


Figure B.2: Intersections with the Correlation baseline algorithm (based on Linear Regression).

B.3 Additional Gene Set Intersections

There were 68 stable genes inferred by the `RandomForestRegression` models across folds. Intersections with causal gene sets and differential expression are shown in Fig. B.3. Fig. B.4 and Fig. B.5 show within causal graph and differential expression intersections respectively. There are a core set of "invariant" genes across dose rates for each method which may correspond to genes involved in fundamental radiation response processes. The invariant gene set is larger for causal graphs than differential expression (438 genes compared to 329 genes).

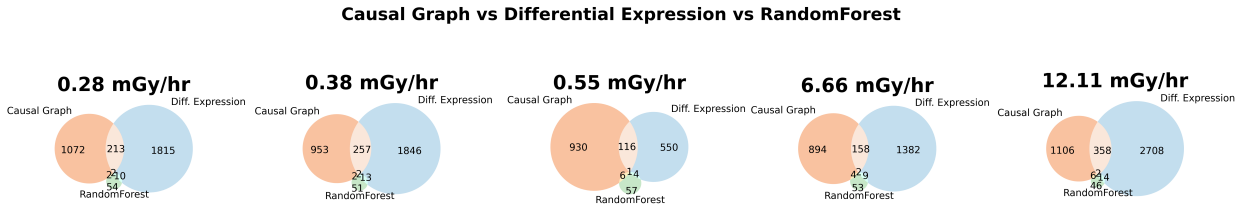


Figure B.3: Intersections with the Random Forest gene set which is dose invariant.

B.4 Random and Correlative Gene Sets

Pathway enrichment $-\log_{10}p$-values were surprisingly large for causal gene sets (Fig. 5.4). To check that these values were meaningful, we ran a simple test with `gProfiler` with random

Table B.1: Top 10 Enriched Pathways — Correlation Gene Set

Pathway	$-\log_{10}(p)$	**Term Size**	**Intersection Size**
Cell Cycle Checkpoints REAC:R-HSA-69620	8.28	289	24
Regulation of TP53 Activity through Phosphorylation REAC:R-HSA-6804756	7.70	92	14
Regulation of TP53 Activity REAC:R-HSA-5633007)	7.18	160	17
G1 to S cell cycle control WP:WP45	7.15	64	12
Cell cycle WP:WP179	6.96	120	15
Mitotic G1 phase and G1/S transition REAC:R-HSA-453279	6.82	148	16
Retinoblastoma gene in cancer WP:WP2446	6.79	89	13
Cell cycle KEGG:04110	6.30	157	16
DNA damage response WP:WP707	6.02	69	11
DNA replication GO:0006260	5.84	279	20

