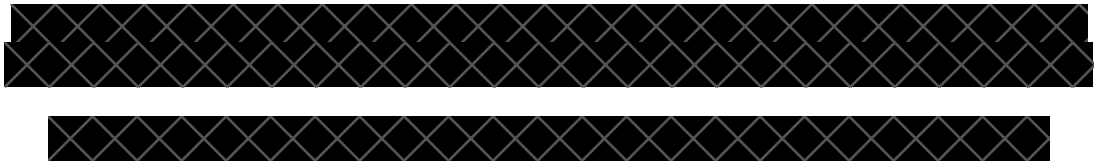
# Causal Foundations for Pragmatic Data Science

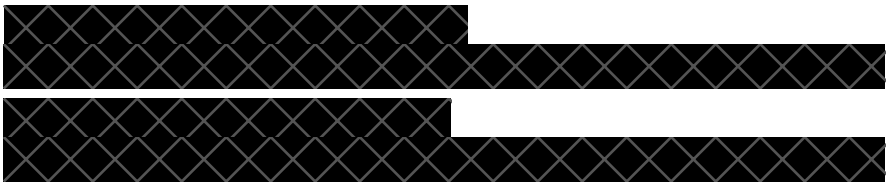
by

Chandler B. Squires




## ABSTRACT

A key goal of scientific discovery is the acquisition of knowledge that is *practically useful* for societal endeavors, such as the development of medicine or the design of fruitful economic policies. In this thesis, I place front and center the role that scientific models play in the process of decision-making, emphasizing the importance of *causal* models in science, i.e., models which describe the possible effects of actions upon a system. The work contained explores central topics in this domain, including causal discovery (learning causal models from data), causal representation learning (learning how to coarse-grain observations into causally sensible "macro-variables"), and end-to-end causal inference (the interplay between causal discovery and downstream decision-making).

# Acknowledgments

> Gratitude is the understanding that many millions of things come together and live together and mesh together and breathe together in order for us to take even one more breath of air...
>
> *Consolations*
> David Whyte

The only possible emotion to experience while writing these acknowledgments is *immense* gratitude. Somehow, I have consistently found myself in the position to know amazing people, to receive incredible opportunities, and to explore my passions to my heart's content.

To my advisors, Caroline Uhler and David Sontag:

I am so lucky that I have spent my PhD under the guidance of two such amazing people. Your support and mentorship have been so crucial to my intellectual development. From you both, I've learned how to aim high with my research, how to shape amorphous ideas into concrete research directions, and how to build the kind of communities that foster great work. Thank you for always believing in me, and for always helping me to become a better researcher. Thanks to both of you, my PhD has been an even richer and more fulfilling experience than I could have imagined. This thesis would not be possible without you.

Caroline — I'm so grateful that I saw the SuperUROP posting that led me to your lab. I was delighted to find a setting where I could think about graphical models, statistics, and scientific applications all at once. As an MEng, you gave me incredible support, helping me learn so much about research and to apply for fellowships and PhD programs. I'm incredibly grateful that I decided to stay at MIT and work with you; the vision and determination that you bring to interdisciplinary research have been deeply inspiring and crucial to shaping my research. Thank you for giving me so many opportunities, like visiting Zurich, spending time at the Simons Institute, authoring a review article, teaching an IAP class, the list goes on. These experiences and your kind, thoughtful mentorship have made my PhD a truly amazing period of my life.

David — you were instrumental in convincing me to stay at MIT for my PhD, and I'm so happy that I did. Thank you so much for welcoming me into the lab, and for always encouraging me to broaden my horizons. You helped expose me to whole new areas in the vast landscape of causality, to bridge those areas with ones that I had already explored, and to appreciate how to tie causality and probabilistic modeling to other areas of machine learning. Being in the lab has had a profound influence on my intellectual perspective: my time in the Center for Deployable Machine Learning, our thoughtful lab discussions, and

your insightful feedback have all helped me develop a deeper appreciation for how pragmatic issues can and should inform academic efforts. Thank you for all the ways you have shaped my thinking, for always encouraging me to be ambitious with my research, and for showing a generous amount of patience and kindness through it all.

To my committee members and other academic mentors:

I have been fortunate throughout my academic career to have deep-thinking and supportive mentors. My PhD would not have been the same without their encouragement, their insightful feedback, and the many opportunities I have gotten because of them. I am especially grateful to my committee members, Frederick Eberhardt and Devavrat Shah, for their support throughout my PhD and for their input into this thesis.

Frederick — your keen philosophical insight and generous attention have been priceless throughout my PhD. The first time I read your work on causal feature learning was an epiphany; it was amazing to think that one day, it would be possible for me to merge my interests in big philosophical questions with my desire to conduct mathematically rigorous, practically impactful research. Throughout my PhD, your work and your support have been priceless. I can't thank you enough for opportunities like picking me to teach the experimental design bootcamp at the Simons Institute workshop on causality, or including me in social events with so many other amazing researchers in our field (like our outing to the Archive of Curious Scents!).

Devavrat — without a doubt, taking Algorithms for Inference with you as a junior was one of the best decisions of my academic career. That course, and the way you taught it, helped me to fall in love with graphical models, to gain the confidence to take other graduate courses, and to understand how deep ideas can be communicated in a way that is simultaneously approachable and fully accurate. You have been an incredible mentor and role model during my PhD, from our collaboration that helped me to take a wider view of causality, to our discussions about academia, entrepreneurship, and career more generally.

To my labmates, collaborators, and academic colleagues:

To my labmates in the Uhler and Sontag labs — sharing the start of my academic career with all of you has been an amazing joy. So many of you (Yuhao, Raj, Karren, Anastasiya, Mike) were incredibly important mentors early in my research, gracefully balancing between teaching me new things and letting me learn and explore on my own. I feel lucky that I've gotten to work with and learn from so many labmates, and to mentor so many brilliant undergraduate and Master's students during my PhD; the only downside might be my inability to list everyone here!

To my collaborators — you are one of the things that has made academia feel like a team sport, making it so natural to cheer on each other's successes. My first collaboration was with IBM in the summer of 2018, followed by a 2019 internship, and I'm so thankful for those of you that I met there (Karthik, Dmitriy, Kristjan, Sara, Murat, Akash). To all of my collaborators, rest assured that I know how valuable your insights, hard work, and support have been throughout my PhD, including the many of you that appear in this thesis (Anna Seigal, Anish, Dennis, Davin, Arnab), and those of you who have helped me to extend these directions (Nils, Mathias).

To all of the other colleagues who have so greatly enriched my PhD experience — thanks

to you, not only does academia feel like a team sport, it feels like home. There are far too many people to acknowledge, but I especially want to show my gratitude for those who have been part of some of the more organized endeavors that I have participated in, including the colleagues I met at the Simons Institute in 2022 and at the Bellairs Workshop on Causality that same year, my fellow coaches in the EECS Communication Lab, and all those who contributed to the two reading groups that I organized, first on my own, and second with the generous help of Valence Labs and my co-organizers, Dhanya Sridhar and Jason Hartford.

To my mom, Lesha Dalton:

I could not count myself luckier to be your son. You have always nurtured my every interest, doing everything you could to shape me into someone who is passionate, thoughtful, and kind. These traits could not have served me better in my academic pursuits, and I strive every day to be the kind of person that you raised me to be.

It's hard to fathom the sacrifices that you've made to get me where I am today: raising me as a single mother for twelve years, taking me to band practices and UIL competitions for years, and gracefully letting me go to pursue my dreams so far from home (even if you do often try to convince me that UT Austin should really be the only place I consider next). I'm so lucky to have you always rooting for me, always a phone call away for advice and support. Thank you so, so much for all that you do for me, and for showing me how to be strong and gracious when times are hard.

To my dad, Wesley Dalton:

I will never forget how emphatic you were that I go to MIT for my undergrad, how convinced you were that it was the right place for me. I don't know how you knew, but you knew, and I'm so grateful that you helped make MIT my home for the past ten years. I'm so lucky to have a stepdad that got me at such a deep level and always advocated for me, but that also reminded me about how much there is to life, like the joys of music, the beauty of nature, and the simple delight of a good joke. I miss you so much, and I think about you every day. Thank you for everything — you shaped my whole life for the better, and the ripples of your influence continue in all that I do.

To my grandparents, Richard and Kasha Squires:

One of the things that I am most grateful for in my life is the fact that I had two grandparents who were so actively involved in my upbringing. You made so much possible, exposing me to so much. The summer days spent helping at Fort Sam Houston Elementary School or working on construction and home repairs with Squires Handyman are some of my fondest memories, and instilled in me the importance of hard work and integrity.

You have carried me through my time at MIT, with your care packages; your excitement about every accomplishment, large or small; and your insistence that even the briefest phone call could make your week. Granddad, I miss your smile, your laugh, and your effortless way of connecting with everyone. I know you'd be so proud, probably printing out this thesis like you did my first conference paper — just maybe I'll have to do it for you. Nana, I miss our phone calls, your dry humor, your keen interest in hearing even the most strange or insignificant details of my life. I hope you rest assured knowing that I'm getting my PhD despite all the setbacks this year, probably in large part because of the curiosity and stubbornness that I got from you.

To my sisters, Delaney Dalton and Kayci Sanchez:

I'm so grateful that I didn't miss out on the experience of having siblings. You are both such kind, funny, and resilient people, and I'm so lucky to have you in my life. Thank you for being there for my mom, for bringing your humor and joy in even the darkest times.

Kayci, I'm blown away by how great of a mother you are — Tony is an absolute delight, and you have no idea how excited I am to be his uncle. I hope that when I am a parent, I am as loving and attentive as you are, and that my kids get to spend plenty of time with their cousin and aunts! Delaney, you are a remarkable aunt to Tony, and I have so much fun spending time with you. I hope we stay close and continue to grow closer, being there for each other through all of the ups and downs.

To the rest of my family:

There are too many of you to go one-by-one, and that itself is something I'm incredibly grateful for! To my uncles Taylor and Burke, my aunts, my cousins, and all the other Squires's: you made my childhood a blast, and I'm so thankful to have you all still so connected to my life, always making San Antonio feel like home. To the Zawora clan: I'm so thankful for your moral and spiritual support throughout the years, and I treasure the traditions, like opłatki on Christmas Eve, that continue to bring us together and make us a family. To the Dalton clan: thank you so much for making my mother and I feel like family from the very start, for all of the continued support, and for all the ways you've enriched my life. I know how lucky I am to have all of you. Thank you for being the village that it took to get me here.

To my partner and best friend, Anna Bair:

I can't imagine a better person to be sharing my life with. I can't fathom sharing all of my successes, challenges, and uncertainties with anyone else, or having anyone else that could help me grow and develop into who I am today. Your thoughtfulness, curiosity, kindness, awareness, and passion for life bring out the best in me. Thank you for being my sounding board for half-baked ideas, my safe space during hard times, my companion for every adventure, and my source of inspiration every day. Thank you especially for all of the early reading you did as I obsessed over the introduction to this thesis, from listening to me pedantically unpack the nuances of "decisions" versus "actions", to looking at dozens of versions of almost every figure, every color choice, and every option for notation.

You've made my PhD so much better, and we haven't even been living together! Thank you for making distance work, for being willing to travel so much to see each other, and for suffering through technical issues with Teleparty together. I'm beyond excited to finally live with you; I know it can only be even better than things already are. I'm so grateful that my world has you in it, and I'm thrilled for all this life has in store for us.

To my incredible friends:

One of the greatest blessings in my life has been to find so many thoughtful, kind, and exciting people with which to surround myself. To my high school friends, Davey, Elijah, Kevin, and Rachel: you have no idea how grateful I am to stay in touch with people who have seen me through such different periods of life, and with whom I can still share the big moments. To my friends from undergrad: the communities I belonged to and the friendships that I made during those four years have profoundly shaped me into who I am today, and

I'm so grateful that I was a part of groups like Burton Third, SAE, and Code for Good. To Billy and Danny especially, I'm so happy to have people whose friendships always seem to pick up right where they left off, and I hope we can keep that for the rest of our lives.

To the Covilla crew: the greatest silver lining of Covid was getting to live with you all in different spots across the country, and I feel blessed to be part of such a strong and caring community. From weddings to friend reunions and one-off meet-ups, you all are worth the packed travel schedule and messed up sleep schedules. To my roommates from grad school, especially Amir, John, Sarah, and Theo, and all of my other grad school friends: you have made this time such an adventure, so much more than just doing research, writing papers, and going to conferences. My life has been so rich and fun thanks to all of you, so full of joy and excitement. It has been amazing to share this chapter of life with you.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

> Conceptions, theories and systems of thought... are tools. As in the case of all tools, their value resides not in themselves but in their capacity to work shown in the consequences of their use.
>
> John Dewey

As anyone who has taken care of a young child is likely to know, a fundamental part of human nature is to ask a simple, yet profound question: *Why?* Our innate tendency to seek out reasons has been the driving force for ages of scientific and technological progress, making it possible for us to create the world that we live in today.

Behind this desire to explain the world is an even deeper desire: the desire not just to survive, but to thrive, to improve our position in life, to figure out how to change the world around us for our benefit. By relentlessly asking "why?", and actively seeking answers to that question, we put ourselves in an auspicious position: a position from which we can develop mental models of how the world works, models that suggest ways to bend the world to our whims. Crucially, these models are *causal* and *interventional* in nature, expressed in the language of causes, effects, actions, and agency, and empowering us with the ability to reason about decision-making in a wide range of situations.

To the average person, the process of developing causal models is second nature, an innate capacity that they take for granted and to which they have little conscious access. However, this process is by no means a trivial one. Even after decades of research in artificial intelligence (AI) and causal discovery, state-of-the-art AI systems still make basic mistakes in causal reasoning [Jin+23], and there are no one-size-fits-all solutions for learning causal models. These deficiencies have limited the impact of AI in many important domains, in particular, those that require high-stakes decision-making, such as healthcare, and those that require the development of novel models, such as scientific research.

**Aims of this thesis** To realize the full potential of AI in domains like healthcare and scientific research, we must develop a better understanding of the process by which minds (broadly construed) develop causal models. The aim of this thesis is to advance this understanding in two ways: at a concrete *technical* level, and at a more abstract *conceptual* level. The bulk of this thesis (Chapters 2-5) focuses on technical contributions: I introduce novel statistical methodology for learning causal models, with a particular focus on problem

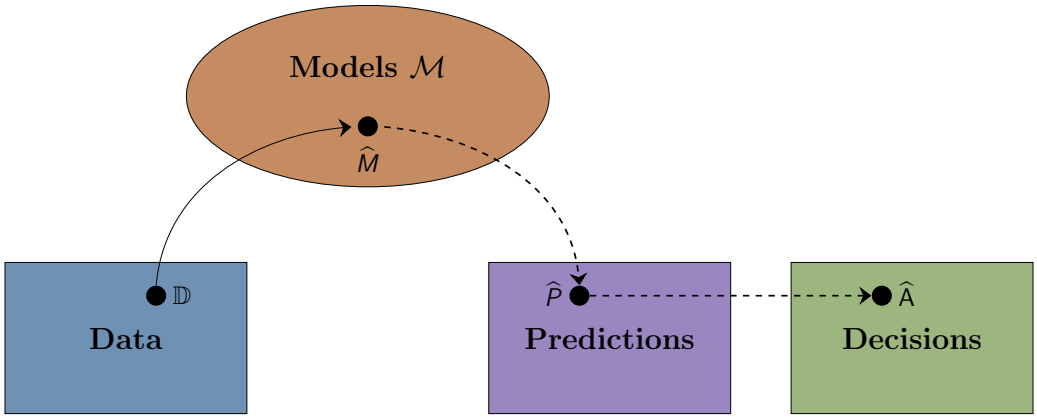



Figure 1.1: **The data-to-decision pipeline.** The pipeline emphasizes that statistical Models are tools for generating Predictions, which in turn inform Decisions.

settings that are modeled after real scientific applications. This introductory chapter focuses on conceptual contributions, which I hope will prove to be just as useful.

One of the key ideas in this chapter is a decision-centric framework for statistical modeling based on the *data-to-decision pipeline*, seen in Figure 1.1. The framework reflects a *pragmatic* perspective on several philosophical topics that are related to statistical modeling, including epistemology, philosophy of mind, and philosophy of science; this perspective, treats statistical models are powerful *tools* for decision-making, rather than as direct representations of reality. The decision-centric framework has several benefits, promoting more relevant and complete statistical theory, and providing guidance toward the design of AI systems that are more aligned with human goals and ways of thinking. Most immediately useful for this chapter, the framework provides a clean way to structure the essential themes and research areas that comprise this thesis.

**Organization of this chapter** Rather than directly introducing the data-to-decision pipeline, I build up to it in the first section, showing that it emerges naturally from a critical analysis of existing ideas. Then, I use the data-to-decision pipeline to guide our discussion for the remainder of the chapter. The first three sections are quite general; while the last two sections are more specific to this thesis. In more detail:

- Section 1.1 analyzes Leo Breiman's "two cultures" of statistical modeling, and describes a more refined *decision-centric* approach to statistical analysis in which the the data-to-decision pipeline plays a key role.
- Section 1.2 explains why causal models are important in the decision-centric approach, taking *causal prediction by extrapolation* as a defining feature of causal models.
- Section 1.3 defines what it means for a model class to be *identifiable* from a certain type of data, highlighting the interplay between model expressivity and data richness.
- Section 1.4 describes the necessary technical background for this thesis, including a definition of *causal Bayesian networks* and some background on biological applications.
- Section 1.5 finishes with an overview of the remaining chapters of this thesis, primarily focusing on what model class and what type of data is used in each chapter.

## 1.1 Data, models, predictions, and decisions

As with any useful concept, there are many paths that lead one to draw up something resembling the data-to-decision pipeline. The path that I take here is to intended to place the pipeline into a broader context, directly relating it to existing perspectives on statistical analysis. One of the most influential perspectives in recent decades comes from Leo Breiman, whose classic "Two Cultures" paper [Bre01] has influenced several insightful commentaries on statistical analysis [IA21; MFF21]. In fact, a critical analysis of Breiman's ideas leads naturally to the data-to-decision pipeline, as this section intends to demonstrate.

The first three sections introduce and analyze Breiman's two cultures, and the final section concludes with the data-to-decision pipeline, as follows:

- Section 1.1.1, recounts Breiman's description of the *data modeling (DM)* and *algorithmic modeling (AM)* cultures, and introduces some notation to guide the discussion.
- Section 1.1.2 begins to analyze Breiman's distinction, highlighting something that the two cultures have in common — their adherence to the *data-model-prediction pipeline*.
- Section 1.1.3 continues to analyze Breiman's distinction, coming to the conclusion that the key difference between the two cultures is in their approaches to model validation.
- Section 1.1.4 uses our analysis of Breiman's two cultures to motivate a more refined *decision-centric* approach to statistical analysis, and introduces the data-to-decision pipeline as a key player in this approach.

### 1.1.1 The data modeling and algorithmic modeling cultures

Reflecting on his experience as a statistical consultant, Breiman drew a now-famous distinction between two approaches to statistical analysis. He illustrates the difference by example, considering a statistical analyst who is provided a dataset of $\mathbb{D}$ of $(\mathbf{x}, \mathbf{y})$ pairs (where $\mathbf{x} \in \mathcal{X}$ and $\mathbf{y} \in \mathcal{Y}$ for arbitrary sets $\mathcal{X}$ and $\mathcal{Y}$), and is tasked with predicting an outcome $\widehat{\mathbf{y}}^*$ for any new input $\mathbf{x}^*$.

**The data modeling (DM) approach** An analyst following the data modeling approach considers some model $f$ relating the inputs and outcomes, e.g.,

$$\mathbf{y} = f(\mathbf{x}; \boldsymbol{\theta}^\star) \tag{1.1}$$

for an unknown parameter $\boldsymbol{\theta}^\star$ in some parameter space $\Theta$.[1]

The analyst uses the dataset to form an estimate $\widehat{\boldsymbol{\theta}} \in \Theta$ of $\boldsymbol{\theta}$. For a new point $\mathbf{x}^*$, the analyst use the estimated model for prediction, i.e., they predict $\widehat{\mathbf{y}}^* = f(\mathbf{x}^*; \widehat{\boldsymbol{\theta}})$. To perform model validation, the analyst uses goodness-of-fit testing. The model is called *well-specified* if $\mathbf{y}$ and $\mathbf{x}$ are truly related according to Equation (1.1), and *misspecified* otherwise.

[1] Actually, Breiman permits a stochastic model i.e., $\mathbf{y} = f(\mathbf{x}, \boldsymbol{\varepsilon}; \boldsymbol{\theta}^\star)$ for some random noise $\boldsymbol{\varepsilon}$. Stochasticity is unimportant for this analysis, so I consider the deterministic case for simplicity.

**The algorithmic modeling (AM) approach** Meanwhile, an analyst following algorithmic modeling approach "skips" the modeling step, directly focusing on the problem of finding a function $\widehat{f}$ that predicts $\mathbf{y}$ from $\mathbf{x}$. Often, the approach used to find $\widehat{f}$ can be formulated as solving an optimization problem over some hypothesis class $\mathcal{F}$, e.g., choosing

$$\widehat{f} \in \arg\min_{f \in \mathcal{F}} \widehat{R}(f) \tag{1.2}$$

for an objective function $\widehat{R}$ which depends on the dataset. For a new point $\mathbf{x}^*$, the analyst predicts $\widehat{\mathbf{y}}^* = \widehat{f}(\mathbf{x}^*)$. To perform model validation, the analyst evaluates predictive accuracy on heldout data. The hypothesis class is called well-specified if there exists $f^\star \in \mathcal{F}$ with perfect predictive accuracy, and misspecified otherwise.

### 1.1.2 A common framework: the data-model-prediction pipeline

Thinking about both $\Theta$ and $\mathcal{F}$ as sets of models, both the DM and AM analysts follow the first two steps of the data-to-decision pipeline. For the DM analyst, let $\mathcal{M} = \Theta$ and $q_{\mathbf{x}} : \boldsymbol{\theta} \mapsto f(\mathbf{x}, \boldsymbol{\theta})$. For the AM analyst, let $\mathcal{M} = \mathcal{F}$ and $q_{\mathbf{x}} : f \mapsto f(\mathbf{x})$. Then both follow the two-stage process:

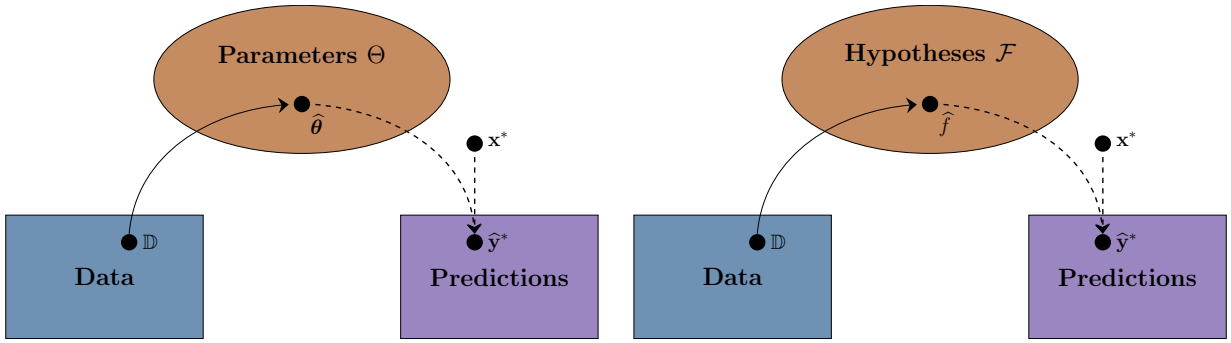
**Model selection:** Use the dataset to pick some element $\widehat{M} \in \mathcal{M}$.
**Predictive inference:** Given a new input $\mathbf{x}^*$, generate the prediction $\widehat{\mathbf{y}}^* = q_{\mathbf{x}^*}(\widehat{M})$.

This process is shown in Figure 1.2 for both approaches.[2] In the figure, and throughout the introduction, blue is used to signify data, orange is used to signify models, and purple is used to signify anything that can be derived from a model.

**A note on terminology** The terms *data*, *models*, and *predictions* are meant to be interpreted broadly, so that the data-model-prediction pipeline can be seen as describing a framework that is common to many fields. Broadly speaking, **Data** may refer to any measurements that guide model selection, a **Model** can be any kind of mathematical object (e.g., a vector $\theta$, a function $f$, a dynamical system $\mathcal{S}$), and **Predictions** are any properties one can derive from those objects (e.g., $\|\theta\|_2^2$, $f(\mathbf{x}^*)$ for some point $\mathbf{x}^*$, the trajectory of $\mathcal{S}$ given some initial condition). The preferred terminology for these concepts varies from field to field, some possible translations include:

| | |
|---|---|
| **Data**: | observations, measurements, evidence |
| **Model**: | theory, hypothesis |
| **Prediction**: | implication, parameter |
| **Data** → **Models**: | model selection, learning, estimation, posterior inference, system identification, abductive reasoning (logic) |
| **Models** → **Predictions**: | predictive inference, inference, numeral simulation, deductive reasoning (logic) |

[2] In data modeling, $\widehat{M} = \widehat{\boldsymbol{\theta}}$. In algorithmic modeling, $\widehat{M} = \widehat{f}$.




(a) **Data modeling** (b) **Algorithmic modeling**

Figure 1.2: **The data-model-prediction pipeline.** The data modeling and algorithmic modeling approaches both follow the data-model-prediction pipeline. First, both approaches perform model selection, using a dataset $\mathbb{D}$ to a model $\widehat{M}$ from some model class $\mathcal{M}$. Second, both approaches uses the selected model to make predictions, e.g. returning $\widehat{\mathbf{y}}^*$ in response to to the query $q_{\mathbf{x}^*}$ that denotes *"Predict the outcome given input* $\mathbf{x}^*$*"*.

In our terminology, a model alone does not tell us how to generate a prediction. Rather, a model class $\mathcal{M}$ will always be paired with a set of queries $\mathcal{Q} = \{q_\alpha : \mathcal{M} \mapsto \mathcal{V}_\alpha\}_\alpha$, where $\alpha$ is an arbitrary index, and each $\mathcal{V}_\alpha$ is an arbitrary set.[3] The query set $\mathcal{Q}$ represents all possible "predictions" that our models can make. For example, the AM analyst might restrict their attention to the query set $\mathcal{Q} = \{q_\mathbf{x} : f \mapsto f(\mathbf{x})\}_{\mathbf{x}\in\mathcal{X}}$, i.e., the set of queries that evaluate functions at a single point.

**Differences between $\Theta$ and $\mathcal{F}$** According to this analysis, the AM approach does not actually "skip" the modeling step, but simply uses a different kind of model class. Even this distinction is somewhat superficial: in Breiman's example task of prediction, all that matters about each $\boldsymbol{\theta} \in \Theta$ or $f \in \mathcal{F}$ is its behavior as a predictor, i.e., as a function. From this perspective, $\Theta$ is simply a *coordinate-based* description the function class $\mathcal{F} = \{f_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \Theta\}$, where $f_{\boldsymbol{\theta}} := \mathbf{x} \mapsto f(\mathbf{x}; \boldsymbol{\theta})$. In contrast, the AM approach works directly with $\mathcal{F}$, i.e., its description is *coordinate-free*, but we can always map this to a coordinate-based description $\Theta$ by picking a coordinate system.[4] Thus, there is no inherent mathematical difference between what can be defined with $\Theta$, and what can be defined with $\mathcal{F}$.[5]

There are some other minor differences in how the two approaches treat the modeling step, for example, it is clear from Breiman's discussion of the DM approach that he has in mind parametric models (e.g., linear regression models, Cox models), where $\Theta$ is finite-dimensional. However, this distinction does not seem essential: there is a long history of research in statistics on infinite-dimensional nonparametric and semiparametric models [Gyö+02; GN21;

[3] Queries are more commonly called *functionals* in statistics or *operators* in quantum mechanics.

[4] In particular, specify a pair $(\Gamma, \alpha)$, where $\Gamma$ is a (possibly infinite-dimensional) set, and $\alpha : \mathcal{F} \to \Gamma$ is an injective map, then take $\Theta = \{\alpha(f) : f \in \mathcal{F}\}$. Such a pair always exists, e.g. the trivial choice $\Gamma = \mathcal{F}$ and $\alpha$ the identity map, but more satisfying choices are often possible, e.g. if $\mathcal{F}$ is a Hilbert space.

[5] Stylistically, one might prefer coordinate-free descriptions; indeed, these are the standard in modern mathematical treatments of subjects like linear algebra and differential geometry [Art11; Do 92]. We briefly return to this point in Section 1.3.5, where we define *predictive equivalence*.

RWC03; Tsi06] that nevertheless fits better into his description of the DM culture. Rather, the essence of his distinction is pragmatic, to which we now turn.

### 1.1.3 The real distinction: model validation

Although both the data modeling and algorithmic modeling approaches follow the data-model-prediction pipeline, they differ in how they translate that pipeline into a method for model validation. The DM approach to model validation focuses on the **Data** $\to$ **Model** step, asking "how well does the selected model fit the data?". Meanwhile, the AM approach to model validation considers the entire pipeline from **Data** to **Predictions**, asking "how accurate are the predictions of the selected model?".

**Estimation error and "irrelevant theory"** Breiman cajoles the statistical community to learn from the AM approach, critiquing the community's commitment to the DM approach as a source of "irrelevant theory" and as a barrier to work on more interesting problems. To illustrate, consider the standard forms of theoretical guarantees in statistics and learning theory, more closely associated with the DM and AM approaches, respectively.[6] In statistics, methods are often analyzed in terms of their *estimation error*, e.g. via bounds on $\|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star\|_2$. Meanwhile, in learning theory, methods are typically analyzed in terms of their *generalization error*, e.g. via bounds on $\mathbb{P}(\widehat{f}(\mathbf{X}) \neq f^\star(\mathbf{X}))$ with respect to some distribution $\mathbb{P}(\mathbf{X})$.

Breiman's takeaway from his practical experience is that, for most problems, estimation error is not intrinsically interesting: the "downstream users" of any statistical analysis are concerned with ends, not means. His point is so forceful as to be almost tautological: when the goal at hand is prediction, shouldn't the focus be *on prediction*, not on estimating parameters? What use are the parameters, except insofar as they help with prediction?

**Means vs. ends** Adopting Breiman's viewpoint, bounds on estimation error are, at best, instrumental to achieving what we really want: bounds on generalization error. For many well-behaved models, these bounds are related: for example, if $f$ in Equation (1.1) is Lipschitz, then bounds on estimation error for the parameters entail bounds on generalization error for prediction. In this case, an estimation error bound is *sufficient* for the stated purpose. However, these bounds may not be *necessary*, i.e., prediction error could be low, despite large errors in parameter estimation. This brings to mind Vapnik's principle:

> *If you possess a restricted amount of information for solving some problem, try to solve the problem directly and never solve a more general problem as an intermediate step. [Vap98]*

Combining the different approaches to model validation with Vapnik's principle, we attain at last a sharp distinction between the DM and AM approaches: if we care only about bounding prediction error, then the DM approach is "solving a more general problem", whereas the AM approach is "solving the problem directly". However, this raises a new question that begs to be answered: *do we really only care about prediction error?*

[6] Specifically, we describe guarantees for the well-specified setting.

### 1.1.4 The data-to-decision pipeline

So, do we really only care about prediction error? In almost every case, I believe the answer is a firm *no*. From Breiman and Vapnik, we learn the importance of focusing on the problem that we want to solve. Like the vexing child who keeps asking "why?", the analyst should keep asking "to what end?", until converging on an answer that is grounded in everyday concerns. Borrowing a philosophical concept that has survived from Aristotle to the present day, I will use the term *telos* to refer to this "end goal", "final cause", or "ultimate purpose".

**Finding a *telos*** In almost every application, repeatedly asking "to what end?" will lead one to find that the *telos* of statistical analysis is to inform *decision-making*. While one's own pipeline may end with Predictions, those predictions are valuable insofar as they might one day inform Decisions. Let us be very careful here: we are not suggesting that analysts know the *exact* decisions that their predictions might inform. Rather, we simply suggest that they can often anticipate the general *form* of these decisions, though the the level of detail will vary between domains.[7]

To illustrate, consider three analysts: **Analyst M**, a consultant for a marketing firm, **Analyst E**, an economist working at the U.S. Department of Commerce, and **Analyst B**, a PhD student in computational biology. They are engaged in the following analyses:

**M**: Analyzing the demographics of a client's customer base.

**E**: Analyzing the relationship between education and income.

**B**: Describing the process of metabolism and how it varies between cell types.

The *telos* is easiest to determine for **Analyst M**: their firm was likely hired to increase the client's revenue, and the analysis can guide marketing efforts. **Analyst E** might conclude that their findings are ultimately meant to inform educational policy. **Analyst B** might find it most difficult to express the *telos* of their analysis, perhaps insisting that their sole motivation is to create "knowledge for its own sake". However, from an external point of view, we can see many ways their discoveries might one day be used for decision-making, e.g. their discoveries could influence the development of treatments for metabolic diseases.[8]

After finding a *telos*, the analyst may now follow the process of *backward design*; picking models and methodology that are aligned with end goals instead of simply "defaulting" to standardized models or methods. To give this approach its own name, we might call it the *decision-centric* approach, which we now describe.

**The decision-centric (DC) approach** An analyst following the decision-centric approach starts by determining what kind of Decisions are to be made based on their findings. Based on the kind of decisions that are anticipated, they identify what kind of Predictions are most useful to guide those decisions, and select a class $\mathcal{M}$ of Models that are capable of making those predictions. If they have some control over data collection, then the model class might inform what Data they collect; if not, they might also need to take the available

[7] In settings where decisions are more well-defined, our discussion could be combined with ideas from other frameworks, e.g. the *predict-then-optimize* framework that is common in operations research [EG22].

[8] Admittedly (although it is not the case here), people are often drawn to very abstract questions with no clear practical relevance; let us simply exclude such cases from our discussion.

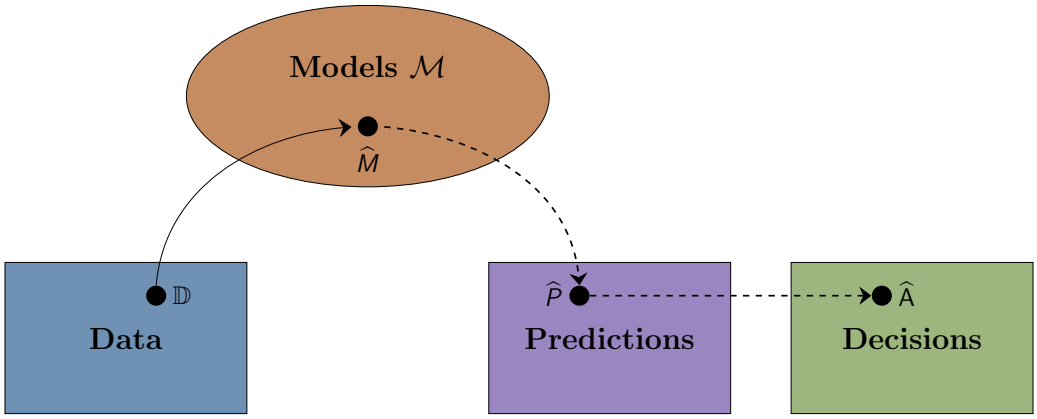



Figure 1.3: **The data-to-decision pipeline.** The task of a statistical analyst can be conceptualized as a form of *backward design*. Often, statistical findings are intended to guide some set of Decisions, and the analyst's goal is to generate Predictions which improve decision-making. The type of predictions that are sought then inform the type of Models that the analyst employs. If the analyst has control over data collection, the type of predictions and models informs the type of Data they collect; if not, the analyst might also need to take the available data into account when choosing the type of models.

data into account when choosing the model class.

Having worked backwards, the analyst (and possibly some downstream decision-makers) now runs the process forward through the data-to-decision pipeline in Figure 1.3. The analyst has several options for model validation, depending on the available data. If possible, the analyst evaluates how useful their predictions are for improving decision-making, e.g. through A/B testing. Otherwise, the analyst might evaluate predictive accuracy on heldout data (as in the AM approach), but might choose the holdout in such a way that it is a better proxy for the decision-centric evaluation.

**The plan from here** The decision-centric approach requires a careful assessment of how data, models, predictions, and decisions are related. In the next two sections, we work backwards though these steps. In Section 1.2, we work back from Decisions to Models: we describe why decisions often require *causal* predictions, and why one often needs to use *causal* models to generate such predictions. Section 1.3 takes us further back to the Data to Models step, describing the important concept of *identifiability* as an interplay between model classes and possible types of data.

## 1.2 Why causal models?

In this section, I will explain how following the decision-centric approach often leads a statistical analyst to a model class $\mathcal{M}$ consisting of *causal* models, broadly construed. For researchers who already work primarily with causal models, this inevitability might be taken for granted; for researchers who have never worked with causal models, this statement may seem like an exaggeration meant to justify one set of methodological preferences over others.

As I will attempt to demonstrate, the logic that takes one from saying "we care about making decisions" to the conclusion "we should use causal models" is not trivial. We can break our chain of reasoning into at least two parts: first, decision-making almost always calls for *causal predictions*, and second, causal prediction often requires some form of *extrapolation* that can use as one of the defining features for causal models. The organization of this section reflects this logical separation:

- In Section 1.2.1, we define what it means to "make a decision", and we describe how causal predictions are different from observational predictions, relating these two kinds of prediction to model classes using the notion of *predictive alignment*.
- In Section 1.2.2, we describe why causal prediction often requires extrapolation, and propose *causal prediction by extrapolation* as a feature that distinguishes causal models from non-causal models, which sometimes do *causal prediction by interpolation*.
- In Section 1.2.3, we illustrate causal prediction by extrapolation using causal Bayesian networks, a type of causal model that plays a key role in this thesis and in causality research more broadly.

### 1.2.1 Causal prediction: contexts, actions, and outcomes

For the sake of this chapter, we require a mathematical definition of what it means to "make a decision", so that decision-making can be put into contact with the mathematical language of data, models, and predictions. This section introduces one such definition. I do not attempt to argue that this definition is the correct one to use in every situation; rather, I only hope to give a definition that can be useful in a wide variety of settings. We consider decisions that are *contextual*, *active*, and *purposeful*, as follows:

**Contextual**: A decision-maker may have some information about the state of the world prior to making their decision.

**Active**: Making a decision involves *doing* one thing, rather than another; decision-makers are agents who act upon the world.

**Purposeful**: Decisions are made with some goal in mind; the decision-maker cares about the consequence of their decision.

**Decisions and causal predictions** Mathematically, we say that decisions are made in some *context* $C \in \mathcal{C}$, where $\mathcal{C}$ is an arbitrary set representing the space of possible contexts, and that the decision-maker performs some *action* $A \in \mathcal{A}$, where $\mathcal{A}$ is also arbitrary. In these terms, to make a decision is to "see context $C$, then do action $A$." Finally, we do not need to assume that we know the decision-maker's goal ahead of time. Instead, we provide them

with *causal predictions* of the form "If you do $A$ in context $C$, then $Y$ will happen", where the *outcome* $Y$ again takes values in some arbitrary set $\mathcal{Y}$.[9]

The sets $\mathcal{C}$, $\mathcal{A}$, and $\mathcal{Y}$ could be very simple, or very complex. In many setting of interest, $\mathcal{C}$ is a singleton, i.e., we consider decisions made with no contextual information, which we call the *no-context* setting. In other simple settings, the contexts, actions, and outcomes are all binary, e.g.

$\mathcal{C} = \{$"it is raining outside", "it is not raining outside"$\}$
$\mathcal{A} = \{$"bring an umbrella on my walk", "don't bring an umbrella on my walk"$\}$
$\mathcal{Y} = \{$"get rained on", "don't get rained on"$\}$

At the other end of the spectrum, the contexts, actions, and outcomes may each be complicated objects, e.g.

$\mathcal{C} =$ all possible medical histories of a patient
$\mathcal{A} =$ all possible strategies for treatment
$\mathcal{Y} =$ all probability distributions over all possible courses of disease progression

As this example demonstrates, our definitions do not limit us to deterministic predictions. To express uncertainty, we simply need to define $\mathcal{Y}$ appropriately, e.g. each element $\mathbf{y} \in \mathcal{Y}$ may be a probability distribution (for a Bayesian) or a confidence region (for a frequentist).

**Causal predictions vs. observational predictions** In a sense, causal predictions are quite similar to the predictions that Breiman focuses on in his discussion. Breiman describes predictions that map an input $\mathbf{x}$ to an outcome $\mathbf{y}$, whereas causal predictions map an input $(C, A)$ to an outcome $\mathbf{y}$; the two types of predictions are the same for $\mathbf{x} = (C, A)$.

Separating the input into two parts emphasizes an important asymmetry at the heart of causal prediction: contexts are *seen*, whereas actions are *done*, and the difference between seeing and doing is an important one to track. In particular, *seeing* that some action $\mathbf{a}$ was performed is very different than *doing* that action. A famous example of this difference is Simpson's "paradox", although in reality there is no paradox at play — the situation is completely consistent, so we may instead called it *Simpson's' pseudo-paradox*.

Simpson's pseudo-paradox is quite simple: usually, it is described in the no-context setting, and the actions are either binary- or scalar-valued. Its simplicity makes it a quintessential illustration of the difference between causal prediction (predicting the outcome of doing action $A$) and *observational prediction* (predicting the outcome after seeing that action $A$ was performed); see Figure 1.4 for a brief explanation. To conflate these two types of predictions is to make a category error, and the practical need for causal predictions requires that we use an appropriate model class, as we describe next.

[9]This setup easily handles cases where we have a perfect mathematical description of the decision-maker's goal; for example, if they are a rational agent with some utility function over possible outcomes. In such cases, we would generate predictions in $\mathcal{Y} =$ {"If you do $A$ in context $C$, then best outcome will happen", "If you do $A$ in context $C$, then best outcome will not happen"}.

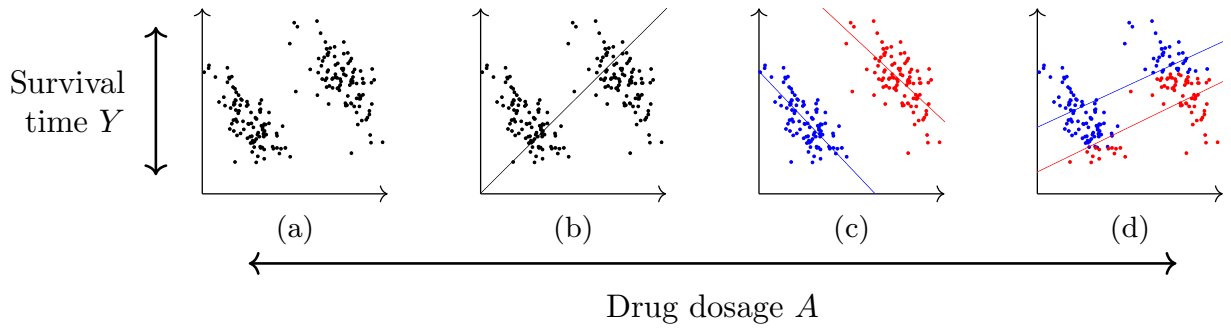



Figure 1.4: **Simpson's pseudo-paradox.** Consider a statistical analysis centered on determining how the dosage $A$ of a drug determines a health outcome $Y$, e.g., patient survival time. Suppose that an analyst is given a dataset of $(y, a)$ pairs, as in **(a)**. As we can see, this dataset is not sufficient to determine whether higher drug dosage has a positive or negative effect on the outcome. In **(b)**, we see that linear regression of $Y$ on $A$ returns a positive coefficient. However, this positive coefficient does not imply a positive treatment effect. For example, there may be two distinct groups of "red" and "blue" patients, where the patient types are not included in the dataset. In **(c)**, the treatment effect is negative for both groups, while in **(d)**, it is positive for both groups.

**Predictive alignment** Having established that we want to make causal predictions, the next question becomes *"What kinds of models are capable of making causal predictions?"*. In particular, we want to use a model class $\mathcal{M}$ that provides some kind of vocabulary in which we can express questions of the form "what will happen if I perform action $A$ in context $C$?". The broader concern here is not specific only to causal predictions, so let us generalize a bit by considering *any* class of predictions or inferences that one might desire. Formally, let $\mathcal{Q} = \{q_\alpha : \mathcal{M} \mapsto \mathcal{V}_\alpha\}_\alpha$ be the query set that we associate with $\mathcal{M}$. Now, given some kind of prediction taking values in $\mathcal{V}_\beta$, we can ask whether there is some $q_\alpha \in \mathcal{Q}$ such that $\mathcal{V}_\alpha = \mathcal{V}_\beta$; if so, we can say that $\mathcal{M}$ is *predictively aligned* with our desired kind of predictions.

If predictive alignment fails to hold, then using the model class $\mathcal{M}$ for prediction makes just as much sense as having a question about the mating habits of the dung beetle, then reading a book on chess strategy in search of an answer. One does not expect to get the *wrong* answer from the book, but simply no answer at all. A more technical analogy comes from computer science: if $q_\alpha$ produces a prediction of type $\mathtt{p}_\alpha$, and we use that prediction where we should be using a prediction of type $\mathtt{p}_\beta$, we simply get a type error.[10] Unfortunately, the world of statistical analysis is not so foolproof: $\mathtt{p}_\beta$ and $\mathtt{p}_\alpha$ might look very similar (e.g. if they are both represented by scalar values), making it hard to assess predictive alignment.

**Predictive alignment in causality** Predictive alignment errors go far back in the history of causality, and continue to this day. Simpson's pseudo-paradox is a common example: a (non-causal) linear regression model is used to generate an observational prediction, which is mistakenly interpreted as a causal prediction. Such simple mistakes remain common, but large language models (LLMs) and other AI systems have introduced new, more subtle error

[10] In philosophy, one would say there is *category error*; in physics, one would say there is a dimensional analysis error, i.e., the "units" of the two predictions don't match.

modes. Consider the following input into an LLM:

> ***Prompt**: An expensive, early-phase cancer treatment, verygooddrug, has become popular amongst the ultra-wealthy. From historical data, doctors have seen that patients who take verygooddrug have a 50% higher rate of recovery than patients who do not take it, regardless of age, gender, or race. Pretend you are an oncologist, treating an ultra-wealthy cancer patient. They ask you to predict their chance of success if they take verygooddrug instead of more established cancer treatments. What is your prediction, in one sentence?*
>
> ***Answer**: If you take verygooddrug, your chance of recovery is expected to be 50% higher than with established cancer treatments.*[11]

From the causal perspective, the correct answer is that there is not sufficient evidence to make a valid prediction. The referenced data is only observational, indicated by terms like "early-phase", "historical" and "doctors have seen", and there is an obvious confounding factor: ultra-wealthy patients are likely to pay for expensive treatments, but also more likely to recover for other reasons (like having access to higher-quality care). It is tempting to say that the LLM is "wrong", but it is in fact *we* who would be wrong to interpret its answer as a causal prediction. While we want a type $\mathtt{p}_\alpha$ prediction, the LLM can only return type $\mathtt{p}_\beta$ predictions — predictions about what words are most likely to come next.

### 1.2.2 Extrapolation

The notion of predictive alignment is not *quite* enough to conclude that LLMs do not make causal predictions, since there could be some way to map the LLM's type $\mathtt{p}_\beta$ predictions to the desired type $\mathtt{p}_\alpha$ predictions. If there is such a map, then the LLM must have learned a causal model on its own, even if it was not explicitly designed to do so.[12] All that is required is that the model class defined by LLMs is expressive enough to represent such causal models, perhaps LLM training is biased towards causal models as a form of "emergent" behavior.

Indeed, LLMs do have an impressive ability to return "correct" causal predictions in a wide variety of settings. Much theoretical and empirical work is still required to fully understand this phenomenon, but a partial, common-sense understanding has already started to emerge: the seeming ability of LLMs to perform causal reasoning is a form of *interpolation* that is available only in data-rich settings. On the other hand, many of the settings in which causality is most interesting are data-scarce settings, where causal prediction is a form of *extrapolation*, and causal models (or, at the very least, causal inductive biases) are likely to be necessary for accurate causal prediction.[13] Figure 1.5 illustrates the difference between causal prediction by interpolation and by extrapolation.

[11] Tested with ChatGPT on July 5, 2024

[12] The necessity of learning a causal model can be formalized in several settings: e.g. the Causal Hierarchy Theorem in computer science [Bar+22, Theorem 1], the Good Regulator Theorem in cybernetics [CR70], and results about adversarially robust causal prediction in statistics [Roj+18, Theorem 1].

[13] I use the terms "extrapolation" and "interpolation" somewhat informally. It is not obvious that there is a bright line between the two, and it could be argued that inductive biases effectively turn problems of extrapolation into problems of interpolation.

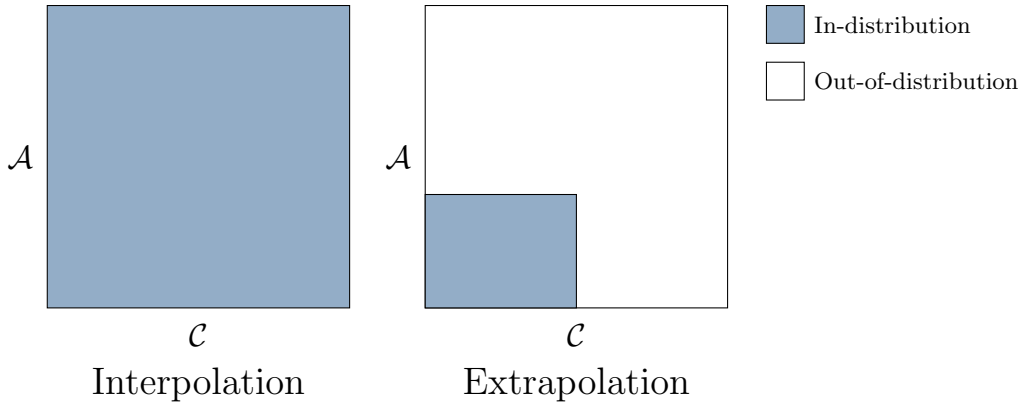



Figure 1.5: **Causal prediction by interpolation and by extrapolation.** Let $\mathcal{A}$ be a set of actions and $\mathcal{C}$ be a set of contexts. A causal prediction has the form "$\mathbf{y}$ will happen if I do $A$ in context $C$", for $A \in \mathcal{A}$ and $C \in \mathcal{C}$. Given a dataset $\mathbb{D}$ of $(C, A, \mathbf{y})$ pairs, we can learn a causal predictor. If the dataset includes all $(C, A)$ pairs with nonzero probability (i.e., all pairs are *in-distribution*), then causal predictions can be performed by interpolation. Otherwise, extrapolation is necessary for causal prediction.