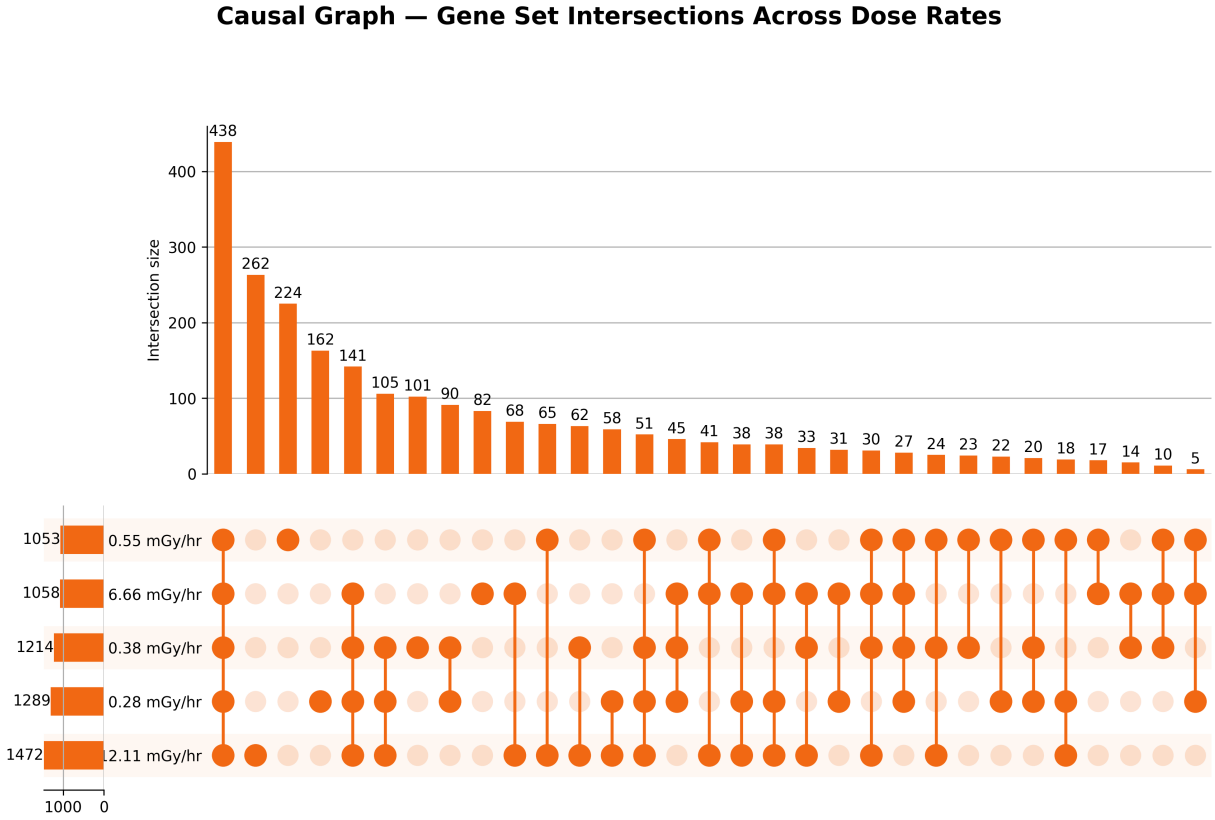


Figure B.4: Intersections of causal graph gene sets across each dose rate. There is an "invariant" set of 438 genes which is the intersection across all dose rates.

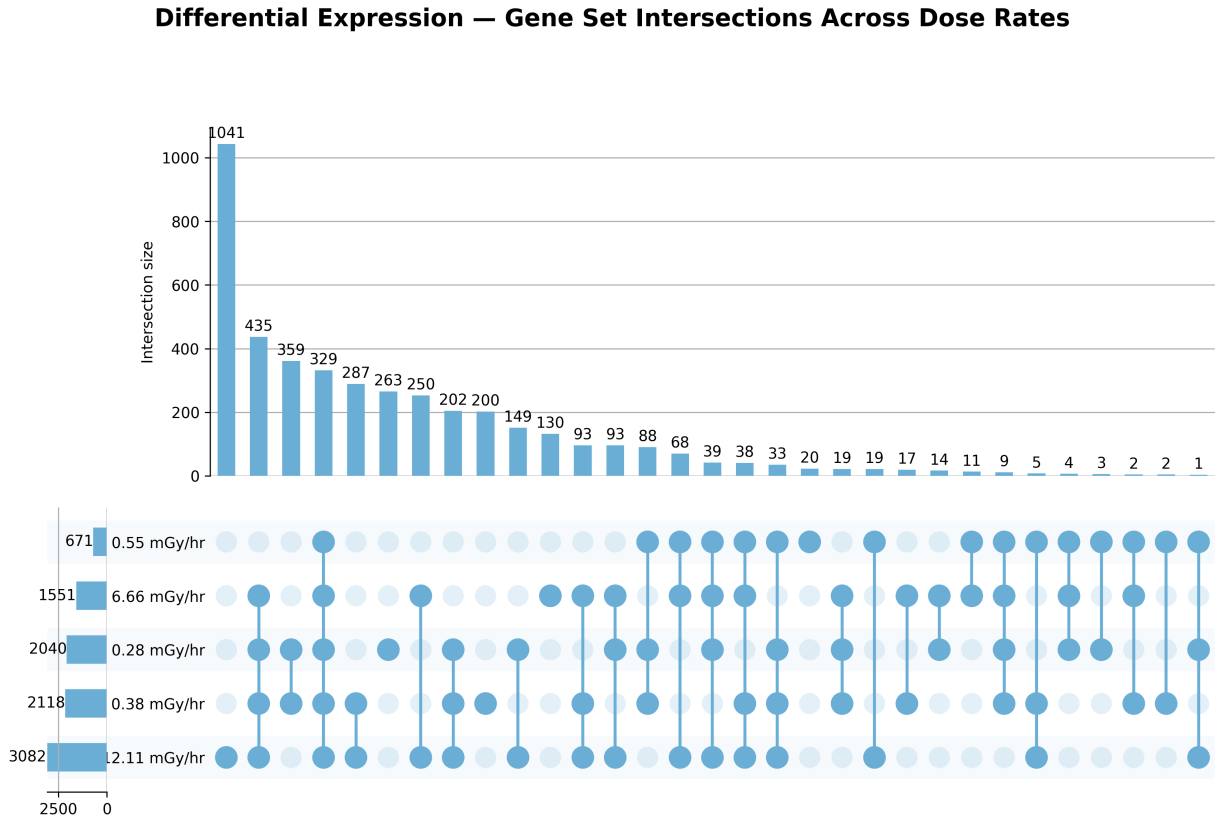


Figure B.5: Intersections of differential expression graph sets across each dose rate. There is an "invariant" set of 329 genes which is the intersection across all dose rates.

genes taken from the background gene set (all genes measured in the experiment). In Fig. B.6 we report the distribution of the top 10 enriched pathways (with 5 repeated runs). We verify random gene sets do not enrich any pathways. We also check our correlative linear model and see that these gene sets do enrich pathways, but not at the level of the causal gene sets. We already demonstrate in Fig. 5.4 that the causal gene sets perform better compared to differential expression and `RandomForest` gene sets.

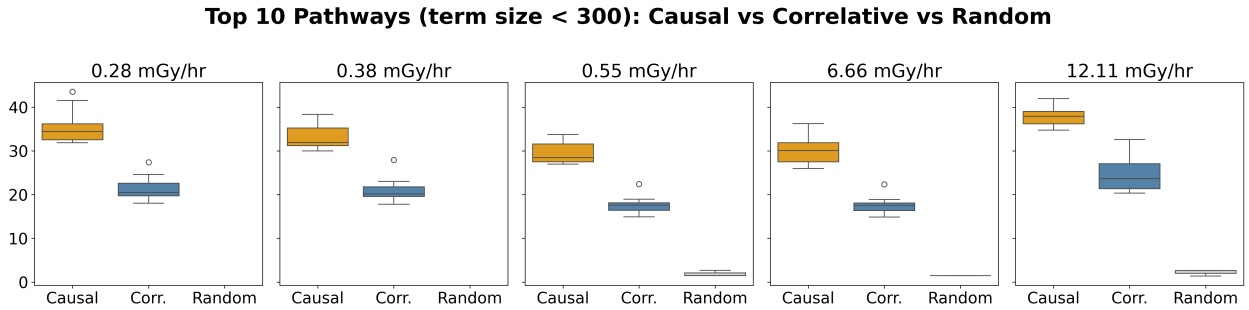


Figure B.6: As a sanity check, we computed pathway enrichment for the top 10 pathways from the causal, correlative, and random gene sets.

B.5 Top 10 Pathways for Differential Expression Invariant Gene sets

We report the top 10 pathways for the differential expression invariant gene set (329 genes) in Table B.2.