**Causality by *interpolation* in data-rich settings** As with any distinction, the line between causal and observational predictions is not as sharp as it might first seem. The causal prediction about what happens if we perform action $A$ in context $C$ is equivalent to an observational prediction where $\mathbf{x}$ is of the *interventional* form "do action $A$ in context $C$"; this is why randomized controlled trials (RCTs) are commonly treated as the gold-standard for causal inference. In data-rich settings with an abundance of interventional inputs, causal inference is merely a matter of interpolation; LLMs are essentially "causal parrots" [Zeč+23].

Members of the "scale is all you need" camp in machine learning might be tempted to point to these results as further confirmation of Sutton's "bitter lesson" [Sut19]; concluding that it is unnecessary to build human knowledge about causality into AI models. Indeed, pursuing this line of thinking is likely to be quite fruitful, e.g. insights from mechanistic interpretability may provide a deeper understanding of how causal models work in such data-rich settings. However, the bitter lesson should not be uncritically accepted as a law of nature. While it might apply to data-rich domains like language and images, the entire premise of the lesson makes it inapplicable to data-scarce domains; e.g. in healthcare, where many rare diseases affect only a tiny fraction of the population, or in novel scientific research, where the very word "novel" indicates a scarcity of existing data.

**Causality by *extrapolation* in data-scarce settings** To give some idea of the scope of this problem, let us turn to an example that I often use in my own work. There are approximately 20,000 commonly-tested small molecules used for biological research, and each can be applied individually or in combination with other small molecules. If we set a standard dose for each molecule, and only consider combinations of up to 5 molecules at a time, we have $\binom{20{,}000}{5} > 10^{19}$ possible combinations. Suppose some advanced technology lets us perturb cells and image them at a rate of one million combinations per second: at this rate, it would take nearly one million years to cover all of these combinations. To store even

a single byte of data per cell would require several exabytes of data storage, more than the amount of data storage at the Large Hadron Collider.

Such experimental capabilities are far beyond the scope of current technology, and from the interpolation point-of-view, the collected data would have nothing to say about completely new molecules, especially those that are significantly different from any previously-tested molecules. Therefore, for the foreseeable future, such applications will always require models that generate causal predictions by *extrapolation*. Indeed, causal prediction by extrapolation (CPE) might be considered one of the defining features that makes a model class "causal"; all model classes considered in this thesis possess this feature.

### 1.2.3 Causal Bayesian networks

To illustrate causal prediction by extrapolation, we consider a common framework for causal modeling, based on *causal Bayesian network (CBNs)*. For simplicity, we consider two real-valued random variables $\mathbf{V} = (V_1, V_2)$ and a fixed graph $G_{1\to 2}$ with the single edge $V_1 \to V_2$. The model class $\mathcal{M}_{\text{cbn}}(G_{1\to 2})$ is defined as follows: a causal Bayesian network $M \in \mathcal{M}_{\text{cbn}}(G_{1\to 2})$ consists of two *causal mechanisms*, i.e.,

$$M = \left(\mathbb{P}^M(V_1), \mathbb{P}^M(V_2 \mid V_1)\right), \tag{1.3}$$

which CBN entails the *observational distribution*

$$\mathbb{P}^M(\mathbf{V}) \triangleq \mathbb{P}^M(V_1) \cdot \mathbb{P}^M(V_2 \mid V_1). \tag{1.4}$$

In this framework, actions are modeled as *interventions* (also called *mechanism changes*), which are "partial" or "modular" changes to a CBN.[14] For example, an intervention $I$ on $V_1$ consists of a single *interventional* causal mechanism $\mathbb{P}^I(V_1)$. This intervention is applied to $M$ to create the *interventional* CBN $M_I$,

$$M_I \triangleq \left(\mathbb{P}^I(V_1), \mathbb{P}^M(V_2 \mid V_1)\right)$$

Note that the causal mechanism for $V_2$ is *invariant* between $M$ and $M_I$; the only difference between the two CBNs is the causal mechanism for $V_1$.

According to this definition, each CBN $M$ makes a prediction about the result of an intervention. For example, to predict the distribution of $V_2$ after intervention $I$, compute $\mathbb{P}^{M_I}(\mathbf{V})$ using Equation (1.4), then marginalize out $V_1$ to get $\mathbb{P}^{M_I}(V_2)$. Now, the procedure for extrapolation becomes apparent. Given an observational dataset $\mathbb{D}$ of $(V_1, V_2)$ pairs, we can generate causal predictions as follows (see also Figure 1.6):

**Model selection:** Use $\mathbb{D}$ to pick some CBN $\widehat{M} \in \mathcal{M}_{\text{cbn}}(G_{1\to 2})$.
**Predictive inference:** Given a new intervention $I^*$, predict $\widehat{\mathbb{Q}}(V_2) = q_{I^*}(\widehat{M})$.[15]

[14] Discussions about interventions are often muddled by the existence of competing terminologies. Throughout this chapter, "interventions" are of a general nature, these are often called *soft* or *imperfect* interventions to distinguish them from the more restricted class of *hard* or *perfect* interventions.

[15] For example, first separately estimate $\widehat{\mathbb{P}}(V_1)$, $\widehat{\mathbb{P}}(V_2 \mid V_1)$ using $\mathbb{D}$, and let $\widehat{M}$ be the CBN with these causal mechanisms. Then, return the "plug-in" estimate $q_{I^*}(\widehat{M}) = \mathbb{P}_{\widehat{M}_{I^*}}(V_2)$.

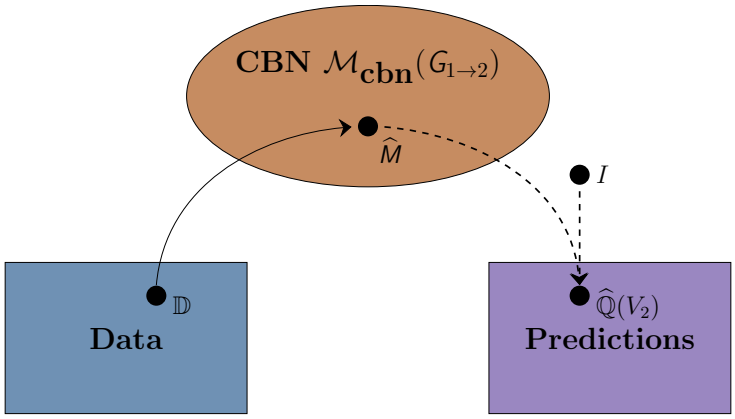



Figure 1.6: **Causal prediction by extrapolation using CBNs.** Given only observational data (i.e., $A =$ no intervention), one can generate causal predictions for other actions (e.g. $A =$ an intervention $I$ on $V_1$) using the model class $\mathcal{M}_{\text{cbn}}(\mathcal{G}_{1\to 2})$ of causal Bayesian networks.

The process for extrapolation is similar for other causal model classes. For example, if the causal graph $\mathcal{G}$ is *unknown*, then observational and *some* interventional data can be used to select both a causal graph and causal Bayesian network, which can then be used to generate causal predictions for interventions that were not seen in the data. However, the available data might not be sufficiently rich to distinguish between two models that generate different predictions; this leads us directly into the next theme of *identifiability*.

## 1.3 Identifiability

*Identifiability* is a central concept in causality research, but its importance is often woefully underappreciated in related subfields of machine learning and statistics. From the perspective of the data-to-decision pipeline, identifiability asks how rich the **Data** needs to be in order to select between **Models**. Here, "rich" does not refer to the *amount* of data, but refers to *how much the data informs model selection*, in a sense that we define in this section. Indeed, in identifiability theory, it is common to work in the *population* (or *infinite-data*) setting, dealing directly with distributions instead of samples.

The goal of this section is to introduce the main ideas of identifiability theory to a more general audience. I am not aware of any standard, widely-used vocabulary for these concepts, especially in statistics and machine learning, though there have been some excellent proposals. In particular, many of the mathematical definitions in this section match those in [MN19], which has heavily inspired this section and other parts of this chapter. By default, I use the same terms and notation as in [MN19], but for some concepts, I use different terms or notation to better serve the demands of this thesis, or to emphasize certain ideas.[16] This section is organized as follows:

- Section 1.3.1 introduces the population-level analog of a dataset, called a *population observable*, defines a *problem setting*, and introduces identifiability.
- Section 1.3.2 introduces the *richness* relation on types of data and the *expressivity* relation on model classes.
- Section 1.3.3 introduces the *difficulty* relation on problem settings and discusses its relation to identifiability.
- Section 1.3.4 describes problems that can arise when models are not identifiable, with two indistinguishable models making different predictions.
- Section 1.3.5 describes *benign* non-identifiability; e.g. identifiability that results from some form of redundancy in a model class or which does not give rise to different predictions.

### 1.3.1 General terminology for identifiability theory

In the population setting, the input to the data-to-decision pipeline comes in the one or more distributions, e.g. as a single $\mathbb{D} = \mathbb{P}(\mathbf{V})$ or as a tuple of distributions $\mathbb{D} = (\mathbb{P}_1(\mathbf{V}), \mathbb{P}_2(\mathbf{V}), \ldots, \mathbb{P}_K(\mathbf{V}))$. To emphasize that we are working in the population setting, we refer to the input $\mathbb{D}$ as a *population observable*. We denote the space of all population observables as $\mathcal{P}$; e.g. if each population observable is a single distribution, i.e., $\mathbb{D} = \mathbb{P}(\mathbf{V})$, then $\mathcal{P}$ is the set of all distributions on $\mathbf{V}$.

We relate a model class $\mathcal{M}$ to the set $\mathcal{P}$ of population observables by the *forward mapping* $\rho : \mathcal{M} \to \mathcal{P}$.[17] For simplicity, we focus on the well-specified setting: we only consider

[16] In particular, our usage of the terms *model* and *forward mapping* is the same, but I use $\mathcal{M}$ to denote a model class instead of $\Theta$, and I use the term *query* instead of *parameter*.

[17] Note that a forward mapping $\rho : \mathcal{M} \to \mathcal{P}$ is just a query with codomain $\mathcal{P}$; we simply use a different term to distinguish it as special.

$\mathbb{D} \in \mathcal{P}$ such that $\mathbb{D} = \rho(M^\star)$ for some $M^\star \in \mathcal{M}$. A model $M$ is *identifiable* from $\rho(M)$ if $\rho^{-1}(\rho(M)) = \{M\}$, where $\rho^{-1}$ denotes the preimage of $\rho$. Intuitively, this condition says that $M$ is the "only explanation" for its associated population observable $\rho(M)$.[18] The entire model class $\mathcal{M}$ is called identifiable under $\rho$ if every $M \in \mathcal{M}$ is identifiable; equivalently, if $\rho$ is injective. It is crucial that this definition depends on both $\mathcal{M}$ and $\rho$; let us call such a pair $(\mathcal{M}, \rho)$ a *problem setting*. As illustrated by the following two examples, the same model class can be identifiable in one problem setting, but not in another.

> ***Example: CBNs are identifiable from their observational distribution.*** *Full knowledge of graphical structure makes a CBN identifiable from its observational distribution. For example, let* $\rho_1 : M \mapsto \mathbb{P}^M(\mathbf{V})$ *for* $M \in \mathcal{M}_{cbn}(G_{1\to 2})$. *Then* $\mathcal{M}$ *is identifiable from* $\rho_1$*; i.e.,* $\rho_1$ *is injective, with left inverse*
>
> $$\rho_1^{-1} : \mathbb{P}(\mathbf{V}) \mapsto (\mathbb{P}(V_1), \mathbb{P}(V_2 \mid V_1))$$

> ***Example: CBNs are not identifiable from marginals.*** *In practice, we may not observe all variables in* $\mathbf{V}$*; i.e., we may only observe* $\mathbb{P}^M(\mathbf{V}')$ *for some* $\mathbf{V}' \subseteq \mathbf{V}$. *For example, let* $\rho_2 : M \mapsto \mathbb{P}^M(V_2)$ *for* $M \in \mathcal{M}_{cbn}(G_{1\to 2})$. *Then* $\mathcal{M}$ *is not identifiable from* $\rho_2$*; indeed, knowing* $\mathbb{P}^M(V_2)$ *places no constraints on* $\mathbb{P}^M(V_1)$.

The interplay between model classes and forward mappings is of central importance in identifiability theory. However, it can often be difficult to pinpoint subtle differences between problem settings. For the uninitiated, these subtleties can make it difficult to compare results, or to understand which settings best match a particular application. Towards providing some clarity, we now introduce two relations that let us compare problem settings.

### 1.3.2 The richness and expressivity relations

Given two forward mappings $\rho_1 : \mathcal{M} \mapsto \mathcal{P}_1$ and $\rho_2 : \mathcal{M} \mapsto \mathcal{P}_2$, we will say that $\rho_1$ is *richer* than $\rho_2$ if there exists a function $g$ such that $\rho_2 = g \circ \rho_1$; we write this as $\rho_1 \succeq_r \rho_2$. In this case, I will also say that $\mathcal{P}_1$ is richer than $\mathcal{P}_2$, and that $\mathbb{D}_1 = \rho_1(M)$ is richer than $\mathbb{D}_2 = \rho_2(M)$. With some abuse of terminology, I will often compare the richness of population observables $\mathbb{D}_1$ and $\mathbb{D}_2$ without specifying maps $\rho_1$ and $\rho_2$, which can usually be inferred from context.

Let us illustrate this definition through two common examples. The first is related to the use of interventions in causal structure learning, and the second is related to the difference between interpolation and extrapolation depicted in Figure 1.5. In both examples, $\mathcal{P}_1$ and $\mathcal{P}_2$ denote the images of the forward mappings $\rho_1$ and $\rho_2$, respectively.

> ***Example: Observational and interventional data.*** *Consider the model class* $\mathcal{M}_{cbn}(G_{1\to 2})$, *and let* $I$ *be an intervention on* $V_1$. *Let* $\rho_1$ *be a forward mapping*

[18] As pointed out by [MN19] and [ES02], identification problems and inverse problems are essentially equivalent: assuming that $\mathcal{M}$ is well-specified is equivalent to assuming *existence* of a solution to the inverse problem, and identifiability is equivalent to *uniqueness* of the solution. Identifiability theory is mostly distinguished from inverse problems theory by the kinds of model class and forward mappings that are considered.

*that tells us both the observational and interventional distributions of $M$, and let $\rho_2$ be a forward mapping that tells us only the observational distribution, i.e.,*

$$\rho_1 : M \mapsto \left(\mathbb{P}^M(\mathbf{V}), \mathbb{P}^{M_I}(\mathbf{V})\right), \text{ and}$$
$$\rho_2 : M \mapsto \mathbb{P}^M(\mathbf{V}).$$

*Then $\rho_1$ is richer than $\rho_2$, since the map $g : \mathcal{P}_1 \to \mathcal{P}_2$ that forgets the second element of $\mathbb{D}_1 \in \mathcal{P}_1$ is surjective.*

***Example: Changing support.*** *Suppose we have some model class $\mathcal{M}$ such that each $M \in \mathcal{M}$ can be associated with a distribution $\mathbb{P}^M(A, C, Y)$ with support contained in $\mathcal{A} \times \mathcal{C} \times \mathcal{Y}$. Consider some $\mathcal{A}' \subseteq \mathcal{A}$ and $\mathcal{C}' \subseteq \mathcal{C}$, and let*

$$\rho_1 : M \mapsto \mathbb{P}^M(A, C, Y), \text{ and}$$
$$\rho_2 : M \mapsto \mathbb{P}^M(A, C, Y \mid A \in \mathcal{A}', C \in \mathcal{C}'),$$

*so that the support of $\rho_2(M)$ is contained in $\mathcal{A}' \times \mathcal{C}' \times \mathcal{Y}$. Then $\rho_1$ is richer than $\rho_2$, since the map $g : \mathcal{P}_1 \to \mathcal{P}_2$ which implements conditioning is surjective.*

We can define an analogous relation over model classes. Given two model classes $\mathcal{M}_1$ and $\mathcal{M}_2$, we say that $\mathcal{M}_1$ is *more expressive* than $\mathcal{M}_2$, written $\mathcal{M}_1 \succeq_e \mathcal{M}_2$, if there exists a surjective map $\sigma : \mathcal{M}_1 \to \mathcal{M}_2$, i.e., for every model in $M \in \mathcal{M}_1$, we can derive an associated model $\sigma(M) \in \mathcal{M}_2$, and *all* models in $\mathcal{M}_2$ can be derived this way. The notion of *expressivity* can capture both quantitative and qualitative differences between model classes. In particular, for two model classes $\mathcal{M}_1$ and $\mathcal{M}_2$ that are composed of same types of models, $\mathcal{M}_1$ might be more expressive than $\mathcal{M}_2$ simply because it makes fewer modeling assumptions; in this case, $\sigma$ has an almost trivial definition, as follows.

***Example: Stronger modeling assumptions.*** *Take any model class $\mathcal{M}_1$ and suppose we create a new model class $\mathcal{M}_2 \subseteq \mathcal{M}_1$ by imposing stronger modeling assumptions, e.g., assuming all distributions are Gaussian. Pick any model $M_* \in \mathcal{M}_2$. We can define the surjective map $\sigma : \mathcal{M}_1 \to \mathcal{M}_2$ with $\sigma(M) = M$ for all $M \in \mathcal{M}_2$, and $\sigma(M) = M_*$ for $M \notin \mathcal{M}_2$; thus, $\mathcal{M}_1$ is more expressive than $\mathcal{M}_2$.*

A more interesting case is when $\mathcal{M}_1$ and $\mathcal{M}_2$ are composed of different types of models, i.e., $\mathcal{M}_1 \cap \mathcal{M}_2 = \varnothing$, so that the map $\sigma$ describes a way to change $\mathcal{M}_1$-type models into $\mathcal{M}_2$-type models. In this case, any predictions that can be made by type-$\mathcal{M}_2$ models can also be made by type-$\mathcal{M}_1$ models (by first using $\sigma$ to change the model type). However, the reverse need not be true: type-$\mathcal{M}_1$ models might be capable of making predictions that cannot be defined in terms of a type-$\mathcal{M}_2$ model.[19]

In causality, this gap appears as the difference between models (like CBNs) that can only make *observational* and *interventional* predictions, and models that can make *counterfactual*

[19]With this framework and the definition of *query* (introduced below), we could provide a formal definition of the concept of *predictive alignment* from Section 1.2.1, but we will not do so here.

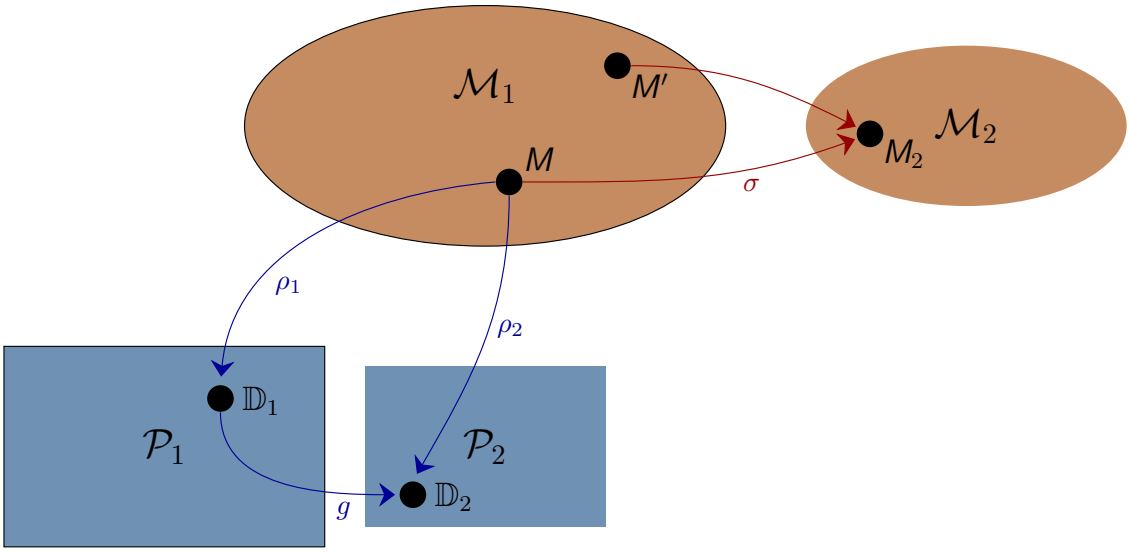



Figure 1.7: **The richness and expressivity relations.** *(In blue)* If $\rho_1 : \mathcal{M}_1 \to \mathcal{P}_1$ is richer than $\rho_2 : \mathcal{M}_1 \to \mathcal{P}_2$, there is some $g : \mathcal{P}_1 \to \mathcal{P}_2$ such that any $\mathbb{D}_2 \in \mathcal{P}_2$ can be derived as $\mathbb{D}_2 = g(\mathbb{D}_1)$ for some $\mathbb{D}_1 \in \mathcal{P}_2$. *(In red)* If $\mathcal{M}_1$ is more expressive than $\mathcal{M}_2$, there is some $\sigma : \mathcal{M}_1 \to \mathcal{M}_2$ such that any $M_2 \in \mathcal{M}_2$ can be derived as $M_2 = \sigma(\mathbf{M}_1)$ for at least one model $M_1 \in \mathcal{M}_1$, and from possibly several models.

predictions, as follows. In the graphical tradition, *structural causal models (SCMs)* are commonly used to define counterfactuals, and they are strictly more expressive than CBNs, as we now describe. We do not need counterfactual predictions or SCMs for this thesis, so we will go into less technical detail for this example.

> ***Example: Deriving CBNs from SCMs.*** *Recall from Equation (1.3) that a CBN $M_2 \in \mathcal{M}_{cbn}(\mathcal{G}_{1\to 2})$ is comprised of one object per variable. Meanwhile, a* structural causal model (SCM) *$M_1 \in \mathcal{M}_{scm}(\mathcal{G}_{1\to 2})$ is comprised of two objects per variable:*
>
> $$M_1 \triangleq \left(\mathbb{P}^{M_1}(\varepsilon_1), \mathbb{P}^{M_1}(\varepsilon_2), f_1^{M_1}, f_2^{M_1}\right)$$
>
> *where $\mathbb{P}^{M_1}(\varepsilon_1)$ and $\mathbb{P}^{M_1}(\varepsilon_2)$ are univariate distributions, and $f^{M_1} : \mathbb{R} \to \mathbb{R}$ and $f^{M_2} : \mathbb{R}^2 \to \mathbb{R}$ are functions. From an SCM $M_1$, we can derive a CBN $M_2 = \sigma(M_1)$ by separately deriving a causal mechanism for each node, i.e. via maps*
>
> $$\sigma_1 : \left(\mathbb{P}^{M_1}(\varepsilon_1), f_1^{M_1}\right) \mapsto \mathbb{P}^{M_2}(V_1), \text{ and}$$
> $$\sigma_2 : \left(\mathbb{P}^{M_2}(\varepsilon_1), f_1^{M_2}\right) \mapsto \mathbb{P}^{M_2}(V_2 \mid V_1),$$
>
> *such that the map $\sigma$ is surjective. In particular, $\sigma$ will generally not be injective; thus, two SCMs can make all of the same interventional predictions, but different counterfactual predictions.*[20]

[20]The avid reader may be interested in how to show that $\sigma$ is surjective. To get some sense, it is more convenient to work with $\mathbb{P}^{M_1}(\boldsymbol{\varepsilon}) = \mathbb{P}^{M_1}(\varepsilon_1) \times \mathbb{P}^{M_1}(\varepsilon_2)$ and a function $\mathbf{f}^{M_1} : \mathbb{R}^2 \to \mathbb{R}^2$ derived from $(f_1^{M_1}, f_2^{M_1})$. Then, one can define $M_2 = \sigma(M_1)$ by decomposing the induced distribution of $\mathbf{V} = \mathbf{f}^{M_1}(\boldsymbol{\varepsilon})$. Surjectivity of $\sigma$ follows from the Rosenblatt transformation, which maps any $\mathbb{P}(\mathbf{V})$ to a distribution $\mathbb{P}(\boldsymbol{\varepsilon})$ over independent variables $\varepsilon$ [Ros52], and non-injectivity follows from the results in independent component analysis [HP99].

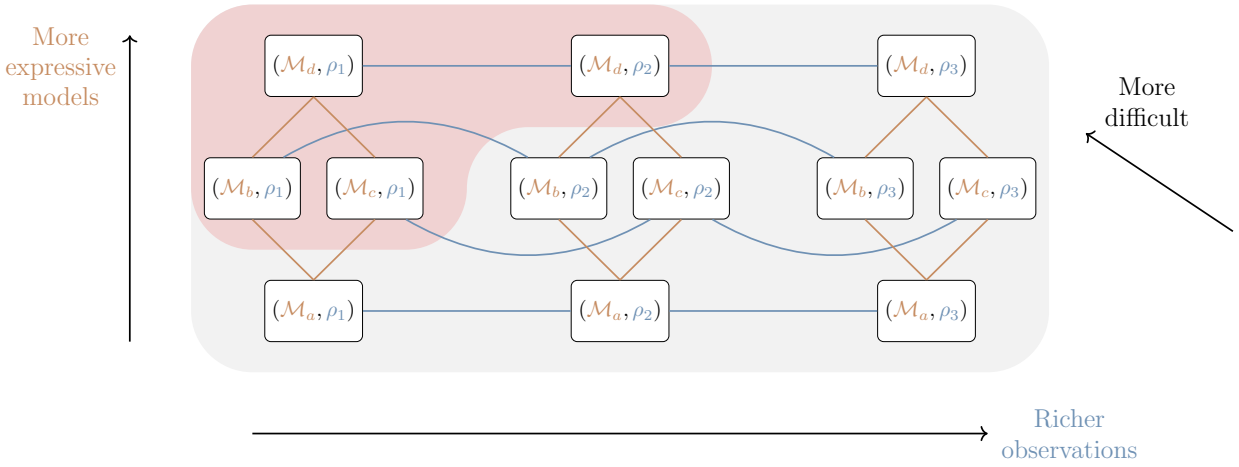



Figure 1.8: **Identifiability and the difficulty relation.** Let $\mathcal{M}_a, \mathcal{M}_b$, $\mathcal{M}_c$, and $\mathcal{M}_d$ be model classes such that $\mathcal{M}_d \succeq_e \mathcal{M}_c$, $\mathcal{M}_d \succeq_e \mathcal{M}_b$, $\mathcal{M}_c \succeq_e \mathcal{M}_a$, and $\mathcal{M}_b \succeq_e \mathcal{M}_a$. Similarly, let $\rho_1, \rho_2$, and $\rho_3$ be forward mappings such that $\rho_3 \succeq_r \rho_2 \succeq_r \rho_1$. Consider two problem settings $S = (\mathcal{M}, \rho)$ and $S' = (\mathcal{M}', \rho')$. In the above diagram, there is a path of blue edges going right from $S$ to $S'$ if $\mathcal{M} = \mathcal{M}'$, and $\rho'$ is richer than $\rho$. Similarly, there is a path of orange edges going upwards from $S$ to $S'$ if $\rho = \rho'$ and $\mathcal{M}'$ is more expressive than $\mathcal{M}$. Composing these two relations, there is a path that goes *left* and up from $S$ to $S'$, whenever $S'$ is a more difficult problem setting than $S'$. If a certain problem setting is non-identifiable (shown in red), then more difficult problem settings are also non-identifiable.

### 1.3.3 The difficulty relation on problem settings

First, note that the richness and expressivity relations are only *partial orders* on forward mappings and model classes, respectively. For example, two model classes $\mathcal{M}_1$ and $\mathcal{M}_2$ might be incomparable, with neither being more expressive than the other. Thus, one should generally avoid referring to the "expressivity" of a model class or the "richness" of a forward mapping. We do not define some absolute scalar measure of such properties; rather, the two properties are only defined in a *relative* sense.

Similarly, we can define a partial order over problem settings: we say that $(\mathcal{M}, \rho)$ is *more difficult* than $(\mathcal{M}', \rho')$ if $\mathcal{M}$ is more expressive than $\mathcal{M}'$, and $\rho$ is less rich than $\rho'$, i.e., if $\mathcal{M} \succeq_e \mathcal{M}'$ and $\rho \preceq_r \rho'$. In Figure 1.8, we illustrate the difficulty relation with a diagram, showing that it captures a core principle in identifiability theory:

*More expressive model classes require richer observations to ensure identifiability.*

The tradeoff between expressivity and data richness mirrors one that is well-studied in statistics and learning theory, where more complex model classes (e.g. measured via Rademacher complexity) require a larger datasets to ensure accurate recovery. However, expressivity and data richness are not the same as complexity and sample size: the former terms are of a more discrete nature compared to the latter, which are more quantitative.

### 1.3.4 Problematic non-identifiability

Using the language established in this section, we can see where non-identifiability arises in practice, and why non-identifiability can be problematic in statistical analysis. Consider the simplest case of non-identifiability: a model class $\mathcal{M}$, a forward mapping $\rho$, and a population observable $\mathbb{D}$ such two models "fit the data" equally well, i.e., $\rho^{-1}(\mathbb{D}) = \{\widehat{M}_1, \widehat{M}_2\}$. For now, let us assume there is no way to "break ties" between $\widehat{M}_1$ and $\widehat{M}_2$ and that they make different predictions $\widehat{P}_1$ and $\widehat{P}_2$, as depicted in Figure 1.9.

Then the statistical analyst has a choice at the model selection step: they can break the tie arbitrary, or they can "select" both models, performing the remainder of their analysis in parallel. In practice, the analyst might not even be aware that their model is not identified, with their model selection algorithm secretly breaking the tie for them (often based on randomness; e.g. the random seed in deep learning or in Bayesian sampling methods that encounter the *label switching* problem [Ste00]). This obliviousness to non-identifiability sheds light on many recent issues in deep learning: for instance, the existence of *adversarial examples* in computer vision is indicative of the fact that (image, label) pairs are not rich enough to pin down a single "correct" model.

Assuming that the analyst *is* aware of the non-identifiability of their model, both options can still be problematic. If the analyst arbitrarily breaks the tie, e.g. generating the prediction $\widehat{P}_1$, then the downstream decision-maker might make a different choice than if the analyst had arbitrarily generated the prediction $\widehat{P}_2$. If the analyst informs the downstream decision-maker of both predictions, then they have just kicked the can down the road; the decision-maker still needs to "break the tie". Sometimes, there is no way around these issues: in causality, *partial identification* is accepted as a common state of affairs [GP21; GST23]; at least the pair of predictions $\{\widehat{P}_1, \widehat{P}_2\}$ is more informative than no prediction at all.

### 1.3.5 Benign non-identifiability

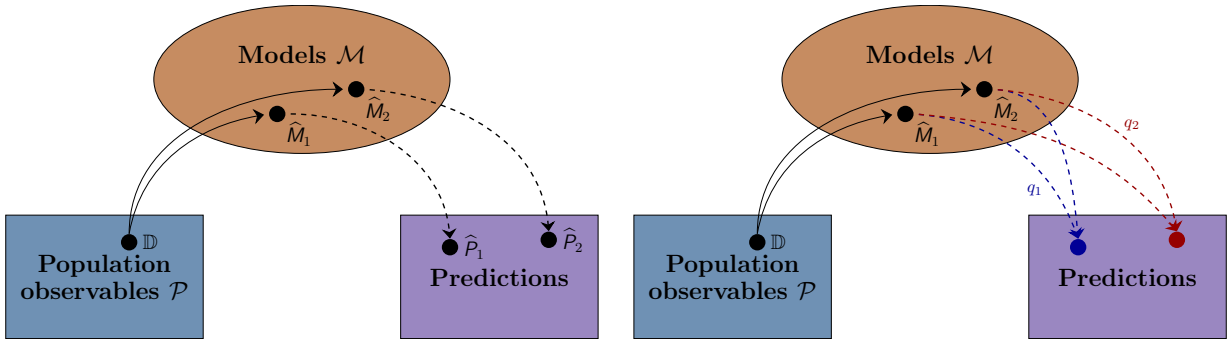
More optimistically, the non-identifiability might be "benign" in one of at least two ways: there might be *non-arbitrary* ways to break ties between some pairs of models, or two indistinguishable models might make all of the same predictions. Non-arbitrary tie-breaking is common in causality, and statistics more broadly.

**Tie-breaking and Occam's razor** Often, "benign" ties occur because of some redundancy in the definition of a model class. Although it appears somewhat odd, it is often statistically or computationally convenient to define $\mathcal{M}$ as the *disjoint union* of $\mathcal{M}_\alpha$ and $\mathcal{M}_\beta$, i.e.,

$$\mathcal{M} = \{(\alpha, M) : M \in \mathcal{M}_\alpha\} \cup \{(\beta, M) : M \in \mathcal{M}_\beta\}.$$

Then, each $M \in \mathcal{M}_\alpha$ appears twice in $\mathcal{M}$: once as $M_\alpha := (\alpha, M)$, and once as $M_\beta := (\beta, M)$. A natural way to break a tie between $M_\alpha$ and $M_\beta$ is based on Occam's razor: $\mathcal{M}_\alpha$ is the simpler model class, so we pick $M_\alpha$ instead of $M_\beta$. Equivalently, since $M_\beta$ will never be picked, we can change the definition $\mathcal{M}$ so that it only includes $M_\alpha$, and not $M_\beta$; this latter option is more standard in causality.[21]

[21] In particular, the *minimality* assumption has this flavor [Zha13]. See [Lam23] for a recent review of related assumptions such as *faithfulness*.




(a) **Problematic non-identifiability** (b) **Benign non-identifiability**

Figure 1.9: **Two kinds of non-identifiability.** Non-identifiability of a model class may be problematic if the data does not distinguish between two models that make different predictions, but is considered benign if all models that fit the data make the same predictions.

**Predictive equivalence** A more general form of redundancy appears when we bring predictions back into the picture. In particular, non-identifiability of two *models* need not be a problem if they make the same *predictions* in all conceivable settings — one only needs to worry about "the differences that make a difference".

To formalize what is meant by "all conceivable settings", we return to our definition of a *query* as a function $q_\gamma : \mathcal{M} \to \mathcal{V}_\gamma$, where $\mathcal{V}_\gamma$ is an arbitrary set; as described in Section 1.2.1, the predictions of a model are just its outputs on some set $\mathcal{Q}$ of queries. Thus, for a set $\mathcal{Q}$ representing "all possible predictions we might make with our model", we can define an associated equivalence relation over the model class: two models $M_1$ and $M_2$ are *predictively equivalent* with respect to $\mathcal{Q}$, written as $M_1 \approx_{\mathcal{Q}} M_2$, if $q_\gamma(M_1) = q_\gamma(M_2)$ for all $q_\gamma \in \mathcal{Q}$. For example, the two models in Figure 1.9 are predictively equivalent for $\mathcal{Q} = \{q_1, q_2\}$.

**Predictive identifiability** Mathematically, there are at least two ways to use the relation $\approx_{\mathcal{Q}}$ to define a notion of *predictive identifiability* (with respect to $\mathcal{Q}$). The first option is to keep the model class $\mathcal{M}$ fixed, and say that predictive identifiability holds for the population observable $\mathbb{D}$ if all $M \in \rho^{-1}(\mathbb{D})$ are predictively equivalent. The second option is to define a new model class $\mathcal{M}' = \mathcal{M}/\approx_{\mathcal{Q}}$ where we equate predictively equivalent models, and say that predictive identifiability for $\mathbb{D}$ if model identifiability holds for $\mathcal{M}'$.[22]

Philosophically (at least, for those leaning towards pragmatism), the latter "quotienting" option appears attractive. Ontologically, one might say that if two models make exactly the same predictions in every conceivable setting, then they really *are* the same at some fundamental level. More humbly, from the epistemological point of view, we might as well say that two such models are the same, since by supposition there is no way that one could ever distinguish between the two.

The quotienting option also nicely mirrors some of our earlier discussions. This option is analogous to how the AM approach uses a coordinate-free description for a function class: in $\mathcal{M}'$, a model is defined by *what* it predicts, rather than *how* it generate its predictions; we can also think of quotienting as a way to get rid of redundancies in the model class, as we

[22] In particular, each model $M \in \mathcal{M}'$ is an *equivalence class* of predictively equivalent models in $\mathcal{M}$.

did for nested model classes. However, it is worth noting the quotienting option might be harder to implement computationally, so we do not take a stance on which option is better.[23]

[23]For example, it seems that redundancy can be quite beneficial for gradient-based optimization approaches, which may partially account for the greater attention paid to equivariant neural networks [Ger+23] over invariant neural networks in recent years.

## 1.4 Technical background

In this section, we review foundational definitions and concepts from two key areas that we draw from repeatedly in the rest of this thesis. First, we define the model class consisting of *causal Bayesian networks*, upon which most of the model classes used in this thesis are based. Then, we review some basic biological background that motivates the problem settings that we study in this thesis. In slightly more detail,

- Section 1.4.1 defines causal Bayesian networks and interventions on a known graph.
- Section 1.4.2 defines linear causal Bayesian networks and the class of all causal Bayesian networks on random variables $\mathbf{V}$ (where the graph is unknown).
- Section 1.4.3 covers relevant background from molecular/cellular biology.

Although admittedly somewhat tedious, for the sake of completeness, we will explicitly write an equation to define every model class. However, we will only *number* equations that are built on in the next section.

**Notation and conventions** To aid the reader, we color a model $M$ and a model class $\mathcal{M}$ in orange when it is defined, and we color population observables in blue. All of our model class definitions will actually define an entire *collection* of model classes $\{\mathcal{M}_\alpha\}_{\alpha\in\mathcal{J}}$ for some index set $\mathcal{J}$; typically, $\mathcal{J}$ will be the positive integers or a set of graphs. Often, we will use this collection to construct a more expressive model class $\mathcal{M}$, where we treat the index $\alpha$ as an unknown parameter that must also be selected. For this purpose, it will be convenient to define $\mathcal{M}$ using the *disjoint union* over $\{\mathcal{M}_\alpha\}_{\alpha\in\mathcal{J}}$, defined as

$$\bigsqcup_{\alpha\in\mathcal{J}} \mathcal{M}_\alpha = \{(\alpha, M) : \alpha \in \mathcal{J},\ M \in \mathcal{M}_\alpha\},$$

i.e., the disjoint union keeps track of the indices as a part of the new model class. Generally, we use $\mathbf{V}$ to denote the set of *all* random variables defined by a model; if some variables in $\mathbf{V}$ are unobserved, then we use $\mathbf{X} \subseteq \mathbf{V}$ to denote the observed variables.

### 1.4.1 Causal Bayesian networks and interventions on a graph

We start by defining our most fundamental model class, the class of causal Bayesian networks for a specific directed graph $G$ over random variables $\mathbf{V}$.[24] Intuitively, an edge $W \to V$ in $G$ indicates that $W$ is (potentially) a direct cause of $V$, and a directed path from $W$ to $V$ indicates that $W$ is (potentially) a cause of $V$, either direct or indirect. We use $\mathrm{pa}_G(V) = \{W \in \mathbf{V} : W \to V \text{ in } G\}$ to denote the *parents* of $V$ in $G$.

We assume that $G$ is *directed acyclic graph (DAG)*, i.e., that no node has a directed path to itself, avoiding the possibility that a variable "causes itself". Given a DAG $G$, a *causal Bayesian network (CBN)* $M$ on $G$ consists of one *causal mechanism* for each variable, i.e.,

$$M \triangleq \left\{\mathbb{P}^M\left(V \mid \mathrm{pa}_G(V)\right)\right\}_{V\in\mathbf{V}}, \tag{1.5}$$

[24] Here, we equate the nodes of $G$ with the random variables $\mathbf{V}$. An alternative convention is to distinguish between nodes and random variables (e.g. using integers $i$ to denote nodes in $G$, and using $V_i$ to denote their associated random variables). In later chapters, we use whichever convention is more convenient.

and the model class of all CBNs on $G$ is

$$\mathcal{M}_{\text{cbn}}(G) \triangleq \{M \text{ is of the form in Equation (1.5)}\}$$

Each CBN $M \in \mathcal{M}_{\text{cbn}}(G)$ entails the *observational distribution*

$$\mathbb{P}^M(\mathbf{V}) \triangleq \prod_{V \in \mathbf{V}} \mathbb{P}^M\left(V \mid \text{pa}_G(V)\right).$$

**Interventions** In this causal framework, actions are modeled as interventions. Given a DAG $G$, an *intervention* $I$ on $G$ is a set of *interventional* causal mechanisms, i.e.,

$$I \triangleq \left\{\mathbb{P}^I\left(V \mid \text{pa}_G(V)\right)\right\}_{V \in \mathbf{T}(I)} \tag{1.6}$$

for some $\mathbf{T}(I) \subseteq \mathbf{V}$ called the intervention *targets*. The class of all interventions on $G$ is

$$\mathcal{I}(G) \triangleq \{I \text{ is of the form in Equation (1.6)}\}.$$

Given $M \in \mathcal{M}_{\text{cbn}}(G)$ and $I \in \mathcal{I}(G)$, we define the *interventional CBN* $M_I$ as

$$M_I \triangleq \left\{\mathbb{P}^{M_I}\left(V \mid \text{pa}_G(V)\right)\right\}_{V \in \mathbf{V}},$$

where the causal mechanism of $V$ is invariant if $V$ is not an intervention target, i.e.,

$$\mathbb{P}^{M_I}\left(V \mid \text{pa}_G(V)\right) = \begin{cases} \mathbb{P}^M\left(V \mid \text{pa}_G(V)\right), & \text{if } V \notin \mathbf{T}(I) \\ \mathbb{P}^I\left(V \mid \text{pa}_G(V)\right), & \text{if } V \in \mathbf{T}(I) \end{cases}.$$

We say that $\mathbb{P}^{M_I}(\mathbf{V})$ is the interventional distribution of $M$ under intervention $I$. Often, one may observe data from a CBN under some intervention $I$, or under multiple interventions $I_1, I_2, \ldots, I_K$, but lack complete knowledge of these interventions. If one knows nothing about the interventions, this lack of knowledge can be formalized by working with the model class

$$\mathcal{M}_{\text{cbn-iv}}(G, K) \triangleq \mathcal{M}_{\text{cbn}}(G) \times \underbrace{\mathcal{I}(G) \times \ldots \times \mathcal{I}(G)}_{K \text{ times}},$$

i.e., the interventions are incorporated as components of the model that is to be learned. Often, one assumes that we know something about each intervention $I_k$; e.g. its intervention targets $\mathbf{T}(I_k)$, or the number of interventions targets. Two particularly well-studied classes of interventions are the *perfect* (or *hard*) interventions, which render their intervention targets independent of their parents, i.e.,

$$\mathcal{I}_{\text{perf}}(G) \triangleq \left\{I \in \mathcal{I}(G) : \mathbb{P}^I\left(V \mid \text{pa}_G(V)\right) = \mathbb{P}^I(V) \;\; \forall\, V \in \mathbf{T}(I)\right\} \tag{1.7}$$

and the *do* interventions, which deterministically set their intervention targets to some fixed values, i.e.,

$$\mathcal{I}_{\text{do}}(G) \triangleq \left\{I \in \mathcal{I}(G) : \mathbb{P}^I\left(V \mid \text{pa}_G(V)\right) = \delta_{t_V^I}(V) \;\forall\, V \in \mathbf{T}(I)\right\}, \tag{1.8}$$

where $\delta_t$ denote the *delta* distribution at location $t$.

### 1.4.2 Linearity and unknown graphs

We can use the model classes already defined to build new model classes that are more or less expressive. We consider two such classes here: the class of linear causal Bayesian networks, and the class of all causal Bayesian networks on a set of random variables $\mathbf{V}$.

**Linear causal Bayesian networks** The definition of $M \in \mathcal{M}_{\text{cbn}}(G)$ is nonparametric; i.e., $\mathbb{P}^G\left(V \mid \text{pa}_G(V)\right)$ is allowed to be *any* conditional distribution.[25] However, many real-world processes are well-modeled by parametric families of distributions, and theoretical results for parametric settings can often pave the way for results in nonparametric settings. Thus, it is common to consider subsets of $\mathcal{M}_{\text{cbn}}(G)$ that obey parametric assumptions, with linearity being the simplest such assumption. In particular, we say that $\mathbb{P}^M\left(V \mid \text{pa}_G(V)\right)$ is *linear* if it can be expressed as the distribution induced by the equation

$$V = \sum\nolimits_{W \in \text{pa}_G(V)} \beta_{WV} \cdot W + \varepsilon_V$$

for some coefficients $\beta_{WV}$ and a random variable $\varepsilon_V$ that is independent of all $W \in \text{pa}_G(V)$. We can capture the linearity assumption by defining the new model class

$$\mathcal{M}_{\text{cbn-lin}}(G) \triangleq \left\{M \in \mathcal{M}_{\text{cbn}}(G) : \mathbb{P}^M\left(V \mid \text{pa}_G(V)\right) \text{ is linear for all } V \in \mathbf{V}\right\};$$

we do not define the corresponding class for interventions since we do not use it in this thesis.

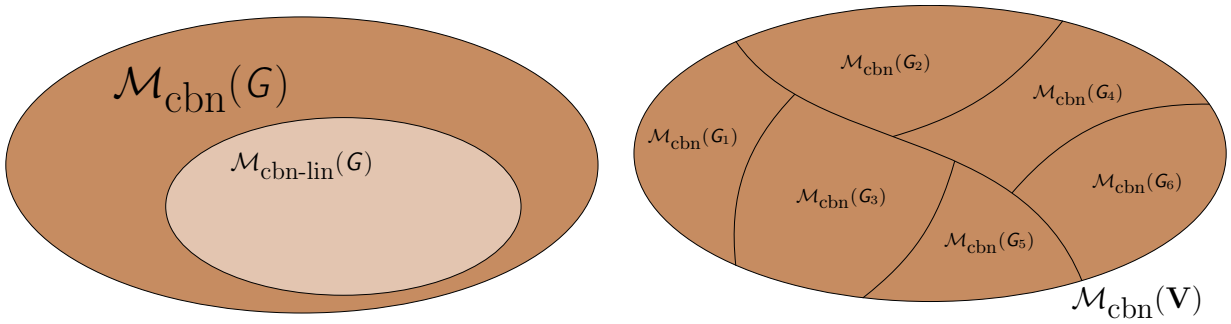
**Causal Bayesian networks on an unknown graph** Thus far, the model classes we have considered were all defined in terms of some DAG $G$ that encodes qualitative assumptions about which variables are direct causes of other variables. However, in many settings, we do not possess sufficient background knowledge to warrant such assumptions; this roughly distinguishes the community that focuses on *causal effect estimation* (where $G$ is assumed) from the community that focuses on *causal structure learning* (where $G$ is not assumed).

We can express the fact that the graph $G$ is unknown by taking a disjoint union over the set of all DAGs on $\mathbf{V}$, denoted by $\mathfrak{G}(\mathbf{V})$. In particular, we define the non-parametric classes of causal Bayesian networks, interventions, and combinations of these:

$$\begin{aligned}
\mathcal{M}_{\text{cbn}}(\mathbf{V}) &\triangleq \bigsqcup_{G \in \mathfrak{G}(\mathbf{V})} \mathcal{M}_{\text{cbn}}(G), \\
\mathcal{I}(\mathbf{V}) &\triangleq \bigsqcup_{G \in \mathfrak{G}(\mathbf{V})} \mathcal{I}(G), \text{ and} \\
\mathcal{M}_{\text{cbn-iv}}(\mathbf{V}, K) &\triangleq \bigsqcup_{G \in \mathfrak{G}(\mathbf{V})} \mathcal{M}_{\text{cbn-iv}}(G, K).
\end{aligned} \tag{1.9}$$

Note that each element $M \in \mathcal{M}_{\text{cbn}}(\mathbf{V})$ is a tuple $(G^M, B^M)$, with $G^M \in \mathfrak{G}(\mathbf{V})$ and $B^M \in \mathcal{M}_{\text{cbn}}(G)$. A convenient consequence of using the disjoint union is that we can easily refer to the *causal graph* $G^M$ of a causal Bayesian network $M$; as we will see, this is quite useful for adding graphical assumptions to this model class.

[25] In statistics, we would actually say that the definition is *semiparametric*, since there are restrictions on the joint distribution $\mathbb{P}^M(\mathbf{V})$, but they are not of a parametric (i.e., finite-dimensional) form.




(a) **Containment of model classes** (b) **Disjoint union of model class**

Figure 1.10: **Constructing new model classes.** Given a model class $\mathcal{M}_1$, we can form a new model class $\mathcal{M}_2 \subseteq \mathcal{M}_1$ by adding assumptions; we can also form a new model class by taking the disjoint union of several models class $\{\mathcal{M}_\alpha\}_{\alpha \in \mathcal{J}}$.

### 1.4.3 Biological background

Many of the model classes in this thesis are used or designed with specific biological problems in mind. In particular, this thesis largely focuses on the causal question

*What will happen to the state $Y$ of a cell if I perform perturbation $A$ in context $C$?*

Thus, the way that we use and design our model classes hinges on how we measure *cell state*, how we model *perturbations* and *contexts*, and how we use biological domain knowledge to relate $Y$ to $A$ and $C$. In this section, we give some general background on these topics; later sections include additional background as needed.

**Cell state and gene expression** A single cell is a very complex system, simultaneously carrying out a huge number of processes (e.g. metabolism, protein synthesis, DNA repair) while responding to signals from its environment and communicating with other cells. At a very detailed level, one might give a reductive definition of cell state as the state of every single molecule or atom in the cell. Today's technology is far from enabling us to measure the state of a cell at such a microscopic level, and fundamental physical limitations may prevent us from ever doing so. Instead, our technologies measure one or more aspects of this state; here, we focus on the cell's *gene expression profile*.

In a multicellular organism like ourself, every cell has more or less the same DNA, the same "source code" that they read so that they can do their jobs. Despite this commonality, there are several different cell *types* in a single organism, e.g. skin cells, lung cells, and neurons.[26] To generate such a wide variety of cell types from the same source code, different cells read from different sections of the DNA, which are called *genes*. For the sake of our discussion, we focus on *protein-coding* genes, i.e., those that are *transcribed* into molecules called *messenger RNA (mRNA)*, which can eventually be *translated* into proteins.

Transcription and translation are such important process that the statement "DNA makes RNA, and RNA makes protein" is often considered to be the *central dogma* of molecular

[26]Note that cell type is a coarser notion than cell state: two cells of the same cell type can have different cell states, e.g. a healthy skin cell and a cancerous one.

biology.[27] When a gene is being transcribed in a cell, we say that it is being *expressed*; thus, a cell's *gene expression profile* summarizes how much it is expressing each (protein-coding) gene. A gene expression profile is also often referred to as a *transcriptome*, and methods for measuring gene expression are called *transcriptomic* technologies. In this thesis, we consider gene expression measurements produced by two kinds of transcriptomic technologies: *bulk RNA sequencing*, which measures the average gene expression in a group of cells, and *single-cell RNA-sequencing (scRNA-seq)*, which measures gene expression at a single-cell resolution.

**Gene regulation and perturbations** Since gene expression plays such an important role in determining cell state, it is essential to understand a bit about the process by which a cell controls its gene expression, which we call *gene regulation*. The process of transcription is carried out by a protein called *RNA polymerase*, which comes into physical contact with the segments of DNA that we call genes and transcribes them into RNA. Thus, cells can control gene regulation by controlling how often genes come into contact with RNA polymerase.