B.6 Cluster Functional Analysis

We performed a cluster analysis of the causal graphs at each dose rate; an example clustering is shown in Fig. B.7. To understand if each cluster is biologically meaningful and encodes distinct processes, we again perform pathway enrichment on each gene subset. We also compute overlaps to CORUM protein complexes which are manually annotated, experimentally validated, high quality annotations. In Table B.3 we see significant enrichment of

Table B.2: Top 10 Enriched Pathways — Differential Expression Invariant Gene Set

Pathway	$-\log_{10}(p)$	**Term Size**	**Intersection Size**
cellular response to interleukin-1 GO:0071347	9.73	84	13
response to interleukin-1 GO:0070555	9.40	110	14
cellular response to tumor necrosis factor GO:0071356	7.68	208	16
glycosaminoglycan binding GO:0005539	7.15	249	17
response to tumor necrosis factor GO:0034612	7.09	229	16
glomerulus development GO:0032835	6.97	71	10
active monoatomic ion transmembrane transporter activity GO:0022853	6.71	117	12
Diabetic cardiomyopathy KEGG:05415	6.45	202	15
regulation of angiogenesis GO:0045765	6.42	293	17
nephron development GO:0072006	6.40	160	13

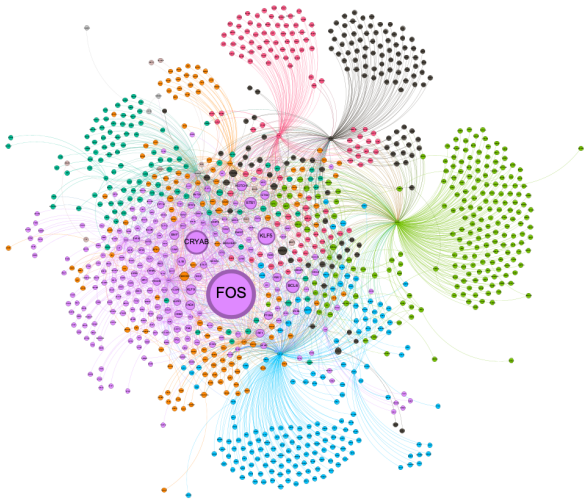


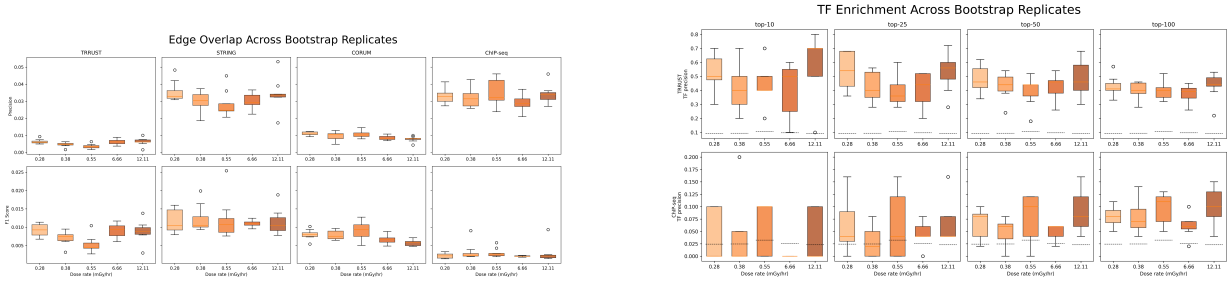
Figure B.7: Example of the causal graph at dose rate 6.66 mGy/hr colored by cluster ID.

Table B.3: Cluster enrichment analysis for dose rate 6.66 mGy/hr. For each cluster, we measure the gene overlap with CORUM complexes and identify the protein complex with highest overlap (and associated exact Fisher test p-value). Top pathways for each cluster and identified with `gProfiler`.

Cluster	**CORUM Complex**	**p-value**	**Top Pathway**	**p-value**
C0	DNA binding; chromosome; DNA topological change	4.81×10^{-5}	Antigen presentation: folding, assembly, and peptide loading of class I MHC	8.10×10^{-21}
C1	Cell cycle checkpoint signaling; DNA binding; nucleus	1.26×10^{-4}	Pathogenic *Escherichia coli* infection	5.09×10^{-7}
C2	Cytoplasm; translation	2.94×10^{-24}	Eukaryotic translation elongation	6.66×10^{-22}
C3	ATP binding; chromosome; DNA replication	7.66×10^{-10}	ATP-dependent activity, acting on DNA	8.70×10^{-9}
C4	mRNA splicing via spliceosome; spliceosomal complex	1.93×10^{-5}	Mitotic G1 phase and G1/S transition	1.79×10^{-16}
C5	Protein targeting; intracellular transport; endocytosis	1.82×10^{-2}	NRF2 pathway	3.91×10^{-6}
C6	Myeloid cell differentiation; regulation of apoptosis	9.58×10^{-9}	Integrated breast cancer pathway	9.83×10^{-10}

pathways and CORUM complexes in each cluster for dose rate 6.66 mGy/hr. However, only C2 (Eukaryotic translation) and C3 (ATP-DNA binding) show clear consensus between the top pathway and the CORUM complex.

B.7 Causal Discovery Bootstrap Results



(a) (b)

Figure B.8: Bootstrap `DAG-GNN` results. (a) The distribution of edge overlap with knowledge bases for all graphs across 10 bootstrap runs. (b) The percentage of top-k hub nodes that correspond to known transcription factors for all graphs in the 10 bootstrap runs.

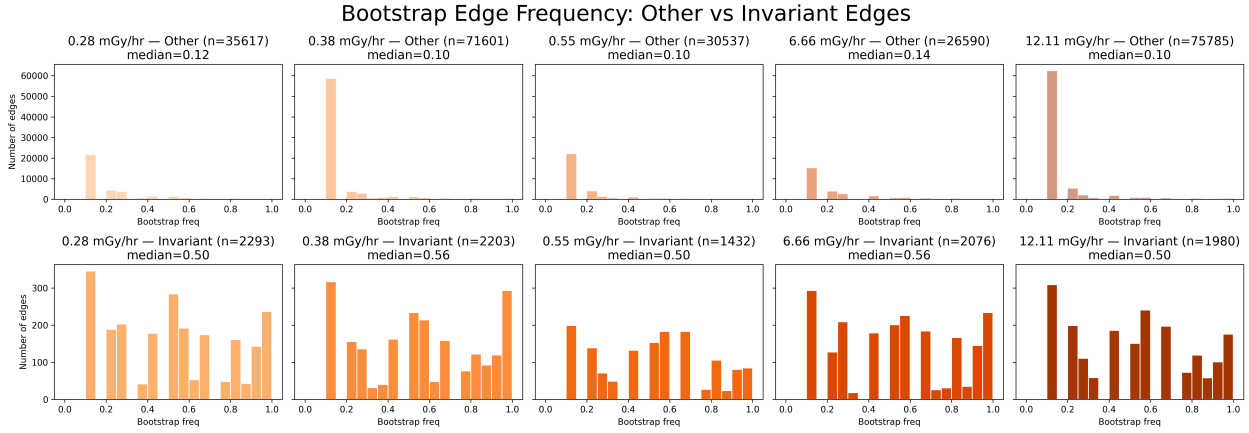


Figure B.9: Top: The distribution of edge frequencies at each dose rate across bootstraps. A frequency of 1.0 means that the edge appeared in all 10 of the DAGs. Bottom: The distribution of edge frequencies at each dose rate subgraph induced by the causal invariant gene set (438 genes).

Here, we show the `DAG-GNN` metrics across bootstrap runs. The results in the main paper report metrics for consensus graphs formed by filtering for edges that co-occur in 50% or more

of the bootstrap runs. We ran DAG-GNN with 10 bootstraps and metrics are shown in Fig. B.8. Compared to Table 5.2 and Fig. 5.5, we see a high variance distribution and overall poorer results which supports our choice of choosing consensus edges for downstream analysis. In Fig. B.9, we visualize the distribution of edge frequencies for subgraphs induced by the invariant causal gene set. We see that this subgraph has a higher density of "perfect" edges that appear in all bootstraps. This shows that the invariant gene sets and subgraph edges have a higher confidence, and this is supported by enrichment of only radiation pathways. These subgraphs and high frequency edges can be used as a starting point for hypothesized novel pathways, however, this requires experiments to prove.