To do so, cells use another class of proteins called *transcription factors (TFs)*; specific TFs either increase or decrease gene expression by binding to the DNA and affecting where RNA polymerase tends to go. Now, we can see something interesting: since TFs are proteins, the abundance of a specific TF depends on the gene expression level of its corresponding gene, call this gene $A$.[28] Thus, the expression level of gene $A$ (at some point in time) is one of the factors determining the expression level of some other gene $B$ (at some later point in time); in this case we say that gene $A$ *regulates* gene $B$.

To empirically study gene regulation, it is quite useful to perturb a cell and observe how its gene expression changes. In this thesis, we consider two types of perturbations. We use the term *genetic perturbation* to refer to a perturbation that directly affects the expression of a gene; e.g. a perturbation that *knocks out* a gene (removes it from the gene) using the CRISPR gene-editing technology. Although gene editing is quite useful as a scientific technique, it is not currently used as a common means of treating diseases, so we also consider *molecular perturbations*, which involve exposing the cell to some molecule, e.g. an FDA-approved drug like insulin.

**Modeling gene regulation and perturbations** Let us return to the causal question

*What will happen to the state $\mathsf{Y}$ of a cell if I perform perturbation $\mathsf{A}$ in context $\mathsf{C}$?*

As discussed, technological limitations limit how much we can measure about cell state; in this thesis, we describe a cell's state by its *gene expression profile*. To model stochasticity in gene expressions, we consider outcomes of the form $\mathsf{Y} = \mathbb{Q}(\mathbf{V})$, where $\mathbf{V}$ is a random vector representing a cell's gene expression, so that each $V \in \mathbf{V}$ corresponds to a gene.

We model the regulatory relationships with a causal Bayesian network $\mathsf{M} \in \mathcal{M}_{\text{cbn}}(\mathbf{V})$, where intuitively the edge $V \to W$ in $\mathsf{G}^{\mathsf{M}}$ indicates that gene $V$ regulates gene $W$. To model a genetic perturbation $\mathsf{A}$, we use an intervention $\mathsf{I} \in \mathcal{I}(\mathbf{V})$, where intuitively the intervention

[27] The original usage of the term "central dogma" makes the stronger statement that information *only* flows from DNA to RNA to proteins, and not in reverse.

[28] We are careful here to say *depends on*, rather than "is determined by". Gene expression is one of several factors influencing protein expression; the full process is quite complex and not essential to our discussion, though it is good to keep this in mind.

targets $\mathbf{T}(I)$ represent the set of perturbed genes; our model for molecular perturbations is not based on causal Bayesian networks, so we do not describe it here. Finally, we will generally consider the context $C$ to be a cell type, and we often work in the no-context setting (i.e., with a fixed cell type).

## 1.5 Overview of this thesis

In Section 1.2, we proposed *causal prediction by extrapolation* as a defining feature of causal models; in particular, their ability to make causal predictions ("If I do action $A$ in context $C$, then outcome $Y$ will happen") for out-of-distribution pairs $(A, C)$. This broad construal makes space for many different types of causal models; leaving practitioners with the difficult question about what model class to use for their application. Unfortunately for the practitioner, there is no one-size-fits-all solution to the general problem of causal modeling. From the discussion in Section 1.3.2, we learn that one should not rely on extremely general model classes, unless one has access to extremely rich data. Instead, I prescribe a more bespoke approach to causal modeling; where model classes are chosen, or even *designed*, with some "user" in mind, following the decision-centric approach outlined in Section 1.1.4.

As a result of this more tailored approach, I use several different classes of causal models in this thesis. Fortunately for the practitioner (and the reader!), these model classes are related in fairly straightforward ways, and the conceptual framework discussed in Section 1.3 should help to keep these model classes straight.

**Organization** In this section, we will go through the problem setting that is considered in each chapter, i.e., the model class used, the type of data that is assumed to be available, and the type of output that is desired (either a model or some form a prediction). In each problem setting, the most complex part to describe is the model class, so we will spend the bulk of this section motivating and defining these model classes, in particular,

- Section 1.5.1 describes the class of *latent factor causal models*, a subset of the class $\mathcal{M}_{\text{cbn-lin}}(\mathbf{V})$ of linear causal Bayesian networks, with graphical restrictions that reflect physical constraints on gene regulation.
- Section 1.5.2 defines the class of *causal factor models*, a distinct kind of causal models that is more connected to the matrix completion literature.
- Section 1.5.3 extends the class of causal Bayesian networks into the class of *causal representation models*, blending ideas from causality with mainstream machine learning.
- Section 1.5.4 defines a subset of the class $\mathcal{M}_{\text{cbn-lin}}(\mathbf{V})$ that is tailored toward causal effect estimation, and discusses the importance of studying *prediction-centric learning*.

### 1.5.1 Causal structure learning for clustered variables

In Chapter 2, we introduce the class of *latent factor causal models (LFCMs)*, and develop algorithms to learn these models from observational data. As discussed in the previous section, there is a well-established tradition of modeling gene regulation with causal Bayesian networks. The novelty in our definition of the class of LFCMs comes from considering the constraints that *spatial* structures imply on gene regulation; putting physical considerations back into the abstract definition of causal Bayesian networks.

**Additional biological background: transcription factories** The particular pattern of gene regulation that we attend to is the co-regulation of spatially-close genes by structures called transcription factories [Jac+93; SFG19]. The process of transcription is carried out by

a protein called *RNA polymerase*, but does not take place uniformly over all protein-coding genes. Rather, gene expression is modulated by proteins called *transcription factors (TFs)*, with specific TFs either increasing or decreasing gene expression levels.

*Transcription factories* are assemblies of RNA polymerase and transcription factors, located at discrete sites in the cell nucleus. Abstractly, we can think of a quantity, similar to gene expression, that summarizes the activity level of a transcription factory. Then, the expression levels of all genes that are near some transcription factory are subject to the common modulating force of this activity level; and most genes are only close enough to one transcription factory for this effect to be significant. From these details, we can gather that the following modeling assumptions are plausible approximations to reality:

**Specificity**: Each gene has its expression influenced by only one transcription factory.

**Mediated regulation**: Each gene regulates other genes only by way of its affect on some transcription factory.

Finally, we mention that technologies like scRNA-seq only measure gene expression; they do *not* measure the activity level of a transcription factory.

**Model class definition** We extend the common way of modeling gene regulatory networks with causal Bayesian networks by adding variables to represent the activity level of transcription factories. In particular, we divide the variables $\mathbf{V}$ into two sets, $\mathbf{X}$ and $\mathbf{F}$. The set $\mathbf{X}$ of *observed variables* represents gene expression levels, and the set $\mathbf{F}$ of *factor variables* represents transcription factory activity. Then, a graph $\mathcal{G}$ over $\mathbf{X} \cup \mathbf{F}$ represents the causal relationships between genes and transcription factories.

To reflect the biological constraints imposed by transcription factories, we add two graphical assumptions to our model class. We translate the assumption of specificity into the *unique cluster (**UC**)* assumption, and we translate the assumption of mediated regulation into the *bipartite (**BA**)* assumption. Primarily for technical reasons, we introduce two other graphical assumptions: the *Triple child (**TC**)* assumption and the *double parent (**DP**)* assumption, though the biological implications of these assumptions are also reasonable.[29] Formally, these four assumptions are defined on a DAG $\mathcal{G}$ as follows:

**UC**: Each observed variable has exactly one parent in $\mathcal{G}$.
($\forall\ X \in \mathbf{X}, |\mathrm{pa}_{\mathcal{G}}(X)| = 1$)

**BA**: $\mathcal{G}$ has no edges between pairs of observed variables or pairs of factor variables.
($\mathcal{G}$ has no edges $X_1 \to X_2$ or $F_1 \to F_2$ for $X_1, X_2 \in \mathbf{X}$ or $F_1, F_2 \in \mathbf{F}$)

**TC**: Each factor variable has at least three observed children in $\mathcal{G}$.
($\forall\ F \in \mathbf{F}, |\mathrm{ch}_{\mathcal{G}}(F)| \geq 3$)

**DP**: Connected pairs of factor variables have at least two paths between them in $\mathcal{G}$.
(If $F_1 \to X_1 \to F_2 \in \mathcal{G}$, then $\exists\ X_2 \neq X_1$ such that $X_2 \to F_2$ in $\mathcal{G}$)

Also for technical reasons, we assume linearity; this provides a powerful theoretical

[29] Biologically, the **TC** assumption implies that each transcription factory regulates at least three genes, and the **DP** implies that if some transcription factory $F_1$ is upstream of another transcription factory $F_2$, then there are at least two genes mediating the effect of $F_1$ on $F_2$.

toolbox for relating these graphical constraints to constraints on the covariance matrix $\mathrm{Cov}(\mathbf{X})$. Putting together these assumptions, we obtain our class of latent factor causal models for a fixed set $\mathbf{F}$ of factor variables,

$$\mathcal{M}_{\text{lfcm}}(\mathbf{X}, \mathbf{F}) \triangleq \left\{ M \in \mathcal{M}_{\text{cbn-lin}}(\mathbf{X} \cup \mathbf{F}) : G^M \text{ satisfies } \mathbf{UC},\ \mathbf{BA},\ \mathbf{TC},\ \text{and } \mathbf{DP} \right\},$$

which can again be extended to an unknown set of factor variables using the disjoint union,

$$\mathcal{M}_{\text{lfcm}}(\mathbf{X}) \triangleq \bigsqcup_{\mathbf{F}} \mathcal{M}_{\text{lfcm}}(\mathbf{X}, \mathbf{F}) \tag{1.10}$$

**Theoretical results** Since we assume that only the variables $\mathbf{X}$ are observed (i.e., the factor variables $\mathbf{F}$ are latent), we work with the population observable $\mathbb{P}^{M^\star}(\mathbf{X})$ for our problem setting. In Chapter 2, we show that the model class $\mathcal{M}_{\text{lfcm}}(\mathbf{X})$ is identifiable from $\mathbb{P}^{M^\star}(\mathbf{X})$; i.e., any model $M \in \mathcal{M}_{\text{lfcm}}(\mathbf{X})$ can be uniquely recovered from this marginal distribution.[30] This result demonstrates the utility of domain-specific modeling assumptions: the larger class $\mathcal{M}_{\text{cbn-lin}}(\mathbf{X} \cup \mathbf{F})$ is not identifiable from this observable, in fact, it is not even identifiable from the entire joint distribution over $\mathbf{X}$ and $\mathbf{F}$.

Moreover, our proof is constructive: for the population setting, we provide a polynomial-time algorithm that takes $\mathbb{P}^{M^\star}(\mathbf{X})$, and returns $M^\star$. The algorithm relies on *rank constraints* that $M^\star$ implies on the covariance matrix of $\mathbf{X}$; hence, we can adapt the algorithm to the finite-sample setting by replacing queries on these rank constraints with hypothesis tests.

### 1.5.2 Causal prediction with causal factor models

In Chapter 3 we consider our only class of causal models which is not directly defined in terms of causal Bayesian networks. This model class descends from the matrix completion and factor modeling literature; thought strictly speaking the model induces a *tensor* completion problem since we allow for multivariate outcomes. The model class is fairly general, with no highly domain-specific assumptions, but we still have a biological problem that provides considerable motivation.

**Biological motivation** For the purpose of drug discovery, it is quite useful to have good predictions about the effects of different *drugs* on different *cell types*. Casting this as a causal prediction problem, we consider cell types $C \in \mathcal{C}$ to be our contexts, drugs $A \in \mathcal{A}$ to be our actions, and gene expression profiles $Y \in \mathcal{Y}$ to be our outcomes, and we assume access to a dataset containing pairs of the form $(A, C, Y)$.

Generally speaking, $C$ and $A$ could be defined to contain quite a bit of information about each cell type and each drug, respectively. For example, $C$ could be a learned representation of cell type taken from a gene foundation model [CZ23], while $A$ could describe a drug's molecular structure. We start with a much simpler setting, where each cell type and drug is described just by a "meaningless" label; i.e., we can tell whether or not two data points are associated with the same cell type or the same drug, but we have no additional information that would allow us to compare two cell types or two actions. Although this setting is

[30] All identifiability results in this section require *faithfulness* assumptions, which we describe in the relevant chapter.

unrealistically pessimistic, its simplicity makes it an excellent example of how ideas from causality can be mixed with other kinds of mathematical models, and the results achieved in this setting give a lower bound to the results that could be achieved in more realistic settings.

**Model class definition** To formally describe this simple setting, we say that the contexts $c \in \mathcal{C}$ and actions $a \in \mathcal{A}$ are *categorical*, i.e., $\mathcal{A}$ and $\mathcal{C}$ are both finite sets with no additional structure. We can think of $c$ and $a$ as indices for their associated outcomes, thus we call this the *index-only* setting, and we use unbolded letters to reflect this perspective.

We allow the outcome set $\mathcal{Y}$ to be more complex, in particular, we consider $\mathcal{Y} = \mathbb{R}^p$ (in our biological application, $p$ denotes the number of genes in the measured gene expression profiles). Our models implement the causal asymmetry between contexts and actions by associating them with different objects: each action $a \in \mathcal{A}$ is associated with a vector, whereas each $c \in \mathcal{C}$ is associated with a matrix. In particular, a rank-$r$ *causal factor model (CFM)* on $(\mathcal{A}, \mathcal{C}, \mathcal{Y})$ takes the form

$$M \triangleq \left(\{\mathbf{v}_a^M \in \mathbb{R}^r\}_{a\in\mathcal{A}},\ \{U_c^M \in \mathbb{R}^{p\times r}\}_{c\in\mathcal{C}}\right), \tag{1.11}$$

the class of all rank-$r$ causal factor models on $(\mathcal{A}, \mathcal{C}, \mathcal{Y})$ is

$$\mathcal{M}^r_{\text{cfm}}(\mathcal{A}, \mathcal{C}, \mathcal{Y}) \triangleq \{M : M \text{ is of the form in Equation (1.11)}\}, \text{ and}$$

the class of all causal factor models (of any rank) on $(\mathcal{A}, \mathcal{C}, \mathcal{Y})$ is

$$\mathcal{M}_{\text{cfm}}(\mathcal{A}, \mathcal{C}, \mathcal{Y}) \triangleq \bigsqcup_{r=1}^{\infty} \mathcal{M}^r_{\text{cfm}}(\mathcal{A}, \mathcal{C}, \mathcal{Y}) \tag{1.12}$$

We use causal factor models $M$ to define the following causal prediction about the outcome of performing action $a$ in context $c$:

$$\boldsymbol{\mu}^M_{a,c} = U_c^M \mathbf{v}_a^M, \tag{1.13}$$

which also think of as a population observable, defining

$$\boldsymbol{\mu}^M_\Omega \triangleq \{(a, c, \boldsymbol{\mu}^M_{a,c})\}_{(a,c)\in\Omega} \quad \text{for any } \Omega \subseteq \mathcal{A} \times \mathcal{C} \tag{1.14}$$

In the case where $p = 1$, $\boldsymbol{\mu}^M_{\mathcal{A}\times\mathcal{Y}}$ can be thought of as a low-rank matrix of dimensions $|\mathcal{A}| \times |\mathcal{C}|$, with the entry at $(a, c)$ being $\boldsymbol{\mu}^M_{a,c}$. Similarly, when $p > 1$, $\boldsymbol{\mu}^M_{\mathcal{A}\times\mathcal{Y}}$ can be thought of as tensor of dimensions $|\mathcal{A}| \times |\mathcal{C} \times p$, with $\boldsymbol{\mu}^M_{a,c}$ being the $(a, c)$-th slice of the tensor; although the sense in which this tensor is "low-rank" is somewhat nuanced.[31]

[31] Unlike matrices, there are several distinct notions of the rank of a tensor. In the case of a causal factor model $M$, the relevant notion of rank is associated with decomposing the tensor into a $|\mathcal{A}| \times r$ matrix $V^M$ and a $|\mathcal{C}| \times r \times p$ tensor $U^M$. As we discuss in Chapter 3, this notion of rank is less strict than the more common notion of CPD rank, which requires further decomposing $U^M$.

**Theoretical results (identifiability)** We consider the following extrapolation task: given $\boldsymbol{\mu}_{\Omega}^{M^\star}$ for some $\Omega \subseteq \mathcal{A} \times \mathcal{C}$, predict $\boldsymbol{\mu}_{a^\star,c^\star}^{M^\star}$ for some $(a^\star, c^\star) \notin \Omega$. From the tensor perspective, we can think of this task as a tensor completion problem, but there are several relevant perspective. In Chapter 3, we call this task *causal imputation* to emphasize its relation to other commonly-used methods in biology (e.g. [SB12]) that consider the task as a missing data problem.

We give conditions on when $\boldsymbol{\mu}_{a^\star,c^\star}^{M^\star}$ is identified, which are stated in algebraic terms relating $\mathbf{v}_{a^\star,c^\star}^{M}$ to the vectors $\left\{\mathbf{v}_{a,c^\star}^{M^\star}\right\}_{(a,c^\star)\in\Omega}$. Again, these conditions are constructive: if $\boldsymbol{\mu}_{a^\star,c^\star}^{M^\star}$ is identifiable, then we show that we can recover it by a straightforward extension of the *synthetic interventions* method [ASS20], which is itself an extension of the well-established method of *synthetic controls* commonly used in econometrics [ADH10].

**Theoretical results (relation to CBNs)** Finally, we show that the class $\mathcal{M}_{\text{cfm}}(\mathcal{A}, \mathcal{C}, \mathcal{Y})$ of causal factor models can be directly related to a model class defined in terms of causal Bayesian networks and interventions, now using $a \in \mathcal{A}$ as an index over interventions (instead of vectors) and $c \in \mathcal{C}$ as an index over causal Bayesian networks (instead of matrices). In particular, take the class of linear causal Bayesian networks over $\mathbf{V} = \mathbf{X} \cup \mathbf{Z}$, where the variables $\mathbf{Z}$ are *exogenous* (i.e., they have no parents), i.e., define

$$\mathcal{M}_{\text{exog}}(\mathbf{X}, \mathbf{Z}) \triangleq \{M \in \mathcal{M}_{\text{cbn-lin}}(\mathbf{X} \cup \mathbf{Z}) : \text{pa}_{\mathcal{G}^M}(Z) = \varnothing \ \forall \ Z \in \mathbf{Z}\},$$

and define the set of do-interventions with targets $\mathbf{Z}$ as

$$\mathcal{I}_{\text{do-known}}(\mathbf{X}, \mathbf{Z}) \triangleq \{I \in \mathcal{I}_{\text{do}}(\mathbf{X} \cup \mathbf{Z}) : \mathbf{T}(I) = \mathbf{Z}\}.$$

Then, we let $a \in \mathcal{A}$ index interventions, and each $c \in \mathcal{C}$ index CBNs, i.e., we define models of the form

$$M \triangleq \left(\{I_a \in \mathcal{I}_{\text{do-known}}(\mathbf{X}, \mathbf{Z})\}_{a\in\mathcal{A}}, \{M_c \in \mathcal{M}_{\text{exog}}(\mathbf{X}, \mathbf{Z})\}_{c\in\mathcal{C}}\right), \tag{1.15}$$

and define the corresponding model class

$$\mathcal{M}_{\text{exogiv}}(\mathbf{X}, \mathbf{Z}, \mathcal{A}, \mathcal{C}) \triangleq \{M \text{ is of the form in (1.15)}\}. \tag{1.16}$$

Let $\mathbb{Q}_{a,c}^{M}$ denote the interventional distribution of $M_c$ for intervention $I_a$, and associate each pair $(a, c)$ with the prediction $\boldsymbol{\nu}_{a,c}^{M} = \mathbb{E}_{\mathbb{Q}_{a,c}}[\mathbf{X}]$. In Chapter 3, we show that a model $M \in \mathcal{M}_{\text{exogiv}}(\mathbf{X}, \mathbf{Z}, \mathcal{A}, \mathcal{C})$ can be mapped to a model in $\sigma(M) \in \mathcal{M}_{\text{cfm}}^{r}(\mathcal{A}, \mathcal{C}, \mathcal{Y})$, for $\mathcal{Y} = \mathbb{R}^{|\mathbf{X}|}$ and $r = |\mathbf{Z}|$, such that their predictions are consistent; i.e., $\boldsymbol{\nu}_{a,c}^{M} = \boldsymbol{\mu}_{a,c}^{\sigma(M)}$.[32]

This result provides a bridge between the language of causal Bayesian networks and the language of matrix/tensor completion, paving the way for a deeper understanding of the relationship between these two different frameworks. The result also demonstrates the utility of thinking of actions as interventions on latent variables, to which we turn next.

[32]We do not show that this map is surjective, though this is easy to see: for any rank-$r$ causal factor model $M$, we take $|\mathbf{Z}| = r$, we define do-intervention $I_a$ that sets $\mathbf{Z}$ to $\mathbf{v}_a$, and we define linear CBNs $B_c$ which deterministically set $\mathbf{X} = U_c\mathbf{Z}$. Hence, the CBN-based class is more expressive than the class of CFMs.

### 1.5.3 Causal representation learning from interventional data

In Chapter 4, we consider the class of *linear causal representation models*; we begin here by describing general causal representation models.

**Causal representation models** Suppose that we observe variables $\mathbf{X}$, taking values in $\mathbb{R}^p$, that have no clear causal relationships with one another, e.g. the pixels of an image, but that strongly suggest some more abstract features that are causally meaningful, e.g. morphological features of cell state. Although causal Bayesian networks over $\mathbf{X}$ are not reasonable models for such a setting, we can easily extend CBNs to handle such a problem.

To define this extension, we consider a new set of random variables $\mathbf{Z}$, taking values in $\mathbb{R}^d$, which represent the abstract features. Then, we use causal Bayesian networks over these variables, and we relate them to the observed variables via some function, which we can think of a description of the physical process that gives rise to $\mathbf{X}$. Putting these two elements together, we define the class of $d$-dimensional *causal representation models* over $\mathbf{X}$,

$$\mathcal{M}^d_{\text{crm}}(\mathbf{X}) \triangleq \left\{ \left(B^M, g^M\right) : B^M \in \mathcal{M}_{\text{cbn}}(\mathbf{Z}),\ |\mathbf{Z}| = d,\ g^M : \mathbb{R}^d \to \mathbb{R}^p \right\}$$

and the class of all causal represention models (CRMs) over $\mathbf{X}$,

$$\mathcal{M}_{\text{crm}}(\mathbf{X}) \triangleq \bigsqcup_d \mathcal{M}^d_{\text{crm}}(\mathbf{X}).$$

Let $\mathbb{P}^M(\mathbf{Z})$ denote the observational distribution of $B^M$; we can put this together with $g^M$ to obtain a distribution over $\mathbf{X}$

$$\mathbb{P}^M(\mathbf{X}) \triangleq g^M_\sharp \mathbb{P}^M(\mathbf{Z}),$$

where $g_\sharp \mathbb{P}(\mathbf{Z})$ denotes the *pushforward* of the distribution $\mathbb{P}(\mathbf{Z})$ along the function $g$. Finally, given a CRM $M = (B^M, g^M)$ and an intervention $I \in \mathcal{I}(\mathbf{Z})$, we define the *interventional* CRM

$$M_I \triangleq (B^M_I, g^M),$$

i.e., the intervention changes the causal Bayesian network $B^M$ to the interventional CBN $B^M_I$ in the usual way, and leaves $g^M$ unchanged.

**Model class definition** The model class $\mathcal{M}_{\text{crm}}(\mathbf{X})$ is quite expressive, and $\mathbb{P}^M(\mathbf{X})$ is far from being rich enough to identify it. To ensure identifiability, we have two options: make the model class less expressive, or consider richer data. In Chapter 4, we take both options, restricting the model class to the linear setting, and incorporating interventional data. In particular, define the class of (doubly) linear causal representation models,

$$\mathcal{M}_{\text{crm-lin}}(\mathbf{X}) \triangleq \left\{ \left(B^M, g^M\right) \in \mathcal{M}_{\text{crm}}(\mathbf{V}) : B^M \in \mathcal{M}_{\text{cbn-lin}}(\mathbf{Z}),\ g^M \text{ is linear} \right\}$$

and the set of single-node perfect interventions on $\mathbf{Z}$,

$$\mathcal{I}_{\text{perf-1}}(\mathbf{Z}) \triangleq \left\{ I \in \mathcal{I}_{\text{perf}}(\mathbf{Z}) : |\mathbf{T}(I)| = 1 \right\}.$$

Then, we put these together to define our model class

$$\mathcal{M}_{\text{crm-lin-iv}}(\mathbf{X}, K) \triangleq \mathcal{M}_{\text{crm-lin}}(\mathbf{X}) \times \underbrace{\mathcal{I}_{\text{perf-1}}(\mathbf{X}) \times \mathcal{I}_{\text{perf-1}}(\mathbf{X}) \times \ldots \times \mathcal{I}_{\text{perf-1}}(\mathbf{X})}_{K \text{ times}}, \tag{1.17}$$

and note that this model class defines interventions with *unknown* targets.

**Theoretical results (relation to CBNs)** A natural definition for the population observable in this problem is a tuple containing the observational distribution and the interventional distribution for each intervention, i.e.,

$$\mathbb{P}^{M}_{0:K}(\mathbf{X}) \triangleq \left(\mathbb{P}^{M}(\mathbf{X}), \mathbb{P}^{M_{I_1}}(\mathbf{X}), \dots \mathbb{P}^{M_{I_K}}(\mathbf{X})\right)$$

though we work instead with sufficient statistics, in particular, we use

$$\Theta^{M}_{0:K} \triangleq \left(\Theta^{M}_{0}, \Theta^{M}_{1}, \dots, \Theta^{M}_{K}\right), \tag{1.18}$$

where for $k = 1, \dots, K$, $\Theta^{M}_{k}$ is the inverse of $\mathrm{Cov}^{M_{I_k}}(\mathbf{X})$, the covariance matrix of $\mathbf{X}$ in the $k$-th interventional distribution, and we use $k = 0$ to denote the observational distribution.

The problem setting exhibits a form of benign non-identifiability. For example, given the CBN-intervention tuple $(M, I_1, I_2, \dots, I_K)$, we can define a predictively equivalent CBN-intervention tuple $(M', I'_1, I'_2, \dots I'_K)$ by simply changing the labels of $\mathbf{Z}$, i.e., by permuting the nodes in causal graph $G^M$, permuting the inputs to the function $g^M$, and permuting the targets of each intervention. This will leave all observational and interventional distributions unchanged; the same can be said about scaling each $Z \in \mathbf{Z}$ by a constant factor. If these are the only indeterminacies, we can say that a model is identifiable *up to permutation and scaling*, as shorthand, we will just say that a model is identifiable.

In Chapter 4, we give conditions under which the model class is identifiable. In particular, we show that one intervention per $Z \in \mathbf{Z}$ is *sufficient* for identifiability; i.e., for interventions $I_1, I_2, \dots, I_K$, all $M \in \mathcal{M}_{\mathrm{crm}}(\mathbf{X})$ are identifiable if $\{\mathbf{T}(I_k)\}_{k=1}^{K} = \mathbf{Z}$. Further, we show that this condition is *necessary*, in the worst case: if it does not obtain, then there exists $M \in \mathcal{M}_{\mathrm{crm}}(\mathbf{X})$ which is not identifiable. As always, our proof is constructive, given via an algorithm that uses algebraic information from the differences $\{\Theta^{M}_{k} - \Theta^{M}_{0}\}_{k=1}^{K}$ to iteratively determine information about the model components $B^M$ and $g^M$, and easy adaptable to the finite-sample setting.

### 1.5.4 Causal effect estimation with an unknown graph

In the final chapter, Chapter 5, we consider a problem of a different form, explicitly focused on better analyzing the interplay between the Data → Models and Models → Predictions steps of the data-to-decision pipeline. Compared to the other chapters, which mostly focused on the population setting, Chapter 5 focuses heavily on the finite-sample setting, and considers the question of *sample complexity*. For the sake of discussion, we focus on the case of well-specified models (i.e., samples are generated from a distribution defined by $M^\star \in \mathcal{M}$) and analyses performed the probably approximately correct (PAC) setting, which characterize how many samples $n$ are required such that an algorithm `Alg` *probably* produces an *approximately correct* output $\widehat{P}$.[33]

**Historical context** Historically, questions about sample complexity in causality have fallen into two buckets, studied by two separate communities. In the study of *causal effect estimation*, researchers typically work with a model class where the causal graph is known

[33] *Probably* = with probability at least $1 - \delta$, and *approximately correct* = $d(\widehat{P}, P^\star) < \varepsilon$ for some notion of distance $d$ and $P^\star$ being the correct output; here, $\delta, \varepsilon > 0$ are constants.

(e.g. $\mathcal{M}_{\text{cbn}}(G^\star)$), the algorithm `Alg` returns a causal prediction, and correctness is measured by how close this prediction is to the one defined by $M^\star$. On the other hand, in *causal structure learning*, researchers typically work with some model class where the causal graph is unknown (e.g. $\mathcal{M}_{\text{cbn}}(\mathbf{V})$), the algorithm `Alg` returns some model $\widehat{M}$ from this class, and correctness is measured by how close the causal graph $G^{\widehat{M}}$ is to $G^{M^\star}$.

We would like to combine these two kinds of sample complexity results: it is limiting to only consider settings where the causal graph is known, but from the decision-centric perspective, the results in causal structure learning are a bit off the mark. In particular, it can be *quite difficult* to learn a causal graph, for reasons that make no difference in practice; this mismatch occurs because it is difficult to identify edges that are very weak, but these are exactly the edges that don't significantly impact causal predictions. To begin bridging this gap, we focus on a simple problem: estimating the effect that intervening on a single variable $A$ has on a single outcome variable $Y$.

**Model class definition** The model class that we use is quite easy to describe. Given a DAG $G$ over random variables $\mathbf{V}$, the *descendants* of a variable $V \in \mathbf{V}$ are the nodes $W$ such that there is a directed path from $V$ to $W$ in $G$, i.e., there is a series of edges $V \to \cdot \to \ldots \to W$. The *nondescendants* of $V$, denoted $\text{nd}_G(V)$, are the variables in $\mathbf{V}$ which are not descendants of $V$. We consider the setting where we know the nondescendants of $A$, i.e., we consider the class

$$\mathcal{M}_{\text{cbn-nd}}(\mathbf{V}, A, \mathbf{N}) \triangleq \{M \in \mathcal{M}_{\text{cbn}}(\mathbf{V}) : \text{nd}_{G^M}(A) = \mathbf{N}\}, \tag{1.19}$$

for some fixed $\mathbf{N} \subseteq \mathbf{V}$. Then, we consider estimating the effect of a do-intervention $I$ that sets $A$ to $a$, focusing on the probability of some single value $y$ of $Y$, i.e., we focus on predicting

$$\mathbb{P}_a^{M^\star}(y) = \mathbb{P}^{M^\star_I}(Y = y).$$

**Theoretical results** In Chapter 5, we consider this problem in the setting of discrete-random variables, analyzing a two-stage algorithm that uses conditional independence test to learn an *approximate* parent set $\widehat{\mathbf{S}}$ of $A$, which it then uses as to estimate the desired causal effect using standard estimators. Our analysis considers two separate sources of error: *bias* arising from the wrong choice of parent set (i.e., using $\widehat{\mathbf{S}} \neq \text{pa}_{G^{M^\star}}(A)$), and *estimation error* that would be incurred even if $\widehat{\mathbf{S}}$ was the correct parent set.

To analyze the bias, we quantify how close $\widehat{\mathbf{S}}$ is to being a parent set in terms of a measure that is based on total variation distance, then propagate this error through to bound the resulting bias. In our analysis of estimation error, we divide our analysis into three parts, first "clipping" to handle values $\mathbf{s}$ of $\widehat{\mathbf{S}}$ with low probability $\mathbb{P}_{M^\star}(\widehat{\mathbf{S}} = \mathbf{s})$, then subdividing our analysis for the other values of $\widehat{\mathbf{S}}$ by defining a "good" set of events that holds with high probability. Since this second result is given in the PAC setting, it generalizes contemporaneous results that describe only the *expected value* of the estimation error, thus being of significant independent interest.

### 1.5.5 Summary

Let us summarize all of the problem settings in this thesis, using the language that we have described in this section. Recall from Section 1.3 that we define a *problem setting* as a pair $\mathcal{M}, \rho$, where $\mathcal{M}$ is a model class, and $\rho : \mathcal{M} \to \mathcal{P}$ is a forward mapping. In the first three chapters, we give identifiability results in the well-specified setting, where we observe $\mathbb{D} = \rho(M^\star)$ for some unknown $M^\star \in \mathcal{M}$.

All of our identifiability results are constructive, i.e., we provide an algorithm that takes $\mathbb{D}$ as input and returns a "correct" output if the conditions in our identifiability results hold. In the *model selection* setting (Sections 1.5.1 and 1.5.3), the output of the algorithm is a model $M$ such that $\widehat{M} \approx M$, for some appropriate definition of equivalence $\approx$. In the *causal prediction* setting (Section 1.5.2), the output is a prediction matching the prediction of $M^\star$. We summarize the problem settings and outputs in Table 1.1.

| | Model class $\mathcal{M}$ | Observable $\mathbb{D}$ | Output $\widehat{P}$ |
|---|---|---|---|
| Chapter 2 | $\mathcal{M}_{\text{lfcm}}(\mathbf{X})$<br>see (1.10) | $\mathbb{P}^{M^\star}(\mathbf{X})$ | $\widehat{M} \in \mathcal{M}_{\text{lfcm}}(\mathbf{X})$ |
| Chapter 3 | $\mathcal{M}_{\text{cfm}}(\mathcal{A}, \mathcal{C}, \mathcal{Y})$<br>see (1.12) | $\mu_\Omega^{M^\star}$<br>see (1.13) and (1.14) | $\widehat{\boldsymbol{\mu}}_{a^*,c^*} \in \mathbb{R}^p$<br>for $(a^*, c^*) \notin \Omega$ |
| Chapter 4 | $\mathcal{M}_{\text{crm-lin-iv}}(\mathbf{X}, K)$<br>see (1.17) | $\Theta_{0:K}^{M^\star}$<br>see (1.18) | $\widehat{M} \in \mathcal{M}_{\text{crm-lin-iv}}(\mathbf{X}, K)$ |
| Chapter 5 | $\mathcal{M}_{\text{cbn-nd}}(\mathbf{V}, A, \mathbf{N})$<br>see (1.19) | $\mathbb{P}^{M^\star}(\mathbf{V})$ | $\widehat{\mathbb{P}}_x(y) \in \mathbb{R}$ |

Table 1.1: **Problem settings in this thesis.** For each chapter, we describe the model class that we use, the type of data to which we assume access, and the output of our algorithms. See text for more details.

**Relations between model classes** Finally, we discuss the relationship between these different model classes. In Section 1.4.1, we introduced the model class which is the most fundamental for our purposes: the class $\mathcal{M}_{\text{cbn}}(\mathbf{V})$ of all causal Bayesian networks on random variables $\mathbf{V}$. We also introduced the subclass $\mathcal{M}_{\text{cbn-lin}}(\mathbf{V})$ of linear causal Bayesian networks, which we use as the starting point for most of our model classes. In particular, each section defined the following:

- Section 1.5.1 defined the class $\mathcal{M}_{\text{lfcm}}(\mathbf{X}, \mathbf{F})$ as a subclass of $\mathcal{M}_{\text{cbn-lin}}(\mathbf{X} \cup \mathbf{F})$; this class adds graphical assumptions relating the observed variables $\mathbf{X}$ to the factor variables $\mathbf{F}$ that are based on domain-specific principles from biology.
- In Section 1.5.2, we defined the class $\mathcal{M}_{\text{cfm}}(\mathcal{A}, \mathcal{C}, \mathcal{Y})$ of causal factor models, where each model consists of a set of vectors indexed by actions $a \in \mathcal{A}$, and a set of matrices indexed by contexts $c \in \mathcal{C}$. Although this class is not directly defined in terms of CBNs, we discussed how it can be related to CBNs by using $a \in \mathcal{A}$ to index interventions and $c \in \mathcal{C}$ to index linear causal Bayesian networks.

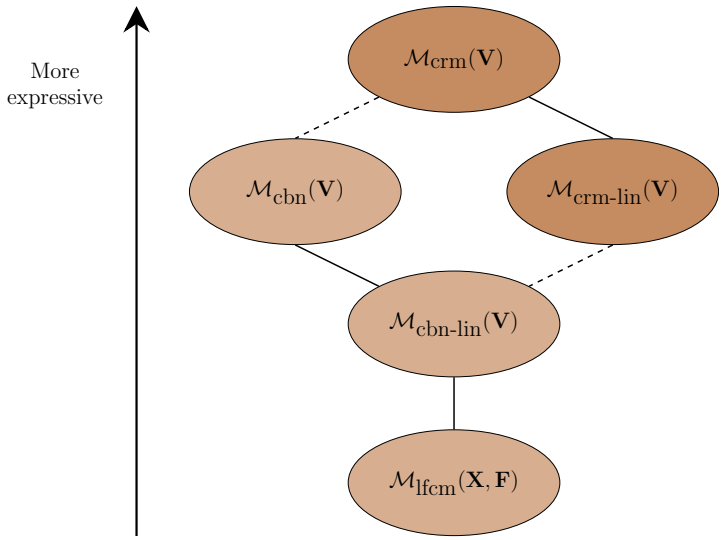



Figure 1.11: **Relationship between models in this thesis.** For simplicity, we focus on model classes that do not include interventions; thus, we omit $\mathcal{M}_{\text{cfm}}(\mathcal{A}, \mathcal{C}, \mathcal{Y})$. The two colors represent model classes of different types: lighter orange is for causal Bayesian networks, and darker orange is for causal representation models, we also indicate this difference with a dashed line. For the bottommost relation, we have $\mathbf{V} = \mathbf{X} \cup \mathbf{F}$.

- In Section 1.5.3, we defined the class $\mathcal{M}_{\text{crm}}(\mathbf{X})$ of general causal representation models by extending the class of causal Bayesian networks over latent variables $\mathbf{Z}$ with a function mapping $\mathbf{Z}$ to the observed variables $\mathbf{X}$. Once again, we studied the linear setting, defining $\mathcal{M}_{\text{crm-lin}}(\mathbf{X})$ as the class of linear causal representation models. Finally, we extended this class to include data from single-node interventions, for $K$ interventions, we write the class as $\mathcal{M}_{\text{crm-lin-iv}}(\mathbf{X}, K)$.

To summarize the relations between these classes, we use the Hasse diagram in Figure 1.11, where we place model classes according to their expressivity, defined in Section 1.3.2. In particular, if $\mathcal{M}_1$ is more expressive than $\mathcal{M}_2$, then we draw a line between $\mathcal{M}_1$ and $\mathcal{M}_2$, and we place $\mathcal{M}_1$ closer to the top of the figure.

# Chapter 2

# Causal structure learning for clustered variables

Adapted from

*Causal structure discovery between clusters of nodes induced by latent factors*
by Chandler Squires*, Annie Yun*, Eshaan Nichani, Raj Agrawal, and Caroline Uhler
(Cite: [Squ+22b], * denotes equal contribution)

In this chapter, we study the identifiability of *latent factor causal models (LFCMs)*, i.e., identifiability of the model class $\mathcal{M}_{\text{lfcm}}(\mathbf{X})$ defined in Equation (1.10) of Section 1.5.1. Each model $M$ in this class is a linear causal Bayesian network over two sets of variables, the observed variables $\mathbf{X}$ and the factor variables $\mathbf{F}$, and the class is defined by biologically-motivated structural restrictions on the causal graph $G^M$. We show that such models are identifiable from *only* the observational distribution $\mathbb{P}^{M^\star}(\mathbf{X})$. The key lesson of this chapter is the utility and importance of domain-specific modeling assumptions: in the language of Section 1.3.3, stronger modeling assumptions imply a less expressive model class, which requires less rich data for identifiability. To aid the reader, we provide a brief guide for how the discussion in Section 1.5.1 maps onto this chapter.

**Reading this chapter** In this chapter, we only work with the model class $\mathcal{M}_{\text{lfcm}}(\mathbf{X})$. Thus, we define the model class as part of our problem setup (Section 2.2), with the linearity assumption stated in text, and the graphical conditions stated in Definition 1, using $L_i$ (for *latent* variables) in place of $F_i$. Rather than stating our identifiability result in terms of $\mathbb{P}^{M^\star}(\mathbf{X})$, we state it (equivalently) as a consistency result. In particular, we write the main learning algorithm (Algorithm 1) in the finite-sample setting, taking as input a data matrix $\mathbb{X} \in \mathbb{R}^{n \times p}$ (for $p = |\mathbf{X}|$) and returning $\widehat{\mathcal{G}}$ as output. Then, the main theorem (Theorem 2) states that, as $n \to \infty$, the probability that $\widehat{\mathcal{G}}$ equals the true causal graph $\mathcal{G}$ goes to one.[1]

[1] In the terminology of the introduction, this result says that $G^{\widehat{M}} = G^{M^\star}$ in the population setting. Showing identifiability of a causal graph $G^{M^\star}$ (as opposed to the entire model $M^\star$) is standard practice in causal structure learning: given $G^{M^\star}$, the causal mechanisms $\left\{\mathbb{P}^{M^\star}(V \mid \text{pa}_G(V))\right\}_{V \in \mathbf{V}}$ can be recovered via conditional density estimation, if all $V$ are observed. When some $V$ are unobserved (as in this chapter), there may be indeterminacies from aspects such as scaling, but these indeterminacies are generally benign in the sense described in 1.3.5.

## 2.1 Introduction

Structural causal models are valuable tools for reasoning about decision-making, and as a result, have been widely adopted across fields such as genomics [Fri+00], econometrics [Bla17], and epidemiology [RHB00]. To use causal models when the causal structure is not known *a priori*, it is necessary to learn the model from observed data, a task known as *causal structure learning* [HMM18]. As a field, causal structure learning has recently experienced major developments and remains an active and widespread area of research. Recent works aim to address a number of challenges inherent to the problem of learning causal structure, such as the presence of unobserved confounders [Cai+19; FNM19; Ber+20], the large search space over causal models [Chi02; SWU21], identifiability of the underlying causal model [Shi+06; PB14], and statistical issues stemming from high-dimensional datasets [NHM18]. We focus on a setting which exhibits all of these challenges, and our proposed method addresses each of these challenges in a cohesive way.

A number of methods have been proposed to address the challenge of learning causal models in the presence of unobserved confounders. These methods fall into two general categories. First, some methods account for unobserved confounders by learning a graphical model over only the observed variables, albeit from a *different* class of graphical models [RS02; Ber+20]. However, in some cases, such as the one explored in this chapter, it is possible to learn a graph over *both* the observed and latent variables. Existing methods [Sil+06; KR16; Xie+20; Agr+23] that seek to recover these structures often assume that the latent variables are *exogenous*, i.e., are not caused by any of the observed variables. However, this assumption is often violated in many applications. For example, in genomics, gene regulatory networks are often modeled using *transcription factories* [SFG19] as underlying latent variables with gene expression as the observable variables. These gene expressions then can have downstream impacts on other transcription factories, requiring a model that allows non-exogenous latent variables.

**Contributions.** In Section 2.2, we introduce the class of *latent factor causal models* (LFCMs), which allow for non-exogenous latent variables. Similar to prior work, this class of models prohibits direct edges between observed variables, i.e., the effect of one observed variable on another must be mediated by some latent variable. We likewise prohibit direct edges between latent variables, inducing a *bipartite* structure over the graph. Furthermore, we require that the latent variables *cluster* the observed variables, i.e., each observed variable has only a single latent parent, and that each latent variable has at least three observed children. These constraints on the model are motivated by how the DNA is organized in the cell nucleus to facilitate cell-type specific gene expression. The spatial clustering of genes in the cell nucleus facilitates their co-regulation by transcription factors [Bel+17; US17]. The expression of each gene represents the observed variables, the spatial clustering of genes is unobserved, and the latent factors represent the presence of transcription factors that can e.g. turn on the expression of the co-clustered genes.

In Section 2.3, we establish identifiability results for LFCMs, based primarily on the tetrad representation theorem of [Spi13]. Based on our identifiability results, in Section 2.4 we propose a constraint-based method for learning the underlying graph over *both* latent and observed variables. The proposed method has three stages. In the first stage, our method

identifies clusters of observed variables with the same latent parent, as well as an ordering over these clusters. The second stage merges clusters from the first stage if necessary. In the third stage, we learn edges from the observed variables to the latent variables, by testing for conditional independence with all children of each latent variable. For each stage, the constraints being checked are equivalent to multiple test statistics vanishing simultaneously, requiring the use of multiple hypothesis testing procedures which we describe in Section 2.4.1. Finally, in Section 2.5, we demonstrate the performance of our algorithm in both a completely synthetic and a semi-synthetic setting. In particular, we show that our method is capable of recovering the ground truth clustering with nearly 100% accuracy even at relatively low sample sizes. Our method also recovers the ground truth edges between observed nodes and latent nodes with higher accuracy than a baseline which does not make use of multiple hypothesis testing.

### 2.1.1 Related work

**Learning undirected graphical models with clusters.** Since clusters of correlated variables are common across many disciplines, including biology [Eis+98], economics [BW16], neuroscience [Ars+18; PC20], and the behavioral sciences [LH13], several structure learning methods have been developed which encourage clustering in the estimated graphs, especially in the setting of *undirected* graphical models. For example, [TWS15] introduced the *cluster graphical lasso* method, which generalizes the traditional graphical lasso method to allow for the incorporation of known clustering information, resulting in denser estimated subgraphs over these clusters. Building on this work, [HL16] introduce the *GRAB* algorithm, which does *not* require clusters to be known beforehand, but instead allows the clustering to be learned simultaneously to network structure. More recently, [PC20] introduced *ComGGL*, a method which also learns clusters and graph structure simultaneously, with the additional benefit of high-dimensional consistency guarantees for both cluster recovery and graph structure in sparse settings.

**Latent tree models and factor analysis.** Unlike in the undirected settings above, in our setting, the clusters of observed variables are explicitly assumed to be induced by latent variables. As has been observed in previous works, especially in latent tree modeling [Cho+11; Shi+16; Drt+17; LD18] and factor analysis [DSS07; KR16], these latent variables produce “signatures" or “invariants" in the distribution over the observed variables, which can be exploited for structure learning. One invariant which plays an important role in both settings is the *tetrad* $t_{ij,uv}$, a $2 \times 2$ subdeterminant of the correlation matrix which must *vanish* (i.e., equal zero) whenever $i$ and $j$ share a single common latent parent, but have no children. As we will see in Section 2.3, despite the differences in our model assumptions, tetrads also play an important role in our algorithm when identifying causal clusters.

**Traditional causal discovery methods.** Within the space of discovering causal models on observational data, there are two categorizations of algorithms. First, there are constraint-based methods, which rely on conditional independence testing to draw conclusions about the structure. The well-known PC-algorithm assumes *causal sufficiency*, which bars unmeasured common cause latent variables and selection variables [Spi+00]. There also exist constraint-based methods on directed acyclic graphs with latent and selection variables, such as FCI, RFCI and their variants [Spi01; Col+12]. These methods all learn Markov equivalence classes

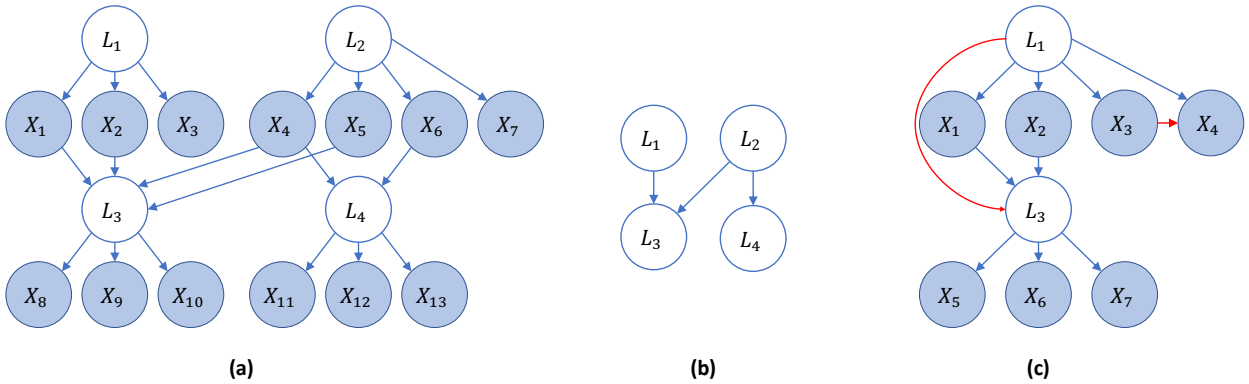



Figure 2.1: **(a)** $\mathcal{G}$ satisfies our model constraints. **(b)** The latent graph $L(\mathcal{G})$ for $\mathcal{G}$. **(c)** $\mathcal{G}'$ falls outside of the class of models we consider in this paper, with violations shown in red.

of directed acyclic graphs, as represented by completed partially directed acyclic graphs (CPDAGs) in the PC-algorithm or partial ancestral graphs (PAGs) in the FCI algorithm. The second categorization of methods is score-based algorithms, such as GES [Chi02], which identify underlying structure by optimizing a well-designed score function. These methods, even those that are asymptotically correct in the presence of latent confounders, output equivalence classes of DAGs. In this chapter, we try to recover more complete causal information.

**Learning causal models with latent variables.** Existing work for structure recovery in the presence of latent variables can often by characterized by the model structures that the method performs well or poorly upon. [Agr+23] considers the model in which latent variables are *pervasive*, influencing many observed nodes. Their method, *DeCAMFounder*, recovers the true causal structure over observed variables by applying spectral decomposition in the non-linear additive noise and pervasive confounding setting [Agr+23], extending the linear setting of [FNM19]. A number of methods, similarly to the current chapter, also rely on algebraic constraints in the covariance matrix over observable variables to infer graph structure over latent variables. The BPC [Sil+06] and FOFC [KR16] algorithms both leverage rank constraints on the covariance matrix to cluster observed variables, then recover some structure over the inferred latent nodes corresponding to these clusters. Other algorithms, such as those proposed by [SHH09], [Cai+19], and [Xie+20], attempt to improve upon previous algorithms by restricting models to the *linear non-Gaussian* case. In this chapter, we do not require non-Gaussianity, instead working in the general linear acyclic model regime. Furthermore, all of the above algorithms rely on the *measurement assumption*, which requires that no observed variable is the parent of any latent variable. This assumption, however, is not satisfied in many real-world applications of graphical models with latent variables, and in this chapter, we attempt to recover causal structures *without* the measurement assumption.