B.8 Other Graph Algorithms

We considered two additional non-causal, graph inference methods, GENIE3 and Lasso. Both formulate network inference as an ensemble of regression problems, where candidate parent genes are treated as covariates for predicting the expression of each target gene. In GENIE3, candidate parent genes are pre-filtered to only include genes that code for transcription factors in humans. GENIE3 uses a RandomForest regression model, while Lasso uses lasso regression.

We report pathway enrichment results of GENIE3 and Lasso compared to DAG-GNN and DESeq2 in Fig. B.10. Here we visualize the average p-value over the manually selected radiation response pathways in Fig. 5.4 as a function of the size of the gene set. Genes are filtered according to LFC p-value (DESeq2), or edge weight (DAG-GNN, Lasso, GENIE3). This measures the discriminatory power of each method in identifying important genes for radiation response; ideally, a method ranks these radiation-specific genes highest, with the smallest gene set size. We see that GENIE3 and Lasso have the highest discriminatory power as smaller gene sets are enriched in radiation response pathways at a higher rate.

We further compare GENIE3 and Lasso to knowledge bases in Fig. B.11. For equal

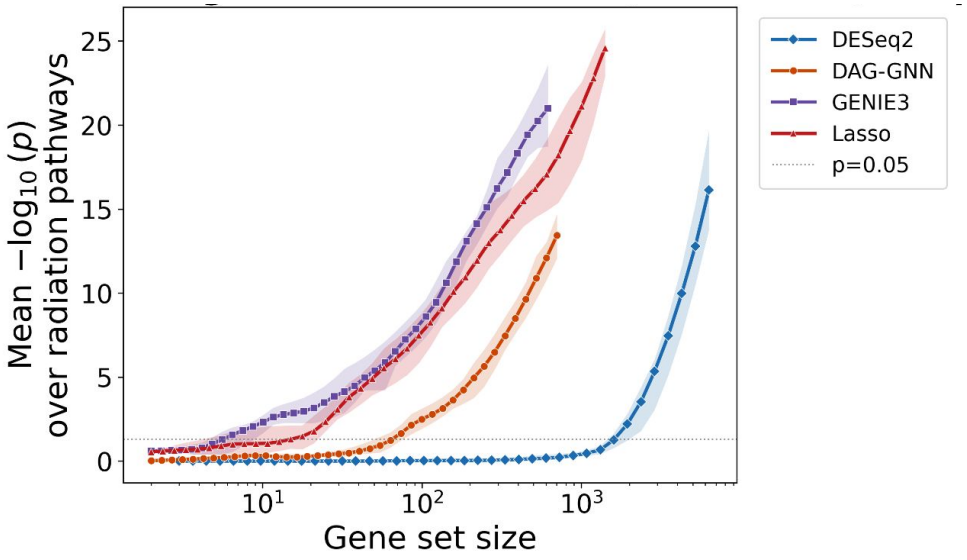


Figure B.10: The average enrichment as $-\log p$ of radiation specific pathways as a function of gene set size for each method.