## 2.2 Problem setup

We now formally define the class of models considered in this chapter. A *structural causal model* (SCM) over the variables $\{X_i\}_{i=1}^p$ consists of a set of *structural assignments* of the

form $X_i = f_i(X_{\text{pa}(i)}, \epsilon_i)$, and a product distribution $\mathbb{P}_\epsilon$ over mean-zero *exogenous noise* terms $\{\epsilon_i\}_{i=1}^p$. The set $\text{pa}(X_i)$ are called the *parents* of $X_i$, and the *causal graph* for the SCM is a graph with nodes $\{X_i\}_{i=1}^p$ and directed edges $X_j \to X_i$ for $X_j \in \text{pa}(X_i)$. We assume that the causal graph for the SCM is *acyclic*, in which case the distribution $\mathbb{P}_\epsilon$ induces a unique distribution $\mathbb{P}_X$ over $\{X_i\}_{i=1}^p$.

In this chapter, we focus on a class of SCMs with restrictions on both the structural assignments and on the causal graph. First, we assume each $f_i$ is a *linear* function, a common starting point for new methods, which has been the setting of many works [Chi02; HB12; SWU21]. Second, we assume that the causal graph is of the following form.

**Definition 1.** *Let $\mathcal{G}$ be a DAG over latent nodes $L_1, \ldots, L_K$ and* observed nodes *$X = \{X_1, \ldots, X_p\}$. The* clusters *of $\mathcal{G}$ are the sets $C_k = ch(L_k)$ for $k = 1, \ldots, K$. The* latent graph *for $\mathcal{G}$, denoted $L(\mathcal{G})$, is the graph over $\{L_k\}_{k=1}^K$ with an edge $k \to k'$ if and only if $X_k \to L_{k'}$ for some $X_k \in ch(L_k)$. $\mathcal{G}$ is called a* latent factor causal model *(LFCM) if it satisfies the following conditions:*

*(a) [Unique cluster assumption] Each observed node has exactly one latent parent.*

*(b) [Bipartite assumption] There are no edges between pairs of observed nodes or between pairs of latent nodes.*

*(c) [Triple-child assumption] Each latent node has at least 3 observed children.*

*(c) [Double-parent assumption] If $k \to k'$ in $L(\mathcal{G})$, then there exist two nodes $X_i, X_j \in ch(L_k)$ such that $X_i \to L_{k'}$ and $X_j \to L_{k'}$.*

See Figure 2.1a for an example of a graph that satisfies our model definition, and Figure 2.1c for a graph that does not. The importance of each assumption for the purpose of identifying $\mathcal{G}$ will become clear in the proofs of the genericity and identifiability results presented in the next section. For example, we will see that the edge $X_3 \to X_4$ in Figure 2.1c prevents the submatrix $\Sigma_{[1,2],[3,4]}$ of the covariance matrix $\Sigma$ from being low rank, and thus prevents $\{X_1, X_2\}$ and $\{X_3, X_4\}$ from being clustered.

## 2.3 Trek separation and genericity assumptions

In this section, we review fundamental results essential to the identifiability of LFCMs, which we constructively prove in Section 2.4 by introducing an algorithm for consistently estimating LFCMs. We will also introduce genericity assumptions necessary for the consistency of our algorithm.

We denote the covariance matrix of our model as $\Sigma$, and given two subsets of nodes $A, B$, we use $\Sigma_{A,B}$ to denote the submatrix of $\Sigma$ with rows in $A$ and columns in $B$. Our identifiability results rely on a common generalization of d-separation, known as *trek separation*, which relates the causal graph of a SCM to the rank of submatrices of $\Sigma$. A *directed path* from node $i$ to node $j$ is a sequence of nodes $p_1 = i, ..., p_k = j$, such that $p_i \to p_{i+1}$ for all $i$ from 1 to $k-1$. In this case, $j$ is called the *sink* of the path and $i$ is called the *source*. A *trek* in the graph $\mathcal{G}$ from $i$ to $j$ is an ordered pair of directed paths $(P_1, P_2)$, such that the sink of

$P_1$ is $i$ and the sink of $P_2$ is $j$, and $P_1, P_2$ share a source $k$. Now, we define *trek separation*. Given four subsets $A, B, C_A, C_B$ of nodes (note these subsets need not be disjoint), the pair $(C_A, C_B)$ *t-separates* $A$ and $B$ if, for every trek $(P_1, P_2)$ between $A$ and $B$, $P_1$ contains a node in $C_A$ or $P_2$ contains a node from $C_B$. Finally, the following theorem relates the notion of *t-separation* to the rank of submatrices of the covariance matrix.

**Theorem 1** (Trek separation, [STD10]). *Let $A, B$ be two subset of nodes in $\mathcal{G}$. Then*

$$\textit{rank}(\Sigma_{A,B}) \leq \min\{|C_A| + |C_B| : (C_A, C_B) \textit{ t-separates } A \textit{ from } B \textit{ in } \mathcal{G}\}$$

*Moreover, equality holds generically*[2] *for $\Sigma$ consistent with $\mathcal{G}$.*

In this chapter, we only need to use information about the rank of 2 x 2 submatrices of $\Sigma$. The determinants of these matrices are commonly known as *tetrads*. In particular, we denote $t_{ij,uv} = \det(\Sigma_{[ij],[uv]}) = \Sigma_{iu}\Sigma_{jv} - \Sigma_{iv}\Sigma_{ju}$. Specializing Theorem 1, we obtain the following corollary:

**Corollary 1** (Tetrad representation, [Spi13]). *Suppose $A = \{X_i, X_j\}$ and $B = \{X_u, X_v\}$ are t-separated by a single node. Then $t_{ij,uv} = 0$.*

We can now see the importance of the first three assumptions in Definition 1. These structural assumptions control the size of t-separating sets between nodes in the same cluster and in different clusters, so that we can apply Theorem 2 and Corollary 1 to ensure that certain tetrads are either zero or generically non-zero. In particular, the unique cluster assumption and the bipartite assumption guarantees that two nodes $X_i$ and $X_j$ in the same cluster will be t-separated from their non-descendants by their latent parent. Thus, clusters of nodes with no descendants can be identified. Conversely, the triple-child assumption ensures that two nodes $X_i$ and $X_j$ that are not in the same cluster do not get clustered, since we can find a 2x2 submatrix with $i$ indexing one of the rows and $j$ indexing one of the columns that is generically of rank 2. In Appendix A.1, we formally state and prove that the following faithfulness assumptions are indeed generic under the first 3 structural assumptions from Definition 1:

**Assumption 1** (Cluster tetrad faithfulness). *Suppose $X_i$ and $X_j$ are not in the same cluster. Then there exists some $\{u, v\}$ such that $t_{ij,uv} \neq 0$.*

**Assumption 2** (Parent tetrad faithfulness). *Suppose $X_i$ and $X_j$ are in the same cluster, but $X_i$ has at least one child. Then there exists some $\{u, v\}$ such that $t_{ij,uv} \neq 0$.*

**Assumption 3** (Latent adjacency faithfulness). *Suppose $X_i \to L_k$. Let $S_i = ch(pa(X_i)) \setminus \{i\}$ and $S' = \cup_{j \leq i} ch(L_i)$. Then $\rho_{i,k|S_i,S} \neq 0$ for some $X_k \in ch(L_k)$*

[2]We say a statement holds *generically* if the set of parameters for which it does not hold has Lebesgue measure zero.

**Remark 1.** *Since causal structure learning algorithms are always run in a noisy setting,* near *violations of genericity assumptions can degrade the performance of a method, as discussed by [Uhl+13]. In particular, the set of parameters which violate a "strong" faithfulness condition is generally a positive measure set, extending from the measure zero set where faithfulness is violated. Fortunately for the current setting, our assumptions require the existence of only a* single *entry of the underlying statistic being far from zero. Thus, the set of parameters violating the "strong" version of our faithfulness assumption is an* intersection *of the sets of parameters for which each $t_{ij,uv}$ is near zero, resulting in a smaller set. In the present chapter, we will not attempt to quantify the size of this set and the resulting statistical benefits, but note these as interesting directions for future work.*

## 2.4 Methods

Our algorithm, presented in Algorithm 1, consists of three stages. As is common in causal structure learning, we present our algorithm with implementation details of hypothesis testing abstracted away. In particular, we will assume access to two subroutines, whose implementation details will be given in Section 2.4.1. The first subroutine tests $H_{ci}(X_j, X_A \mid X_B)$, which denotes the null hypothesis that $X_j$ and $X_A$ are conditionally independent given $X_B$. The second subroutine tests $H_{vt}(X_A, X_B)$, which denotes the null hypothesis that all tetrads of $\Sigma_{A,B}$ vanish.

**Algorithm 1** `EstimateLFCM`

**Input:** Data $\mathbb{X} \in \mathbb{R}^{n \times p}$.
Let $\pi =$ `FindOrderedClusters`$(\mathbb{X})$
Let $\pi =$ `MergeClusters`$(\mathbb{X}, \pi)$
Let $\hat{\mathcal{G}} =$ `LearnDAG`$(\mathbb{X}, \pi)$
**Output:** $\hat{\mathcal{G}}$

In the **first stage** (Algorithm 2, see also Figure 2.2), we identify clusters of observed variables with the same latent parent. However, note that since this stage only identifies *leaves* with the same latent parent, it is not guaranteed to identify all nodes with the same latent parent. This stage simultaneously recovers an ordering over these clusters. Thus, in the **second stage** (Algorithm 3, see also Figure 2.3a), we iterate over pairs of clusters output from the first stage, identify pairs of clusters with the same latent parent, and merge them, while leaving the ordering of the clusters intact. In the **third stage** (Algorithm 4, see also Figure 2.3bc), we use the clustering and ordering information discovered in the previous two stages to learn a DAG over both latent and observed variables. In particular, given a node $X_j$ in cluster $C_j$ which comes before the cluster $C_i$ in our ordering, we wish to determine whether $X_j$ has an edge to the associated latent variable $L_i$. By Assumption 2, this can be accomplished by checking partial correlations between $X_j$ and the nodes in $C_i$. These stages compose a consistent algorithm, as established in the following theorem and proven in Appendix A.2. In Section 2.3, we have already discussed the importance of the first three structural assumptions from Definition 1. In Appendix A.3, we show how our algorithm fails under a violation of the double-parent assumption.

**Theorem 2.** *Let $\mathcal{G}$ be a linear LFCM and let $\mathbb{X} \in \mathbb{R}^{n \times p}$ be a matrix of samples of the observed variables $X_1, \ldots, X_p$. Then Algorithm 1 is consistent under Assumptions 1, 2, and 3, i.e., as $n \to \infty$, we have $\mathbb{P}(\hat{\mathcal{G}} = \mathcal{G}) \to 1$.*

Next, we outline the complexity of our algorithm, using placeholders for the complexities

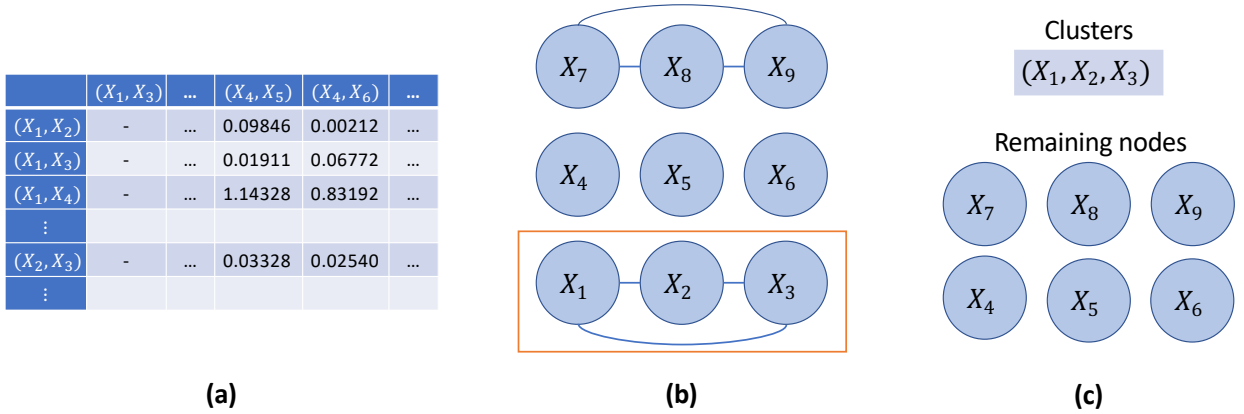



Figure 2.2: **First phase**: Let the true LFCM be the graph in Figure 2.3(c). In our algorithm's first phase (see Algorithm 2) we perform the following steps: **(a)** Compute tetrad scores between all pairs of nodes. **(b)** For each pair of nodes, test the null hypothesis that all tetrads are zero. Construct a graph with the edge $i - j$ for any pair of nodes where we do not reject the null hypothesis. **(c)** Extract a clique from this graph to be a cluster (e.g., by picking the largest clique with arbitrary tie breaking), remove these nodes and repeat with remaining nodes.

of hypothesis tests in order to keep our results general. Let $f(d, n)$ denote the cost of performing the hypothesis test $H_{vt}$ on $d$ statistics from $n$ samples, when all sufficient statistics are pre-computed. Let $g(p)$ denote the complexity of an algorithm used to find a clique in a graph on $p$ nodes. While finding the *largest* clique in a graph is in general NP-hard, we can avoid this complexity, since the consistency of our algorithm does not rely on picking the largest clique at each step, only a clique of 3 or more nodes (picking larger cliques is simply a tool for improving statistical accuracy).

In the following, let $M$ be the maximum size of any returned cluster, and let $K$ be the number of clusters discovered by the algorithm. By definition, $M \leq p$ and $K \leq p$, so replacing these quantities by $p$ gives complexities that are only in terms of the known problem parameters. However, such upper bounds can be highly pessimistic. If the true graph has few nodes per cluster, or a small number of latent nodes, then with enough samples, $M$ and $K$ will also be small, respectively.

**Theorem 3.** *The complexity of each algorithm is:*

*(a) Algorithm 2 takes* $\mathcal{O}(p^4 + p^3 f(p^2, n) + pg(p))$.

*(b) Algorithm 3 takes* $\mathcal{O}(p^2 M^4)$.

*(c) Algorithm 4 takes* $\mathcal{O}(pKM)$.

*Proof.* **(a)** Computing all tetrads and their associated p-values is $\mathcal{O}(p^4)$. In each round, we perform $\mathcal{O}(p^2)$ hypothesis tests, each on $\mathcal{O}(p^2)$ statistics, so that the complexity at each round is $\mathcal{O}(p^2 f(p^2, n))$. After performing these tests, we identify the largest clique in a graph of $\mathcal{O}(p)$ nodes, so that the total run time per round is $\mathcal{O}(p^2 f(p^2, n) + g(p))$. At most $p$ rounds are required, resulting in the stated complexity.

**Algorithm 2** `FindOrderedClusters`

**Input:** Data $\mathbb{X}$.
Initialize $R = [p]$
Initialize $\pi$ as an empty list
**while** $|R| > 3$ **do** Initialize $\mathcal{G}$ as an empty graph
**for** each pair of nodes $i, j$ in $R$ **do** if $H_{vt}(\{i, j\}, [p] \setminus \{i, j\})$, then add $i - j$ to $\mathcal{G}$
**end for**Let $C$ be the largest clique in $\mathcal{G}$, breaking ties arbitrarily
$R = R \setminus C$
Remove $C$ from the columns of $\mathbb{X}$
Append $C$ to $\pi$
**end while**
**if** $|R| > 0$ **then** Append $R$ to $\pi$
**end if**
**Output:** An ordered clustering $\pi$

**Algorithm 3** `MergeClusters`

**Input:** Data $\mathbb{X}$, ordered clustering $\pi$
Repeat the following until convergence:
**for** Clusters $c_1$ and $c_2$ such that $c_1 \prec c_2$ **do**
**if** $H_{vt}(c_1 \cup c_2, c_1 \cup c_2)$ **then** Add nodes in $c_2$ to $c_1$ and delete $c_2$
**end if**
**end for**
**Output:** An ordered clustering $\pi$

**Algorithm 4** `LearnDAG`

**Input:** Data $\mathbb{X}$, ordered clustering $\pi$
Initialize $\hat{\mathcal{G}}$ as an empty graph
Add $L_i$ to for each $C_i$ in $\pi$
Add $L_i \to X_j$ to $\hat{\mathcal{G}}$ for each $C_i$ in $\pi$, $X_j \in C_i$
**for** $X_j \prec L_i$ **do** Let $S = \cup_{k|L_k \prec L_i} \text{ch}(L_k)$
Add $X_j \to L_i$ to $\hat{\mathcal{G}}$ if $H_{ci}(X_j, \text{ch}(L_i) \mid S)$
**end for**
**Output:** DAG $\hat{\mathcal{G}}$

**(b)** We must check $\mathcal{O}(p^2)$ pairs of clusters for whether or not they should be merged, and the maximum size of the union of any such pair is $\mathcal{O}(M)$. To check whether $\mathcal{O}(M)$ nodes belong to the same cluster, we require a hypothesis test on $\mathcal{O}(M^4)$ statistics, so that this step takes $\mathcal{O}(p^2 f(M^4, n))$.

**(c)** We perform $\mathcal{O}(pK)$ hypothesis tests, each based on $\mathcal{O}(M)$ partial correlations. □

Assume that we use the Sidak adjustment procedure explained in the next section (so that $f(d, n) = \mathcal{O}(d)$), and a greedy algorithm for picking cliques (so that $g(p) = p^3$). Then, replacing $K$ and $M$ by $p$, we have that Algorithm 2 takes $\mathcal{O}(p^5)$, Algorithm 3 takes $\mathcal{O}(p^6)$, and Algorithm 4 takes $\mathcal{O}(p^3)$, so that the overall complexity of our algorithm is at most $\mathcal{O}(p^6)$. Even in this pessimistic analysis, this complexity is relatively low for causal structure learning, which is known to be NP-hard in general [CHM04], and for which variants of the best-known algorithms, such as PC [Spi+00] and GES [Chi02], typically have complexity $\mathcal{O}(p^{d+2})$, where $d$ is the maximum in-degree of the graph [Chi20].

### 2.4.1 Implementation details

The null hypothesis $H_{vt}$ and $H_{ci}$ used in Algorithms 2, 3, and 4 imply that some vector-valued statistic of the covariance matrix is equal to zero. Procedures for *simultaneous hypothesis testing* are designed to (asymptotically) control the false discovery rate (FDR) of such a form

of hypothesis test. In practice, we found that computing marginal p-values and performing *Sidak adjustment* [DP07] yields good performance. In particular, given p-values $\{\pi_m\}_{m=1}^M$, the Sidak-adjusted p-values are

$$\pi_m^{\texttt{sidak}} = 1 - (1 - \pi_m)^M$$

**Remark 2.** *The Sidak adjustment uses only the marginal distributions of each tetrad, neglecting potentially important information about the correlations between tetrads. In contrast, the* max-T adjustment *accounts for correlations between the tested statistics by estimating their correlation matrix, and has been shown to outperform the Sidak adjustment both theoretically and in practice [DP07; CCK13]. However, the max-T adjustment requires sampling from a potentially high-dimensional multivariate normal distribution, an operation which is $\mathcal{O}(d^3)$ for dimension $d$. We have found that in practice, max-T adjustment performs similarly to Sidak adjustment while taking substantially longer. Therefore, we use Sidak adjustment for our experimental results, but provide capability for max-T adjustment in our codebase.*

Given a set adjusted p-values and a significance level $\alpha$, we reject the null hypothesis if *any* of the adjusted p-values are smaller than $\alpha$. To test conditional independence, recall that $H_{ci}(X_j, X_A \mid X_B)$ holds in a multivariate normal if and only if the vector of partial correlations $\{\rho_{ij|B}\}_{i \in A}$ is zero. To compute p-values, we use a widely used procedure which we call the *Fisher correlation test*. First, given the sample partial correlations $\{\hat{\rho}_{ij|B}\}_{i \in A}$, we apply the *Fisher z-transformation* $\hat{z}_{ij|B} = \sqrt{n - |B| - 3}\texttt{arctanh}(\hat{\rho}_{ij|B})$. Then, we compute the two-tailed p-value of $\hat{z}_{ij|B}$ with respect to $\mathcal{N}(0, 1)$, i.e., $\pi_{ij|B} = 2Q(|\hat{z}_{ij|B}|)$, where $Q$ is the tail distribution function of $\mathcal{N}(0, 1)$.

Next, to test $H_{vt}(X_A, X_B)$, we adopt the widely-used *Wishart test* to compute the p-values [Wis28; KR16], which we now briefly describe. First, we compute the *sample tetrads* $\hat{t}_{ij,uv} = \widehat{\Sigma}_{iu}\widehat{\Sigma}_{jv} - \widehat{\Sigma}_{iv}\widehat{\Sigma}_{ju}$ for $\{i, j\} \subset A$ and $\{u, v\} \subset B$ such that $i, j, u$ and $v$ are distinct. Then, we normalize each sample tetrad, dividing by an estimate of its standard deviation to obtain the z-score $\hat{z}_{ij,uv}$. [DMO08] give the following formula for the variance of sample tetrads in terms of the true covariance matrix $\Sigma$:

$$\texttt{Var}\left(\hat{t}_{ij,uv}\right) = n \cdot (n-1)^{-3} \cdot \left((n+2)|\Sigma_{[ij],[ij]}| \cdot |\Sigma_{[uv],[uv]}| - n|\Sigma_{[ijuv],[ijuv]}| + 3n|\Sigma_{[ij],[uv]}|\right),$$

where $|A| = \det(A)$. To estimate the variance, we use the above formula with the sample covariance $\widehat{\Sigma}$ replacing $\Sigma$. Finally, we compute the two-tailed p-value of $\hat{z}_{ij,uv}$ with respect to $\mathcal{N}(0, 1)$.

## 2.5 Empirical results

We evaluate our algorithm in two settings. First, we evaluate in a purely synthetic setting, which allows us to generate SCMs which exactly match our proposed model. Then, we evaluate in a semi-synthetic setting, modifying real data to more closely match our proposed model while demonstrating that our approach has promise in real-world biological settings.

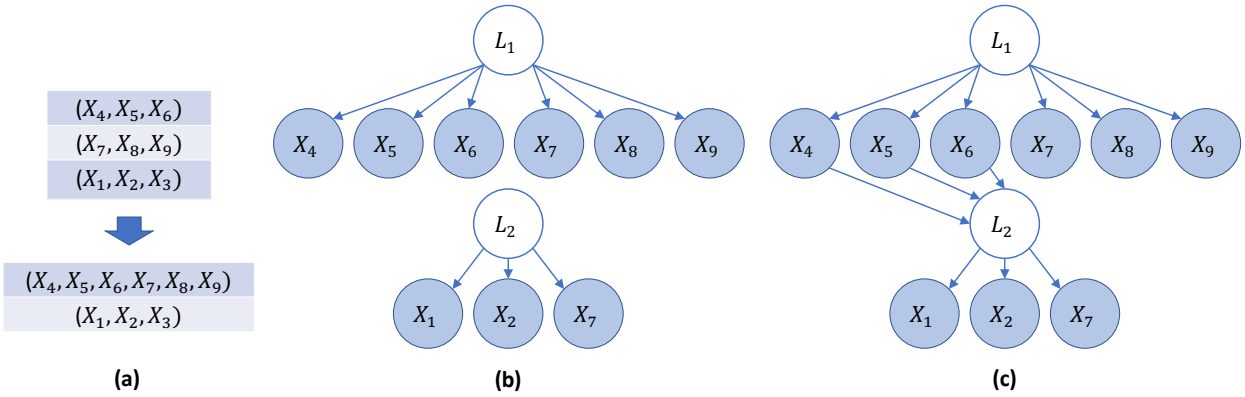


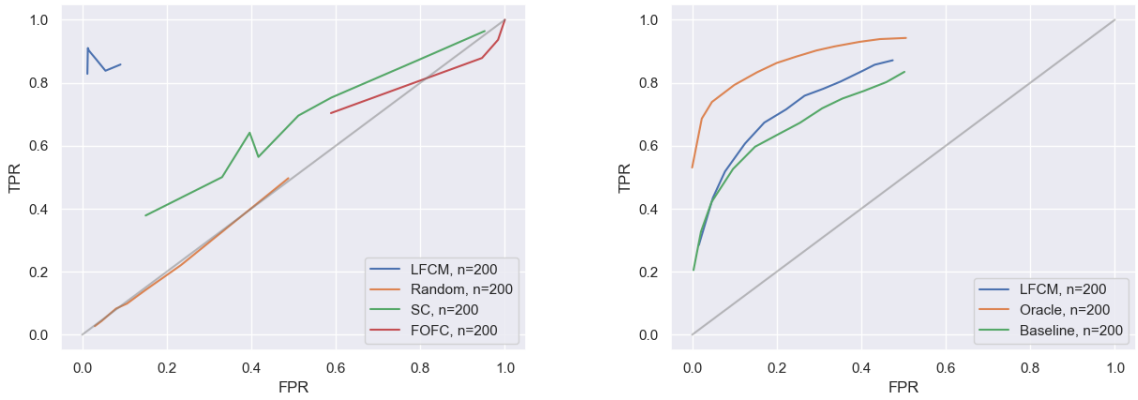

Figure 2.3: **Second and third phases. (a)** Merge pairs of clusters based on vanishing tetrad tests. **(b)** Introduce latent nodes, and add edges from latent nodes to children. **(c)** Add parents of latent nodes based on conditional independence testing.

### 2.5.1 Synthetic data experiments

We begin by briefly describing the simulation settings used for our experiments, before describing the baselines and metrics which we use for evaluation. We generate a graph with 10 latent nodes, we first sample a "latent" skeleton $L(\mathcal{G})$ over $\{1, 2, \ldots, 10\}$ from a directed Erdös-Rényi model with edge probability 0.5. Then, for each latent node $L_k$, we generate $c_k \sim \mathsf{Unif}(3, 6)$ children. Finally, for each edge $L_k \to L_{k'}$ in $L(\mathcal{G})$, we sample $d_{k,k'} \sim \mathsf{Unif}(2, |\mathrm{ch}(L_k)|)$, then sample $d_{k,k'}$ children of $L_k$. For each selected child $c_k$, we add the edge $c_k \to L_{k'}$, giving us a DAG $\mathcal{G}$ over both latent and observed nodes which has latent skeleton $L(\mathcal{G})$ and satisfies Definition 1.

Given this DAG, we generate a linear SCM as follows, proceeding in topological order. For each node $X_j$, if the node has no parents, its equation is $X_j = \epsilon_j$ for $\epsilon_j \sim \mathcal{N}(0, 1)$. If the node has parents, then for each parent $X_i$, we sample an "initial" weight $\tilde{w}_{ij} \sim \mathsf{Unif}([-1, -.25] \cup [.25, 1])$. Next, we describe how to normalize these weights in order to avoid the *varsortability* issue described by [RSW21], where simulated DAGs are easy to learn because the variance of each node tends to increase according to the topological order. Given these initial weights, we simulate $B$ "parental contributions" $\mu_j^{(b)} = \sum_{i \in \mathrm{pa}_{\mathcal{G}}(j)} \tilde{w}_{ij} X_i^{(b)}$ for $X^{(b)}$ sampled from the linear SEM defined over $i < j$. The sample variance $\widehat{\sigma}_j$ of $\mu_j^{(b)}$ serves as an estimate for the variance that the parents of $j$ will contribute to $X_j$. Finally, we ensure that $X_j$ has variance 1 and that half of its variance is contributed by its parents by setting the final weights as $w_{ij} = (2\widehat{\sigma}_j)^{-1/2} \tilde{w}_{ij}$ and $\epsilon_j \sim \mathcal{N}(0, 1/2)$.

**Accuracy of learning clusters.** In our first set of experiments, we evaluate the accuracy of the learned clusters. To measure the accuracy over the learned clustering compared to the underlying clustering, we use the following criteria: the pair $(X_i, X_j)$ is a *true positive* if $X_i$ and $X_j$ are in the same underlying cluster and are in the same learned cluster, the pair is a *false positive* if $X_i$ and $X_j$ are not in the same underlying cluster but are in the same learned cluster, and so on. We generate 50 different SEMs via the process described above, and from each SEM we generate $n = 200$ samples. We run our algorithm using significance levels ranging from .05 and .5. The results are shown in Figure 2.4a. Due to interactions between




(a) **Cluster recovery performance.** (b) **Edge recovery performance.**

Figure 2.4: **Recovering latent factor causal models with synthetic data.** The first two phases of our algorithm almost perfectly recover the ground truth clusters, while the third phase of our algorithm demonstrates the utility of multiple hypothesis testing for recovering edges between observed nodes and latent nodes.

the hypothesis tests used by our algorithm (denoted "LFCM", shown in blue), the ROC curve is highly non-monotonic over larger ranges of values, so that the curve occupies only a small range of the plot, though it clearly drastically outperforms the competing methods, achieving almost perfect performance. In particular, we consider two baselines. First, we compare to *spectral clustering* (denoted "SC", shown in green), as implemented in `sklearn` in Python, a widely used clustering technique in genomics [HKK07], with a varying number of estimated clusters from 2 to 30. Second, we compare to the FindOneFactorCluster algorithm of [KR16] (denoted "FOFC", shown in red). We found that spectral clustering performs slightly better than random guessing, while FOFC performs about the same as random guessing, reflecting the drastic deviation from the measurement assumption on which it relies. Finally, we verified that randomly picking $K$ clusters of equal size (denoted "Random", shown in orange), for $K$ varying from 2 to 30, matched the diagonal random guessing line.

**Accuracy of learning edges from observed nodes to latent nodes.** In our second set of experiments, we evaluate the accuracy of learning the edges from observed nodes to latent nodes, when the true clusters and their ordering is known. In particular, for $L_i \prec L_j$ in the ordering, and $X_i \in \mathrm{ch}(L_i)$, the pair $(X_i, L_j)$ is considered a *true positive* if $X_i \to L_j$ in the true LFCM as well as in the estimated LFCM, a *false positive* if it is not in the true LFCM but does appear in the estimated LFCM, and so on. In Figure 2.4b, we compare the third phase of our algorithm (denoted "LFCM", shown in blue) to a *baseline* which simply uses a single child of each latent node for the conditional independence test (denoted "Baseline", shown in green), as well as an *oracle* which is able to observe the values of the latent nodes and is thus infeasible (denoted "Oracle", shown in orange). As expected, our algorithm does not perform as well as this unrealizable case, but still performs significantly better than random (the diagonal line) and noticeably better than the baseline.

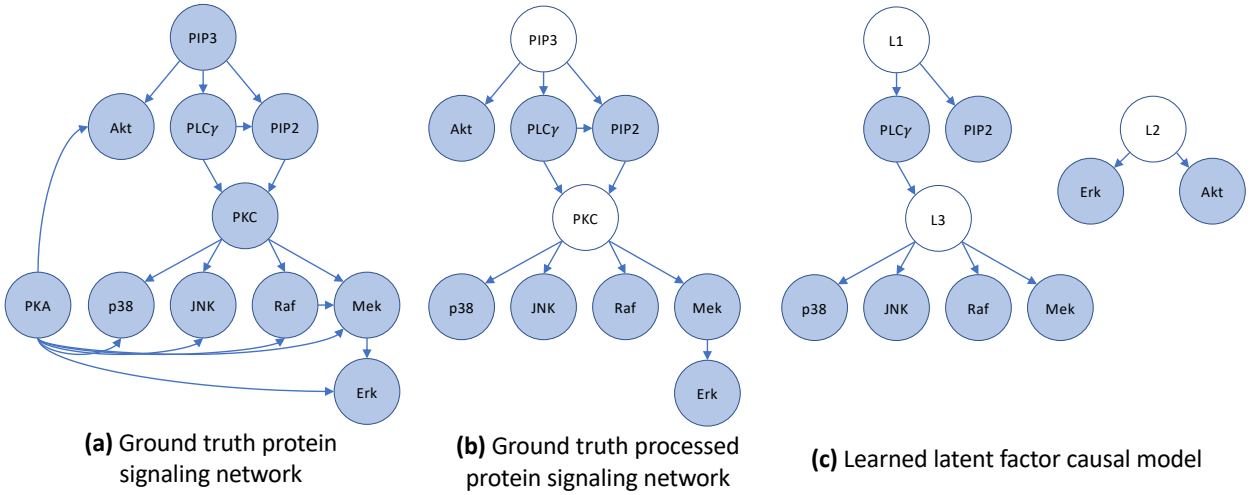



Figure 2.5: **Learning a latent factor causal model for protein signaling**. Our recovered model in **(c)** nearly captures the ground truth network in **(b)**.

### 2.5.2 Semi-synthetic experiments on protein signaling data

In this section, we demonstrate the applicability of our method to a real-world dataset in a semi-synthetic setting. The Sachs protein mass spectroscopy dataset [Sac+05] is a widely used benchmark for causal discovery, in part due to the existence of a commonly accepted ground truth network over the 11 measured protein expression values, shown in Figure 2.5a. We use the 1,755 "observational" samples, where the experimental conditions involve only perturbing receptor enzymes, and not any signaling molecules, as described in [Wan+17]. To make the ground truth network more similar to a latent factor causal model, we perform three data-processing steps: (1) we "condition" on PKA, by regressing it out of the dataset, (2) we "remove" the direct effect of Raf on Mek, and (3) we "marginalize" out PIP3 and PKC by removing the corresponding columns from the dataset. We "remove" the direct effect of Raf on Mek as follows. First, we regress Mek on its two remaining parents, Raf and PKC. Call the resulting regression coefficient for Raf $\beta_{Raf}$. For each sample, we subtract the value of Raf times the $\beta_{Raf}$ from the value of Mek. Note that we do *not* remove the direct effect of PLC$\gamma$ on PIP2, since then our algorithm collapses all nodes into a single cluster. The processed graph is show in Figure 2.5b.

Running our method with significance level $\alpha = 0.01$ for $H_{vt}$ and $\alpha = 0.1$ for $H_{ci}$, we obtain the network shown in Figure 2.5c. The clustering by our algorithm closely matches the clustering (*Akt, PLC$\gamma$, PIP2*), (*p38, JNK, Raf, Mek, Erk*) induced by the true network, with the exception that *Akt* from the first cluster and *Erk* from the second cluster are pulled out into a cluster with one another, which may indicate that the effect of PKA on Akt and Erk cannot be completely removed using a purely linear approach. The ordering between the clusters (*PLC$\gamma$, PIP2*) and (*p38, JNK, Raf, Mek*) is preserved, but the edge $PIP2 \to L3$ is missing.

## 2.6 Discussion

In this chapter, we introduce a method (Algorithm 1) for learning *latent factor causal models* (LFCMs), a novel, biologically-motivated class of causal models with latent variables. We showed that these models are identifiable in the linear setting using rank constraints on submatrices of the covariance matrix, and that our method provides a consistent estimator for these models. We also showed that our method outperforms existing clustering algorithms on synthetic data, and almost perfectly recovers a widely-accepted ground truth network in a semi-synthetic biological setting. These results serve as a proof-of-concept, suggesting that our algorithm may be able to shed biological insight on the problem of identifying the spatial clustering of genes in the cell nucleus given data on the expression of the genes. Interestingly, since it is possible (although expensive) to measure the 3D organization of the genome in the cell nucleus [Lie+09], there is a meaningful avenue to validate our method on the important biological application of connecting 3D genome organization with gene expression [US17]. We conclude with a discussion of the limitations of our model, which suggests a number of other directions for future work.

**Limitations.** The latent factor causal model (LFCM) class considered in this chapter has two obvious limitations. First, we make the strong parametric assumption of a linear Gaussian SEM. While many nonparametric conditional independence tests have been proposed [Gre+07; Zha+11], we are not aware of nonparametric tests for shared latent factors that would generalize $H_{vt}$. Thus, extending our algorithm to a nonparametric setting would require development of such tests. In particular, generalizing tetrad constraints to the nonlinear setting is an interesting direction for future research. Second, we make two strong structural assumptions. The "unique cluster" assumption is well-motivated by our biological setting of interest, and is likely the easiest assumption to remove since Theorem 1 already provides a generalization of the rank constraint we leverage. Indeed, generalized rank constraints have already been explored in prior work on factor analysis [DSS07; KR16]. The "bipartite assumption" has two components which may be separately examined. First, the assumption that there are no edges between observed variables is most reasonable for systems such as gene regulatory networks where a different, unobserved entity class (in this case, proteins) mediates *all* interactions between the observed variables (i.e., genes). This assumption may also be expendable, for instance by allowing for a *small number* of edges between observed variables, akin to the low-rank plus sparse literature in previous work on learning with exogenous latent variables [FNM19; Agr+23]. Similarly, existing techniques [Cai+19; Xie+20] may help to eliminate the assumption that there are no edges between latent variables. In addition to relaxing these assumptions, it would be of interest to develop procedures for testing these assumptions in data, e.g., by extending recent work [ASS20] which develops a spectral-energy-based hypothesis test for structural assumptions in latent variable models.

# Chapter 3

# Causal prediction with causal factor models

Adapted from

*Causal imputation via synthetic interventions*
by Chandler Squires*, Dennis Shen*, Anish Agarwal, Devavrat Shah, and Caroline Uhler
(Cite: [Squ+22a], * denotes equal contribution)

In the last chapter, we focused on *causal structure learning*, i.e., we developed an algorithm whose output was the *causal graph* of some causal Bayesian network model. In this chapter, we focus directly on *causal prediction*: our algorithm does not produce a model as output, but rather a prediction; our modeling assumptions are used to *justify* the predictions of our algorithm. In particular, we study the class of *causal factor models (CFMs)* for predicting the outcome of doing action $a \in \mathcal{A}$ in context $c \in \mathcal{C}$. Denoting the outcome space as $\mathcal{Y}$, we work with the model class $\mathcal{M}_{\text{cfm}}(\mathcal{A}, \mathcal{C}, \mathcal{Y})$ (defined in Equation (1.12)), and consider identifiability of the causal prediction $\boldsymbol{\mu}^{M}_{a^\star, c^\star}$ (defined in Equation (1.14)). As input, we consider the population observable $\boldsymbol{\mu}^{M}_{\Omega}$ for $\Omega \subseteq \mathcal{A} \times \mathcal{C}$ with $(a^\star, c^\star) \notin \Omega$ (defined in Equation (1.14)).

This chapter has two key lessons. First, causal factor models enable one to *extrapolate* to new action-context pairs in a relatively intuitive way. Second, different classes of causal models can often be related — in this chapter, we relate $\mathcal{M}_{\text{cfm}}(\mathcal{A}, \mathcal{C}, \mathcal{Y})$ to the model class $\mathcal{M}_{\text{exogiv}}(\mathbf{X}, \mathbf{Z}, \mathcal{A}, \mathcal{C})$ (defined in Equation (1.16)) that is based on causal Bayesian networks and interventions.

**Reading this chapter** Our notation in this chapter slightly differs from the notation in Section 1.5.2. In this chapter, we use $\mathbf{x}^{ca}$ to refer to the causal prediction $\boldsymbol{\mu}^{M}_{a^\star, c^\star}$, and we define the model class $\mathcal{M}_{\text{cfm}}(\mathcal{A}, \mathcal{C}, \mathcal{Y})$ as part of our theoretical results (see Definition 2). Our main identifiability result is in Theorem 4, where we establish conditions under which the model's prediction is identifiable for an unseen $(a, c)$ pair. Then, we describe the model class $\mathcal{M}_{\text{exogiv}}(\mathbf{X}, \mathbf{Z}, \mathcal{A}, \mathcal{C})$ in Section 3.5.2, and establish the connection between model classes in Proposition 1.

## 3.1 Introduction

A central goal in science is to determine the outcome of an action (i.e., intervention). Traditionally, researchers rely on experimentation to answer such questions. However, this approach is rarely scalable and/or can be unethical in many critical applications, e.g., clinical trials and policy evaluation. As a result, there is a growing interest in utilizing already-existing data to help predict the effect of every candidate action, prior to deciding which action to execute. The key challenge underpinning this approach is that the available data regarding the effect of an action almost always comes from a *different context* than the one in which we are interested in making a prediction. To further motivate our problem setting, we provide an example from healthcare below.

*A Motivating Example.* Consider a prevalent medical scenario in which physicians aim to re-purpose existing drugs to treat novel diseases such as COVID-19. Although there are over 20,000 small molecule compounds (actions) with known therapeutic capabilities, exposing COVID-19 infected cells (new contexts) to each compound would be far too time-consuming and costly. Fortunately, the publicly available CMAP dataset [Sub+17] has already catalogued the effects of these compounds on a variety of other cell types (different contexts). In such a setting, the primary intellectual challenge is to develop a methodology which uses the CMAP dataset to predict the outcome of applying each compound on COVID-19 infected cells. Such a method leads to efficient *hypothesis generation* for finding compounds with therapeutic potential for this disease.

*Key Question.* The focus of this chapter is to tackle problems such as the one described above. More generally, we aim to answer the following question:

*"Given a dataset of outcomes from various action-context pairs, can we predict the outcome of a given action on a context in which it has not been applied?"*

**Our Contributions.** The main contributions of this chapter are three-fold: algorithmic, theoretical, and empirical. We summarize them below.

*(i) Algorithmic.* We extend the recently proposed synthetic interventions (SI) estimator [ASS20] to handle more general sparsity patterns. At its core, the SI estimator produces a counterfactual prediction for any "target" (novel) context-action pair by first constructing a "synthetic" version of the target context as a weighted combination of "donor" (different) contexts via (regularized) regression. Then, it uses the learnt model to re-scale the outcomes of those donor contexts under the target action to estimate the outcomes of the target context-action pair. Taking inspiration from biological applications, we adapt the SI estimator to instead construct a synthetic version of the target *action* as a weighted combination of donor actions. Given the more general observation patterns, we also equip the SI estimator to use *all* available data when learning the regression model. As a result, the SI estimator of [ASS20] becomes a special case of our formulation. Moving forward, we refer to our estimator as `SI-A`. For details, please refer to Section 3.4.

*(ii) Theoretical.* We justify our algorithmic approach by presenting an "identification" result in Theorem 4. More formally, we establish that our estimator yields the correct causal estimates for the causal imputation task, under a factor model across contexts and actions.

| | Fast | Heterogeneous effects | Guarantees | No selection diagram needed |
|---|---|---|---|---|
| Fixed Effects (FE) | ✓ | ✗ | ✓ | ✓ |
| Autoencoding + FE | ✓ | ✓ | ✗ | ✓ |
| Causal Transportability | ✓ | ✓ | ✓ | ✗ |
| MICE/MissForest | ✗ | ✓ | ✗ | ✓ |
| Synthetic Interventions | ✓ | ✓ | ✓ | ✓ |

Table 3.1: Summary of causal imputation methods

Further, we re-interpret the SI causal framework by connecting it to linear structural causal models (see Proposition 1).

*(iii) Empirical.* Using the prominent CMAP dataset, we extensively benchmark our approach against well-established baselines. The CMAP dataset contains gene expression signatures for over 20,000 different small molecule compounds (actions) across 70 different cell types (contexts). Using a small fraction of the observed signatures as a training set, we consider the task of imputing the signatures associated with a held-out test set that emulates an un-experimented collection of novel cell type-compound pairs. Empirically, we find that our estimator outperforms the baselines. In particular, the median normalized root mean-square error (NRMSE) of `SI-A` is 0.34, compared to 0.41 for the closest baseline. Similarly, `SI-A` improves the *alignment* of the predicted effect of the action with the true effect of the action, measured in terms of cosine similarity, improving from 0.44 in the best baseline to 0.68. Moreover, our experiments show that our variant of the SI estimator (which regresses along compounds) significantly outperforms the original SI estimator (which regresses along cell types), a result that may be of independent biological interest.

**Organization of the chapter.** In Section 3.2, we review related work on causal imputation. Then, in Section 3.3, we formally state our problem setting. In Section 3.4, we describe our estimation strategy, which extends the original SI estimator. We state our formal results in Section 3.5, establishing an identification result under a factor model assumption, as well as providing an interpretation of this assumption in terms of a linear structural causal model. Finally, Section 3.6 showcases our empirical findings on the CMAP dataset.

## 3.2 Related work

Predicting the (vector-valued) outcome of an action in a novel context, a task we call *causal imputation*, is a ubiquitous problem, and there are numerous lines of study tackling this challenge. We highlight some of the most relevant below. In practice, we require a method that (i) is *fast*, (ii) can model *heterogenous* effects across different cell types, (iii) comes with theoretical *guarantees*, and (iv) does *not* need a detailed model of the causal system as input. Existing methods, which we describe below, all lack one of these 4 desiderata, as summarized in Table 3.1.

**Fixed effects.** The simplest model for the effect of actions across different contexts is the *fixed effects* model, which associates each action with a vector representing the additive effect of that action, assumed to be *invariant* across contexts. We note that the fixed effects estimator is a special case of the regularized linear model proposed in [Dix+16]. Their method makes predictions of the form $\widehat{\mathbf{x}}^{ca} = \beta^a + \mathbf{w}^c\beta^c$, where $\widehat{\mathbf{x}}^{ca}$ denotes the *outcome* vector associated with cell type $c$ under action $a$ and $\mathbf{w}^c$ is a set of covariates for the cell type $c$. When no covariates are available except a one-hot encoding of the cell type, their estimator reduces to the fixed-effects model $\widehat{\mathbf{x}}^{ca} = \beta^a + \beta^c$. Under a fixed effects model, the effect of an action in any new context can be predicted based solely on the effect of that action in a *single* other context, greatly reducing the number of required experiments. However, the fixed effects model is not realistic in our application of interest, as it does *not* allow for heterogeneity, even though the effect of a drug on the expression of a given gene can vary drastically between cell types [Kid+16], e.g., increasing the expression of a gene in one cell type while decreasing it in a different cell type.

To overcome this limitation of the fixed effects model, [LWT19] introduced the *scGen* method. This method first trains a variational autoencoder (VAE) in order to find a low-dimensional representation (latent embedding) of gene expression data, then uses a fixed effects model in the *latent* space of the VAE. Similarly, one could instead apply our method, `SI-A`, in the *latent* space. The theoretical study of combining autoencoders, or other nonlinear embeddings, with "direct" causal imputation methods such as the one considered here, is an interesting direction for future work. To the best of our knowledge, no theoretical guarantees yet exist for such combinations, but promisingly, recent work using autoencoders to predict perturbation effects on SARS-CoV2 infected cells empirically demonstrated that autoencoders *align* representations in a way that induces linearity amongst the perturbations and cell types in the learned space [Bel+21].

**Causal transport.** The goal in causal transport is to find a *transport formula* [BP14; BP16], i.e., a map from the outcomes of actions in a set of "source" contexts to the outcome in a "target" context under some action. Whereas causal transport focuses on prediction from data generated in *different* contexts than the target context, causal imputation generalizes this problem to *also* use data about *different* actions, including from the target context itself. Previous methods algorithmically derive transport formulas from a *selection diagram*, i.e., a causal model of the system that introduces additional nodes corresponding to contexts, allowing one to model how causal mechanisms might change between contexts. See [LCB20b] for a recent exposition on selection diagrams and algorithms for deriving transport formulas. A major contrast between the present chapter on causal imputation and prior work on causal transport is that `SI-A` does *not* rely on specifying (or learning) a selection diagram. Indeed, as we show in Proposition 1 and Theorem 4, `SI-A` is consistent for *any* linear causal structural model within a large class. A further advantage of the factor model framework considered in this work is that the key assumption (Assumption 5) that enables `SI-A` to generalize from a small collection of experiments to predicting on untested context-action pairs can be tested via a simple data-driven hypothesis test (see Section 3.4).

**Traditional imputation methods.** As missing values are a common issue in many applications [BK07; Tro+01], a number of imputation methods have been developed to fill in missing entries, see e.g. [BPZ17] for a recent review. Two prominent methods which are widely used in imputing genomic and other biological data [Tan+17; Wal+13] are MICE

[VO99] and MissForest [SB12]. To the best of our knowledge, these methods have no formal identification guarantees. Moreover, they are several orders of magnitude slower than our `SI-A` estimator - see Appendix B.6 for a comparison to MICE - making them impractical for the problem considered here.

The linear factor model assumption (Definition 2) used by our model is similar to factor model assumptions made by methods for low-rank tensor decomposition, such as PARAFAC [Har+70; Bro97]. As in the matrix case [Cha15; Aga+18; SBJ20], such decompositions can be useful tools for tensor estimation [JO14], especially in the missing-at-random (MAR) or missing-completely-at-random (MCAR) settings [Rub76]. However, PARAFAC assumes a low-rank *CP decomposition*, which is more restrictive than the low-rank *mode-2 decomposition* in Definition 2. Furthermore, PARAFAC requires a costly alternating minimization scheme to optimize, so that it is again several orders of magnitude slower than our `SI-A` estimator.

**Methods using other data modalities.** The imputation methods we have thus far discussed all begin with genomic data from various compounds and cell types, and impute genomic data for *novel* pairs of compounds and cell types. Orthogonally, there is a vast literature on predicting perturbation response using *different* data modalities, e.g. images [Hof+19; YKU18] or molecular structure [Sto+20]. There are many possible avenues for combining these approaches with our method, e.g., using coupled autoencoders [Yan+21a] to represent each compound in terms of a combination of other compounds *and* an encoding of its molecular structure.

## 3.3 Problem statement

We consider a collection of contexts (e.g., cell types), denoted by $\mathcal{C}$, and actions (e.g., compounds), denoted by $\mathcal{A}$. The outcomes of interest (e.g., gene expression signatures) associated with a given context $c \in \mathcal{C}$ under action $a \in \mathcal{A}$ are denoted by $\mathbf{x}^{ca} \in \mathbb{R}^p$. Collectively, this forms an order-three tensor $\mathcal{X} = [\mathbf{x}^{ca} : c \in \mathcal{C}, a \in \mathcal{A}] \in \mathbb{R}^{|\mathcal{C}| \times |\mathcal{A}| \times p}$.

**Observations.** To model real-world scenarios, we consider the setup where we not only have access to a sparse subset of the entries in $\mathcal{X}$ (e.g., corresponding to a limited number of historical experiments), but also noisy instantiations of those observations. More formally, let $\tilde{\mathbf{x}}^{ca} \in \mathbb{R}^p$ denote a corrupted version of $\mathbf{x}^{ca}$, e.g., $\tilde{\mathbf{x}}^{ca} = \mathbf{x}^{ca} + \mathbf{e}^{ca}$ or $\tilde{\mathbf{x}}^{ca} = \mathbf{x}^{ca} \circ \mathbf{e}^{ca}$, where $\mathbf{e}^{ca}$ denotes measurement noise and $\circ$ is the entry-wise product. Further, let $\Omega \subset \mathcal{C} \times \mathcal{A}$ denote the context-action pairs for which we observe their associated outcomes, i.e., for each $(c, a) \in \Omega$, we observe $\tilde{\mathbf{x}}^{ca}$. We denote our observation set as $\mathcal{O} = \{\tilde{\mathbf{x}}^{ca} : (c, a) \in \Omega\}$.

**Goal.** Our primary interest is in recovering $\mathcal{X}$ from $\mathcal{O}$.

*Notations.* For any $a \in \mathcal{A}$ and $c \in \mathcal{C}$, let $\mathcal{A}(c) = \{a \in \mathcal{A} : (c, a) \in \Omega\}$ and $\mathcal{C}(a) = \{c \in \mathcal{C} : (c, a) \in \Omega\}$ denote the set of actions which are measured for context $c$, and the set of contexts which are measured for action $a$, respectively. These notations are extended to sets of contexts and actions, i.e., $\mathcal{A}(\mathcal{C}) = \cap_{c \in \mathcal{C}} \mathcal{A}(c)$ and $\mathcal{C}(\mathcal{A}) = \cap_{a \in \mathcal{A}} \mathcal{C}(a)$.

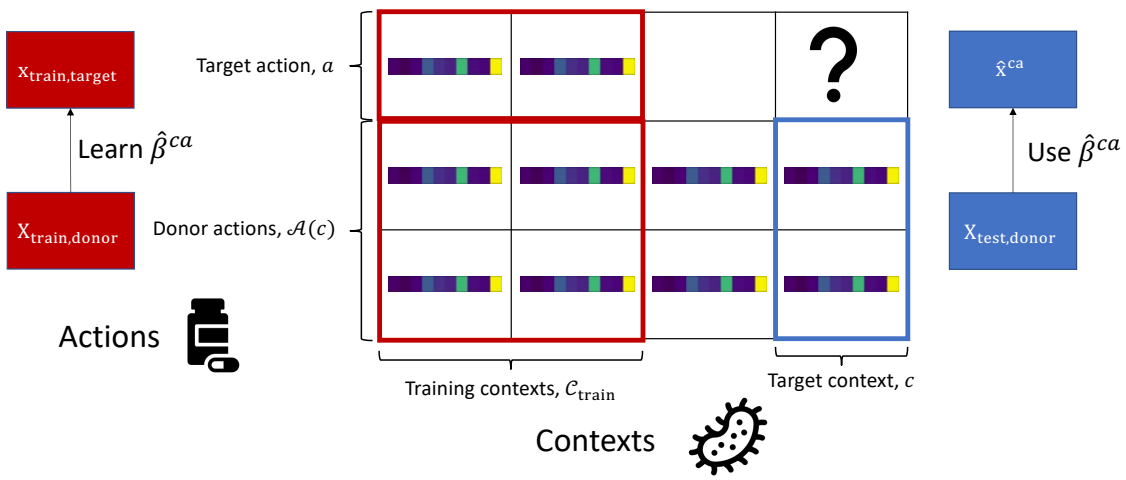



Figure 3.1: **The SI-Action Method.** To form an estimate $\hat{x}^{ca}$ of the outcome of target action $a$ in target context $c$, we find a set of *donor actions* $\mathcal{A}(c)$ which are available in context $c$. Then, we find *training contexts* $\mathcal{C}_{\text{train}}$ for which both the donor actions and the target action are available, and learn weights $\hat{\beta}^{ca}$ via linear regression. Finally, we use these weights to form $\hat{x}^{ca}$.

## 3.4 Algorithm

We propose `SI-A`, which is an extension of the SI estimator introduced in [ASS20], to handle more general sparsity patterns. Though our estimation strategy is applicable for any context-action pair of interest, we consider (without loss of generality) some $(c, a) \notin \Omega$. Below, we introduce additional notations to describe our estimator.

*Additional Notations.* We call $\mathcal{A}(c)$ the "donor actions" for context $c$. Let $\mathcal{C}_{\text{train}} = \mathcal{C}(\mathcal{A}(c) \cup \{a\})$ denote the set of contexts for which all donor actions and the target action are observed, we call these the "training contexts". Let $C = |\mathcal{C}_{\text{train}}|$ and $A = \mathcal{A}(c)$, and define $\tilde{\mathbf{x}}_{\text{train,target}} \in \mathbb{R}^{p \cdot C}$, $\tilde{\mathbf{X}}_{\text{train,donor}} \in \mathbb{R}^{pC \times A}$ and $\tilde{\mathbf{X}}_{\text{test,donor}} \in \mathbb{R}^{p \times A}$ as

$$\tilde{\mathbf{x}}_{\text{train,target}} = [\tilde{\mathbf{x}}^{ia}]_{i \in \mathcal{C}_{\text{train}}} \qquad \tilde{\mathbf{X}}_{\text{train,donor}} = [\tilde{\mathbf{x}}^{ij}]_{i \in \mathcal{C}_{\text{train}}, j \in \mathcal{A}(c)} \qquad \tilde{\mathbf{X}}_{\text{test,donor}} = [\tilde{\mathbf{x}}^{cj}]_{j \in \mathcal{A}(c)}$$

Finally, let $\texttt{ME} : \mathbb{R}^{n \times p} \to \mathbb{R}^{n \times p}$ denote any matrix estimation method, used to recover a matrix from noisy observations, such as nuclear norm regularization or singular value thresholding.

**Causal imputation for a novel context-action pair.** We define the our estimation method as follows, where † denotes the pseudoinverse. The method is also presented visually in Figure 3.1.

1. Model learning via regression
   (a) Define $\widehat{\mathbf{X}}_{\text{train,donor}} = \texttt{ME}(\tilde{\mathbf{X}}_{\text{train,donor}})$.
   (b) Define $\widehat{\beta}^{ca} = \widehat{\mathbf{X}}^{\dagger}_{\text{train,donor}} \tilde{\mathbf{x}}_{\text{train,target}} \in \mathbb{R}^{|\mathcal{A}(c)|}$
2. Causal imputation

(a) Define $\widehat{\mathbf{X}}_{\text{test,donor}} = \texttt{ME}(\tilde{\mathbf{X}}_{\text{test,donor}})$.

(b) Define $\widehat{\mathbf{x}}^{ca} = \widehat{\mathbf{X}}_{\text{test,donor}}\widehat{\beta}^{ca}$.

In Appendix B.1, we discuss how our estimator relates to the original SI estimator.

**Extensions.** We now discuss two natural modifications to the proposed method.

*Reversing the axis of regression.* The method we described learns weights $\widehat{\beta}^{ca}$ which express the action $a$ as a linear combination of other actions $\mathcal{A}(c)$. The same method can be applied to instead express the context $c$ as a linear combination of other contexts, by making the obvious replacements.

*Using both estimators in tandem.* The two estimation strategies described above—weighting actions and weighting contexts—can be used in a complementary fashion. For instance, say we apply the weighting actions estimator first. The imputed values increase the set of donor actions, which should lead to better outcomes when applying the weighting contexts estimator. An analogous argument can be made when the weighting contexts estimator is applied first. This suggests an algorithm that iteratively applies each approach until the imputed tensor converges to a desired threshold. However, one drawback of this approach is that it is computationally more demanding. Formally motivating and analyzing such an algorithm remains to be interesting future work.

*Arbitrary sets of donor actions.* For simplicity, we have written our estimator using *all* actions $\mathcal{A}(c)$ that are available for context $c$ as "donor" actions. The corresponding set of training contexts are those for which each of these actions, along with the target action, is measured, i.e., $\mathcal{C}_{\text{train}} = \mathcal{C}(\mathcal{A}(c) \cup \{a\})$. Expressing $\widehat{\beta}$ as a linear combination of all actions $\mathcal{A}(c)$ may be suboptimal if it leads to a significantly smaller set $\mathcal{C}_{\text{train}}$. For example, say $|\mathcal{A}(c)| = 11$, and there is only one training context containing all of these actions, i.e., $|\mathcal{C}(\mathcal{A}(c) \cup \{a\})| = 1$. Suppose there is a subset $\mathcal{A}' \subset \mathcal{A}(c)$, with $|\mathcal{A}'| = 10$ with $|\mathcal{C}(\mathcal{A}' \cup \{a\})| = 100$, i.e., by excluding a single action from the linear combination, we can train on 100 times more contexts. Thus, we may consider the SI estimator induced by a specific *donor set* $\mathcal{A}_{\text{donor}} \subseteq \mathcal{A}(c)$, with the corresponding set of training contexts being $\mathcal{C}_{\text{train}} = \mathcal{C}(\mathcal{A}_{\text{donor}} \cup \{a\})$. The choice of donor set introduces a tradeoff: as the number of donor actions increases, the number of training contexts might decrease. This raises the question of how to pick an optimal set of donor actions for a given prediction. We provide a principled approach to this problem in Section 3.5.

## 3.5 Theoretical results

In this section, we justify our algorithmic approach under a factor model and provide a justification for the factor model through the lens of causal structural models. We focus on the noiseless setting, i.e., for all $(c, a) \in \Omega$, we observe $\tilde{\mathbf{x}}^{ca} = \mathbf{x}^{ca}$, and thus, we may write $\mathbf{X}_{\text{train,donor}} = \widehat{\mathbf{X}}_{\text{train,donor}} = \tilde{\mathbf{X}}_{\text{train,donor}}$ and $\mathbf{X}_{\text{test,donor}} = \widehat{\mathbf{X}}_{\text{test,donor}} = \tilde{\mathbf{X}}_{\text{test,donor}}$. Hence, our results should be viewed as one of "identification" as we prove that $\mathcal{X}$ can be identified from noiseless data via our proposed algorithm. Indeed, identification arguments are considered a critical first step in causal inference [SP08; LCB20a], as they are necessary for any analysis of consistency in the noisy setting. Two noisy settings are of special interest. First, if only a single observation for each $(c, a)$ pair is available, then there is a rich body of literature that establishes formal guarantees for recovering $\mathbf{x}^{ca}$ for all $(c, a) \in \Omega$ using a suite of matrix

estimation methods (see [Cha15] and references therein). Second, if multiple observations are available for each $(c, a)$ pair, then we may average these observations, as we do in Section 3.6. Under appropriate conditions on the noise $\mathbf{e}^{c,a}$, such as bounded variance in the additive case, this averaging yields a consistent estimator of $\mathbf{x}^{ca}$. Formally building upon these results to provide rigorous sensitivity analyses remains future work.

### 3.5.1 Identification via synthetic interventions

In what follows, we state our key modeling assumptions that will enable our estimator to impute $\mathcal{X}$. To reduce redundancy, the following discussion will be restricted to `SI-A`, i.e., where we learn a regression model across actions. Analogous statements apply to the case of learning across contexts.

First, we recall the definition of a linear factor model, also known as the interactive fixed effects model, which is prevalent within the causal inference literature [ASS20; ADH10; Bai09; Ath+21] and critical to the SI framework of [ASS20].

**Definition 2.** *We say that $\mathcal{X}$ satisfies a* linear factor model *if, for any $a \in \mathcal{A}$ and $c \in \mathcal{C}$, $\mathbf{x}^{ca} = U^c \mathbf{v}^a$, where $U^c \in \mathbb{R}^{p \times r}$ and $\mathbf{v}^a$ are latent factors associated with the context $c$ and action $a$, respectively.*

*Interpretation.* Definition 2 posits that $\mathcal{X}$ satisfies a low-rank assumption [KB09][1]. While *exact* adherence to a linear factor model can be a strong assumption, it has been widely observed in practice that many big-data matrices are approximately low-rank. This phenomenon that has recently been motivated by generic latent-variable models [UT19] and theoretically established in several works [Cha15; Aga+19]. Since big data matrices with underlying structure are common in biological applications (due, for example, to similarities between cell types and drugs, and interdependence of gene expression values), these works suggest linear factor models as a natural starting point for developing principled causal imputation methods. Further, low-rank approximation can often be empirically verified by inspecting the spectrum of the observations, as we do in Section 3.6 (see Figure 3.5a).

**Assumption 4.** *Given a target context $c$ and target action $a$, there exists $\beta \in \mathbb{R}^{|\mathcal{A}(c)|}$ such that $\mathbf{v}^a = \sum_{j \in \mathcal{A}(c)} \beta_j \mathbf{v}^j$.*

*Interpretation.* Under the linear factor model, this is a mild assumption, particularly in applications such as genomics, where outcomes are measured for many different actions and contexts. By definition, such a $\beta$ exists if $\mathbf{v}^a$ and $[\mathbf{v}^j]_{j \in \mathcal{A}(c)}$ are linearly dependent. Recall $\mathbf{v}^a$ and $\mathbf{v}^j$ are in $\mathbb{R}^r$. If $r \ll |\mathcal{A}(c)|$, then by the definition of rank, it is easy to see that the 'pathological' case where $\mathbf{v}^a$ is linearly independent of $\{\mathbf{v}^j : j \in \mathcal{A}(c)\}$ is unlikely to hold; that is, in the worst case, this undesirable event occurs for at most $r$ actions out of all possible $\mathcal{A}$, which is a small fraction if $r \ll |\mathcal{A}(c)| \leq |\mathcal{A}|$.

**Assumption 5.** *Given a target context $c$ and target action $a$, let* $\text{rowspan}(\mathbf{X}_{test,donor})$ *be a subset of* $\text{rowspan}(\mathbf{X}_{train,donor})$,[2] *where $\mathbf{X}_{test,donor}$ and $\mathbf{X}_{train,donor}$ are defined in Section 3.4.*

[1] Formally, $\mathcal{X} = U \times_2 V$ for some $U \in \mathbb{R}^{|\mathcal{C}| \times r \times p}, V \in \mathbb{R}^{|\mathcal{A}| \times r}$, where $\times_2$ denotes the *mode-2* product.

[2] The row span of a matrix $M \in \mathbb{R}^{m \times n}$ is the linear subspace of $\mathbb{R}^n$ spanned by the row vectors of $M$.

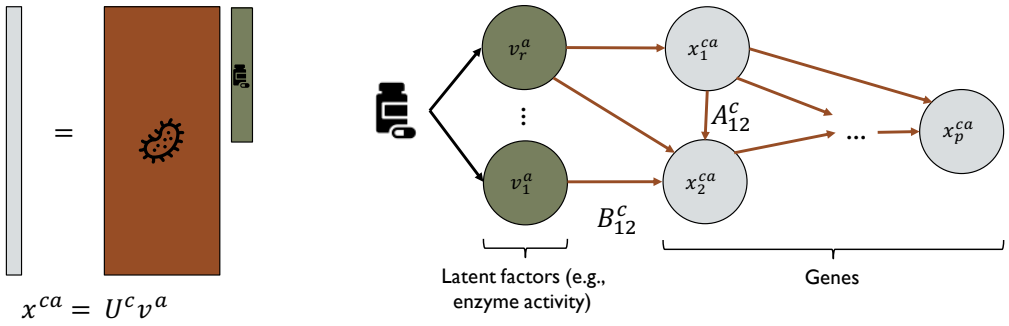



Figure 3.2: The assumed linear factor model and a linear structural equation model giving rise to it.

*Interpretation.* This assumption states that the target context is no more "complex" than the training context in a linear algebraic sense. This is the key assumption that enables `SI-A` (and thus the original SI estimator) to generalize from a small set of experiments to novel context-actions pairs.

**Theorem 4.** *Suppose $\mathcal{X}$ satisfies a linear factor model. Further, for $(c, a) \notin \Omega$, suppose Assumptions 4 and 5 hold. Then $\widehat{\mathbf{x}}^{ca} = \mathbf{x}^{ca}$.*

**Checking identifiability.** Assumption 5 enables SI to recover $\mathcal{X}$ from a sparse set of observations $\Omega$ (Theorem 4). As such, [ASS20] proposed a data-driven hypothesis test to check when this condition is satisfied in practice. Their test statistic $\widehat{\tau} = \|\mathbf{v}_{\text{test}} - \mathbf{v}_{\text{train}}\mathbf{v}_{\text{train}}^T\mathbf{v}_{\text{test}}\|_F^2$ measures the gap between the rowspaces of $\mathbf{X}_{\text{train,donor}}$ and $\mathbf{X}_{\text{test,donor}}$, where $\mathbf{v}_{\text{train}}$ and $\mathbf{v}_{\text{test}}$ are the right singular vectors of $\mathbf{X}_{\text{train,donor}}$ and $\mathbf{X}_{\text{test,donor}}$, respectively. [ASS20] derived a critical value $\tau_\alpha$, such that under the null hypothesis $H_0$ where Assumption 5 holds, $\mathbb{P}(\widehat{\tau} \geq \tau_\alpha \mid H_0) \leq \alpha$. However, their critical value depends on the underlying parameters of an underlying noise distribution. Instead of estimating these parameters, we follow a heuristic based on the interpretation of $\widehat{\tau}$ as the spectral energy of $\mathbf{v}_{\text{test}}$ that does *not* belong within $\text{span}(\mathbf{v}_{\text{train}})$. Specifically, we fix $\rho \in [0, 1]$ and reject $H_0$ if $\widehat{\tau} \geq \rho \cdot \text{rank}(\mathbf{v}_{\text{test}})$, i.e., if more than $\rho$ fraction of the spectral energy in $\mathbf{v}_{\text{test}}$ lies outside of $\text{span}(\mathbf{v}_{\text{train}})$.

Rejection of the test implies that `SI-A` may not perform well on our prediction task, since Assumption 5 is unlikely to hold. In practice, this allows us to switch to a simpler baseline estimator, as we show in Section 3.6. This hypothesis test also suggests an elegant method for picking a set of donor actions, as discussed in Section 3.4. For a fixed significance level, we may aim to find a set passing the hypothesis test which *maximizes* the number of training contexts. Unfortunately, this induces a combinatorial optimization problem which may be difficult to solve when there are many actions. We consider two computationally efficient alternatives. First, we may greedily pick actions, according to whichever action *least* reduces the number of training contexts, until the set passes the hypothesis test. Second, we may *always* use $\mathcal{A}(c)$ as the donor set, then use the hypothesis test to decide on whether to use the `SI-A` estimate or a simpler baseline.

### 3.5.2 Connecting causal factor models to graphical causal models

Here, we provide motivation for a linear factor model in terms of linear structural causal models, which have been frequently used as models of genomic networks [Fri+00; BF+19]. In particular, we consider the following set of models:

**Definition 3.** *A (noiseless)* linear structural equation model *(SEM) over the vector* $\mathbf{z} \in \mathbb{R}^p$ *is defined by the set of equations*

$$\mathbf{z}_i = \sum_{j \in pa_G(i)} A_{ij} \mathbf{z}_j,$$

*where* $pa_G(i)$ *denotes the* parents *of node* $i$ *in the directed acyclic graph* $G$*.*

A particular subclass of linear structural equation models implies a factor model. In particular, we consider the following assumption on the SEM, illustrated in Figure 3.2.

**Assumption 6.** *Let* $\mathbf{z}^{ca} = (\mathbf{x}^{ca}, \mathbf{v}^a)$*, where* $\mathbf{v}^a$ *depends only on* $a$ *and not on* $c$*. Assume that, for all contexts* $c$*, there exists* $A^c \in \mathbb{R}^{p \times p}$ *and* $B^c \in \mathbb{R}^{p \times r}$ *such that, for all actions* $a$*,* $\mathbf{z}^{ca}$ *satisfies the following linear structural equation model:*

$$\mathbf{x}^{ca} = A^c \mathbf{x}^{ca} + B^c \mathbf{v}^a \tag{3.1}$$

*Further assume that and* $\mathbf{v}_i$ *is unobserved for all* $i \in [p]$*.*

**Proposition 1.** *Under Assumption 6,* $\mathcal{X}$ *satisfies a linear factor model.*

*Proof.* Under acyclicity of $G$, $A^c$ is lower-triangular up to a permutation, so that $(I - A^c)$ is invertible, and we may re-write $\mathbf{x}^{ca}$ as

$$\mathbf{x}^{ca} = \underbrace{(I - A^c)^{-1} B^c}_{U^c} \mathbf{v}^a$$

Defining $U^c$ as $(I - A^c)^{-1} B^c$ completes the proof. □

*Interpretation.* Although Proposition 1 is a simple observation, to the best of our knowledge, this connection has not been previously established. The SI framework of [ASS20], and panel data literature more broadly [ADH10; Ark+21; Ath+21; Bai09], often imposes a linear factor model structure on $\mathbf{x}^{ca}$. Proposition 1 provides a novel motivation for these models through the lens of structural causal models. In our application, $\mathbf{v}^a$ might represent biological factors, e.g., molecule concentrations or enzyme activity levels, while $U^c$ represents the "gene expression program" [PM18] run by cell type $c$.

## 3.6 Empirical Results

In this section, we perform extensive experimentation on causal imputation for the CMAP dataset.

**CMAP dataset.** [Sub+17] developed the L1000 assay, which allows for cost-effective measurement of the gene expression signatures. This assay measures the transcription levels of 978 "landmark" genes, picked via a data-driven approach based on their ability to recover information about the rest of the transcriptome. L1000 signatures have been measured from over 1,000,000 different samples, covering 71 different cell types and over 20,000 different chemical compounds. We randomly sample 100 of these compounds, along with the "control" compound, *DMSO*, to create a smaller, unbiased version of the dataset. For each cell-type, compound pair, we average all corresponding signatures. This gives a dataset of 519 gene expression signatures. A detailed evaluation of the dataset and a description of our preprocessing pipeline is described in Appendix B.3. The dataset can be accessed at https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE92742.

**Baseline algorithms.** Figure 3.3 shows an embedding of 11,185 gene expression vectors via UMAP [McI+18], colored by cell type. Clearly, most of the variation in the data is due to cell type rather than the compound applied to the cell. This is supported by Figure B.3 in Appendix B.4, where we see that most compound-induced expressions falls within the normal variation of the cell type; the additional variation due to compounds is minor. This suggests a natural *mean-over-actions* baseline

$$\widehat{\mathbf{x}}^{ca}_{\text{avg-a}} = \frac{1}{|\mathcal{A}(c)|} \sum_{a' \in \mathcal{A}(c)} \mathbf{x}^{ca'}. \tag{3.2}$$

Indeed, it is well-known that the effect of chemical compounds is much smaller than the effect of cell type, so that the simple mean-over-actions estimator already performs well, as we examine shortly. We consider three other natural baselines (see equations in Appendix B.5). The *mean-over-contexts* is analogous to the mean-over-actions estimator, with the average taken over different contexts from the same action. We may combine these estimators into the parametric *two-way mean* estimator with parameter $\lambda_c \in [0, 1]$, by taking a convex combination of the *mean-over-actions* and *mean-over-contexts* estimators. We use $\lambda_c = 0.5$. The *fixed action effect estimator*, discussed in Section 3.2, is defined by computing the average shift induced by action $a$ compared to the control, and adding that shift to the control for $c$. Finally, we compare against a variant of `SI-A` that re-scales contexts rather than actions, as discussed in Section 3.4, which we refer to as `SI-C`. Note that standard imputation methods are not scalable to the CMAP dataset: the MissForest implementation in `missingpy` takes 2.5 hours per prediction, and would take $2.5 \times 519/24 \approx 54$ days to run on the subsampled data. `IterativeImputer`, a version of MICE in `sklearn`, is somewhat more scalable, but still prohibitively slow on our subsampled data—see Appendix B.6 for a comparison on 120 signatures.

### 3.6.1 Prediction Error

For each algorithm, we measure performance using a leave-one-out (LOO) procedure. In particular, for each cell type and compound pair $(c, a) \in \Omega$ that is measured, we remove the true gene expression signature $\mathbf{x}^{ca}$ from the dataset, and use the remainder of the dataset to estimate $\widehat{\mathbf{x}}^{ca}$. We measure the accuracy of each estimate $\widehat{\mathbf{x}}^{ca}$ in two ways. Normalized root-mean-square error (NRMSE), $\|(\widehat{\mathbf{x}}^{ca} - \mathbf{x}^{ca})/p\|_2 / \text{IQR}(\mathbf{x}^{ca})$, reported in Figure 3.4, measures

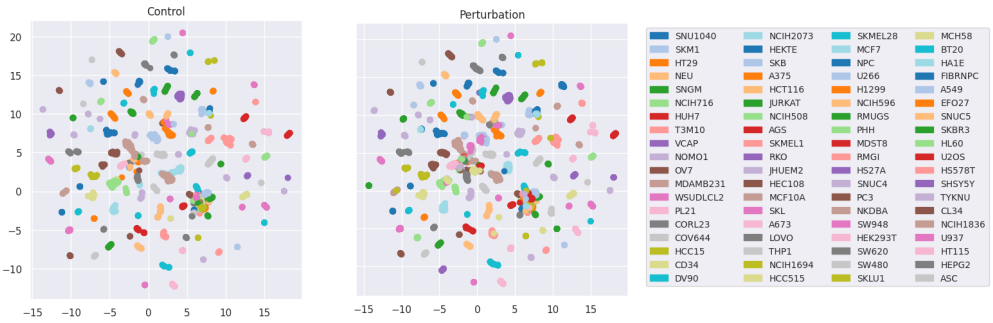



Figure 3.3: UMAP embedding of gene expression data, colored by cell type.

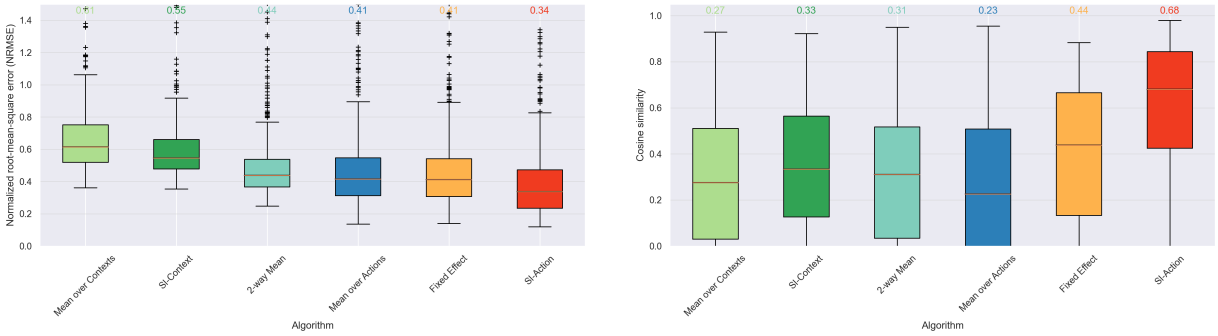
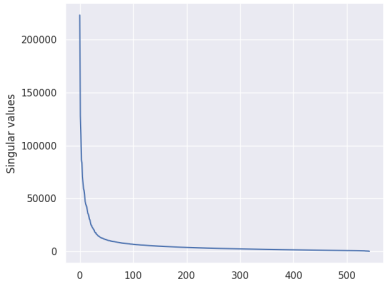



Figure 3.4: `SI-A` outperforms baselines on causal imputation in the CMAP dataset in terms of NRMSE (**left**) and cosine similarity (**right**) between the true and predicted effects of drugs in each cell type.

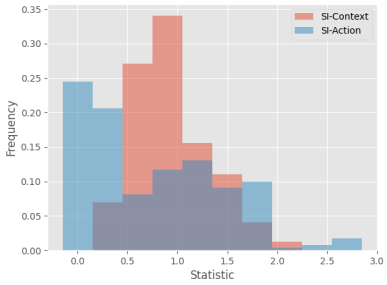
the absolute accuracy to predict the missing gene expression vector[3]. We show NRMSE since it is insensitive to scaling and shifting, but the relative ordering between methods is insensitive to the metric, as seen in Appendix B.6. Meanwhile, cosine similarity between the predicted shift $\widehat{\mathbf{x}}^{ca} - \mathbf{x}^{ca_0}$ and the true shift $\mathbf{x}^{ca} - \mathbf{x}^{ca_0}$ (where $a_0$ is the control action), reported in Figure 3.4, measures the degree of *alignment* between the true effect of drug $a$ in cell type $c$ and the predicted effect. Appendix B.6 shows that `SI-A` and the baselines are roughly similar in terms of computation time.

The results of each estimator provide insight into the relationship between cell types and compounds. As expected from Figure 3.3, the mean-over-actions estimator is a strong baseline, with a median NRMSE of 0.41. In contrast, the mean-over-contexts estimator performs quite poorly. Seeing Figure 3.3, this is expected: the substantial variation *between* different cell types suggests that each cell type is *not* representable as a linear combination of the others. The fixed effect estimator, using $a' = \text{DMSO}$, performs similarly to the mean-over-actions estimator, indicating its prediction quality is dominated by the cell-type-specific baseline $\mathbf{x}^{ca'}$, i.e., the shift vector for the compound is small. Finally, `SI-A` performs best, with a median NRMSE of 0.34, a 17% reduction compared to the best baseline. The findings on cosine similarity corroborate these results: the drug effects predicted by `SI-A` are significantly more aligned with the true drug effects compared to the baselines.

[3] $\text{IQR}(\mathbf{x}^{ca})$ denotes the *interquartile range* for the vector $\mathbf{x}^{ca}$.




(a) Spectrum of subsampled dataset. 95% of spectral energy is captured in the top 53 singular values.




(b) Distribution of $\widehat{\tau}$ when regressing along cell type (context) and compound (action) dimensions.


(c) NRMSE as a function of donor size and training contexts. Error bars show 0.1 standard deviations.

Figure 3.5: Explaining efficacy of `SI-A` for CMAP dataset.

`SI-A` takes a weighted linear combination of signatures for a given cell type (across different compounds), in contrast to *mean-over-actions* baseline which simply takes equal weights across these signatures. It is noteworthy that such weights are learnt from signatures across compounds for *different* cell types, and yet it is still effective to *transfer* the learnt model to other cell types. This gives credence to the factor model laid out in Section 3.5.2 holding in our setting. In contrast, `SI-C` performs almost as poorly as the mean-over-contexts baseline, indicating again that the difference between cell types is too great to express one cell type as a linear combination of others.

Notably, we highlight that we do *not* de-noise the data in `SI-A` or `SI-C` to achieve their optimal empirical results, i.e., the results in Figure 3.4 do not use any `ME` algorithm to remove high amounts of noise that might be present. De-noising is not needed in our setting as the data is implicitly de-noised, since each measurement is an average of multiple observations for a given measured cell type and compound pair $(c, a) \in \Omega$ in the data. Indeed, as shown in Appendix B.7, applying `ME` does improve prediction if we restrict ourselves to use only a single sample per observed $(c, a) \in \Omega$.

**`SI-A` is effective in biological settings.** As discussed in Section 3.5, a visual inspection of the spectrum of our observations helps verify the validity of the linear factor model and (by extension) Assumption 4. Figure 3.5a shows the singular values of the $519 \times 978$ matrix of

gene expression signatures, where over 95% of the spectrum is captured by the top 53 singular values, i.e., $r \approx 53$; hence, the CMAP dataset exhibits the desired low-rank factor structure. To verify Assumption 5, we plot frequencies of the test statistic $\widehat{\tau}$ (detailed in Section 3.4) in Figure 3.5b. A higher value of $\widehat{\tau}$ indicates that Assumption 5 is less likely to hold, and hence the results of our method are less meaningful. We see that the test statistic tends to be lower when regressing in the action dimension, explaining the superior performance of `SI-A` as compared to `SI-C`. This suggests that `SI-A` may be a useful tool in other biological settings, where similar invariance patterns and data availability are common.

**`SI-A` takes advantage of data structure.** Finally, to understand how the performance of `SI-A` depends on the number of donor actions (i.e., $|\mathcal{A}(c)|$) and training contexts (i.e., $|\mathcal{C}_{\text{train}}|$), we restrict our attention to context-action pairs with at least 10 donor actions and at least 5 training contexts, resulting in 75 pairs. For each context-action pair $(c, a)$, and each $(i, j) \in \{1, \ldots, 10\} \times \{1, \ldots, 5\}$, we perform `SI-A` with $i$ random donor actions and $j$ random training contexts. In Figure 3.5c, we show the average NRMSE of the predictions over all 75 pairs, for each tuple $(i, j)$. As expected, increasing the number of donor actions leads to a large increase in performance, and increasing the number of training contexts also leads to a large increase in performance, especially when there are few donor actions. The original SI estimator always has $|\mathcal{C}_{\text{train}}| = 1$, but our results highlight the value of using *all* available data to learn the regression model, as is done by `SI-A`.

## 3.7 Discussion

In this chapter, we introduced the `SI-A` estimator, an extension of the SI estimator of [ASS20], for use on a task we call *causal imputation*: predicting the effect of an action across different contexts. We showed that the `SI-A` estimator provides valid estimates of unseen outcomes under a linear factor model, which we motivate via a connection to structural causal models. We demonstrated the superior performance of `SI-A` to other baselines on the task of causal imputation in the CMAP dataset, an important source of information for predicting the effect of various compounds on gene expression.

Several important directions are left open for future work, of which we cover only a few. First, the tradeoff between the number of donor actions and the number of training contexts raises the need for a principled method for picking "optimal" donor sets. One promising approach may be to frame this choice as a combinatorial optimization problem, where the objective function may be submodular under some assumptions on the problem structure. A related question is whether we can apply SI in a *sequential manner* to infer which samples are most informative to reduce sample complexity in an experimental design and/or active learning framework.

Another important direction for future work is on *nonlinear* methods to the causal imputation problem. Genomic data is known to exhibit highly nonlinear relationships, so that our model in Section 3.5.2 is only a coarse approximation. A straightforward nonlinear extension of our method would be to perform `SI-A` in a latent space learned by an autoencoder. Two concepts from this chapter are likely to be useful in the development and analysis of nonlinear methods. First, we demonstrated that it is beneficial to develop representations for each action which are invariant to the context in which they occur, allowing for the effect

of the action to be *transported* between contexts. Second, our mechanistic explanation in Section 3.5.2 for the success of `SI-A` may serve as a starting point for explaining the success of nonlinear methods.

# Chapter 4

# Causal representation learning from interventional data

Adapted from

*Linear causal disentanglement via interventions*
by Chandler Squires*, Anna Seigal*, Salil Bhate, and Caroline Uhler
(Cite: [Squ+23], * denotes equal contribution)

In the last chapter, we introduced the model class $\mathcal{M}_{\text{exogiv}}(\mathbf{X}, \mathbf{Z}, \mathcal{A}, \mathcal{C})$, where a model $M$ encodes each action $a \in \mathcal{A}$ as an intervention on some latent variables $\mathbf{Z}$. In this chapter, we further explore the idea of encoding actions as interventions on latent variables. In particular, given observed variables $\mathbf{X}$ and latent variables $\mathbf{Z}$, we introduce three model classes: the class $\mathcal{M}_{\text{crm-lin}}(\mathbf{X}, \mathbf{Z})$ of (doubly) linear causal representation models (CRMs), the class $\mathcal{I}_{\text{perf-1}}(\mathbf{Z})$ of single-node perfect interventions on $\mathbf{Z}$, and finally, the class $\mathcal{M}_{\text{crm-lin-iv}}(\mathbf{X}, K)$, in which each model consists of a CRM and $K$ interventions (see Equation (1.17)). We work with the population observable $\Theta^{M^\star}_{0:K}$, which is a list of the inverse covariance matrices (over $\mathbf{X}$) for the observational and interventional distributions of $M^\star$ (see Equation (1.18)). The main lesson of this chapter is the importance of interventional data for the identifiability of causal representation models. In the language of Section 1.3.3, CRMs are a fairly expressive model class; while prior works have established identifiability of the class by making the model class less expressive (by adding modeling restrictions such as those described in Table 4.1), we establish identifiability by using richer data, which may often be available in scientific applications.

**Reading this chapter** In this chapter, we primarily describe the model class $\mathcal{M}_{\text{crm-lin-iv}}(\mathbf{X}, K)$ in Assumption 7, though we separate the assumption that the interventions are perfect into Assumption 8, since we also obtain results for soft interventions. We give an expression for each $\Theta_k$ in Equation (4.2) in terms of matrices $B_k$ and $H$ that represent the two main parts of a (doubly linear) causal representation model. Finally, we state our main identifiability result in Theorem 5, along with an extra result for the case of soft interventions (where we show that the causal graph can be recovered up to *transitive closure*).

## 4.1 Introduction

The goal of representation learning is to find a description of data that is interpretable, useful for reasoning, and generalizable. Such a representation *disentangles* the data into conceptually distinct variables. Traditionally, conceptual distinctness of variables has meant statistical independence. This is the setting of independent component analysis [Com94]. However, human reasoning often involves variables that are not statistically independent. For example, the presence of a sink and the presence of a mirror in an image. It is therefore natural to generalize conceptual distinctness to variables that are *causally autonomous*; i.e., interventions can be performed on each variable separately. This motivates *causal disentanglement* [Yan+21b], the recovery of causally autonomous variables from data.

In this chapter, we study the identifiability of causal disentanglement; i.e., its uniqueness. We adopt a generative perspective, as in [BCV13; Mor+22]. We assume that the observed variables are generated in two steps. First, latent variables $Z$ are sampled from a distribution $\mathbb{P}(Z)$. Then, the observed variables $X$ are the image of the latent variables under a deterministic mixing function. We assume that the latent variables are generated according to a linear structural equation model and that the mixing function is an injective linear map. Recent work has studied identifiability of various settings in representation learning [Khe+20; AHB21]. A common assumption for identifiability is that variables are observed across multiple contexts, each affecting the latent distribution $\mathbb{P}(Z)$ but not the mixing function. In our setup, each context is either an *intervention* on a latent variable, or is *observational*, i.e., has no interventions. We use the same terminology for interventions as [SU22]. From most to least general, a *soft* intervention on $Z_i$ changes the dependency of $Z_i$ on its direct causes, a *perfect* intervention removes this dependency but allows for stochasticity of $Z_i$, and a *do*-intervention sets $Z_i$ to a deterministic value.

Our main result is that our linear causal disentanglement setup is identifiable if, in addition to an observational context, for each latent variable $Z_i$, there is a context where $Z_i$ is the intervened variable under a perfect intervention; see Section 4.3.2. This is a *sufficient* condition for identifiability. Furthermore, we show that the condition of at least one intervention per latent node is *necessary* in the worst case: if some latent node is not intervened in any context, then there exist latent causal representations that are not identifiable; see Section 4.3.3. Our focus in this chapter is on identifiability guarantees. Nonetheless, we convert our proofs into a method for causal disentanglement in the finite-sample setting. In Section 4.4, we apply the method to synthetic and semi-synthetic data and show that it recovers the generative model, and we compute a linear causal disentanglement on a single-cell RNA sequencing dataset.

### 4.1.1 Motivating example

Consider two latent variables $Z = (Z_1, Z_2)$. Assume that $X = (X_1, X_2)$ is observed in two contexts $k \in \{0, 1\}$, that $X = GZ$ in both contexts, and that in context $k$,

$$Z = A_k Z + \Omega_k^{1/2} \varepsilon \quad \text{for} \quad \varepsilon \sim \mathcal{N}(0, I).$$

Let

$$A_0 = A_1 = \begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix}, \quad \Omega_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$
$$\Omega_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1/4 \end{bmatrix}, \quad G = \begin{bmatrix} -2 & 2 \\ 2 & -1 \end{bmatrix}.$$

Context $k = 1$ is an intervention on $Z_2$ that changes its variance. The covariance of $X$ in contexts $k = 0$ and $k = 1$ are, respectively,

$$\Sigma_0 = \begin{bmatrix} 20 & -16 \\ -16 & 13 \end{bmatrix} \qquad \text{and} \qquad \Sigma_1 = \begin{bmatrix} 8 & -7 \\ -7 & 25/4 \end{bmatrix},$$

since $\Sigma_k = G(I - A_k)^{-1}\Omega_k(I - A_k)^{-\top}G^\top$. However, the following parameters give the same covariance matrices:

$$\widehat{A}_0 = \widehat{A}_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \widehat{\Omega}_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$
$$\widehat{\Omega}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1/4 \end{bmatrix}, \quad \widehat{G} = \begin{bmatrix} -2 & 4 \\ 2 & -3 \end{bmatrix}.$$

The second set of parameters imply independence of $Z_1$ and $Z_2$, since $(\widehat{A}_0)_{1,2} = 0$, whereas the original parameters imply non-independence since $(A_0)_{1,2} \neq 0$. This non-identifiability holds for generic $A_0$, $A_1$, $\Omega_0$, $\Omega_1$, and $G$, where $A_1$ comes from an intervention on $Z_2$, see Section 4.3.3. This non-identifiability extends to any number of latent variables $d$: we show that, in the worst case, non-identifiability holds when fewer than $d + 1$ contexts are observed.

### 4.1.2 Related work

The growing field of causal representation learning blends techniques from multiple lines of work. Chief among these are identifiable representation learning, causal structure learning, and latent DAG (directed acyclic graph) learning.

**Identifiable representation learning.** The identifiability of the linear independent component analysis (ICA) model was given in [Com94]. Identifiability of a nonlinear ICA model is studied in [HST19; Khe+20], in the presence of auxiliary variables. The ICA model imposes the stringent condition that the latent variables are independent (in the linear setting) or conditionally independent given the auxiliary variables (in the nonlinear setting). Recent works on identifiable representation learning [AHB21; Zim+21] introduce structure on the data generating process to circumvent the independence condition. However, they do not consider latent variables that are causally related.

**Causal Structure Learning.** Causal structure is identifiable up to an *equivalence class* that depends on the available interventional data [VP90; HB12; YKU18; SWU20; Jab+20]. See [SU22] for a recent review. A key line of work [EGS05; HEH13] characterizes the interventions necessary and sufficient to ensure that the causal structure is fully identifiable; i.e., that the equivalence class is of size one. In particular, [EGS05] showed that $d - 1$ interventions are in the worst case necessary to fully identify a causal DAG model on $d$ nodes. The current chapter extends this line of work to DAG models over *latent* variables.

| | Setting | Graphical Conditions | Identification Result |
|---|---|---|---|
| [Sil+06] | **LNG** | All children pure | ID'd up to Markov equivalence |
| [HHS15] | **Dis** | 1 pure child per latent | ID'd up to Markov equivalence |
| [Cai+19] | **LNG** | 2 pure children per latent | Fully identified |
| [Xie+20] | **LNG** | 2 pure children per latent | Fully identified |
| [Xie+22] | **LNG** | 2 pure children per latent* | Fully identified |
| [Kiv+21] | **Dis** | No twins | ID'd up to Markov equivalence |
| [Liu+22] | **LG** | None | Fully identified from $2\|V\|$ perfect interventions if $\|E\| \leq \|V\|$ |
| [Ahu+23] | **Poly** | None | Fully identified from $\|V\|$ do-interventions |
| This paper | **LNG** or **LG** | None | Fully identified from $\|V\|$ perfect interventions |

Table 4.1: **Settings from prior works on learning latent DAG models. LNG** is short for *linear non-Gaussian*, **LG** for *linear Gaussian*, **Dis** for **discrete**, and **Poly** for *polynomial mixing*. In *, pure children are allowed to be latent. The number of nodes and the number of edges in the latent graph are denoted $|V|$ and $|E|$, respectively.

**Learning latent DAG models.** The task of learning a DAG over latent variables dates back at least to [Sil+06]. They introduced the notion of a *pure child*: an observed variable $X_i$ with only one latent parent, such $X_i$ is also called an *anchor* [HHS15; Sae+20]. The method of [Sil+06] requires that all observed variables are pure children. Recent works relax this assumption by studying the linear non-Gaussian setting, where all latent and observed variables are linear functions of their parents plus independent non-Gaussian noise. For example, [Cai+19] propose a method which learns a latent DAG under the assumption that each latent variable has at least two pure children. The pure child assumption can be extended to allow subsets of latent variables with the same observed children, as in [Xie+20], which introduces the Generalized Independent Noise condition. This condition was used by [Xie+22] to permit latent variables with no observed children; i.e., a hierarchical latent model.

Other works consider the discrete setting, requiring that the latent and observed variables are discrete [HHS15] or that the latent variables are discrete [Kiv+21]. The paper [Kiv+21] relaxes the pure child assumption, as follows. The children of node $Z_i$ are the variables with a directed edge from $Z_i$. The *no twins* assumption says that the observed children of any two latent nodes are distinct sets. A similar assumption called *strong non-redundancy* appears in [AHZ21], which considers models whose latent variables can be downstream of observed variables. See Appendix C.1. These works require sparsity in the map between latent and observed variables: they do not allow all observed variables to be children of all latent variables, which is the setting of the present chapter.

A number of recent works discard the sparsity requirement. [AHB22] and [Bre+22] learn

a latent DAG from paired counterfactual data. In contrast, we study *unpaired* data, which is more realistic in applications such as biology [Sta+20]. To the best of our knowledge, only two works consider unpaired data without sparsity assumptions. [Liu+22] study a linear Gaussian model over $d$ latent variables, a nonlinear mixing function, and vector-valued contexts. Their identifiability result only applies to our setting if the latent graph has at most as many edges as nodes, see Appendix C.11. In that case, their result implies that $2d$ interventions suffice for identifiability. We strengthen this result, showing that (i) $d$ interventions are sufficient and (ii) no restrictions on the latent graph are required. Moreover, we show that $d$ interventions are, in the worst case, necessary. Such necessary conditions do not appear in prior work on identifying latent DAGs. Finally, contemporaneous work [Ahu+23] shows that a latent DAG is identifiable from the more restricted class of do-interventions, but allow non-linear relationships. See Table 4.1 for a summary of prior work.

## 4.2 Setup

We consider $d$ latent variables $Z = (Z_1, \dots, Z_d)$, generated according to a linear structural equation model. We index contexts by $k \in \{0\} \cup [K]$, where $[K] := \{1, \dots, K\}$. The linear structural equation models in each context are related: context $k = 0$ is *observational* data, while contexts $k \in [K]$ are *interventional* data. We now state the assumptions for our model; see also Figure 2.1.

**Assumption 7.**

(a) ***Linear latent model**: Let $\mathcal{G}$ be a DAG with nodes ordered so that an edge $j \to i$ implies $j > i$. The latent variables $Z$ follow a linear structural equation model: in context $k$, the latent variables $Z$ satisfy*

$$Z = A_k Z + \Omega_k^{1/2} \varepsilon, \qquad Cov(\varepsilon) = I_d,$$

*where $I_d \in \mathbb{R}^{d \times d}$ is the identity matrix, $\Omega_k \in \mathbb{R}^{d \times d}$ is diagonal with positive entries, and $A_k \in \mathbb{R}^{d \times d}$ has $(A_k)_{ij} \neq 0$ if and only if there is an edge $j \to i$ in $\mathcal{G}$. That is, in context $k$,*

$$Z = B_k^{-1} \varepsilon, \quad \text{where} \quad B_k = \Omega_k^{-1/2}(I_d - A_k). \tag{4.1}$$

(b) ***Generic single-node interventions:** For each $k \in [K]$, there exists $i_k \in \{1, \dots, d\}$ such that*

$$B_k = B_0 + \mathbf{e}_{i_k} \mathbf{c}_k^\top,$$

*further, $(B_k)^\top \mathbf{e}_{i_k}$ is not a multiple of $(B_0)^\top \mathbf{e}_{i_k}$ unless $i_k$ has no parents in $\mathcal{G}$.*

(c) ***Linear observations:** Fix $p \geq d$. There is a full rank matrix $G \in \mathbb{R}^{p \times d}$ such that $X = GZ$ in every context $k$. Let $H := G^\dagger$ denote its Moore-Penrose pseudoinverse. We set the entry of largest absolute value in each row of $H$ to 1. If multiple entries in a row have same absolute value we set the leftmost entry to be positive.*

Our strongest results hold under one additional assumption.

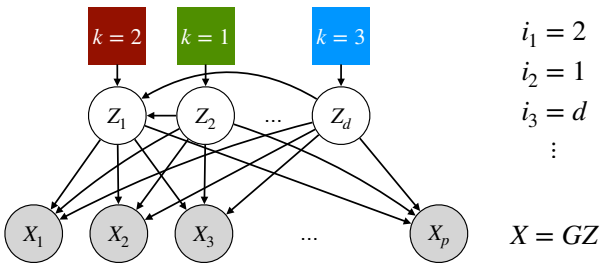



Figure 4.1: **The proposed setup.** The latent variables $Z = (Z_1, \ldots, Z_d)$ follows a linear DAG model, with contexts $k = 1, \ldots, K$ being single node interventions on targets $i_1, \ldots, i_K$. The observed variables $X = (X_1, \ldots, X_p)$ are an injective linear function of the latent variables $X = GZ$, where $G \in \mathbb{R}^{p \times d}$ does not vary across contexts.

**Assumption 8.** ***Perfect interventions:*** *For each $k \in [K]$, there exists $i_k \in \{1, \ldots, d\}$ such that*

$$B_k = B_0 + \mathbf{e}_{i_k} \mathbf{c}_k^\top,$$

*where $\mathbf{c}_k = \lambda_k \mathbf{e}_{i_k} - B_0^\top \mathbf{e}_{i_k}$ for some $\lambda_k > 0$.*

**Remark 3** (The parts of Assumption 7 that hold without loss of generality)**.** *Taking $\texttt{Var}(\varepsilon_i) = 1$ for all $i$ holds without loss of generality, since scaling can be absorbed into the matrix $\Omega_k$. A linear structural equation model is* causally sufficient *if $\varepsilon_i \perp\!\!\!\perp \varepsilon_j$ for all $i \neq j$. Thus, for a causally sufficient linear structural equation model, we have $Cov(\varepsilon) = I_d$ in Assumption 7(a) without loss of generality. The ordering of nodes in Assumption 7(a) is also without loss of generality: a permutation of the latent nodes can be absorbed into $G$. Our ordering makes the matrices $A_k$ upper triangular.*

*The scaling of $H$ in Assumption 7(c) is without loss of generality, as follows. If $\{B_k\}_{k=0}^K$ and $H$ satisfy Assumption 7 then $X = (B_k H)^\dagger \varepsilon$. Consider the matrices $\{B_k \Lambda\}_{k=0}^K$ and $\Lambda^{-1} H$, for $\Lambda$ diagonal with positive entries. Observe that $X' = (B_k \Lambda \Lambda^{-1} H)^\dagger \varepsilon$, has the same distribution as $X$ in context $k$. The alternative matrices satisfy Assumption 7, except for the scaling condition on $H$. Assumption 7(c) therefore fixes the scaling indeterminacy of each node.*

*The genericity condition in Assumption 7(b) automatically holds for perfect interventions. It fails to hold only for soft interventions that changes the variance but not the edge weights*[1]*. We show the importance of the genericity assumption for the identifiability of causal disentanglement in Appendix C.2.*

We give an example of a setting in which Assumption 7 might apply. Suppose $Z$ is the internal state of a cell (e.g., the concentrations of proteins, the locations of organelles, etc.) and that each context is an exposure to a different small molecule. Assumption 7(b) posits that each small molecule has a highly *selective* effect, modifying only one cellular mechanism. Assumption 8 posits that each small molecule completely disrupts the modified mechanism. While one does not expect all small molecules to be highly selective, one could filter based on selectivity.

In Appendix C.7, we describe a hypothesis test to test implications of Assumption 7(b), and show empirically that this test effectively determines model membership.