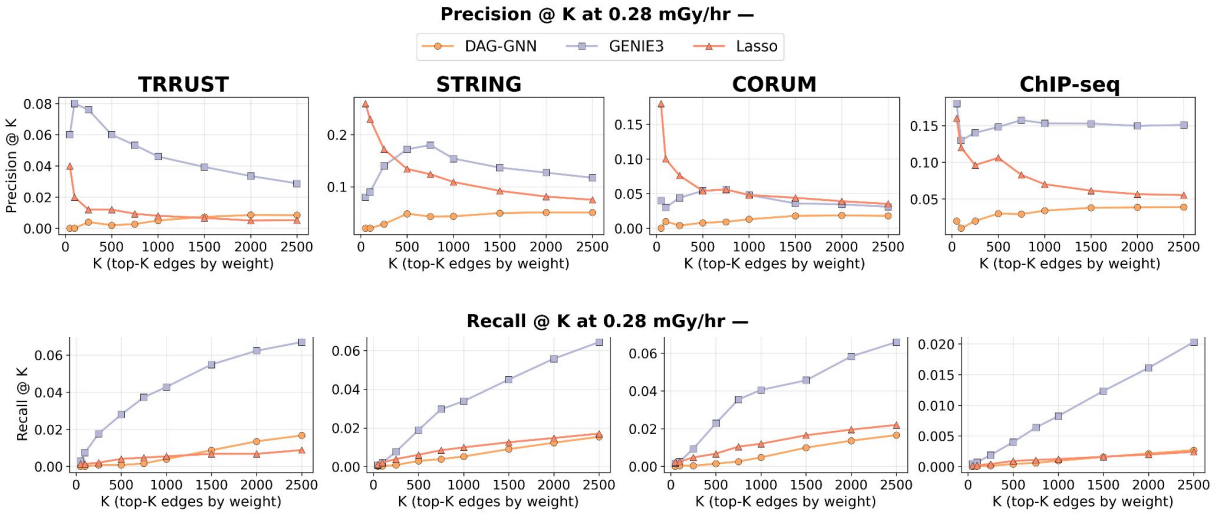


Figure B.11: Edge comparison of all graph inference algorithms to knowledge bases as a function of sparsity of each graph, where edges are pruned by weight.

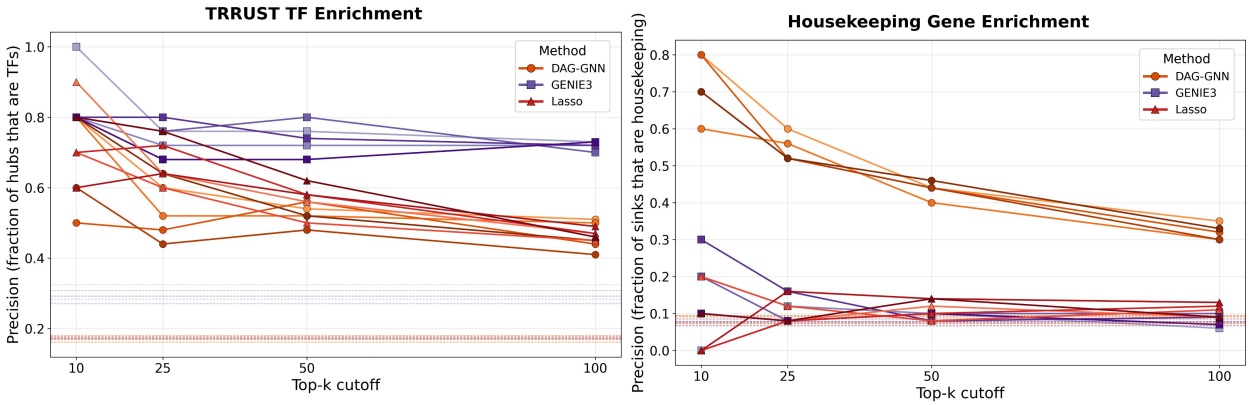


Figure B.12: Left Percent of top-k out-degree nodes (hubs) that are known transcription factors. **Right**
Percent of top-k in-degree nodes (sink) that are housekeeping genes. Light to dark is low to high dose rate for each method. Dashed lines show the perfectage if genes are randomly assigned to nodes. Only `DAG-GNN` assigns housekeeping genes as sink-nodes.

comparison, we control for sparsity by measuring precision and recall as a function of the top-k edges filtered by weight. `GENIE3` is the best performing algorithm, while `DAG-GNN` is the worst performing algorithm for reconstructing knowledge bases.

Finally, we look at the topology of inferred graphs and the biological significance of highly connected nodes in Fig. B.12. All methods assign the majority of hub nodes to transcription factors; `GENIE3` has the highest and most stable enrichment due to the pre-filtering step which only assigns transcription factors as parents. Only `DAG-GNN` assigns the majority of sink nodes as housekeeping genes. The DAG structure tells us that genes that regulate other genes are upstream in the graph whereas genes that are closer to representing the state of the cell are downstream. This suggests that the enforcement of the DAG topology reveals meaningful biological structure. The story is nuanced, cheaper graph algorithms like `GENIE3`

and `Lasso` are much better at recovering known edges however `DAG-GNN` topologies encode higher level biological organization.