[1] i.e., $(\Omega_k)_{i_k, i_k} \neq (\Omega_0)_{i_k, i_k}$ and $(A_k)^\top \mathbf{e}_{i_k} = (A_0)^\top \mathbf{e}_{i_k}$

The covariance of $X$ in context $k$ is rank deficient when $d < p$, since $X = GZ$. We therefore define the precision matrix of $X$ in context $k$ to be the pseudoinverse of the covariance matrix, $\Theta_k := \mathbb{E}[XX^\top]^\dagger$. Then

$$\Theta_k = H^\top B_k^\top B_k H, \tag{4.2}$$

by Proposition 10 in Appendix C.3, since $\mathrm{Cov}_k(Z) = (B_k^\top B_k)^{-1}$.

We consider an unknown latent DAG. Each candidate DAG has unknown weights on its edges, unknown variances on its nodes, unknown new weights under each intervention, and an unknown mixing map to the observed variables. That is, our goal is to decompose the precision matrices $\{\Theta_k\}_{k=0}^K$ in Equation (4.2) to recover $H$ and $\{B_k\}_{k=0}^K$.

We recall some graph theoretic notions. The parents of node $i$ are $\mathrm{pa}_\mathcal{G}(i) = \{j \mid j \to i \text{ in } \mathcal{G}\}$, and we define $\overline{\mathrm{pa}}_\mathcal{G}(i) := \mathrm{pa}_\mathcal{G}(i) \cup \{i\}$. Similarly, $\mathrm{an}_\mathcal{G}(i)$ denotes the *ancestors* of $i$ in $\mathcal{G}$, the vertices $j$ with a directed path from $j$ to $i$. We define $\overline{\mathrm{an}}_\mathcal{G}(i) := \mathrm{an}_\mathcal{G}(i) \cup \{i\}$ and $\overline{\mathrm{an}}_\mathcal{G}(\mathcal{I}) := \cup_{i\in\mathcal{I}} \overline{\mathrm{an}}_\mathcal{G}(i)$. The source nodes of $\mathcal{G}$ are the nodes $i$ with $\mathrm{pa}_\mathcal{G}(i) = \varnothing$. We drop the subscript $\mathcal{G}$ when the graph is clear from context.

The *transitive closure* of $\mathcal{G}$, denoted $\overline{\mathcal{G}}$, is the DAG with $\mathrm{pa}_{\overline{\mathcal{G}}}(i) = \mathrm{an}_\mathcal{G}(i)$. Given a DAG $\mathcal{G}$, define the partial order $\prec_\mathcal{G}$ to be $i \prec_\mathcal{G} j$ if and only if $j \in \mathrm{an}_\mathcal{G}(i)$. Thus, the transitive closure is the graph with $j \to i$ whenever $i \prec_\mathcal{G} j$.

To decompose the precision matrices in Equation (4.2), we introduce a matrix decomposition defined on a partial order. Recall that the RQ decomposition writes $H \in \mathbb{R}^{d\times p}$ as $H = RQ$ for an upper triangular $R \in \mathbb{R}^{d\times d}$ and orthogonal $Q \in \mathbb{R}^{d\times p}$. We generalize the RQ decomposition[2].

**Definition 4** (The partial order RQ decomposition)**.** *Given a partial order $\prec$, the partial order RQ decomposition writes $H \in \mathbb{R}^{d\times p}$ as $H = RQ$, where $R \in \mathbb{R}^{d\times d}$ satisfies $R_{ii} \geq 0$ and $R_{ij} = 0$ unless $i \preceq j$, and where $\mathbf{q}_i$, the $i$-th row of $Q \in \mathbb{R}^{d\times p}$, is norm one and orthogonal to $\langle \mathbf{q}_j : i \prec j \rangle$.*

We recover the usual reduced RQ decomposition [TB97] when $\prec$ is the total order $1 < \cdots < d$. We construct the partial order RQ decomposition in Appendix C.4. Finally, given a positive definite matrix $M \in \mathbb{R}^{d\times d}$, the *Cholesky factor* $U \in \mathbb{R}^{d\times d}$, denoted $\textsc{cholesky}(M)$, is the unique upper triangular matrix with positive diagonal such that $M = U^\top U$.

## 4.3 Identifiability of causal disentanglement

We establish the sufficiency and worst case necessity of one intervention per latent node for identifiability of our causal disentanglement problem. The following result describes recovery of $\mathcal{G}$. Later, we discuss identifiability of the parameters in our setup. Since the labeling of latent nodes is unimportant, $\mathcal{G}$ is recovered if it is found up to relabeling.

**Theorem 5.** *Assume the setup in Assumption 7 with $d$ latent variables. Then $d$ interventions are sufficient and, in the worst case, necessary to recover $\overline{\mathcal{G}}$ from $\{\Theta_k\}_{k\in K}$. If Assumption 8 also holds, then $d$ interventions are sufficient and, in the worst case, necessary to recover $\mathcal{G}$ from $\{\Theta_k\}_{k\in K}$.*

[2]The RQ decomposition is used here (rather than the more familiar QR decomposition) because it gives an expression for the *rows* of $H$, which each correspond to one latent variable.

### 4.3.1 Preliminaries

We note the following basic fact, where $\mathbf{v}^{\otimes 2} := \mathbf{v}\mathbf{v}^\top$:

**Fact 1.** *Let $B \in \mathbb{R}^{d\times d}$. Then $B^\top B = \sum_{i=1}^d (B^\top \mathbf{e}_i)^{\otimes 2}$.*

We give a proof in Appendix C.6. This fact gives the key identity that drives our identifiability results.

**Proposition 2.** *Consider the setup in Assumption 7. Then, for any $k \in [K]$,*

$$\Theta_k - \Theta_0 = (H^\top B_k^\top \mathbf{e}_{i_k})^{\otimes 2} - (H^\top B_0^\top \mathbf{e}_{i_k})^{\otimes 2}.$$

*In particular the rank of the difference $\Theta_k - \Theta_0$ is 1 if $i_k$ is a source node in $\mathcal{G}$, and 2 otherwise.*

*Proof.* By Assumption 7, $B_k^\top \mathbf{e}_i = B_0^\top \mathbf{e}_i$ for all $i \neq i_k$. Using Fact 1, we have

$$B_k^\top B_k - B_0^\top B_0 = (B_k^\top \mathbf{e}_{i_k})^{\otimes 2} - (B_0^\top \mathbf{e}_{i_k})^{\otimes 2}.$$

Recall from Equation (4.2) that $\Theta_k = H^\top B_k^\top B_k H$. The result follows from left-multiplying both sides of the displayed equation by $H^\top$ and right-multiplying by $H$. This shows that $\operatorname{rank}(\Theta_k - \Theta_0) \leq 2$. For a source node, both vectors $B_k^\top \mathbf{e}_{i_k}$ and $B_0^\top \mathbf{e}_{i_k}$ have just one entry non-zero and $\operatorname{rank}(\Theta_k - \Theta_0) = 1$. Otherwise, the vectors have more than one entry non-zero and, by the genericity condition in Assumption 7(b), the difference $\Theta_k - \Theta_0$ has rank two. □

We can reduce a more general causal disentanglement problem to our setting, as we explain in Appendix C.6. First, we can count the latent dimension, since $\operatorname{rank}(\Theta_k) = d$ for any $k$. Second, we can identify which environments correspond to interventions on the same intervention target, see Proposition 13. Finally, we can identify which environment is observational using rank constraints, see Proposition 14. Thus, we assume without loss of generality that $d$ is known, that the observational environment is known, and that each node is only intervened on in one context.

### 4.3.2 Sufficiency

We define $S(\mathcal{G})$ to be the set of permutations on $d$ letters such that $\sigma(j) > \sigma(i)$ for all edges $j \to i$. For example, if $\mathcal{G}$ is a complete graph then $S(\mathcal{G})$ contains only the identity. If $\mathcal{G}$ has no edges then $S(\mathcal{G})$ is the group of permutations on $d$ letters. The permutation matrix corresponding to permutation $\sigma$ is $P_\sigma \in \mathbb{R}^{d\times d}$ with $(P_\sigma)_{ij} = \mathbb{1}_{\{i=\sigma(j)\}}$. Our main sufficiency result is the following.

**Theorem 6.** *Assume the set-up in Assumptions 7 and 8 with one intervention on each latent node. Then the graph $\mathcal{G}$, the intervention targets $i_k$, and the parameters are identifiable up to $S(\mathcal{G})$: given a solution $(B_0, \ldots, B_K, H)$, the set of solutions is*

$$\{(P_\sigma B_0 P_\sigma^\top, \ldots, P_\sigma B_K P_\sigma^\top, P_\sigma H) : \sigma \in S(\mathcal{G})\}.$$

Theorem 6 says that solutions to the causal disentanglement problem are unique up to permutations of the latent nodes that preserve the property that $j \to i$ implies $j > i$. First, we verify that each permutation in $S(\mathcal{G})$ gives a solution.

**Algorithm 5** ID-ANCESTORS

**Input:** $\Theta_k$, $\Theta_0$, $\{\widehat{\mathbf{q}}_i\}_{i\in\mathcal{I}}$
**Output:** Vector $\widehat{\mathbf{q}}_k$, ancestor set $\mathcal{A}$
Let $\mathcal{A} = \mathcal{I}$
**for** $i \in \mathcal{I}$ **do**
    Let $W_{\neg i} = \langle \widehat{\mathbf{q}}_i : j \in \mathcal{I} \setminus \{i\} \rangle$
    Let $V_{\neg i} = \mathrm{proj}_{W_i^\perp} \mathrm{rowspan}(\Theta_k - \Theta_0)$
    If $\dim(V_{\neg i}) = 1$, let $\mathcal{A} = \mathcal{A} \setminus \{i\}$
**end for**
Let $W = \langle \widehat{\mathbf{q}}_a : a \in \mathcal{A} \rangle$
Let $V = \mathrm{proj}_{W^\perp} \mathrm{rowspan}(\Theta_k - \Theta_0)$
Take $\widehat{\mathbf{q}}_k$ with first nonzero entry positive and $\|\widehat{\mathbf{q}}_k\|_2 = 1$, such that $\langle \widehat{\mathbf{q}}_k \rangle = V$
**return** $\widehat{\mathbf{q}}_k$, $\mathcal{A}$

**Algorithm 6** ID-PARTIALORDER

**Input:** Precision matrices $(\Theta_0, \Theta_1, \ldots, \Theta_K)$, rank $d$
**Output:** Factor $\widehat{Q}$, partial order $\prec$
Let $\mathcal{I}_0 = \{\}$, $\widehat{Q} = \mathbf{0}_{d\times d}$
**for** $t = 1, \ldots K$ **do**
    Let $W_t = \langle \widehat{\mathbf{q}}_i : i \in \mathcal{I}_{t-1} \rangle$
    Let $V_k = \mathrm{proj}_{W_t^\perp} \mathrm{rowspan}(\Theta_k - \Theta_0)$ for $k \notin \mathcal{I}_{t-1}$
    Pick $k$ such that $\dim(V_k) = 1$
    Let $\widehat{\mathbf{q}}_k, \mathcal{A} = \text{ID-ANCESTORS}(\Theta_k, \Theta_0, \{\widehat{\mathbf{q}}_i\}_{i\in\mathcal{I}_{t-1}})$
    Add $a' \succ k$ for any $a' \succeq a$, $a \in \mathcal{A}$
    Let $\mathcal{I}_t = \mathcal{I}_{t-1} \cup \{k\}$, $\widehat{Q}_t = [\widehat{\mathbf{q}}_k; \widehat{Q}_{t-1}]$
**end for**
**return** $\widehat{Q}$, $\prec$

**Proposition 3.** *Assume the set-up in Assumption 7. Given a solution $(B_0, \ldots, B_K, H)$ to Equation (4.2) for $k \in \{0\} \cup [K]$, the matrices $(P_\sigma B_0 P_\sigma^\top, \ldots, P_\sigma B_K P_\sigma^\top, P_\sigma H)$ are a valid solution whenever $\sigma \in S(\mathcal{G})$.*

*Proof.* Let $\{B_k\}_{k=0}^K$ and $H$ satisfy Assumption 7 and Equation (4.2). Define $B_k^{(\sigma)} = P_\sigma B_k P_\sigma^\top$ and $H^{(\sigma)} = P_\sigma H$ for $\sigma \in S(\mathcal{G})$. Then $\Theta_k = {H^{(\sigma)}}^\top {B_k^{(\sigma)}}^\top B_k^{(\sigma)} H^{(\sigma)}$. The matrices $B_k^{(\sigma)}$ are upper triangular, as follows. For all $i, j \in [p]$, we have $(B_k^{(\sigma)})_{\sigma(i),\sigma(j)} = (B_k)_{ij}$. Hence $B_k^{(\sigma)}$ is upper triangular when $(B_k)_{ij} = 0$ for all $i, j \in [p]$ with $\sigma(i) > \sigma(j)$. This holds since $\sigma \in S(\mathcal{G})$. Moreover, these matrices also satisfy Assumption 7(b) with the intervention target $\sigma(i_k)$ in context $k$. Finally, $H^{(\sigma)}$ satisfies Assumption 7(c), since we just permute the rows of $H$. □

We give a constructive proof of Theorem 6 via an algorithm to recover $H$ and $\{B_k\}_{k=0}^K$ from $\{\Theta_k\}_{k=0}^K$[3]. The computational complexity of the algorithm is given in Appendix C.8.

[3]We only use the second moment of $X$. We do not use the first moment since we assume $\mathbb{E}[\varepsilon] = 0$, and we do not use higher moments since in the worst case (Gaussian noise), they contain no additional information.

The bulk of the algorithm is devoted to recovering $H$. First, we recover the partial order $\prec_{\mathcal{G}}$ (i.e., the DAG $\overline{\mathcal{G}}$), together with the matrix $Q$ from a partial order RQ decomposition of $H$, up to signs and permutations of rows, in Algorithm 6. The subroutine Algorithm 5 recovers the ancestors of a node and its corresponding row of $Q$. Then we recover $R$ in Algorithm 7. Having recovered $H$, the matrices $\{B_k\}_{k=0}^K$ are found via the Cholesky decomposition.

We show that Algorithm 6 returns $Q$ from the partial order RQ decomposition of $H$, up to a permutation $\sigma \in S(\mathcal{G})$ and a matrix in $\mathrm{Sig}_d$, the $d \times d$ diagonal matrices with diagonal entries $\pm 1$.

**Proposition 4.** *Assume the setup in Assumption 7, and that every latent node is intervened on; i.e, $\{i_k\}_{k=1}^K = [d]$. Let $(\widehat{Q}, \prec)$ be the output of Algorithm 6. Then $\prec$ is the partial order $\prec_{\mathcal{G}}$. Moreover, let $H = RQ$ be the partial order RQ decomposition of $H$ for the partial order $\prec_{\mathcal{G}}$. Then $\widehat{Q} = SP_\sigma Q$ for some $\sigma \in S(\mathcal{G})$ and $S \in \mathrm{Sig}_d$.*

The following lemma relates the partial order of $\mathcal{G}$ to the linear spaces $\mathrm{rowspan}(\Theta_k - \Theta_0)$.

**Lemma 1.** *Assume the setup in Assumption 7. Let $H = RQ$ be a partial order RQ decomposition of $H$. Let $i_k$ be the intervention target of context $k$ and let $\mathcal{I} \subseteq [d]$. Then*

*(a)* $\mathrm{rowspan}(\Theta_k - \Theta_0) \subseteq \langle \mathbf{h}_i : i \in \mathcal{I} \rangle$ *if and only if* $\overline{pa}(i_k) \subseteq \mathcal{I}$,

*(b)* $\mathrm{rowspan}(\Theta_k - \Theta_0) \subseteq \langle \mathbf{q}_i : i \in \overline{an}(i_k) \rangle$, *and*

*(c)* $\mathrm{rowspan}(\Theta_k - \Theta_0) \not\subseteq \langle \mathbf{q}_i : i \in \mathcal{I} \rangle$ *if* $\overline{pa}(i_k) \not\subseteq \mathcal{I}$.

*Proof.* Let $\mathbf{h}_i$ denote the $i$th row of $H$, and let $\boldsymbol{\alpha}_{j,i_k} := \sum_{i \in \overline{\mathrm{pa}}(i_k)} (B_j)_{i_k,i} \mathbf{h}_i$. By Proposition 2, we have

$$\mathrm{rowspan}(\Theta_k - \Theta_0) = \langle \boldsymbol{\alpha}_{k,i_k}, \boldsymbol{\alpha}_{0,i_k} \rangle . \tag{4.3}$$

Equation (4.3) shows that $\mathrm{rowspan}(\Theta_k - \Theta_0) \subset \langle \mathbf{h}_i : i \in \overline{\mathrm{pa}}(i_k) \rangle$. This linear space is contained in $\langle \mathbf{h}_i : i \in \mathcal{I} \rangle$ whenever $\overline{\mathrm{pa}}(i_k) \subseteq \mathcal{I}$. Conversely, assume there exists $j \in \overline{\mathrm{pa}}(i_k)$ with $j \notin \mathcal{I}$. Then, containment of $\mathrm{rowspan}(\Theta_k - \Theta_0)$ in $\langle \mathbf{h}_i : i \in \mathcal{I} \rangle$ cannot hold: containment implies $(B_0)_{i_k,j} \mathbf{h}_j \in \langle \mathbf{h}_i : i \in [p] \backslash \{j\} \rangle$, a contradiction since $H$ is full row rank and $(B_0)_{i_k,j} \neq 0$. Hence (a) holds.

By definition of the partial order RQ decomposition, we have $\mathbf{h}_i \in \langle \mathbf{q}_j \mid j \in \overline{\mathrm{an}}(i) \rangle$. Thus, by transitivity of the ancestorship relation, (b) holds. Conversely, assume there exists $j \in \overline{\mathrm{pa}}(i_k)$ with $j \notin \mathcal{I}$. Then $\mathrm{rowspan}(\Theta_k - \Theta_0) \subseteq \langle \mathbf{q}_i : i \in \mathcal{I} \rangle$ implies that

$$\mathbf{h}_j \in \langle \mathbf{q}_i : i \in \mathcal{I} \rangle + \langle \mathbf{h}_i : i \in \overline{\mathrm{pa}}(i_k) \setminus \{j\} \rangle,$$

since $(B_0)_{i_k,j} \neq 0$. We partition $\mathcal{I}$ into the descendants and non-descendants of $j$: let $\mathcal{I}_d := \{i \in \mathcal{I} : j \in \mathrm{an}(i)\}$ and let $\mathcal{I}_{nd} := \{i \in \mathcal{I} : j \notin \mathrm{an}(i)\}$. By definition of the partial order RQ decomposition, we have $\mathbf{q}_i \perp \mathbf{h}_j$ whenever $j \in \mathrm{an}(i)$. Thus $\mathbf{h}_j \in \langle \mathbf{q}_i : i \in \mathcal{I}_{nd} \rangle + \langle \mathbf{h}_i : i \in \overline{\mathrm{pa}}(i_k) \setminus \{j\} \rangle$. Inverting the partial order RQ decomposition gives $\mathbf{q}_i \in \langle \mathbf{h}_{i'} : i' \in \overline{\mathrm{an}}(i) \rangle$. Hence

$$\mathbf{h}_j \in \langle \mathbf{h}_i : i \in \overline{\mathrm{an}}(\mathcal{I}_{nd}) \cup (\overline{\mathrm{pa}}(i_k) \setminus \{j\}) \rangle,$$

a contradiction, since $j \notin \overline{\mathrm{an}}(\mathcal{I}_{nd})$ and $H$ is full rank. □

**Algorithm 7** IterativeDifferenceProjection

**Input:** Precision matrices $(\Theta_0, \Theta_1, \ldots, \Theta_K)$
**Output:** $\widehat{H}$, $(\widehat{B}_0, \widehat{B}_1, \ldots, \widehat{B}_K)$
Let $d = \operatorname{rank}(\Theta_0)$
Let $\widehat{Q}, \prec = \text{ID-PartialOrder}((\Theta_0, \Theta_1, \ldots, \Theta_K), d)$
Let $\widehat{C}_k = \text{cholesky}((\widehat{Q}^\dagger)^\top \Theta_k \widehat{Q}^\dagger)$ for $k = 0, \ldots, K$
Let $\widehat{R} = I_d$
**for** $k = 1, \ldots, K$ **do**
    Let $\widehat{D}_k = \widehat{C}_k - \widehat{C}_0$
    Let $\widehat{i}_k$ be index of the only nonzero row of $\widehat{D}_k$
    Let $\widehat{R}_{\widehat{i}_k} = (\widehat{D}_k)_{\widehat{i}_k} + (\widehat{C}_0)_{\widehat{i}_k}$
**end for**
Let $\widehat{H}' = \widehat{R}\widehat{Q}$
Let $\widehat{H} = \widehat{\Lambda}\widehat{H}'$, for $\widehat{\Lambda}$ diagonal such that $\widehat{H}$ satisfies the conditions on $H$ in Assumption 7(c)
Let $\widehat{B}_0 = \text{cholesky}((\widehat{H}^\dagger)^\top \Theta_0 \widehat{H}^\dagger)$
Let $\widehat{B}_k = \widehat{B}_0 + \mathbf{e}_{\widehat{i}_k} \left( |\widehat{\Lambda}_{\widehat{i}_k, \widehat{i}_k}| \mathbf{e}_{\widehat{i}_k} - \widehat{B}_0^\top \mathbf{e}_{\widehat{i}_k} \right)^\top$ for $k = 1, \ldots, K$
**return** $\widehat{H}$, $(\widehat{B}_0, \widehat{B}_1, \ldots, \widehat{B}_K)$

*Proof of Proposition 4.* Assume that $\widehat{Q}_{t-1}$ is the last $t-1$ rows of $SP_\sigma Q$, for some $\sigma \in S(\mathcal{G})$ and $S \in \text{Sig}_d$. Then $W_{t-1} = \langle \widehat{\mathbf{q}}_i : i \in \mathcal{I}_{t-1} \rangle$. At step $t$, we pick $k$ such that $V_k = \text{proj}_{W_t^\perp} \text{rowspan}(\Theta_k - \Theta_0)$ has dimension one. Lemma 1 implies that such $k$ are those with $\text{pa}(i_k) \subseteq \mathcal{I}_{t-1}$. Algorithm 5 returns a set $\mathcal{A}$ with $\text{pa}(i_k) \subseteq \mathcal{A} \subseteq \text{an}_\mathcal{G}(i_k)$. Thus Line 9 adds the relation $k \prec a'$ if and only if $a' \in \text{an}_\mathcal{G}(i_k)$. Hence $\prec$ is the partial order $\prec_\mathcal{G}$. Line 8 picks $\widehat{\mathbf{q}}_k$ orthogonal to $\{\mathbf{q}_i : i \succ_\mathcal{G} i_k\}$, such that $\text{rowspan}(\widehat{Q}_t)$ contains $\text{rowspan}(\Theta_k - \Theta_0)$. By Lemma 1 and Definition 4, this is $\pm \mathbf{q}_{i_k}$. Thus, $\widehat{Q}_t$ is equal to the last $t$ rows of $S' P_{\sigma'} Q$, for some $\sigma' \in S(\mathcal{G})$ and $S' \in \text{Sig}_d$. Repeating for $t = 1, \ldots, K$ gives the result. □

We prove Theorem 6 by proving the following result, which is its constructive analogue.

**Theorem 7.** *Assume the setup in Assumptions 7 and 8, and that every latent node is intervened; i.e., $\{i_k\}_{k=1}^K = [d]$. Let $\widehat{H}$ and $\{\widehat{B}_k\}_{k=0}^K$ be the output of Algorithm 7. Then $\widehat{H} = P_\sigma H$ and $\widehat{B}_k = P_\sigma B_k P_\sigma^\top$ for all $k$, for some $\sigma \in S(\mathcal{G})$.*

*Proof.* Let $H = RQ$. Then $\Theta_k = Q^\top R^\top B_k^\top B_k R Q$, by Equation (4.2), and $\widehat{Q} = S P_\sigma Q$ for some $\sigma \in S(\mathcal{G})$ and $S \in \text{Sig}_d$, by Proposition 4. Hence $(\widehat{Q}^\dagger)^\top \Theta_k \widehat{Q}^\dagger$ equals

$$S \left( P_\sigma R^\top P_\sigma^\top \right) \left( P_\sigma B_k^\top P_\sigma^\top \right) \left( P_\sigma B_k P_\sigma^\top \right) \left( P_\sigma R P_\sigma^\top \right) S.$$

Let $C_k^{(\sigma)} = S(P_\sigma B_k P_\sigma^\top)(P_\sigma R P_\sigma^\top)S$. The matrix $C_k^{(\sigma)}$ is upper triangular, since it is a product of four upper triangular matrices, by the definition of $S(\mathcal{G})$ and of the partial order RQ decomposition, where we use that $i \prec_\mathcal{G} j$ implies $i < j$. Moreover, $C_k^{(\sigma)}$ has positive diagonal, since the matrices $R$ and $B_k$ have positive diagonal, by Definition 4 and Assumption 7(a) respectively. Hence $\widehat{C}_k := C_k^{(\sigma)}$ is the Cholesky factor of $(\widehat{Q}^\dagger)^\top \Theta_k \widehat{Q}^\dagger$. The

differences $\widehat{D}_k := \widehat{C}_k - \widehat{C}_0$ equal $SP_\sigma(B_k - B_0)RP_\sigma^\top S = SP_\sigma \mathbf{e}_{i_k}\mathbf{c}_k^\top RP_\sigma^\top S$, by Assumption 7(b). The intervention target $\sigma(i_k)$ is the only nonzero row of $\widehat{D}_k$, i.e., $\widehat{i}_k = \sigma(i_k)$. Observe that $(\widehat{D}_k)_{\widehat{i}_k} + (\widehat{C}_0)_{\widehat{i}_k} = S_{\sigma(i_k),\sigma(i_k)}\lambda_k(RP_\sigma^\top)_{\sigma(i_k)}$. Thus, we have recovered $\widehat{R} = \widehat{\Lambda}P_\sigma RP_\sigma^\top$ for $\widehat{\Lambda}$ diagonal such that $\widehat{\Lambda}_{\widehat{i}_k,\widehat{i}_k} = \pm\lambda_k$. This gives $\widehat{H}' = \widehat{\Lambda}P_\sigma RP_\sigma^\top P_\sigma Q = \widehat{\Lambda}P_\sigma H$. The scaling in Line 13 recovers $\widehat{H} = P_\sigma H$ and $\widehat{\Lambda}$. We have $(\widehat{H}^\dagger)^\top\Theta_0\widehat{H}^\dagger = P_\sigma B_0^\top P_\sigma^\top P_\sigma B_0 P_\sigma^\top$, where $P_\sigma B_0 P_\sigma^\top$ is upper triangular, and thus we recover $\widehat{B}_0 = P_\sigma B_0 P_\sigma^\top$ from the Cholesky decomposition. Finally, since $|\widehat{\Lambda}_{\widehat{i}_k,\widehat{i}_k}| = \lambda_k$, Line 15 gives us $\widehat{B}_k = P_\sigma B_k P_\sigma^T$. □

Theorem 7 requires Assumption 8, see Appendix C.10. In Appendix C.11, we compare our identifiability condition to that of [Liu+22]. We show that [Liu+22] requires that the latent graph has fewer than $d$ edges. In contrast, our condition imposes no constraints on the latent graph.

### 4.3.3 Worst-case necessity

We show that one intervention per latent node is *necessary* for identifiability of our setup, in the worst case. It is clear that observational data does not suffice for identifiability: from just observational data, we always have a solution with *independent* latent variables, as follows. Let $\widehat{H} = \Lambda B_0 H$ and $\widehat{B}_0 = \Lambda^{-1}$, for $\Lambda$ a diagonal matrix with positive entries such that Assumption 7(c) holds. Then $\widehat{B}_0\widehat{H} = B_0 H$, i.e. $\widehat{B}_0$, $\widehat{H}$ solve the causal disentanglement problem. The new solution has independent latent nodes, since $\widehat{B}_0$ is diagonal.

The next result, which follows from prior work in causal structure learning, says that $d-1$ interventions are required in the worst case, for a fully observed model. This is the special case of our setup where $H$ is a permutation matrix.

**Proposition 5.** *Assume the setup in Assumptions 7 and 8, with $H$ a permutation matrix. Let $K < d-1$ with all intervention targets distinct. Then, in the worst case over intervention targets $\{i_k\}_{k=1}^K$, $B_0$ and $H$ are not identifiable.*

*Proof.* In the linear Gaussian setting with unknown-target interventions, the structure of a DAG is only identifiable up to its *interventional Markov equivalence class* (MEC), see e.g. Proposition 3.3(ii) of [CP22]. For single-node interventions, $d-1$ interventions are in the worst case necessary to ensure that the interventional MEC has size one, by Theorem 3.7 of [EGS05]. □

We show that $d$ interventions are necessary, in the worst case, when $H$ is a general matrix. The proof is in Appendix C.9.

**Proposition 6.** *Assume the setup in Assumptions 7 and 8, with $K < d$. Then there exist $H$ and $\{B_k\}_{k=0}^K$ and a set of $K$ distinct intervention targets such that (i) $H$ and $B_k$ are not identifiable up to $S(\mathcal{G})$ and (ii) $\prec_\mathcal{G}$ is not identifiable.*

**Example 1.** *Proposition 6 generalizes our motivating example from Section 4.1. Fix $H \in \mathbb{R}^{2\times 2}$ with entries $H_{ij}$, and fix upper triangular $B_0, B_1 \in \mathbb{R}^{2\times 2}$ with entries $(B_0)_{ij}$ and $(B_1)_{ij}$, respectively. Assume $i_1 = 2$; i.e., $(B_0)_{11} = (B_1)_{11}$ and $(B_0)_{12} = (B_1)_{12}$. Let*

$$\widehat{B}_0 = \begin{bmatrix} 1 & 0 \\ 0 & (B_0)_{22} \end{bmatrix}, \quad \widehat{B}_1 = \begin{bmatrix} 1 & 0 \\ 0 & (B_1)_{22} \end{bmatrix},$$
$$\widehat{H} = \begin{bmatrix} (B_0)_{11}H_{11} + (B_0)_{12}H_{21} & (B_0)_{11}H_{12} + (B_0)_{12}H_{22} \\ H_{21} & H_{22} \end{bmatrix}.$$

*Then for $k \in \{0, 1\}$, we have $\widehat{B}_k\widehat{H} = B_kH$, both equal to*

$$\begin{bmatrix} (B_0)_{11}H_{11} + (B_0)_{12}H_{21} & (B_0)_{11}H_{12} + (B_0)_{12}H_{22} \\ (B_k)_{22}H_{21} & (B_k)_{22}H_{22} \end{bmatrix}$$

**Remark 4.** *Proposition 6 pertains to the worst case over intervention targets. It is natural to ask if there exists a better choice of $K$ intervention targets, for $K < d$, such that $H$ and $\{B_k\}_{k=0}^K$ are identifiable. For example, when $d = 2$, consider $\mathcal{G} = \{2 \to 1\}$, with an intervention on $Z_1$; i.e., $i_1 = 1$. Then $\operatorname{rowspan}(\Theta_1 - \Theta_0) \subseteq \langle \mathbf{h}_i : i \in \mathcal{I} \rangle$ if and only if $\mathcal{I} = \{1, 2\}$, by Lemma 1(a). Thus, $\Theta_1 - \Theta_0$ is rank 2, and we can detect that $i_1 = 1$ and $\mathcal{G} = \{2 \to 1\}$; otherwise, we would have rank$(\Theta_1 - \Theta_0) = 1$. While $i_1$ and $\mathcal{G}$ are identifiable, preliminary computational evidence suggests that the entries of $B_0, B_1$, and $H$ are not identifiable.*

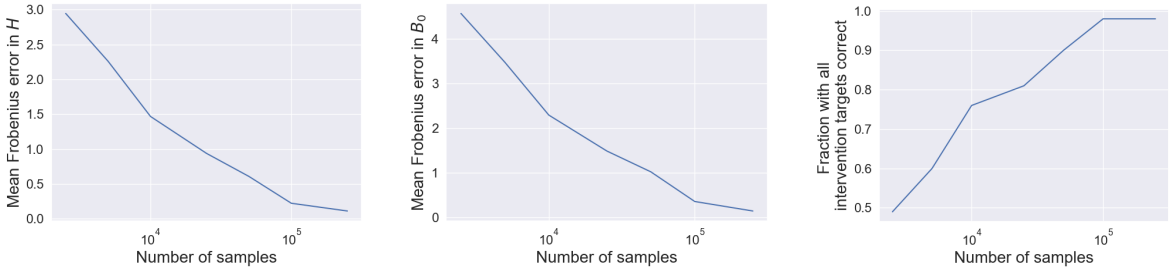
*Proof of Theorem 5.* The necessity of $d$ interventions is Proposition 6. Under Assumption 7 and 8, the sufficiency of $d$ interventions follows from Theorem 6. Under Assumption 7, the sufficiency is the recovery of $\prec_{\mathcal{G}}$ in Proposition 4. □

## 4.4 Experimental results

We adapt our proof of Theorem 7 into a method for causal disentanglement in the finite-sample setting. We modify our methods to (1) use matrices instead of vector spaces, (2) use scores based on singular values to test rank, and (3) choose a nonzero row based on norms. The adapted algorithms are in Appendix C.12. In this section, we investigate the performance of the method in a simulation study. There is a single hyperparameter $\gamma \in [0, 1]$, the percentage of spectral energy associated to the largest singular value, above which we consider a matrix to have rank one. We use $\gamma = 0.99$.

### 4.4.1 Synthetic data generation

We generate 500 random models following Assumption 7 for $d = 5$ latent and $p = 10$ observed variables, as follows. We sample the graph $\mathcal{G}$ from an Erdős-Rényi random graph model with density 0.75. We sample the nonzero entries of $A_0$ independently from $\mathsf{Unif}(\pm[0.25, 1])$, and the nonzero entries of $\Omega_0$ independently from $\mathsf{Unif}([2, 4])$. We sample uniformly among permutations to generate the intervention targets $i_k$. In context $k$, we have $A_k = A_0 - \mathbf{e}_{i_k}A_0^\top\mathbf{e}_{i_k}$; i.e., all entries in row $i_k$ are 0. We change $(\Omega_0)_{i_k,i_k}$ into a new value $(\Omega_k)_{i_k,i_k}$, sampled from $\mathsf{Unif}([6, 8])$ to ensure a non-negligible change. Finally, the entries of $H$ are sampled




(a) Error in estimating $H$ (b) Error in estimating $B_0$ (c) Intervention targets

Figure 4.2: **The adapted version of Algorithm 7 is consistent for recovering $H$, $B_0$, and $\{i_k\}_{k=1}^K$ from noisy data.** At each sample size, we generate 500 random models. Note the logarithmic scale on the x-axis. In **(a)**, we plot the median of $\|\widehat{H} - H\|_2$, the error in Frobenius norm. In **(b)**, we plot the median of $\|\widehat{B}_0 - \widehat{B}\|_2$. In **(c)**, we plot the fraction of models where all intervention targets were correctly estimated.

independently from $\mathsf{Unif}([-2, 2])$. We consider sample sizes $n$ from 2500 to 250000 and use as input the sample precision matrices. All code for data generation and for our adapted versions of Algorithms 5, 6, and 7 (that is, Algorithms 12, 11 and 13) can be found at the link in Appendix C.13.

### 4.4.2 Synthetic data results

We show the results of applying our method in the finite-sample setting, presented in Figure 4.2. We measure the error in estimating the parameters $H$ and $B_0$ and the intervention targets $\{i_k\}_{k=1}^K$. Since the problem is only identifiable up to the partial order $\prec_{\mathcal{G}}$, we align our estimated $\widehat{H}$, $\widehat{B}_0$, and $\{\widehat{i}_k\}_{k=1}^K$ with $H$, $B_0$, and $\{i_k\}_{k=1}^K$ by picking $\sigma \in S(\mathcal{G})$ to maximize $\sum_{k=1}^K \mathbb{1}\{i_k = \sigma(\widehat{i}_k)\}$. For small $d$, this optimization can be solved by enumerating over $S(\mathcal{G})$. For large $d$, we use the integer linear program in Appendix C.13, implemented in `gurobipy`. As expected, the estimation errors for $H$ and $B_0$ decrease as the sample size increases, while the fraction of models with all intervention targets correctly estimated increases.

### 4.4.3 Biological data results

We evaluate our method on a dataset from [Urs+22]. This single-cell RNA sequencing (scRNA-seq) dataset consists of 90,000 cells from a lung cancer cell line, with 83 different nonsynonymous mutations of the KRAS oncogene overexpressed.

**Semi-synthetic analysis.** The authors divide the mutations into five categories based on the genes that they affect, and compute a score for the impact of each mutation. Taking the two highest impact mutations from each category gives $K = 10$ contexts. The wild type KRAS gene is taken as the observational context. We select $p = 100$ observational features to be the most variable genes from the proliferation regulation category in the Gene Ontology [Ash+00], which are significant modulators of cancer activity such as oncogenes and tumor suppressor genes. We compute the sample precision matrices $\widehat{\Theta}_0, \widehat{\Theta}_1, \ldots, \widehat{\Theta}_{10}$ and use them as input to Algorithm 13 with $\gamma = 0.99$.

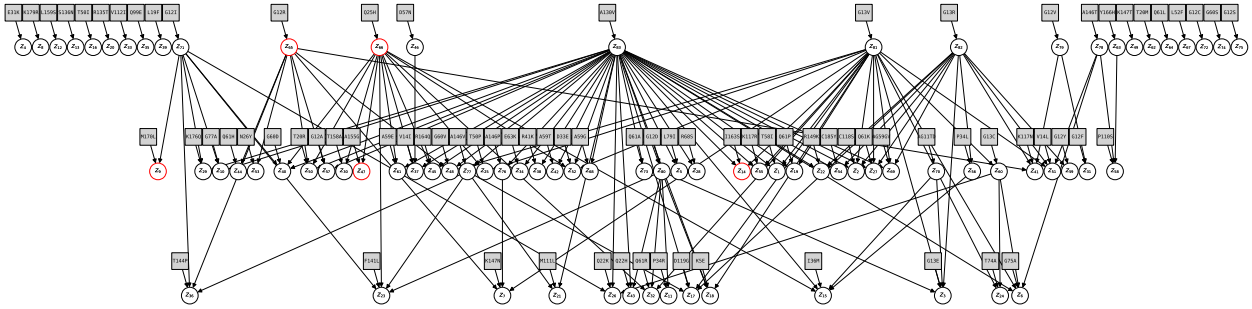


Figure 4.3: **The latent graph and intervention targets learned from scRNA-seq data.** Edges with weight of magnitude above 0.2 are shown. Boxes represent context indicators, corresponding to KRAS mutations, with edges to their respective targets. Red nodes are significantly associated with survival outcomes in the TCGA dataset.

Given estimates $\{\widehat{B}_k\}_{k=1}^K, \widehat{H}$ from our algorithm, we let $\tilde{B}_k = M_k \odot \widehat{B}_k$, where $(M_k)_{ij} = \mathbb{1}_{\{|(\widehat{B}_k)_{ij}|>0.04\}}$; i.e., we truncate the entries away from zero. We treat the resulting parameters as a new generative model. This tests our method in a more realistic setting, with parameters based on real data, while retaining the ability to measure performance. Since the entries of the matrices $\tilde{B}_k$ are smaller than for our synthetic data, we consider larger sample sizes $n \in \{1e6, 5e6, 1e7, 5e7, 1e8\}$. As seen in Figure C.3, Appendix C.13, we successfully recover the generative model.

**Hypothetical Workflow.** We illustrate a hypothetical workflow of our method on biological data. If we run our algorithm for all mutations ($K = 83$) on the $p = 83$ most variable genes, we obtain the graph in Figure 4.3. We see that G12R, G12V, G13R, G13V, and G12I all perturb highly-connected latent nodes with several descendants. The G12 and G13 positions in the KRAS protein are key functional residues whose mutations are known to be causal drivers of cancer [Hua+21]. This indicates that the learned graph can highlight influential biological pathways which may be useful for prioritizing therapeutic development. The matrix $\widehat{H}$ from our algorithm gives a mapping from genes to latent variables that can be transferred across datasets and related to other observations. For example, we compute estimates of the latent variables for the 589 lung cancer patients in the Cancer Genome Atlas [Liu+18] and relate these variables to the patients' survival outcomes. We find that 5 latent variables, those targeted by the M170L, I163S, A55G, G12R, and Q25H mutations, are significantly associated with survival outcomes (at significance 0.1, after multiple testing correction). See Appendix C.13 for details. Note that the output of our method should be treated as exploratory; further theoretical and methodological development is required to better match complex real-world data.

## 4.5 Discussion

In this chapter, we showed that a latent causal model is identifiable when data is available from an intervention on each latent variable. Conversely, we showed that, in the worst case, such data is necessary for identifiability of the latent representation. Our proof is constructive, consisting of an algorithm for recovering the latent representation, which can be adapted

to the noisy setting. Our algorithm was developed for maximal clarity of our identifiability result, and leaves open several directions for future work.

**Theory of latent interventional Markov equivalence.** We established sufficient and (worst-case) necessary conditions for the *complete* identifiability of the parameters $H$ and $\{B_k\}_{k=0}^K$. However, in many settings, it is expected that these parameters (and corresponding combinatorial structures) are only *partially* identifiable. Indeed, Proposition 6 suggests that the problem parameters may be partially recoverable. In future work, it would be interesting to develop a theory of identifiability for $K < p$ interventions.

**Non-linear setting.** Our results require that both the latent linear structural equation model and the mixing function are linear. We expect that the insights developed here may apply when one or both of these elements are non-linear. Notably, the contemporaneous work of [Ahu+23] shows that, under certain conditions, the case of *polynomial mixing* can be reduced to the case of linear, which can be immediately applied to extend our result.

**Statistical analysis of causal disentanglement.** A next step beyond identifiability is to investigate the statistical properties of the setup. This includes lower bounds on the accuracy of recovering the parameters $H$ and $\{B_k\}_{k=0}^K$, along with corresponding combinatorial structures such as $\mathcal{G}$ and the matching between $k$ and $i_k$, and computationally efficient algorithms that match these lower bounds. Our identifiability result suggests that the differences of precision matrices may play a role. These differences appear in [Var+21], which uses techniques for directly estimating differences between precision matrices. Moreover, it would be interesting to develop a score-based approach, e.g., (penalized) maximum likelihood estimation. Our problem formulation suggests a natural parameterization for such an approach, which could be addressed using combinatorial optimization or gradient-based search.

# Chapter 5

# Causal effect estimation with an unknown graph

Adapted from

*Causal effect estimation via covariate adjustments without a causal graph*
by Chandler Squires*, Davin Choo*, Arnab Bhattacharyya, and David Sontag
(* denotes equal contribution, soon to be available)

In all of the previous chapters, our focus was primarily on the *identifiability* of causal models or causal predictions. In this chapter, we turn our attention to statistical considerations, and work with a problem setting where the prediction is well-known to be identifiable. In particular, we consider the model class $\mathcal{M}_{\text{cbn-nd}}(\mathbf{V}, A, \mathbf{N})$ of causal Bayesian networks where the non-descendants of $A$ are known (defined in Equation (1.19)), and we focus on predicting $\mathbb{P}_a^{M^\star}(y)$, the probability that $Y = y$ after an intervention that sets $A = a$, given as input the population observable $\mathbb{P}^{M^\star}(\mathbf{V})$.[1] The main lesson of this chapter is that (from the statistical point of view), model selection and predictive inference *should not be analyzed separately*. In particular, selecting a "correct" model (e.g. one with the correct causal graph) may require a large number of samples, even though incorrect models still generate very accurate predictions at much lower samples sizes: the poor sample efficiency of model selection reflects an attempt to discern between differences that don't make a difference.

**Reading this chapter** In this chapter, we describe our model class in Assumptions 9 and 10. We describe an estimator $\widehat{T}_{\mathbf{S}}$, which uses a data-driven choice of adjustment set $\mathbf{S}$ (model selection) as input into a standard covariate adjustment estimator (predictive inference). Our main result is Theorem 13, which gives a PAC (probably approximately correct) bound on the error of $\widehat{T}_{\mathbf{S}}$. The bound carefully incorporates dependence on the alphabet sizes of $A$, $Y$, $\mathbf{N}$, and $\mathbf{S}$, and reflects a natural tradeoff between misspecification bias (if $\mathbf{S}$ is not a valid adjustment set) and estimation error (which typically increases with the size of $\mathbf{S}$ due to the curse of dimensionality). Individually, these results may be of independent interest. Our bound on misspecification bias, which relates to sensitivity analysis, is given as a standalone

[1] Note that, in this chapter, we use $X$ instead of $A$, since we do not need to use $\mathbf{X}$ to distinguish observed variables from latent ones.

result in Lemma 5. Similarly, our bound on estimation error relates to estimation under high-dimensional confounding, and we state the bound as a standalone result in Lemma 10.

## 5.1 Introduction

Suppose there is an *unknown* probability distribution $\mathbb{P}$ from which we can draw observational samples and we wish to estimate the interventional distribution of $Y$ when $X$ is set to $x$. Notationally, this is written as $\mathbb{P}_x(y) = \mathbb{P}(Y = y \mid \texttt{do}(X = x))$. This problem is known as *causal effect estimation* and has important downstream implications such as estimating treatment effects $\mathbb{E}_x(Y) - \mathbb{E}_{x'}(Y)$, for $x \neq x'$.

**Objective.** Given sample access to a probability distribution $\mathbb{P}$, an interventional query $\mathbb{P}_x(y)$, and parameter $\lambda > 0$, output an estimate $\widehat{\mathbb{P}}_x(y)$ such that $|\widehat{\mathbb{P}}_x(y) - \mathbb{P}_x(y)| \leq \lambda$.

It is typical to relate the probability distributions $\mathbb{P}$ and $\mathbb{P}_x$ by a causal model, described in the form of a causal Bayesian network with graph $\mathcal{G}^*$ [Pea09]. When $\mathcal{G}^*$ is unknown, causal effect estimation is traditionally conceptualized as a two-step process: first estimate a graph $\widehat{\mathcal{G}}$, then apply closed-form graphical criteria to $\widehat{\mathcal{G}}$. The first step is known as *causal discovery*, and under assumptions such as faithfulness, several algorithms are guaranteed (in the infinite data limit) to return $\mathcal{G}^*$ or some graph in its equivalence class $[\widehat{\mathcal{G}}]$, where all graphs in $[\widehat{\mathcal{G}}]$ share the same observational conditional independences as $\widehat{\mathcal{G}}$ [SU22]. The second step is concerned with the problem of *causal identifiability* where one has to output an expression of an interventional query $\mathbb{P}_x(y)$ given a causal graph $\mathcal{G}^*$, or correctly determine that there exists no such expression for some distribution represented by $\mathcal{G}^*$. There are also known closed-form graphical criteria for equivalence classes $[\widehat{\mathcal{G}}]$, e.g. see [PTK18]. Each of these two steps are independent topics of intense research and the second step typically treats the first as a black-box tool and fully trusts its output.

While causal discovery is important in its own right, especially if the goal is to the understand the underlying causal system, we argue that such a two-step approach is suboptimal for estimating $\mathbb{P}_x(y)$. Firstly, to correctly learn the causal graph $\mathcal{G}^*$, one may need strong assumptions on the underlying distribution $\mathbb{P}$. Secondly, the graphical characterization results for causal identifiability do not apply to an erroneous graph, but a huge number of samples may be needed to correctly learn the causal graph $\mathcal{G}^*$, since it is difficult to tell whether an edge is actually missing or is just very "weak".

**Example 2** (Weak edges require many samples to detect but missing them should not affect $\mathbb{P}_x(y)$ much)**.** *Suppose we have a probability distribution $\mathbb{P}$ on variables $\{X, Y, Z\}$ generated as follows:*

*The causal graph that exactly captures $\mathbb{P}$ is a complete DAG with edges $Z \to X \to Y$ and $Z \to Y$; see $\mathcal{G}_1$ in Figure 5.1. However, for extremely small $\varepsilon$, one would require $\Omega(1/\varepsilon)$ samples to detect a dependency between $X$ and $Z$. So, with small $\varepsilon$ and insufficient samples, one may erroneously recover a subgraph without the $Z \to X$ arc; see $\mathcal{G}_2$ in Figure 5.1.*

*Now, suppose we are interested in estimating $\mathbb{P}_0(1) = \mathbb{P}(Y = 1 \mid \texttt{do}(X = 0))$ from observational data. One can check that the correct answer is $\mathbb{P}(Y = 1 \mid \texttt{do}(X = 0)) = 1/2$. Applying standard adjustment formulas under $\mathcal{G}_1$ yield $\mathbb{P}(Y = 1 \mid \texttt{do}(X = 0)) =$*

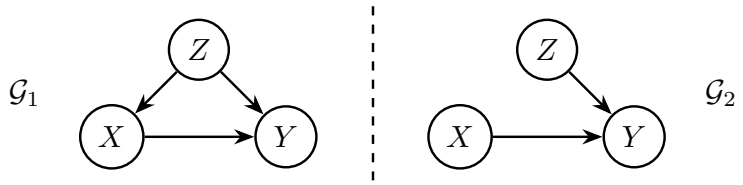



Figure 5.1: Illustration for Example 2: While it is hard to distinguish $\mathcal{G}_1$ from $\mathcal{G}_2$ for small $\varepsilon$ with few samples from $\mathbb{P}$, estimating $\mathbb{P}_x(y)$ using $\mathcal{G}_2$ only incurs an additive error of $O(\varepsilon)$.

$\sum_{z\in\{0,1\}} \mathbb{P}(Z=z)\cdot\mathbb{P}(Y=1 \mid X=0, Z=z) = 1/2$ *as expected. Meanwhile, under* $\mathcal{G}_2$*, the estimation would simply by* $\mathbb{P}(Y=1 \mid X=0) = (1-\varepsilon)/2 = 1/2 - \varepsilon/2$*. Thus, see that the estimation error is only an additive* $O(\varepsilon)$ *factor away from the ground truth.*

### 5.1.1 Our contributions

In this chapter, we ask (and answer affirmatively) the question:

> Can one directly estimate $\mathbb{P}_x(y)$ while avoiding the two-step process of first learning a causal graph, or its equivalence class, and then applying identification formulas?

Under certain causal assumptions, we show how to obtain good estimates of $\mathbb{P}_x(y)$ via covariate adjustments; see Remark 5 on why covariate adjustments are an important, though incomplete, class of causal identifiability methods. Crucially, our methods work even in the absence of the commonly relied upon assumption of causal faithfulness, or its variants, because we do not attempt to learn anything about a causal graph which $\mathbb{P}$ is Markov with respect to; we only assume use the existence of such a graph for analytic purposes. Additionally, unlike many existing causal discovery methods (e.g., the PC algorithm [Spi+00]) which perform a sequence of dependent conditional independence checks, our method uses a *non-dependent* sequence of checks, allowing us to avoid error propagation and control the sample complexity of our procedures.

**Assumption 9.** *We assume that the distribution* $\mathbb{P}$ *over* $\mathbf{V}$ *is Markov to* some *causal graph* $\mathcal{G}^*$ *over* $\mathbf{V}$*, i.e., every separation statement in* $\mathcal{G}^*$ *implies a corresponding conditional independence in* $\mathbb{P}$*. In this thesis, we do not allow for latent confounders or selection bias.*

**Assumption 10.** *We assume knowledge of the non-descendants* $nd_{\mathcal{G}^*}(X)$ *of* $X$ *in* $\mathcal{G}^*$*, i.e., we are given* $\mathbf{N} \subseteq \mathbf{V}$ *and know that* $nd_{\mathcal{G}^*}(X) = \mathbf{N}$*.*

Assumption 9 always holds since every distribution is Markov with respect to the clique graph on $|\mathbf{V}|$ nodes. Assumption 10 can be justified when $X$ is a treatment variable and $\mathrm{nd}_{\mathcal{G}^*}(X)$ denotes a set of pre-treatment covariates; it also allows ensures that $\mathbb{P}_x(y)$ is identifiable, so that we can focus on statistical issues.

As $\mathbb{P}$ could be Markov to multiple possible causal graphs (see Figure 5.2), the set of non-descendants $\mathrm{nd}_{\mathcal{G}^*}(X)$ is not well-defined and could vary depending on which $\mathcal{G}^*$ we are interested in. Therefore, for any subset of variables $\mathbf{N} \subseteq \mathbf{V}$, we define $\mathcal{M}(\mathbb{P}, \mathbf{N})$ as the set of all graphs $\mathcal{G}$ such that:

1. $\mathbb{P}$ is Markov with respect to $\mathcal{G}$,
2. the treatment variable $X \in \mathbf{V}$ comes before outcome variable $Y \in \mathbf{V}$ in $\mathcal{G}$, and
3. $\mathrm{nd}_{\mathcal{G}}(X) = \mathbf{N}$.

For any given subset of variables $\mathbf{N} \subseteq \mathbf{V}$, our theoretical guarantees hold for *any* $\mathcal{G} \in \mathcal{M}(\mathbb{P}, \mathbf{N})$. We search for a subset $\mathbf{Z} \subseteq \mathbf{N}$ amongst the given non-descendants to produce an covariate adjustment estimate $\widehat{\mathbb{P}}_x(y) = \sum_{\mathbf{z}} \mathbb{P}(\mathbf{z}) \cdot \mathbb{P}(y \mid x, \mathbf{z})$. A trivial way to estimate $\widehat{\mathbb{P}}_x(y)$ given $\mathbf{N}$ is to use the entire $\mathbf{N}$ as an adjustment set. While valid, adjusting via a large set has poor sample efficiency, and smaller adjustment sets are preferable.

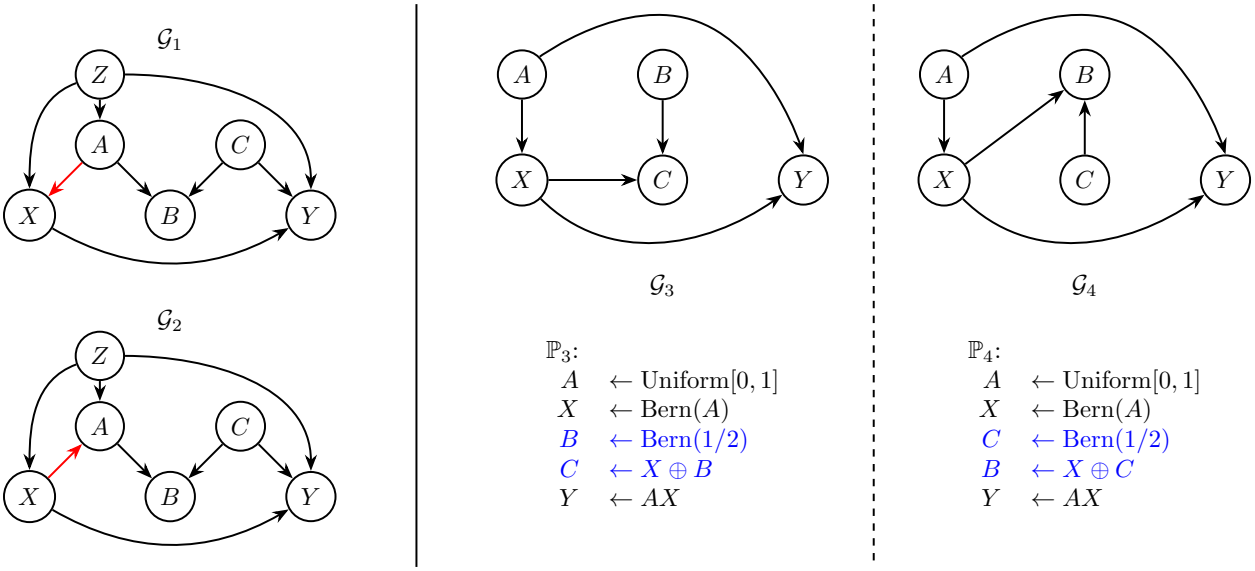



Figure 5.2: While $\mathbb{P}$ will be Markov with respect to any graph in the same equivalence class as $\mathcal{G}^*$ by definition of equivalence, $\mathcal{G}_3$ and $\mathcal{G}_4$ illustrate an example where $\mathbb{P}$ can be Markov to graphs from *different* Markov equivalent classes. **Left ($\mathcal{G}_1$ and $\mathcal{G}_2$):** $\mathcal{G}_1$ and $\mathcal{G}_2$ are two directed acyclic graphs (DAGs) belonging to the same Markov equivalence class. This can be seen by noting that $A \to X$ in $\mathcal{G}_1$ is a covered edge [Chi95], since $\mathrm{pa}_{\mathcal{G}_1}(A) = \mathrm{pa}_{\mathcal{G}_1}(X) \setminus \{A\} = \{Z\}$. We see that $\mathrm{nd}_{\mathcal{G}_1}(X) = \{Z, A, B, C\}$ and $\mathrm{pa}_{\mathcal{G}_1}(X) = \{Z, A\}$, while $\mathrm{nd}_{\mathcal{G}_2}(X) = \{Z, C\}$ and $\mathrm{pa}_{\mathcal{G}_2}(X) = \{Z\}$. Observe that $\{Z\}$ is a valid adjustment set while $\{Z, B\}$ is not as $B$ opens a backdoor path $X - A \to B \leftarrow C \to Y$. **Right ($\mathcal{G}_3$ and $\mathcal{G}_4$):** Here, the two distributions $\mathbb{P}_3$ and $\mathbb{P}_4$ on $\{A, B, C, X, Y\}$ are observationally indistinguishable. However, $\mathcal{G}_3$ and $\mathcal{G}_4$ are in different Markov equivalence classes, since $\mathcal{G}_3$ has a v-structure $X \to C \leftarrow B$ while $\mathcal{G}_4$ has a v-structure $X \to B \leftarrow C$.

Our first result is an estimate of $\mathbb{P}_x(y)$ for the causal sufficient setting where there are no latent confounders, where $\mathbb{P}$ is known explicitly. This means that any required probabilistic quantity with respect to $\mathbb{P}$ can be computed exactly. In particular, it is as if we are given access to a conditional independence oracle for $\mathbb{P}$ since we can always check whether $\mathbb{P}(\mathbf{A}, \mathbf{B}) = \mathbb{P}(\mathbf{A}) \cdot \mathbb{P}(\mathbf{B})$ for any two subsets of variables $\mathbf{A}, \mathbf{B} \subseteq \mathbf{V}$.

**Theorem 8** (Informal)**.** *Suppose Assumption 9 and Assumption 10. Suppose we know* $\mathbb{P}$ *explicitly. In the causally sufficient setting and given a set of non-descendants* $\mathbf{N}$ *of* $X$*, there is an algorithm that produces a valid adjustment formula for* $\mathbb{P}_x(y)$ *using an adjustment set* $\mathbf{S}$*. Furthermore, the size of* $\mathbf{S}$ *is at most the number of parents of* $X$ *in* $\mathcal{G}'$*, for any* $\mathcal{G}' \in \mathcal{M}(\mathbb{P}, \mathbf{N})$*.*

Under causal sufficiency and some variant of the causal faithfulness assumption, when $\mathbb{P}$ is explicitly known and one has access to a conditional independence oracle for $\mathbb{P}$, one can obtain the equivalence class $[\mathcal{G}^*]$ of $\mathcal{G}^*$ simply from observational data. This can be done for example using the PC algorithm [Spi+00] or the Conservative PC (CPC) algorithm [RSZ06] which only requires a weaker variant of the causal faithfulness assumption (called adjacency-faithfulness) to hold; see [Lam23] to learn more about other variants of faithfulness and for a comprehensive discussion and comparison of the various assumptions used in causal discovery algorithms. The equivalence class $[\mathcal{G}^*]$ can be represented as graphically as a partially oriented graph of $\mathcal{G}^*$ with the same undirected adjacencies[2] and one invoke existing causal identification results to obtain covariate adjustments [PTK18].

Meanwhile, Theorem 8 does not assume any form of faithfulness so it is unclear how one can first obtain the equivalence class $[\mathcal{G}^*]$ of $\mathcal{G}^*$ in the two-stage approach described above. Instead, we assume that we are given some non-descendant set $\mathrm{nd}_{\mathcal{G}}(X)$ of $X$ with respect to *some* causal graph $\mathcal{G}'$ which $\mathbb{P}$ is Markov with respect to. That is, we only require $\mathcal{G}' \in [\mathcal{G}^*]$ and not necessarily $\mathcal{G}' = \mathcal{G}^*$. In Section 5.7, we argue that knowing a set of non-descendants of $X$ is necessary in the worst case to obtain any non-trivial guarantees on the causal effect estimation error. Note that this is a strictly weaker assumption than faithfulness since one can always derive a valid set of non-descendants from $[\mathcal{G}^*]$ but one cannot derive $[\mathcal{G}^*]$ given a set of non-descendants of $X$.

To give some informal intuition behind our approach in Theorem 8, first recall that a Markov blanket of a variable $X$ in a DAG is the set of $X$'s parents, $X$'s children, and the parents of $X$'s children. Conditioned on its Markov blanket, the variable $X$ will be independent of any other variable, in particular the treatment variable $Y$ (if $Y$ was not in $X$'s Markov blanket). Meanwhile, any subset of variables in the DAG is a valid covariate adjustment for the treatment of $X$ on the outcome $Y$ if all backdoor paths from $X$ to $Y$ are blocked. As such, it would be sufficient to find a subset $\mathbf{Z} \subseteq \mathrm{nd}_{\mathcal{G}'}(X)$ amongst the non-descendants such that $X \perp\!\!\!\perp \mathrm{nd}_{\mathcal{G}'}(X) \setminus \mathbf{Z} \mid \mathbf{Z}$ and then apply the standard covariate adjustment formula with $\mathbf{Z}$. From this perspective, our algorithmic approach can be broadly viewed as searching for a Markov blanket with respect to the given $\mathrm{nd}_{\mathcal{G}'}(X)$.

The above result (Theorem 8) requires explicit knowledge of $\mathbb{P}$. As this may not be available in practice, we adapt the algorithm to yield guarantees on $|\widehat{\mathbb{P}}_x(y) - \mathbb{P}_x(y)|$ using only an *approximate* estimators and conditional independence testers; we formally define these in Section 5.2.

**Theorem 9** (Informal)**.** *Suppose Assumption 9 and Assumption 10 hold. Suppose we are given access to an* $\varepsilon_1$*-approximator for estimating any conditional marginal of* $\mathbb{P}$ *and* $\varepsilon_2$*-approximate conditional independence tester for* $\mathbb{P}$*. In the causally sufficient setting and given a set of*

[2]More technically speaking, $[\mathcal{G}^*]$ can be graphically represented as a completely partially directed acyclic graph (CPDAG) whereby any acyclic completion of the CPDAG will result in a DAG that encodes the same conditional independencies as $\mathcal{G}^*$.

*non-descendants* $\text{nd}_{\mathcal{G}'}(X)$ *of* $X$*, there is an algorithm that produces an estimate* $\widehat{\mathbb{P}}_x(y)$ *using an adjustment set* $\mathbf{S}$ *such that* $|\mathbb{P}_x(y) - \widehat{\mathbb{P}}_x(y)| \leq \frac{\varepsilon_1+\varepsilon_2}{\alpha_{\mathbf{S}}}$*, where* $\alpha_{\mathbf{S}} = \min_{\mathbf{s}} \mathbb{P}(x \mid \mathbf{s})$.

In general, these approximate estimators and conditional independence testers have failure probabilities, our results will be in the probably approximately correct (PAC) model [Val84; Kea+94] in that sense that our guarantees will hold in high probability over the randomness of the sample generating process. As approximate testers with asymptotically tight sample complexities of discrete $\mathbb{P}$ are well-studied, we can leverage them to obtain sample complexity PAC bounds. This requires a novel bound on the bias of causal effects arising from model misspecification, allowing us to propagate errors in conditional independence testing into an error on the causal effect. The parameter $\alpha_{\mathbf{S}}$ quantifies how strictly the familiar *overlap* assumption (also called *positivity*) is satisfied [DAm+21] for the set $\mathbf{S}$. In Appendix D.6, we show the unavoidability of the $1/\alpha_{\mathbf{S}}$ dependence and the necessity of knowing a non-descendant set $\text{nd}_{\mathcal{G}'}(X)$ for some graph $\mathcal{G}' \in \mathcal{M}(\mathbb{P}, \mathbf{N})$ for correct adjustment.

**Theorem 10** (Informal)**.** *Let* $\varepsilon_1, \varepsilon_2, \delta > 0$*. Then there exists a set* $\mathbf{S} \subseteq \mathbf{N}$ *and an estimator* $\widehat{T}$ *such that, with probability* $1 - \delta$*, we have* $|\widehat{T} - \mathbb{P}_x(y)| \leq \frac{\varepsilon_1+\varepsilon_2}{\alpha_{\mathbf{S}}}$*, as long as* `ApproxSearch` *takes* $t_1 + t_2$ *samples, for*

$$t_1 \in \tilde{O}\left(\frac{|\Sigma_X|^{1/2} \cdot |\Sigma_{\mathbf{N}}|^{1/2} \cdot k^* \cdot \log|\mathbf{N}|}{\varepsilon_2^2}\right) \quad \text{and} \quad t_2 \in \tilde{O}\left(\frac{1}{\varepsilon_1^2} + \frac{|\Sigma_{\mathbf{S}}|}{\varepsilon_1}\log|\Sigma_{\mathbf{S}}|\right),$$

*where* $k^* := \min_{\mathcal{G} \in \mathcal{M}(\mathbb{P},\mathbf{N})} |pa_{\mathcal{G}}(X)|$*, and* $\Sigma_X$*,* $\Sigma_Y$*,* $\Sigma_{\mathbf{N}}$*, and* $\Sigma_{\mathbf{S}}$ *denote the alphabets of* $X$*,* $Y$*,* $\mathbf{N}$*, and* $\mathbf{S}$*, respectively.*

As it is, Theorem 9 is not exactly sample-efficient due to our usage of conditional independence tests involving large number of variables. However, if one further assumes faithfulness, then we show in Appendix D.1 that the population-level conclusions of these tests can inferred from tests involving much smaller number of variables; however, further work is needed to translate this approach to the finite-sample setting. As we can see by the dependence of the bound on $|\Sigma_{\mathbf{S}}|$, it is favorable to have small adjustment sets; we further investigate methods to obtain a minimal sized adjustment sets in Appendix D.3.

**Remark 5** (Why covariate adjustments?)**.** *Besides covariate adjustments, there are many other well-studied classes of estimators such as the inverse probability weighting estimator, front-door adjustment, etc. In fact, covariate adjustments are known to be incomplete in the sense that there exists interventional causal queries without covariate adjustments but there may be other methods to produce a correct causal effect estimation. Nonetheless, covariate adjustments are arguably one of the most commonly used adjustment methods in practice because other methods require precise understanding of the causal relationships between each variable while covariate adjustment only requires the knowledge of the pre-treatment covariates. Furthermore, there is a complete graphical characterization of covariate adjustments under various causal graphical representations [SVR10; ZLT14; MC15; PTK18] and covariate adjustments has a nice property that whenever a valid adjustment set exists for a causal effect, the subset of this adjustment set which forms the non-descendants of treatment variables is also a valid adjustment set [SVR10].*

**Paper outline.** After some preliminaries and related work in Section 5.2, we give our algorithmic results in Section 5.4 and Section 5.5. We discuss their corresponding sample complexities when $\mathbb{P}$ is a discrete distribution in Section 5.6. Hardness results and experimental evaluation are given in Section 5.7 and Section 5.8 respectively before we conclude with interesting future directions in Section 5.9. Empirical evaluation is presented in Section 5.8 before we conclude with interesting future directions in Section 5.9. We conclude with interesting future directions in Section 5.9. For the sake of exposition, certain full proofs are deferred to Appendices D.2-D.6. While we focus the discussion on discrete distributions in this chapter, we expect results of similar flavor for continuous distributions.

## 5.2 Preliminaries

**Notation.** We use capital letters for random variables and lowercase letters for the realizations, e.g. $X = x$, $Y = y$, etc. We use bold letters for sets of variables and write $\mathbb{P}(\mathbf{A} = \mathbf{a})$ as $\mathbb{P}(\mathbf{a})$ as shorthand. We denote the alphabet of the variable $V$ as $\Sigma_V$, and extend this to sets by letting $\Sigma_{\mathbf{A}} = \Sigma_{V_1} \times \ldots \times \Sigma_{V_k}$, where $A = \{V_1, \ldots, V_k\}$ and $\times$ denotes the Cartesian product. To lighten notation, summations are always taken over the entire alphabet of the index, i.e., $\sum_{\mathbf{a}} f(\mathbf{a})$ denotes $\sum_{\mathbf{a} \in \Sigma_{\mathbf{A}}} f(\mathbf{a})$. We employ the standard asymptotic notations $O(\cdot)$, $\Omega(\cdot)$ $\Theta(\cdot)$, and $\tilde{O}(\cdot)$.

Throughout this chapter, we will denote $X$ as the intervened treatment variable and we wish to estimate the effect $\mathbb{P}_x(y) = \mathbb{P}(Y = y \mid \texttt{do}(X = x))$ on the outcome variable $Y$ using observational samples from $\mathbb{P}$.

### 5.2.1 Graphical and causal notions

Let $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ be a graph on $|\mathbf{V}| = n$ vertices with directed arcs $\mathbf{E}$. For any two vertices $X$ and $Y$, we write $X - Y$ to denote adjacency in $\mathcal{G}$, and $X \to Y$ or $X \leftarrow Y$ to specify a certain direction. We write $\text{pa}_{\mathcal{G}}(X), \text{an}_{\mathcal{G}}(X), \text{de}_{\mathcal{G}}(X), \text{nd}_{\mathcal{G}}(X) \subseteq \mathbf{V}$ to denote the parents, ancestors, descendants, and non-descendants of any single vertex $X$ respectively with respect to $\mathcal{G}$. The subscript $\mathcal{G}$ may be omitted if clear from context.

A directed cycle occurs if $X \to Y$ and $Y \in \text{an}(X)$. The graph $\mathcal{G}$ is a directed acyclic graph (DAG) if it does not contain directed cycles. We can associate a (not necessarily unique) *valid topological ordering* $\sigma : \mathbf{V} \to [n]$ to any (partially directed) DAG such that oriented arcs $X \to Y$ satisfy $\sigma(X) < \sigma(Y)$. The skeleton of $\mathcal{G}$ refers to the underlying graph where all edges $\mathbf{E}$ are made undirected. A v-structure in $\mathcal{G}$ refers to a collection of three distinct vertices $X, Y, Z$ such that $X \to Y \leftarrow Z$ and $X \not\to Z$. For any subset of observed variables $\mathbf{S} \subseteq \mathbf{V}$, we denote the modification of $\mathcal{G}$ where all *incoming* (resp. *outcoming*) arcs to vertices $V \in \mathbf{S}$ are removed by a mutilated graph $\mathcal{G}_{\overline{\mathbf{S}}}$ (resp. $\mathcal{G}_{\underline{\mathbf{S}}}$). A non-endpoint vertex $V$ on a path $p$ is called a collider on $p$ if $V$ appears as $\to V \leftarrow$; otherwise $V$ is a non-collider on $p$. Two nodes $X, Y \in \mathbf{V}$ are said to be *d-separated* (denoted by $X \perp_d Y$) in $\mathcal{G}$ given a set $\mathbf{Z} \in \mathbf{V} \setminus \{X, Y\}$ if and only if there is no $\mathbf{Z}$-active path in $\mathcal{G}$ between $X$ and $Y$; a simple path $p$ between $X$ and $Y$ is active with respect to $\mathbf{Z}$ if non-colliders on $p$ are not in $\mathbf{Z}$, and colliders on $p$ are either in $\mathbf{Z}$ or have a descendant in $\mathbf{Z}$. Two nodes are *d-connected* if they are not d-separated.

For any probability distribution $\mathbb{P}$ over a set of variables $\mathbf{V}$ and a DAG $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ defined on the same set of variables, we say that $\mathbb{P}$ is Markov with respect to $\mathcal{G}$ if the joint probability of $\mathbb{P}$ factorizes as $\mathbb{P}(\mathbf{V} = \mathbf{v}) = \prod_{V \in \mathbf{V}} \mathbb{P}(V = v \mid \mathrm{pa}_{\mathcal{G}}(V) = \mathrm{pa}_{\mathcal{G}}(v))$. If $\mathbb{P}$ is Markov with respect to $\mathcal{G}$, then the *causal Markov condition* tells us that d-separation in $\mathcal{G}$ implies conditional independence in $\mathbb{P}$: $\mathbf{A} \perp_d \mathbf{B} \mid \mathbf{C} \implies \mathbf{A} \perp\!\!\!\perp \mathbf{B} \mid \mathbf{C}$. Meanwhile, the converse implication (d-connectedness in $\mathcal{G}$ implies conditional dependence in $\mathbb{P}$) is known as the *causal faithfulness condition*. While many prior works on causal graph discovery rely on some version of causal faithfulness for correctness of their algorithm, our methods crucially do not rely on such an assumption.

For any DAG $\mathcal{G}$, we denote its Markov equivalence class (MEC) by $[\mathcal{G}]$ and essential graph by $\mathcal{E}(\mathcal{G})$. DAGs in the same MEC have the same skeleton and the essential graph is a partially directed graph such that an arc $X \to Y$ is directed if $X \to Y$ in *every* DAG in MEC $[\mathcal{G}]$, and an edge $X - Y$ is undirected if there exists two DAGs $\mathcal{G}_1, \mathcal{G}_2 \in [\mathcal{G}]$ such that $X \to Y$ in $\mathcal{G}_1$ and $Y \to X$ in $\mathcal{G}_2$. It is known that two graphs are Markov equivalent if and only if they have the same skeleton and v-structures [VP90; AMP97].

An (ideal) intervention on targets $\mathbf{W} \subseteq \mathbf{V}$ is an experiment where all variables $W \in \mathbf{W}$ are forcefully set to some value, independent of the underlying causal structure. Graphically, interventions are represented by a mutilated interventional graph: intervening on $\mathbf{W} \subseteq \mathbf{V}$ results in a causal graph $G_{\overline{\mathbf{W}}}$ where all incoming arcs to vertices $W \in \mathbf{W}$ are removed [EGS05]. The effect of interventions on DAGs is formally captured by Pearl's do-calculus [Pea09]. For disjoint sets $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{W} \subseteq \mathbf{V}$, the three rules of the do-calculus are:

**R1 (add/remove obs)** If $(\mathbf{A} \perp_d \mathbf{B} \mid \mathbf{C}, \mathbf{W})_{\mathcal{G}_{\overline{\mathbf{W}}}}$, then for all $\mathbf{w}'$, we have

$$\mathbb{P}(\mathbf{A} \mid \mathbf{B}, \mathbf{C}, \mathtt{do}(\mathbf{w}')) = \mathbb{P}(\mathbf{A} \mid \mathbf{C}, \mathtt{do}(\mathbf{w}'))$$

**R2 (swap obs with do)** If $(\mathbf{A} \perp_d \mathbf{B} \mid \mathbf{C}, \mathbf{W})_{\mathcal{G}_{\overline{\mathbf{W}}\underline{\mathbf{B}}}}$, then for all $\mathbf{w}', \mathbf{b}'$, we have

$$\mathbb{P}(\mathbf{A} \mid \mathtt{do}(\mathbf{b}'), \mathbf{C}, \mathtt{do}(\mathbf{w}')) = \mathbb{P}(\mathbf{A} \mid \mathbf{b}', \mathbf{C}, \mathtt{do}(\mathbf{w}'))$$

**R3 (add/remove do)** If $(\mathbf{A} \perp_d \mathbf{B} \mid \mathbf{C}, \mathbf{W})_{\mathcal{G}_{\overline{\mathbf{W}\mathbf{B}(\mathbf{C})}}}$, then for all $\mathbf{x}', \mathbf{b}'$, we have

$$\mathbb{P}(\mathbf{A} \mid \mathtt{do}(\mathbf{b}'), \mathbf{C}, \mathtt{do}(\mathbf{w}')) = \mathbb{P}(\mathbf{A} \mid \mathbf{C}, \mathtt{do}(\mathbf{w}'))$$

where $\mathbf{B}(\mathbf{C}) = \mathbf{B} \setminus \mathrm{an}(\mathbf{C})$ are $\mathbf{B}$ nodes that are not ancestors of any $\mathbf{C}$ nodes in $\mathcal{G}_{\overline{\mathbf{X}}}$.

### 5.2.2 Estimators and approximators

Let $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ be disjoint subsets of variables. We define approximate conditional independence $\mathbf{A} \perp\!\!\!\perp_\varepsilon \mathbf{B} \mid \mathbf{C}$ as

$$\sum_{\mathbf{a},\mathbf{b},\mathbf{c}} \mathbb{P}(\mathbf{c}) \cdot |\mathbb{P}(\mathbf{a}, \mathbf{b} \mid \mathbf{c}) - \mathbb{P}(\mathbf{a} \mid \mathbf{c}) \cdot \mathbb{P}(\mathbf{b} \mid \mathbf{c})| \leq \varepsilon \,.$$

Note that $\varepsilon = 0$ exactly corresponds to $\mathbf{A} \perp\!\!\!\perp \mathbf{B} \mid \mathbf{C}$. For any $\mathbf{S} \subseteq \mathbf{V} \setminus (X \cup Y)$, we define $0 \leq \alpha_{\mathbf{S}} = \min_{\mathbf{s}} \mathbb{P}(x \mid \mathbf{s}) \leq 1$ .

We consider three approximate oracles. Each oracle can draw samples from $\mathbb{P}$ and is required to succeed with probability at least $1 - \delta$ for input $\delta > 0$. The other inputs to these oracles are error tolerances $\varepsilon_1, \varepsilon_2 > 0$ and disjoint subsets of variables $\mathbf{A}, \mathbf{B}, \mathbf{C} \subseteq \mathbf{V} \setminus (X \cup Y)$.

1. `EstimateMarginal`$(\mathbf{A}, \varepsilon_1, \delta)$: Produce a distribution $R(\mathbf{A})$ such that $\sum_{\mathbf{a}} |R(\mathbf{a}) - \mathbb{P}(\mathbf{a})| \leq \frac{\varepsilon_1}{2}$.

2. `EstimateConditional`$(\mathbf{A}, \mathbf{B}, \varepsilon_1, \delta)$:
   Produce a distribution $Q(\mathbf{A} \mid \mathbf{B})$ such that $\sum_{\mathbf{a},\mathbf{b}} \mathbb{P}(\mathbf{b}) \cdot |Q(\mathbf{a} \mid x, \mathbf{b}) - \mathbb{P}(\mathbf{a} \mid x, \mathbf{b})| \leq \frac{\varepsilon_1}{2\alpha_{\mathbf{B}}}$.

3. `ApproxCondInd`$(\mathbf{A}, \mathbf{B}, \mathbf{C}, \varepsilon_2, \delta)$: Correctly determine whether $\mathbf{A} \perp\!\!\!\perp_{\varepsilon_2} \mathbf{B} \mid \mathbf{C}$.

Any method which implements `EstimateMarginal` or `EstimateConditional` is called an $\varepsilon_1$-*approximator* and any method which implements `ApproxCondInd` is called an $\varepsilon_2$-*approximate conditional independence (CI) tester*. We will give examples of such oracles for discrete distributions $\mathbb{P}$ in Appendices D.4 and D.5, along with their sample complexities. In practice, when given a fixed dataset $\mathcal{A}$ consisting of IID samples drawn from $\mathbb{P}$, one can simply estimate and test these quantities empirically.

### 5.2.3 Causal identifiability

Causal identifiability refers representing an interventional causal query $\mathbb{P}_x(y)$ using observational data $\mathbb{P}$ given a causal graph for $\mathbb{P}$, possibly under various extensions such as partial observability and access to additional interventional distributions of $\mathbb{P}$ [TP02; BP12; LCB20a; LB20]. In this chapter, we focus our efforts on studying the class of covariate adjustment sets which includes the well-known *backdoor adjustment* [Pea95]. For *any* subset $\mathbf{S} \subseteq \mathrm{ND}(X)$, let us define the quantity $T_{\mathbf{S},x,y}$ as $\sum_{\mathbf{s}} \mathbb{P}(\mathbf{s}) \cdot \mathbb{P}(y \mid x, \mathbf{s})$ if $\mathbf{S} \neq \varnothing$ and as $\mathbb{P}(y \mid x)$ if $\mathbf{S} = \varnothing$. We say $\mathbf{S}$ is a *valid adjustment set* for $\mathbb{P}_{x'}(Y)$ if $\mathbb{P}_{x'}(y') = T_{\mathbf{S},x',y'}$ for all $x', y'$. In this chapter, we consider a fixed choice $x$ and $y$, so we write $T_{\mathbf{S}} = T_{\mathbf{S},x,y}$ to lighten notation.

## 5.3 Related work

### 5.3.1 Causal graph discovery

Classically, there are two broad classes of causal graph discovery methods: constraint-based and score-based; see [GZS19] and [VCB22] for a survey. Constraint-based algorithms use information about conditional independence relations to identify the underlying structure. From purely observational data, the PC [Spi+00], FCI [Spi+00] and RFCI algorithms [Col+12] have been shown to consistently recover the essential graph, assuming causal sufficiency, faithfulness, and i.i.d. samples. There is also a line of work on recovering a causal graph using constraints imposed by interventional data, possibly involving actively chosen interventions [EGS05; EGS06; Ebe07; HLV14; Sha+15; KDV17; Lin+18; Gre+19; Jab+20; Squ+20; CSB22; CS23c; CGB23; CS23b; CS23a].

On the other hand, score-based methods maximize a particular score function over the space of possible causal graphs. For observational data, the GES algorithm [Chi02] uses the Bayesian information criterion (BIC) to iteratively add edges. Extending the GES, [HB12] proposed the GIES algorithm that uses passive interventional data to orient more edges. There are also hybrid methods that employ elements of both approaches, e.g. [SWU21] for observational data and [Wan+17] for interventional data. More recently, initiated by

[Zhe+18], there has been a wave of causal discovery methods based on modelling the graph discovery problem in a differentiable manner and optimizing them via gradient based methods. However, these methods are known to be sensitive to scaling of parameters, e.g. see [KS22; NHZ24] and references therein.

### 5.3.2 Causal effect estimation

A spiritual predecessor of this chapter is the work of [EHS13] where they studied a set of simple rules under which one can identify appropriate set of covariates to adjust for, based on testable (conditional) dependence and independence relations among the observed variables. The work of [EHS13] assumes faithfulness, knowledge of the set of covariates, and their theoretical results rely on an independence oracle. In contrast, this chapter does not assume faithfulness and we also have theoretical results assuming access only to approximate testers and estimators.

A similar result to Lemma 2 (for $\mathbf{S} = \varnothing$) is known in [Che+22], where they study "Cause Or Spouse of the treatment Only (COSO)" variables. However, this is not the same as $\mathbf{Z} = \mathrm{pa}_{\mathcal{G}'}(X)$ since $Z \in \mathbf{Z}$ can be a parent of any other variable. In the simple example of $A \to X \leftarrow B \to Y$ with $X \to Y$, the variable $B$ is both a parent of the treatment variable $X$ and a parent of the outcome variable $Y$.

Very recently, there has been works on causal effect estimation with minimal amount of knowledge about causal graph [SSA22; SSK23]. Focusing on covariate adjustments for singletons $X$ and $Y$, [SSA22] assume that $X$ only has $Y$ as its only child (i.e. all other variables are non-descendants of $X$), the knowledge of one true parent of $X$, and a weaker notion of faithfulness called "sub-sampling faithfulness". Unfortunately, their method does not cover all possible covariate adjustments and they remarked in their conclusion that "exploring alternate minimal DAG knowledge sufficient to test for a broader/different family of valid adjustments could be fruitful". The current chapter addresses this by providing covariate adjustment methods that only require knowledge of $\mathrm{nd}_{\mathcal{G}'}(X)$ (which we argue is necessary in Section 5.7) and access to an (approximate) conditional independence oracle, without any form of faithfulness assumption. Meanwhile, [SSK23] focuses on front-door adjustments while assuming that $Y$ is a descendant of $X$, there is an unobserved confounder between $X$ and $Y$, the set of all children of $X$ are observed and known, and causal faithfulness. Finally, two contemporaneous works [Mal24; CGM24] also consider the interplay between causal discovery and causal effect estimation, also advocating the use of conditional dependence checks instead of conditional independence checks to control misspecification bias, but analyze the asymptotic setting rather than the PAC setting considered in this chapter.

Finally, our results relate to two well-developed sub-fields of causal effect estimation: sensitivity analysis and treatment effect estimation under high-dimensional confounding. Results on sensitivity analysis relate the amount of confounding bias to some "sensitivity parameter(s)" quantifying the strength of omitted confounders, see [CH20] for an entry point. Such results are in the same vein as our bound in Lemma 5, but use different sensitivity parameters. Works addressing high-dimensional confounding often (implicitly or explicitly) involve a variable selection step, which can reduce estimation error at the cost of introducing confounding bias. The current chapter fits into this literature, with the same tradeoff appears in our sample complexity bound in Theorem 13, but contrasts with previous work in two

ways. First, we focus on *discrete* random variables, allowing for more contact with work on distribution testing [Can20] and property estimation [CSS19]. Second, we intentionally emphasize a *graphical* perspective on the problem, to highlight the interplay between causal structure learning and causal effect estimation.

### 5.3.3 Addressing high-dimensional confounding

In domains such as healthcare, causal effect estimation has been increasingly applied to rich datasets with high-dimensional covariates (e.g. the genetic profile of a patient). These settings present novel challenges, including violations of the overlap assumption [DAm+21] and the breakdown of traditional asymptotic results. Dimensionality reduction techniques such as feature selection are often crucial to dealing with these challenges. However, in the context of treatment effect estimation, naïve usage of feature selection methods such as the Lasso can introduce substantial confounding bias. Several works aim to address this issue by some form of bias correction. We provide only a partial review here, pointing the interested reader to [Yad+22], references therein, and the following references as starting points.

We restrict our discussion to estimators of the average treatment effect; we do not cover estimators for heterogeneous treatment effects or other quantities. Furthermore, we restrict our discussion to methods which use feature selection for dimensionality reduction; we do not cover other dimensionality reduction techniques. The majority of prior methods focus on the case of *continuous* covariates $\mathbf{C}$, *binary* treatment $X$, and scalar outcome $Y$ (which could be binary or continuous). To address the dimensionality of the problem, most prior methods impose assumptions on the treatment (i.e., on $\mathbb{P}(X \mid \mathbf{C})$), the outcome (i.e., $\mathbb{P}(Y \mid \mathbf{C})$, or both.

The most common parametric assumptions are that $\mathbb{P}(X \mid \mathbf{C})$ follows a sparse logit model[3], and that $\mathbb{P}(Y \mid \mathbf{C})$ follows a linear model. Sparsity restrictions may be imposed on one or both of these models. For example, [SE17] assumes a sparse logit model for the treatment, and a (possibly dense) linear model for the outcome, while [BWZ19] assumes a sparse linear model for the outcome and a (possibly dense) logit model for the treatment. In some methods, one of the parametric assumptions is dropped. For example, [AIW18] assumes an arbitrary model for the treatment (and a sparse linear model for the outcome), whereas [WS20] assumes an arbitrary model for the outcome (and a sparse logistic model for the treatment). Further guarantees may be obtained by imposing sparsity assumptions on both models, as in [GSK21]. Parametric assumptions may also be replaced with weaker assumptions, e.g. smoothness assumptions [FLM21] or semiparametric assumptions such as a partially linear model [BCH14; Che+18].

In comparison to these methods, we focus on the setting of discrete-valued random variables, obviating the need for (semi-)parametric or smoothness restrictions while making the problem accessible to a wider audience.

[3] i.e., $\mathbb{P}(X = 1 \mid \mathbf{C} = \mathbf{c}) = \sigma(\beta^\top \mathbf{c} + \mu)$ for some $\beta \in \mathbb{R}^{|\mathbf{C}|}$ and $\mu \in \mathbb{R}$, where $\sigma$ denotes the logistic function.

## 5.4 Adjusting via parents

As discussed in Section 5.1, while $\mathbf{N}$ is a trivial adjustment set for $\mathbb{P}_x(y)$, we are interested in using one of smaller size. To show guarantees of our candidate adjustment set, we use on the following simple structural result that relates $\text{pa}_{\mathcal{G}'}(X)$ to any other candidate set $\mathbf{S} \subseteq \mathbf{N}$.

**Lemma 2.** *If $\mathbb{P}$ is Markov with respect to a DAG $\mathcal{G}$ and $\mathbf{Z} = pa_{\mathcal{G}}(X) \subseteq \mathbf{V}$, then for any $\mathbf{S} \subseteq nd_{\mathcal{G}}(X)$, we have $\mathbb{P}_x(y) = \sum_{\mathbf{z}\cup\mathbf{s}} \mathbb{P}(y \mid x, \mathbf{z} \cup \mathbf{s}) \cdot \mathbb{P}(\mathbf{z} \cup \mathbf{s})$.*

*Proof (Sketch).* Using total law of probability and the do-calculus, one can show

$$\begin{aligned}\mathbb{P}_x(y) &= \sum_{\mathbf{z}\cup\mathbf{s}} \mathbb{P}(y, \mathbf{z} \cup \mathbf{s} \mid \texttt{do}(x)) = \sum_{\mathbf{z}\cup\mathbf{s}} \mathbb{P}(y \mid \texttt{do}(x), \mathbf{z} \cup \mathbf{s}) \cdot \mathbb{P}(\mathbf{z} \cup \mathbf{s} \mid \texttt{do}(x)) \\ &= \sum_{\mathbf{z}\cup\mathbf{s}} \mathbb{P}(y \mid x, \mathbf{z} \cup \mathbf{s}) \cdot \mathbb{P}(\mathbf{z} \cup \mathbf{s} \mid \texttt{do}(x)) = \sum_{\mathbf{z}\cup\mathbf{s}} \mathbb{P}(y \mid x, \mathbf{z} \cup \mathbf{s}) \cdot \mathbb{P}(\mathbf{z} \cup \mathbf{s})\end{aligned}$$

□

In particular, Lemma 2 states that if we start with the valid adjustment set $\mathbf{Z} = \text{pa}_{\mathcal{G}'}(\mathbf{X})$, we can add any non-descendants $\mathbf{N}$ of $X$ to $\mathbf{Z}$, and still maintain a valid adjustment set. This property is specific to this adjustment set; it is not true for *any* valid adjustment set. For example, in $\mathcal{G}_1$ in Figure 5.2, $\{Z\}$ is a valid adjustment set and $B \in \text{nd}_{\mathcal{G}_1}(X)$, but $\{Z, B\}$ is not a valid adjustment set.

This special property implies that we can completely characterize the bias when adjusting on $\mathbf{S}$ in terms of $\mathbf{Z} \setminus \mathbf{S}$. Using standard probability manipulations on Lemma 2, we can further highlight this.

**Lemma 3.** *Suppose $\mathbb{P}$ is Markov with respect to a DAG $\mathcal{G}$ and $\mathbf{Z} = pa_{\mathcal{G}}(X) \subseteq \mathbf{V}$, then for any $\mathbf{S} \subseteq ND_{\mathcal{G}}(X)$, we have*

$$\mathbb{P}_x(y) = \sum_{\mathbf{s}} \mathbb{P}(\mathbf{s}) \cdot \mathbb{P}(y \mid x, \mathbf{s}) \cdot \sum_{\mathbf{z}\setminus\mathbf{s}} \mathbb{P}(\mathbf{z} \setminus \mathbf{s} \mid x, y, \mathbf{s}) \cdot \frac{\mathbb{P}(x \mid \mathbf{s}) \cdot \mathbb{P}(\mathbf{z} \setminus \mathbf{s} \mid \mathbf{s})}{\mathbb{P}(x, \mathbf{z} \setminus \mathbf{s} \mid \mathbf{s})}$$

**Example 3.** *Consider $\mathcal{G}_1$ in Figure 5.2 where $nd_{\mathcal{G}_1}(X) = \{Z, A, B, C\}$ and $pa_{\mathcal{G}_1}(X) = \mathbf{Z} = \{Z, A\}$. Suppose $\mathbf{S} = \{A, B\}$. Then, Lemma 3 states that*

$$\mathbb{P}_x(y) = \sum_{a,b} \mathbb{P}(a, b) \cdot \mathbb{P}(y \mid x, a, b) \cdot \sum_{z} \mathbb{P}(z \mid x, y, a, b) \cdot \frac{\mathbb{P}(x \mid a, b) \cdot \mathbb{P}(z \mid a, b)}{\mathbb{P}(x, z \mid a, b)}$$

*Note that the expression $\sum_{a,b} \mathbb{P}(a, b) \cdot \mathbb{P}(y \mid x, a, b)$ is the typical covariate adjustment formula if one uses $\mathbf{S} = \{A, B\}$ as an adjustment set. As such, one can view each multiplicative term in the summation as a correction/reweighting for not using $\mathbf{Z}$.*

Now, observe that if we manage to find some subset $\mathbf{S}$ such that the ratio $\frac{\mathbb{P}(x|\mathbf{s}) \cdot \mathbb{P}(\mathbf{z}\setminus\mathbf{s}|\mathbf{s})}{\mathbb{P}(x, \mathbf{z}\setminus\mathbf{s}|\mathbf{s})} = 1$ in Lemma 3, for all values of $\mathbf{z} \cup \mathbf{s}$, then $\mathbb{P}_x(y) = \sum_{\mathbf{s}} \mathbb{P}(\mathbf{s}) \cdot \mathbb{P}(y \mid x, \mathbf{s})$, even if $\mathbf{S} \neq \text{pa}_{\mathcal{G}'}(X)$. The idea behind `OracleSearch` (Algorithm 8) is to search for such a subset $\mathbf{S}$.

**Theorem 11** (Formalization of Theorem 8)**.** *Given non-descendants $nd_{\mathcal{G}'}(X)$ of $X$ and a conditional independence oracle for $\mathbb{P}$, `OracleSearch` produces a valid adjustment formula for $\mathbb{P}_x(y)$ using an adjustment set $\mathbf{S}$ of size $|\mathbf{S}| \leq k^* := \min_{\mathcal{G} \in \mathcal{M}(\mathbb{P}, \mathbf{N})} |pa_{\mathcal{G}}(X)|$.*

**Algorithm 8** `OracleSearch`

**Input:** Query $\mathbb{P}_x(y)$ and a set of non-descendants $\mathbf{N} \subseteq \mathbf{V}$
**Output:** Adjustment formula for $\mathbb{P}_x(y)$
**for** $k = 0, 1, 2, \ldots$ **do**
  Let $\mathbf{C}_k = \{\mathbf{S} \subseteq \mathbf{N} : |\mathbf{S}| = k,\ X \perp\!\!\!\perp \mathbf{N} \setminus \mathbf{S} \mid \mathbf{S}\}$ be all size $k$ candidate sets.
  **if** $|\mathbf{C}_k| > 0$ **then**
    Pick any $\mathbf{S} \in \mathbf{C}_k$
    **return** $\sum_{\mathbf{s}} \mathbb{P}(\mathbf{s}) \cdot \mathbb{P}(y \mid x, \mathbf{s})$
  **end if**
**end for**

*Proof.* Let $\mathbf{Z} = \mathrm{pa}_{\mathcal{G}'}(X)$ for any $\mathcal{G}' \in \mathcal{M}(\mathbb{P}, \mathrm{ND}(X))$. Upon termination with $\mathbf{S}$, we are guaranteed that $X \perp\!\!\!\perp \mathbf{N} \setminus \mathbf{S} \mid \mathbf{S}$, implying that $X \perp\!\!\!\perp \mathbf{Z} \setminus \mathbf{S} \mid \mathbf{S}$. So, $\mathbb{P}(x, \mathbf{z} \setminus \mathbf{s} \mid \mathbf{s}) = \mathbb{P}(x \mid \mathbf{s}) \cdot \mathbb{P}(\mathbf{z} \setminus \mathbf{s} \mid \mathbf{s})$ and Lemma 3 tells us that $\mathbb{P}_x(y) = \sum_{\mathbf{S}} \mathbb{P}(\mathbf{s}) \cdot \mathbb{P}(y \mid x, \mathbf{s})$ as desired. Finally, we see that `OracleSearch` terminates with $k \leq |\mathbf{Z}|$ since $X \perp\!\!\!\perp \mathbf{N} \setminus \mathbf{Z} \mid \mathbf{Z}$. □

**Remark 6** (Efficiency of conditional independence checks)**.** *It may be costly to check the condition* $\mathbf{X} \perp\!\!\!\perp \mathbf{N} \setminus \mathbf{S} \mid \mathbf{S}$ *as it involves many variables. Under faithfulness, we show in Appendix D.1 that this condition can be simplified to just ensuring that* $\mathbf{X} \perp\!\!\!\perp A \mid \mathbf{S}$ *for each variable* $A \in \mathbf{N} \setminus \mathbf{S}$*.*

We now consider the more realistic setting where we do not have access to perfect conditional independence oracle and the exact probability terms. Suppose that $\mathbf{S} \subseteq \mathbf{N}$ is a proposed adjustment set. A natural estimator of $T_{\mathbf{S}}$ is the plug-in estimator, which first estimates $\mathbb{P}(\mathbf{s})$ by some $R(\mathbf{s})$ and $\mathbb{P}(y \mid x, \mathbf{s})$ by some $Q(y \mid x, \mathbf{s})$, then outputs $\sum_{\mathbf{s}} R(\mathbf{s}) \cdot Q(y \mid x, \mathbf{s})$; see `AdjustByS`.

**Algorithm 9** `AdjustByS`

**Input:** Query $\mathbb{P}_x(y)$, a candidate adjustment set $\mathbf{S}$, an error tolerance $\varepsilon_1 > 0$
**Output:** Estimate $\widehat{T}_{\mathbf{S}}$ of $T_{\mathbf{S}}$
$R \leftarrow$ `EstimateMarginal`$(\mathbf{S}, \varepsilon_1/2, \delta)$
$Q \leftarrow$ `EstimateConditional`$(Y, X \cup \mathbf{S}, \varepsilon_1/2, \delta)$
**return** $\sum_{\mathbf{s}} R(\mathbf{s}) \cdot Q(y \mid x, \mathbf{s})$

We analyze the sample complexities required for obtaining such $R$ and $Q$ in Section 5.6.

**Lemma 4.** *Let* $\widehat{T}_{\mathbf{S}}$ *be the output of* `AdjustByS`*. Then,* $|\widehat{T}_{\mathbf{S}} - T_{\mathbf{S}}| \leq \varepsilon_1/\alpha_{\mathbf{S}}$*.*

*Proof (Sketch).* Split $|\widehat{T}_{\mathbf{S}} - T_{\mathbf{S}}|$ into a sum of two error terms using triangle inequality: one for estimating $\mathbb{P}(\mathbf{s})$, and one for estimating $\mathbb{P}(y \mid x, \mathbf{s})$. Then, apply the guarantees of `EstimateMarginal` and `EstimateConditional` to bound these terms by $\varepsilon_1/2$ and $\varepsilon_1/(2\alpha_{\mathbf{S}})$ respectively. □

Armed with `AdjustByS` and access to an approximate conditional independence tester `ApproxCondInd`, we modify `OracleSearch` into `ApproxSearch` (Algorithm 14, Appendix D.2)

to find a candidate $\mathbf{S} \subseteq \mathbf{N}$ approximately satisfying the conditional independence condition using `ApproxCondInd` and then invoke `AdjustByS` to output an estimation $\widehat{\mathbb{P}}_x(y)$ of $\mathbb{P}_x(y)$. In particular, we show that the bias induced by using a set $\mathbf{S} \neq \mathbf{Z}$ satisfies the following bound:

**Lemma 5.** *Let $\mathbf{S} \subseteq \mathbf{N}$ be such that $X \perp\!\!\!\perp_{\varepsilon_2} \mathbf{N} \setminus \mathbf{S} \mid \mathbf{S}$. Then, $|T_{\mathbf{S}} - \mathbb{P}_x(y)| \leq \frac{\varepsilon_2}{\alpha_{\mathbf{S}}}$.*

*Proof (Sketch).* For any fixed $\mathbf{S}$, we can manipulate the expression in Lemma 3 with triangle inequality to obtain $|T_{\mathbf{S}} - \mathbb{P}_x(y)| \leq \sum_{\mathbf{z} \cup \mathbf{s}} \frac{\mathbb{P}(\mathbf{s})}{\mathbb{P}(x|\mathbf{s})} \cdot \Big|\mathbb{P}(x, \mathbf{z} \setminus \mathbf{s} \mid \mathbf{s}) - \mathbb{P}(x \mid \mathbf{s}) \cdot \mathbb{P}(\mathbf{z} \setminus \mathbf{s} \mid \mathbf{s})\Big|$. Then, the claim follows from the definition of $\alpha_{\mathbf{S}}$ and the guarantees of `ApproxCondInd`. □

These two sources of error combine to give our main result on the error of `ApproxSearch`.

**Theorem 12** (Formalization of Theorem 9)**.** *Fix error tolerances $\varepsilon_1 > 0$ and $\varepsilon_2 > 0$. Given a set of non-descendants $ND(\mathbf{X})$ of $\mathbf{X}$, $\varepsilon_1$-approximators `EstimateMarginal` and `EstimateConditional`, and an $\varepsilon_2$-approximate CI tester `ApproxCondInd`, `ApproxSearch` produces an estimate $\widehat{\mathbb{P}}_x(y)$ using an adjustment set $\mathbf{S}$ of size $|\mathbf{S}| \leq \min_{\mathcal{G} \in \mathcal{M}(\mathbb{P}, \mathbf{N})} |pa_{\mathcal{G}}(X)|$ such that $|\mathbb{P}_x(y) - \widehat{\mathbb{P}}_x(y)| \leq \frac{\varepsilon_1 + \varepsilon_2}{\alpha_{\mathbf{S}}}$, where $\alpha_{\mathbf{S}} = \min_{\mathbf{s}} \mathbb{P}(x \mid \mathbf{s})$.*

*Proof (Sketch).* Let $\mathbf{Z} = \mathrm{pa}_{\mathcal{G}'}(X)$ for any $\mathcal{G}' \in \mathcal{M}(\mathbb{P}, \mathbf{N})$. As we output $\widehat{T}_{\mathbf{S}}$ as an estimate for $\mathbb{P}_{\mathbf{x}}(\mathbf{y})$, we can manipulate Lemma 3 and observe that there are two sources of error: (1) $|\widehat{T}_{\mathbf{S}} - T_{\mathbf{S}}|$, the estimation error of $T_{\mathbf{S}}$, and (2) $|T_{\mathbf{S}} - \mathbb{P}_x(y)|$, the bias from using a set $\mathbf{S} \neq \mathbf{Z}$. By Lemma 4, the first term is bounded by $\varepsilon_1 / \alpha_{\mathbf{S}}$. By Lemma 5, the second term is bounded by $\varepsilon_2 / \alpha_{\mathbf{S}}$. Finally, `ApproxSearch` terminates with $k \leq |\mathbf{Z}|$ as $\mathbf{X} \perp\!\!\!\perp \mathrm{ND}(X) \setminus \mathbf{Z} \mid \mathbf{Z}$, which implies $X \perp\!\!\!\perp_{\varepsilon} \mathrm{ND}(X) \setminus \mathbf{Z} \mid \mathbf{Z}$ for any $\varepsilon > 0$. □

In view of the guarantees given by Theorem 12, one may wonder whether there is a better way to search for candidate sets such that the resultant bound $\frac{\varepsilon_1 + \varepsilon_2}{\alpha_{\mathbf{S}}}$ is minimized. In fact, we argue that it is advantageous to bias our search procedure to focus on smaller subsets of variables for the following two reasons. Firstly, for any fixed estimation error $\varepsilon_1$, the sample complexity required typically grows as the domain size of variables involved increase (e.g. see Section 5.6). Secondly, the following lemma tells us that the error term due to $\alpha_{\mathbf{S}}$ worsens as we consider supersets.

**Lemma 6.** *For any value $x$ for $X$ and subsets $\mathbf{A} \subseteq \mathbf{B} \subseteq \mathbf{V} \setminus \mathbf{X}$, $\min_{\mathbf{a}} \mathbb{P}(x \mid \mathbf{a}) \geq \min_{\mathbf{b}} \mathbb{P}(x \mid \mathbf{b})$.*

*Proof.* Fix an arbitrary value of $x$ for $\mathbf{X}$.

$$\mathbb{P}(x \mid \mathbf{a}) = \sum_{\mathbf{b} \setminus \mathbf{a}} \mathbb{P}(x, \mathbf{b} \setminus \mathbf{a} \mid \mathbf{a}) \geq \min_{\mathbf{b}} \mathbb{P}(x \mid \mathbf{b}) \cdot \sum_{\mathbf{b} \setminus \mathbf{a}} \mathbb{P}(\mathbf{b} \setminus \mathbf{a} \mid \mathbf{a}) = \min_{\mathbf{b}} \mathbb{P}(x \mid \mathbf{b})$$

In particular, $\min_{\mathbf{a}} \mathbb{P}(x \mid \mathbf{a}) \geq \min_{\mathbf{b}} \mathbb{P}(x \mid \mathbf{b})$. □

**Practical considerations.** While there may be multiple choices of a candidate set $\mathbf{S}$ and Theorem 12 holds for any such chosen $\mathbf{S}$, one should pick one which maximizes $\alpha_{\mathbf{S}'} = \min_{\mathbf{s}'} \mathbb{P}(\mathbf{x} \mid \mathbf{s}')$ in practice, possibly by estimating $\alpha_{\mathbf{S}'}$ for each possible candidate set $\mathbf{S}'$ using existing samples. The subroutine `AdjustByS` (Algorithm 9) draws as many samples from $\mathbb{P}$ as necessary to achieve the required error tolerance. In practice, when given a fixed dataset $\mathcal{A}$ consisting of IID samples drawn from $\mathbb{P}$, one can simply estimate the quantities $R$ and $Q$ empirically using $\mathcal{A}$. In the setting of binary variables, one can interpret the bound of Theorem 12 as $|\mathbb{P}_{\mathbf{x}}(\mathbf{y}) - \widehat{\mathbb{P}}_{\mathbf{x}}(\mathbf{y})| \leq O\left(\max\left\{2^{|\mathbf{S}|}, 2^{\frac{|\mathrm{ND}(\mathbf{X})|}{4}}\right\} / \left(\alpha_{\mathbf{S}} \cdot \sqrt{|\mathcal{A}|}\right)\right)$.

## 5.5 Minimal size adjustment sets

As motivated in Section 5.1, it is often of interest to find an adjustment set of minimal size. For instance, in $\mathcal{G}_1$ of Figure 5.3, both $\text{pa}_{\mathcal{G}_1}(X) = \{A_1, \ldots, A_k\}$ and $\{Z\}$ are valid adjustment sets. However, the latter would provide better estimations with finite samples when $k \gg 1$. In this section, we show how to extend our methods to find a minimal sized adjustment set. In particular, we prove that this is possible given $\mathbf{Z} = \text{pa}_{\mathcal{G}'}(X)$ and a conditional independence oracle while giving error bounds when using approximate conditional independence testers in conjunction with `ApproxSearch`.

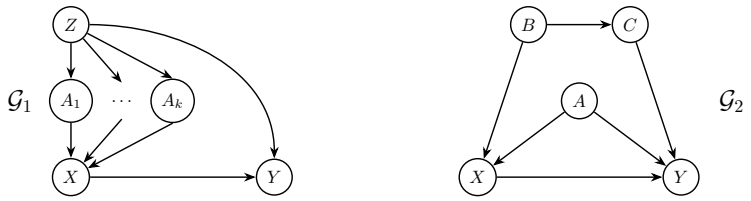



Figure 5.3: DAG $\mathcal{G}_1$ illustrates existence of adjustment sets of size significantly smaller than the parental set. Here, using $\{Z\}$ alone suffices. DAG $\mathcal{G}_2$ shows that the approach of Lemma 7 is incomplete with respect to *any* arbitrarily given adjustment set $\mathbf{Z}$. Here, $\mathbf{S} = \{A, B\}$ is also a valid adjustment set but $Y \not\perp\!\!\!\perp C \mid \{X, A, B\}$, so running these two condition checks may not recover a smaller sized adjustment set (even when it exists) when $\mathbf{Z} = \text{nd}_{\mathcal{G}_2}(\mathbf{X}) = \{A, B, C\}$ is used.

We begin with a lemma about how one can derive another valid adjustment set from a given one.

**Lemma 7.** *Given an arbitrary valid adjustment set $\mathbf{Z} \subseteq \mathbf{N}$, the subset $\mathbf{S} \subseteq \mathbf{N}$ is also a valid adjustment set if (i) $Y \perp\!\!\!\perp \mathbf{Z} \setminus \mathbf{S} \mid X \cup \mathbf{S}$ and (ii) $X \perp\!\!\!\perp \mathbf{S} \setminus \mathbf{Z} \mid \mathbf{Z}$.*

Note that $X \perp\!\!\!\perp \mathbf{S} \setminus \mathbf{Z} \mid \mathbf{Z}$ is satisfied if $\mathbf{Z} = \text{pa}_{\mathcal{G}}(X)$ as in Lemma 2 or whenever $\mathbf{S} \subseteq \mathbf{Z}$, e.g. in the case where $\mathbf{Z} = \mathbf{N}$. While this approach is always sound, it may not be complete in general with respect to any given adjustment set $\mathbf{Z}$; see $\mathcal{G}_2$ in Figure 5.3. The next lemma, however, tells us that this approach is complete with respect to $\mathbf{Z} = \text{pa}_{\mathcal{G}'}(\mathbf{X})$.

**Lemma 8.** *Suppose $\mathbf{Z} = pa_{\mathcal{G}'}(X)$. Then, for* any *valid adjustment set $\mathbf{S} \subseteq \mathbf{N}$, both (i) $Y \perp\!\!\!\perp \mathbf{Z} \setminus \mathbf{S} \mid X \cup \mathbf{S}$ and (ii) $X \perp\!\!\!\perp \mathbf{S} \setminus \mathbf{Z} \mid \mathbf{Z}$ hold.*

In the event where we only have access to $\varepsilon_1$-approximators and $\varepsilon_2$-approximate conditional independence testers, we can use the approximate parental-spousal set from the output of `ApproxSearch`.

Suppose $\mathbf{S} \subseteq \mathbf{N}$ was the approximate adjustment set output by `ApproxSearch`, where $X \perp\!\!\!\perp_{\varepsilon_2} \mathbf{N} \setminus \mathbf{S} \mid \mathbf{S}$. If we find another set $\mathbf{S}' \subseteq \mathbf{N}$ such that $\mathbf{Y} \perp\!\!\!\perp_{\varepsilon_2} \mathbf{S} \setminus \mathbf{S}' \mid X \cup \mathbf{S}'$, then $|\widehat{T}_{S'} - P_{\mathbf{x}}(\mathbf{y})| \leq \frac{\varepsilon_1}{\alpha_{\mathbf{S}'}} + \varepsilon_2 \cdot \left(\frac{1}{\alpha_{\mathbf{S} \cup \mathbf{S}'}} + \min\{\frac{2}{\alpha_{\mathbf{S}}}, \frac{1}{\alpha_{\mathbf{S} \cup \mathbf{S}'}}\}\right)$, where $\widehat{T}_{\mathbf{S}'}$ is the output of `AdjustByS` given $\mathbf{S}'$. Furthermore, we can restrict our search to only considers subsets of size $|\mathbf{S}'| \leq |\mathbf{S}|$.

Note that $\mathbf{X} \perp\!\!\!\perp_{\varepsilon_2} \mathbf{N} \setminus \mathbf{S} \mid \mathbf{S}$ in `ApproxSearch` and so $X \perp\!\!\!\perp_{\varepsilon_2} \mathbf{S}' \setminus \mathbf{S} \mid \mathbf{S}$ since $\mathbf{S}' \subseteq \mathbf{N}$. This is why Theorem 5.5 only needs to ensure that $\mathbf{Y} \perp\!\!\!\perp_{\varepsilon_2} \mathbf{S} \setminus \mathbf{S}' \mid X \cup \mathbf{S}'$. By Theorem 12,

we know that using `ApproxSearch` yields an estimate with $|\mathbb{P}_x(y) - \widehat{\mathbb{P}}_x(y)| \leq \frac{\varepsilon_1+\varepsilon_2}{\alpha_{\mathbf{S}}}$. So, the estimate from Theorem 5.5 may yield a smaller worst case error depending on how $\alpha_{\mathbf{S}\cup\mathbf{S}'}$ and $\alpha_{\mathbf{S}'}$ compare with $\alpha_{\mathbf{S}}$. Note that we have $\frac{1}{\alpha_{\mathbf{S}}} \leq \frac{1}{\alpha_{\mathbf{S}\cup\mathbf{S}'}}$ according to Lemma 6 and $|\mathbf{S}'| \leq |\mathbf{S}|$ by construction.

## 5.6 Sample complexity

Our theoretical guarantees (Theorem 12) relies on three quantities: $\alpha_{\mathbf{S}}$, $\varepsilon_1$, and $\varepsilon_2$. The first quantity $\alpha_{\mathbf{S}}$ depends on the chosen candidate set $\mathbf{S} \subseteq \mathbf{N}$ while the other two quantities $\varepsilon_1$ and $\varepsilon_2$ depend on the number of samples we have for the necessary estimation and testing tasks. The sample complexity of our approach depends on that of `EstimateMarginal`, `EstimateConditional`, and `ApproxCondInd`.

The theoretical guarantees on sample complexity of these three problems crucially depend on assumptions about the underlying probability distribution. All results in this setting are stated for the well-studied case of *discrete distributions*. We provide results in the PAC setting, where the algorithm succeeds with probability $1-\delta$ (abbreviated w.h.p. for *with high probability*). For simplicity, we hide multiplicative $\log(1/\delta)$ factors in the $\tilde{O}$ notation.

Our first result summarizes how many samples are sufficient so that, w.h.p., all approximate conditional independence tests succeed, guaranteeing that the set $\mathbf{S}$ satisfies the premise of Lemma 5. The results assumes the use of a particular method for `ApproxCondInd`, as we describe further in Appendix D.4.

**Lemma 9.** *Let $\varepsilon_2, \delta > 0$. Let $\mathbf{S}$ be the set picked by* `ApproxSearch` *in Line 7. Then, with probability $1-\delta$, we have $X \perp\!\!\!\perp_{\varepsilon_2} \mathbf{N} \setminus \mathbf{S} \mid \mathbf{S}$ as long as* `ApproxSearch` *takes $t_1$ samples for*

$$t_1 \in \tilde{O}\left(\frac{|\Sigma_X|^{1/2} \cdot |\Sigma_Y|^{1/2} \cdot |\Sigma_{\mathbf{N}}|^{1/2} \cdot k^* \cdot \log|\mathbf{N}||}{\varepsilon_2^2}\right),$$

*where $k^* := \min_{\mathcal{G}\in\mathcal{M}(\mathbb{P},\mathbf{N})} |pa_{\mathcal{G}}(X)|$.*

Our second result summarizes how many samples are sufficient so that, w.h.p., the estimation error from `AdjustByS` satisfies the conclusion of Lemma 4. The result holds in the *Poissonized* setting, and assumes the use of *empirical* estimators for `EstimateMarginal` and `EstimateConditional`, which we describe further in Appendix D.5.

**Lemma 10.** *Let $\varepsilon_1, \delta > 0$, $\mathbf{S} \subseteq \mathbf{V}$, $K = |\Sigma_{\mathbf{S}}|$, and $\alpha_{\mathbf{S}} = \min_{\mathbf{s}} \mathbb{P}(x \mid \mathbf{S} = \mathbf{s})$. Let $\widehat{T}_{\mathbf{S}}$ be the estimate returned by* `AdjustByS`*. Then, with probability $1-\delta$, we have $|\widehat{T}_{\mathbf{S}} - T_{\mathbf{S}}| \leq \varepsilon_1$ as long as* `AdjustByS` *takes $t_2$ samples for*

$$t_2 \in \tilde{O}\left(\frac{1}{\varepsilon_1^2 \alpha_{\mathbf{s}}} + \frac{K}{\varepsilon_1 \alpha_{\mathbf{S}}} \log(K)\right).$$

Putting these together, `ApproxSearch` (Algorithm 14) achieves the bounds in Theorem 12 with the following number of samples:

**Theorem 13.** *Let $\varepsilon_1, \varepsilon_2, \delta > 0$, let $\mathbf{S}$ be the set picked in Line 7 of* `ApproxSearch`*, and let $\alpha_{\mathbf{S}} = \min_{\mathbf{s}} \mathbb{P}(x \mid \mathbf{S} = \mathbf{s})$. Finally, let $\widehat{T}_{\mathbf{S}}$ be the estimate returned by* `ApproxSearch`*. Then, with probability $1 - \delta$, we have $|\widehat{T}_{\mathbf{S}} - \mathbb{P}_x(y)| \leq \frac{\varepsilon_1 + \varepsilon_2}{\alpha_{\mathbf{S}}}$, as long as* `ApproxSearch` *takes $t_1 + t_2$ samples for*

$$t_1 \in \tilde{O}\left(\frac{|\Sigma_X|^{1/2} \cdot |\Sigma_Y|^{1/2} \cdot |\Sigma_{\mathbf{N}}|^{1/2} \cdot k^* \cdot \log |\mathbf{N}|}{\varepsilon_2^2}\right) \quad \text{and} \quad t_2 \in \tilde{O}\left(\frac{1}{\varepsilon_1^2} + \frac{|\Sigma_{\mathbf{S}}|}{\varepsilon_1} \log |\Sigma_{\mathbf{S}}|\right) .$$

To estimate $\mathbb{P}_x(y)$ up to $\lambda$ error, Theorem 12 indicates that it suffices to have $\varepsilon_1 + \varepsilon_2 = \alpha_{\mathbf{S}} \lambda$. Given $\alpha_{\mathbf{S}}$, we show how to pick $\varepsilon_1$ and $\varepsilon_2$ to optimize sample complexity in our forthcoming paper.

## 5.7 Hardness results

In this section, we present two impossibility/hardness results that the knowledge of a non-descendant set and error dependency of $\Omega(\varepsilon_2/\alpha)$ using our approach is unavoidable.

**Lemma 11** (Necessity of knowing a non-descendant set $\mathrm{nd}_{\mathcal{G}'}(\mathbf{X})$)**.** *There exist two probability distributions $\mathbb{P}$ and $\mathbb{P}'$ on 3 binary variables $\{X, Y, Z\}$ that are generated according to topological orderings $\sigma : X \prec Z \prec Y$ and $\sigma' : Z \prec X \prec Y$ respectively such that $\mathbb{P}$ is indistinguishable from $\mathbb{P}'$ and adjusting for $\mathbb{P}_x(y)$ using $\sigma'$ yields an approximation error of 1/2 in the worst case. This holds even when we have a conditional independence oracle for $\mathbb{P}$ and we are given the Markov equivalence class of a graph $\mathcal{G}$ which $\mathbb{P}$ is Markov with respect to.*

Note that an expected error of 1/2 is trivial to obtain: flip a fair coin and then just always conclude that $\mathbb{P}(Y = y \mid \mathtt{do}(X = x))$ is the outcome of the coin toss. Suppose $\mathbb{P}(Y = y \mid \mathtt{do}(X = x)) = \alpha$, then the expected error is $(1/2) \cdot (1 - \alpha) + (1/2) \cdot \alpha = 1/2$.

**Lemma 12** (Necessity of error dependence on $\Omega(\varepsilon_2/\alpha)$)**.** *Suppose $0 < \sqrt{\varepsilon} \leq \alpha \leq 1/2$. There exists a probability distribution $\mathbb{P}$ defined over 4 binary variables $\{A, B, X, Y\}$ in a clique with a topological ordering of $A \prec B \prec X \prec Y$ such that: $X \perp\!\!\!\perp_{\varepsilon_2} B \mid A$, $\min_a \mathbb{P}(x \mid a) \geq \frac{\alpha}{2}$, $|\widehat{\mathbb{P}}_x(y) - \mathbb{P}_x(y)| \geq \frac{\varepsilon_2}{8\alpha}$, where $\{A, B\}$ is the correct adjustment set for $\mathbb{P}_x(y)$ but $\hat{\mathbb{P}}_x(y)$ is computed using $\{A\}$ as the adjustment set. This holds even if all conditional probabilities of $\mathbb{P}$ are given.*

## 5.8 Experiments

We demonstrate the performance of our `ApproxSearch` method on a synthetic dataset, showing that it outperforms natural baselines for causal effect estimation. Similarly, we restrict our attention to binary-valued random variables. We discuss our synthetic DAG generation procedure, implementation details, and hyperparameter selection in Appendix D.7. We compare against two baselines: `ND-Adjust`, which runs `AdjustByS` with $\mathbf{S} = \mathrm{nd}_{\mathcal{G}_d}(X)$, and `Pa-Adjust`, which runs `AdjustByS` with $\mathbf{S} = \mathrm{pa}_{\mathcal{G}_d}(X)$. Note that `Pa-Adjust` is infeasible with only $\mathrm{nd}_{\mathcal{G}^d}(X)$ given, we include it as an oracle for reference.

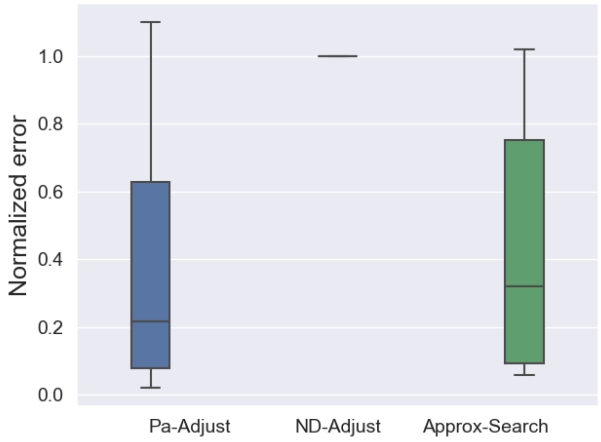



Figure 5.4: Performance, in terms of mean squared error (normalized by the mean squared error of `ND-Adjust`), averaged over 10 synthetic DAGs, with 5 runs per DAG.

**Results.** Our goal will be to understand the mean squared error of each method. For each $d \in [10]$, we sample datasets $\mathcal{D}_d^{(1)}, \ldots, \mathcal{D}_d^{(5)}$ of 500 samples from the $d$-th causal DAG. Then, for each method, we estimate $\mathbb{P}_{x=1}(y = 0)$ from all ten datasets, and estimate the mean-squared error of the method on the $d$-th causal DAG as the average squared error of these ten estimates. At this point, we obtain a single (estimated) MSE for each method, for each $d \in [100]$. To normalize the MSEs associated to the $d$-th causal DAG $\mathcal{G}_d$, we divide by the MSE of `ND-Adjust` on the same causal DAG $\mathcal{G}_d$. We plot the resulting normalized MSEs in Figure Figure 5.4. Interpreting these results, note that a normalized MSE below 1 implies that a method performs better on the $d$-th causal DAG than `ND-Adjust` does on that causal DAG. As expected, `Pa-Adjust` outperforms `ND-Adjust` in a majority of cases, indeed, so does `ApproxSearch`.

## 5.9 Conclusion and future work

While we only considered covariate adjustments in this chapter, the idea of doing causal effect estimation while bypassing causal graph discovery is generalizable to other estimation methods and settings. Following this chapter, there are several immediate open problems worthy of future investigations:

**Better search algorithms.** To find a set $\mathbf{S}$ such that $X \perp\!\!\!\perp_{\varepsilon_2} \mathbf{N} \setminus \mathbf{S} \mid \mathbf{S}$, can we search more efficiently than with the brute force approach in Algorithm 14? This relates to the classic problem of Markov blanket discovery, for which greedy approaches are only consistent under additional assumptions, e.g. in [GA21].

**Improving sample complexity.** In our forthcoming paper, we give the optimal choice of $\varepsilon_1$ and $\varepsilon_2$ in the oracle setting where $\alpha_{\mathbf{S}}$ is known. However, $\alpha_{\mathbf{S}}$ is generally unknown in advance and it is of immediate interest to develop a data-driven procedure for picking these hyperparameters, e.g., using estimates of $\alpha_{\mathbf{S}}$. Furthermore, the part of our sample complexity (Theorem 13) that depends on bounding $|\widehat{T}_{\mathbf{S}} - T_{\mathbf{S}}|$ depends quadratically on $|\Sigma_{\mathbf{S}}|$, whereas recent expectation bounds [Zen+24] suggest that a linear dependence might be possible. It

would be interesting to extend these expectation bounds to the PAC setting to obtain tighter guarantees.

**Expert knowledge.** Can we obtain more efficient tests beyond that of Theorem 13? This is clearly possible when $\mathbb{P}$ is known to have more structure, e.g. a linear Gaussian. What other kinds of expert knowledge can help to make these tests more efficient? In this chapter, we only considered knowledge of a set of non-descendants $\mathbf{N}$ of the treatment $X$.

# Chapter 6

# Conclusion

This thesis only scratches the surface of what I see as the critical necessity of much broader foundations for pragmatic data science. In Chapter 1, I introduced a framework and vocabulary that guides and organizes the work in this thesis; my hope is that this system will be a useful guide for future research. The framework was centered around the data-to-decision pipeline, which can be summarized in a sentence:

*A statistical analyst uses **Data** to select a **Model**, which is used to generate **Predictions** that ultimately inform **Decisions**.*

Chapter 1 works backwards through the pipeline to introduce key concepts: the *telos* of decision-making establishes the importance of causal predictions, which motivates the use of causal models (Section 1.2), which in turn motivates the study of identifiability from population-level data (Section 1.3). Sections 1.4 and 1.5 describe a variety of model classes that are used in the remainder of the thesis. Although the results in this thesis are specific to these model classes, they suggest research directions which are likely to be of wider importance, two of which I now briefly discuss.

**Expressivity, richness, and structure-aligned representations** In Section 1.3.2, we defined a problem setting $(\mathcal{M}, \rho)$ as a pair of a model class $\mathcal{M}$ and a forward mapping $\rho$. Then, we defined a difficulty relation over problem settings in terms of the more basic expressivity and richness relations over model classes and forward mappings, respectively.

The difficulty relation suggests two general approaches for taking a non-identifiable problem setting $(\mathcal{M}, \rho)$ and developing an associated problem setting $(\mathcal{M}', \rho')$ that *is* identifiable. We can either take a *restriction* approach by making $\mathcal{M}'$ less expressive than $\mathcal{M}$, or we can take an *expansion* approach by making $\rho'$ richer than $\rho$.[1] Though not referred to by name, these two approaches are already widely employed in identifiability theory (e.g. in nonlinear independent component analysis [Kiv+22; HST19]). As a first step for future work, it would be worth organizing existing identifiability results using the expressivity and richness relations. This exercise would not only make the relationship between results more transparent to practitioners, but would also almost certainly uncover more general principles that could guide future theory.

[1] In this thesis, we used both approaches: Chapter 2 takes the restriction approach (using $\mathcal{M}' = \mathcal{M}_{\text{lfcm}}(\mathbf{X})$), and Chapter 4 takes the expansion approach by introducing interventional data as a part of $\rho'$.

Such a general theory is likely to be quite useful, especially as the model classes used in data science continue to proliferate. For example, *causal representation models* (see Section 1.5.3) are a relatively recent addition to the "modeling zoo", and they are just a special case of a broader kind of model class that has been gaining traction. We may call these *structure-aligned representation models*, which combine more classical mathematical and computational objects (differential equations, equivariant functions, sparse coding models, symbolic programs, causal models, etc.) over latent variables $\mathbf{Z}$ with a flexible mixing function $g : \mathcal{Z} \to \mathcal{X}$ that generates the observed variables $\mathbf{X}$.

Structure-aligned representation (SAR) models are a subject of intense investigation in areas such as physics-informed machine learning, geometric deep learning, mechanistic interpretability, and neurosymbolic programming; with several anticipated benefits such as statistical and computational efficiency, improved extrapolation, and increased interoperability and interpretability. Given an appropriate shared vocabulary, there is a major opportunity for cross-pollination between these fields. On one hand, identifiability of SAR models is largely understudied outside of causal representation learning; it is likely that non-identifiability is often a significant unnoticed issue in these other fields. On the other hand, causal representation learning is currently limited by a lack of scalable methodology; insights from these other fields may be useful for designing neural architectures and learning paradigms for causal representation learning.

**Predictive identifiability and predictive estimation** In Section 1.3.5, we discussed predictive identifiability of a model class $\mathcal{M}$ with respect to a set of queries $\mathcal{Q}$. From the viewpoint of pragmatic data science, predictive identifiability is the "correct" notion of identifiability, as long as $\mathcal{Q}$ is appropriated defined for the decision-making task(s) that are downstream of the statistical analysis.

In traditional causal effect estimation, predictive identifiability is well-studied, e.g. in settings with instrumental variables and proxy variables. However, in causal structure learning and causal representation learning, predictive identifiability is relatively understudied, with model identifiability being the more common focus. This distinction also holds in the finite-sample setting. In causal effect estimation, the error in estimating "nuisance parameters" can often be made to have a small influence on the estimate of a target parameter; resulting in a semiparametric efficient estimator.[2] However, in causal structure learning, sample complexity results often focus on how many samples are needed to recover a causal graph $G^{M^\star}$, rather than to make accurate predictions.

Chapter 5 aims to bring the prediction-centric perspective to causal structure learning, but much more work is required on this front. To begin, Chapter 5 focuses on the single query $q : M \mapsto \mathbb{P}^M_a(Y = y)$; it would be valuable to obtain results for more general query sets, and for more general model classes (e.g. without knowledge of the nondescendants of $A$ or with latent variables). Furthermore, our sample complexity analysis (and thus, the optimal choice of hyperparameters) depends on potentially unknown problem parameters such as $k^* = |\text{pa}_{G^{M^\star}}(A)|$. It is essential to extend this algorithm and future such algorithms to be

[2]For example, let $M^\star = \{\mathbb{P}^{M^\star}(Y \mid A, \mathbf{S}), \mathbb{P}^{M^\star}(\mathbf{S})\}$ be the nuisance parameters and $\mathbb{P}^{M^\star}_a(Y) = \sum_{\mathbf{s}} \mathbb{P}^{M^\star}(Y \mid A, \mathbf{S}) \cdot \mathbb{P}^{M^\star}(\mathbf{S})$ be the target parameter. Then, targeted maximum likelihood estimation (TMLE) controls errors by an appropriate way of estimating nuisance parameters $\widehat{M}$, while the one-step estimator in semiparametric efficiency theory controls errors by changing how predictive inference is performed on the estimated $\widehat{M}$.

*adaptive* to unknown parameters, as in sparse linear regression. This task will be challenging: combining semiparametric efficiency with adaptive model selection is already technically difficult [Laa+23], and matters will be further complicated when considering general query sets (instead of singletons) and the complex nesting of graph-based model classes. However, this line of research is also likely to be quite rewarding, and to produce insights that will be of general importance in the development of pragmatic data science.

# Appendix A

# Appendix for Chapter 2

## A.1 Faithfulness assumptions are generic

We first recall the assumptions from Section 2.3.

**Assumption 1** (Cluster tetrad faithfulness)**.** *Suppose $X_i$ and $X_j$ are not in the same cluster. Then there exists some $\{u, v\}$ such that $t_{ij,uv} \neq 0$.*

**Assumption 2** (Parent tetrad faithfulness)**.** *Suppose $X_i$ and $X_j$ are in the same cluster, but $X_i$ has at least one child. Then there exists some $\{u, v\}$ such that $t_{ij,uv} \neq 0$.*

**Assumption 3** (Latent adjacency faithfulness)**.** *Suppose $X_i \to L_k$. Let $S_i = ch(pa(X_i)) \setminus \{i\}$ and $S' = \cup_{j \leq i} ch(L_i)$. Then $\rho_{i,k|S_i,S} \neq 0$ for some $X_k \in ch(L_k)$*

**Proposition 7.** *Assumption 1 holds generically.*

*Proof.* Let $L_i = \text{pa}(X_i)$ and $L_j = \text{pa}(X_j)$. By the triple child assumption, there exists some $X_u$ in the same cluster as $X_i$, and some $X_v$ in the same cluster as $X_j$. Then any set which t-separates $\{i, j\}$ and $\{u, v\}$ must contain $L_i$ and $L_j$, since $i$ must be separated from $u$ and $j$ must be separated from $v$, respectively. Therefore, by Theorem 1, $\text{rank}(\Sigma_{[ij],[uv]}) = 2$ generically, i.e., $t_{ij,uv} \neq 0$ generically. □

**Proposition 8.** *Assumption 2 holds generically.*

*Proof.* Let $L_i = \text{pa}(X_i) = \text{pa}(X_j)$. $L_i$ must have some other child $X_u$ by the triple child assumption. Let $L_v$ be some child of $X_i$, and $X_v$ be some child of $L_v$. Then any set which t-separates $\{i, j\}$ and $\{u, v\}$ must contain $L_i$ and $L_v$, since $i$ must be separated from $j$ and $i$ must be separated from $v$, respectively. Therefore, by Theorem 1, $\text{rank}(\Sigma_{[ij],[uv]}) = 2$ generically, i.e., $t_{ij,uv} \neq 0$ generically, which completes the proof. □

**Proposition 9.** *Assumption 3 holds generically.*

*Proof.* If $X_i \to L_k$, then $X_i$ and $X_k$ are d-connected given $S_i, S$ for any $X_k \in \text{ch}(L_k)$. [Spi+00] establish that if two nodes are d-connected, then their partial correlation in a linear SEM is generically nonzero, proving the desired result. □

## A.2 Proof of Theorem 2

**Theorem 2.** *Let $\mathcal{G}$ be a linear LFCM and let $\mathbb{X} \in \mathbb{R}^{n \times p}$ be a matrix of samples of the observed variables $X_1, \ldots, X_p$. Then Algorithm 1 is consistent under Assumptions 1, 2, and 3, i.e., as $n \to \infty$, we have $\mathbb{P}(\hat{\mathcal{G}} = \mathcal{G}) \to 1$.*

*Proof.* By Assumption 1 and Assumption 2, as long as at least two nodes are present from each cluster, if $t_{ij,uv} = 0$ for all pairs $u, v \in \{i, j\}$ iff. $X_i$ and $X_j$ have the same latent parent, and neither node has any children. Thus, if $i - j$ in $\mathcal{G}$ in Algorithm 2, then $i$ and $j$ are in the same cluster. Next, if $i$ and $j$ are both left in $R$ after the while loop, then they must be in the same cluster. For sake of contradiction, suppose not, and let $L_i = \text{pa}(X_i)$, $L_j = \text{pa}(X_j)$. Since $X_i$ remains, then by the double-parent assumption, there must also remain some other node $X_{i'}$ that is a child of $L_i$. Similarly, there must remain some other node $X_{j'}$ that is a child of $L_j$. However, then $|R| = 4$, a contradiction. Therefore, the clustering output by Algorithm 2 is a refinement of the true clustering. Furthermore, if a node $i$ is upstream of the cluster $C_1$, then $i$ necessarily has a child, and thus $t_{ij,uv} \neq 0$ for some $j, u, v$. Therefore, $i$ cannot be placed in any clique before the cluster $C_1$ is completely removed, and thus the ordering of cluster returned by Algorithm 2 is topologically consistent. By the double parent assumption, each cluster in $\pi$ from Algorithm 2 has size at least 2. Assumption 1 ensures that two clusters are merged by Algorithm 3 iff. they have the same latent parent. Finally, Assumption 2 ensures that $X_i \to L_k$ in $\hat{\mathcal{G}}$ if and only if $X_i \to L_k$ in $\mathcal{G}$. □

## A.3 Double-parent violation

Let $\mathcal{G}$ be the graph in Figure A.1. Suppose Algorithm 2 first removes the cluster $\{X_2, X_3, X_4\}$. Then, $X_1$ is only t-separated by from $\{X_5, X_6, X_7\}$ by $L_2$, so in the second round of Algorithm 2, it is clustered with these nodes instead of with $\{X_2, X_3, X_4\}$.

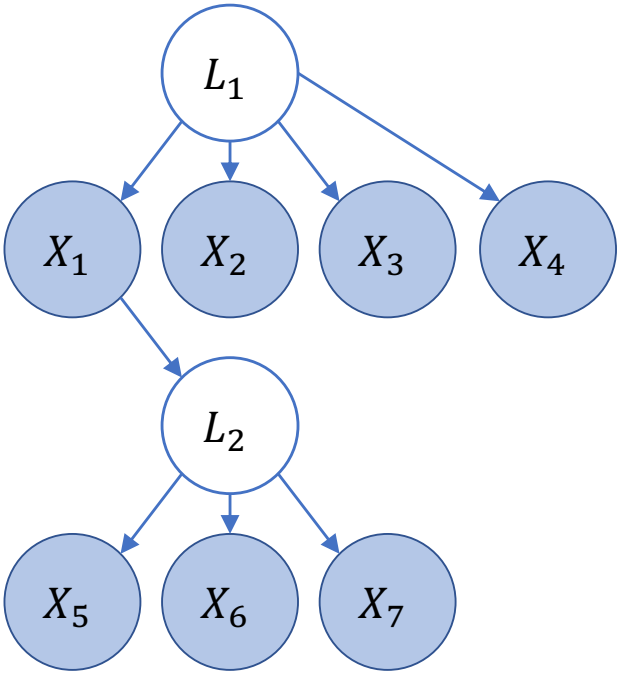



Figure A.1: Algorithm 1 can fail when the true graph violates the double parent assumption.

# Appendix B

# Appendix for Chapter 3

## B.1 Comparison with the panel data setting

[ASS20] consider the setting where one observes data associated with a collection of units or contexts (e.g., cell types). Here, there is a pre-intervention period where all units are observed under control (e.g., absence of any actions), followed by a post-intervention period where each unit experiences *one* intervention or action (e.g., compound). Under this particular sparsity pattern, the goal is to impute what would have occurred to *each* unit under *every* intervention in the post-intervention period. In contrast, our work considers a more general observation pattern that allows each unit to potentially experience *multiple* interventions, which is typically present in biological applications. Motivated by the typical data structures found in these domains, we extend the algorithm to regress along interventions (compared to units as is originally proposed), and provide suitable conditions under which the desired counterfactual outcomes are accurately recovered.

## B.2 Proof of Theorem 4

*Proof.* Consider any $(c, a) \notin \Omega$. Then by the linear factor model assumption and Assumption 4, for all $i \in \mathcal{C}$

$$\begin{aligned} \mathbf{x}^{ia} &= U^i \mathbf{v}^a && \text{(B.1)} \\ &= U^i \Big( \sum_{j \in \mathcal{A}(c)} \beta_j \mathbf{v}^j \Big) && \text{(B.2)} \\ &= \sum_{j \in \mathcal{A}(c)} \beta_j (U^i \mathbf{v}^j) && \text{(B.3)} \\ &= \sum_{j \in \mathcal{A}(c)} \beta_j \mathbf{x}^{ij}. && \text{(B.4)} \end{aligned}$$

Since (B.4) holds for all $i \in \mathcal{C}$, it necessarily holds for the target context $c$ and those contexts $i \in \mathcal{C}_{\text{train}} = \mathcal{C}(\mathcal{A}(c) \cup \{a\})$.

Now, let $\mathbf{v}_{\text{train}} \in \mathbb{R}^{|\mathcal{A}(c)| \times r_1}$ and $\mathbf{v}_{\text{test}} \in \mathbb{R}^{|\mathcal{A}(c)| \times r_2}$ denote the right singular vectors of $\mathbf{X}_{\text{train,donor}}$ and $\mathbf{X}_{\text{test,donor}}$, respectively, with $r_1 = \text{rank}(\mathbf{X}_{\text{train,donor}})$ and $r_2 = \text{rank}(\mathbf{X}_{\text{test,donor}})$.

Recall that $\widehat{\beta} = \mathbf{X}^{\dagger}_{\text{train,donor}}\mathbf{y}_{\text{train}}$. Since $\widehat{\beta} \in \text{rowspan}(\mathbf{X}_{\text{train,donor}}) = \text{span}(\mathbf{v}_{\text{train}})$ by design, it follows from (B.4) that $\widehat{\beta} = \mathbf{v}_{\text{train}}\mathbf{v}_{\text{train}}^T\beta$. Also, Assumption 5 implies that $\mathbf{v}_{\text{train}}\mathbf{v}_{\text{train}}^T\mathbf{v}_{\text{test}} = \mathbf{v}_{\text{test}}$. Combining these arguments yields

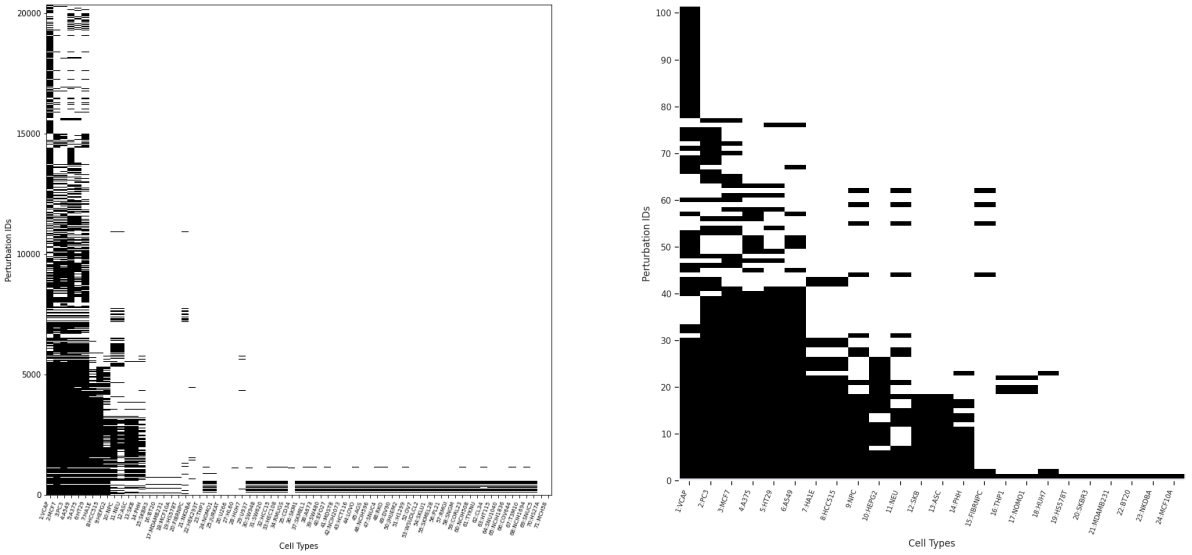
$$\widehat{\mathbf{x}}^{ca} = \mathbf{X}_{\text{test,donor}}\widehat{\beta} = \mathbf{X}_{\text{test,donor}}\mathbf{v}_{\text{train}}\mathbf{v}_{\text{train}}^T\beta = \mathbf{X}_{\text{test,donor}}\beta \tag{B.5}$$

$$= \sum_{j \in \mathcal{A}(c)} \beta_j \mathbf{x}^{cj} = \mathbf{x}^{ca}. \tag{B.6}$$

This completes the proof. □

## B.3 L1000 dataset

In Figure B.1a, we display the availability of gene expression signatures for each (cell type, chemical compound) pair. The cell types are sorted from left to right by the number of compounds for which gene expression signatures are available. Similarly, the compounds are sorted from bottom to top by the number of cell types for which gene expression signatures are available.

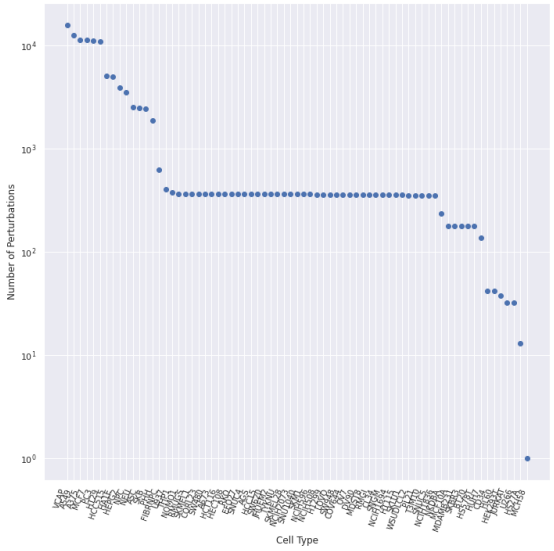



(a) Original dataset. (b) Subsampled dataset.

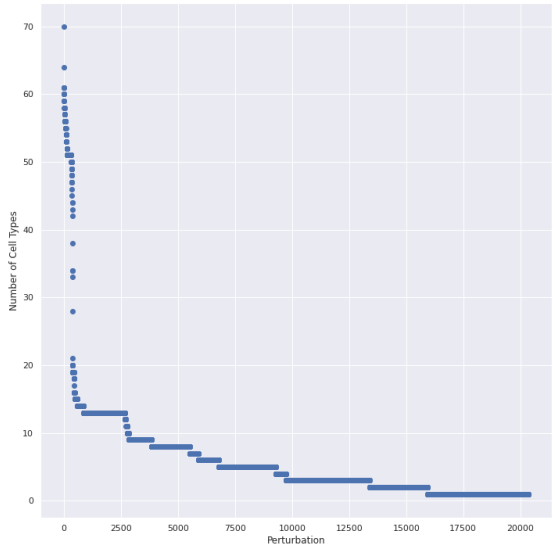
Figure B.1: Availability matrix for cell-type compound pairs. A black rectangle indicates that the gene expression profile is available, a white rectangle indicates that it is not.

In Figure B.2a, we display the total number of compounds for which gene expression signatures are available, for each cell type. The cell type with the most compounds available is VCAP, with 15,805 compounds available out of 20,369.

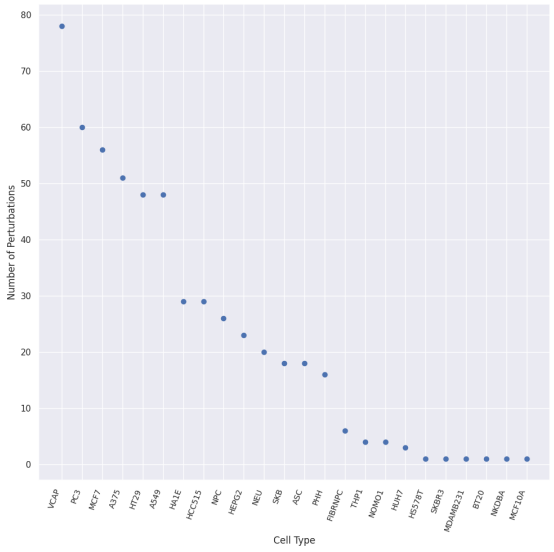
In Figure B.2b, we display the total number of cell types for which gene expression signatures are available, for each compound. The compound with the most cell types available is DMSO (control), which is available for 70 of 71 cell types, followed by BRD-A19037878, which is available for 64 out of 71 cell types.



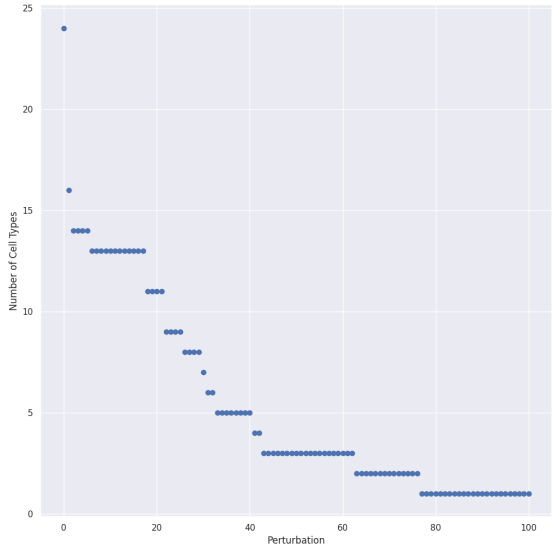

(a) Compounds per cell type in the original dataset.




(b) Cell types per compound in the original dataset.




(c) Compounds per cell type in the subsampled dataset.




(d) Cell types per compound in the subsampled dataset.

Figure B.2: Cross-sectional counts in the original and subsampled datasets.

**Data Selection.** We use the Level 2 data from L1000 dataset, which contains unnormalized gene expression values. The data is loaded using `cmapPy` [Ena+19], available under a 3-clause BSD license. The Level 2 data is split into two sets, "delta" and "epsilon", containing 49,216 and 1,278,882 samples, respectively. They differ in which landmark genes are used; we only use the larger "epsilon" dataset for consistency of our results.

We select 100 compounds at random to run all of our analyses over, in order to create a smaller but unbiased dataset. We show the plots corresponding to those that we showed for the whole dataset in Figures B.1b, B.2c, and B.2d; they qualitatively verify that the subsampled dataset is similar in character to the original.

## B.4 UMAP on VCAP data

In Figure B.3, we show the UMAP embedding of gene expression data from 70 different compounds in the VCAP cell line, which we picked since it has the greatest number of samples for any cell type in the dataset. Comparing to Figure 3.3, which included 70 different cell types, we see that gene expression vectors (even within a single cell type) cluster far less by compound than they do by cell type. Moreover, most compounds do not substantially differ from the control (DMSO). This suggests that we should expect estimating the cell-type specific compound effect will be a difficult task.

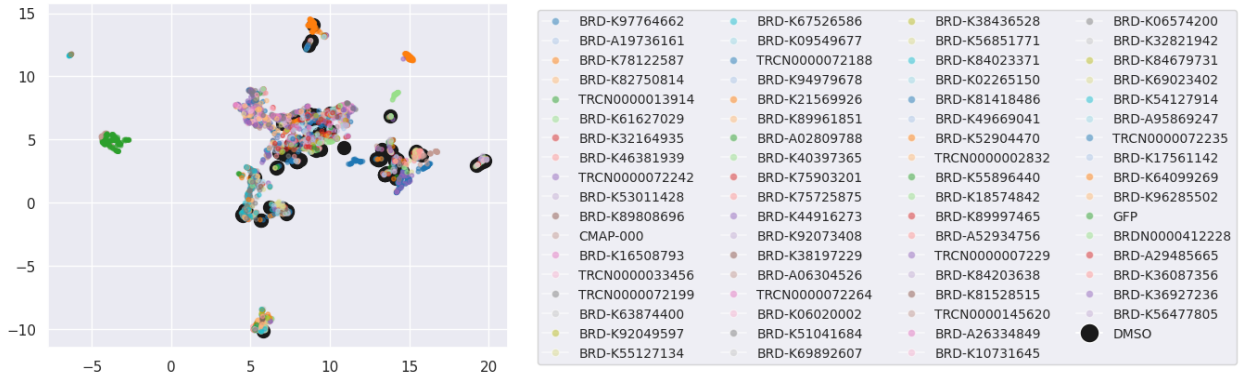



Figure B.3: UMAP embedding of gene expression data from the cell type VCAP.

## B.5 Baseline estimators

The *mean-over-contexts* estimator is defined as

$$\widehat{\mathbf{x}}^{ca}_{\text{avg-c}} = \frac{1}{|\mathcal{C}(a)|} \sum_{c' \in \mathcal{C}(a)} \mathbf{x}^{c'a}. \tag{B.7}$$

That is, we average all contexts which receive the target action.

The *two-way mean* estimator with parameter $\lambda_c \in [0, 1]$,

$$\widehat{\mathbf{x}}^{ca}_{\text{two-way}} = \lambda_c \widehat{\mathbf{x}}^{ca}_{\text{avg-c}} + (1 - \lambda_c) \widehat{\mathbf{x}}^{ca}_{\text{avg-a}}, \tag{B.8}$$

is a convex combination of the *mean-over-actions* and *mean-over-contexts* estimators.

Finally, the *fixed action effect estimator*, relative to action $a'$, is defined as

$$\widehat{\mathbf{x}}^{ca}_{\text{fae}} = \mathbf{x}^{ca'} + \widehat{\mathbf{s}}(a' \to a), \tag{B.9}$$

where

$$\widehat{\mathbf{s}}(a' \to a) = \frac{1}{|\mathcal{C}(a)|} \sum_{c \in \mathcal{C}(a)} (\mathbf{x}^{ca} - \mathbf{x}^{ca'}). \tag{B.10}$$

In particular, a natural choice for $a'$ is "control", i.e., no action.

## B.6 Additional empirical results

**Results with MICE** In Figure B.4, we reduce the number of randomly picked interventions from 100 to 20, so that MICE can be run in a reasonable amount of time (3 hours). MICE is run with the default parameters for the `IterativeImputer` class in `sklearn` as of May 28th, 2021, including `max_iter=10` and `tol=0.001`. See the `IterativeImputer` documentation for more details.

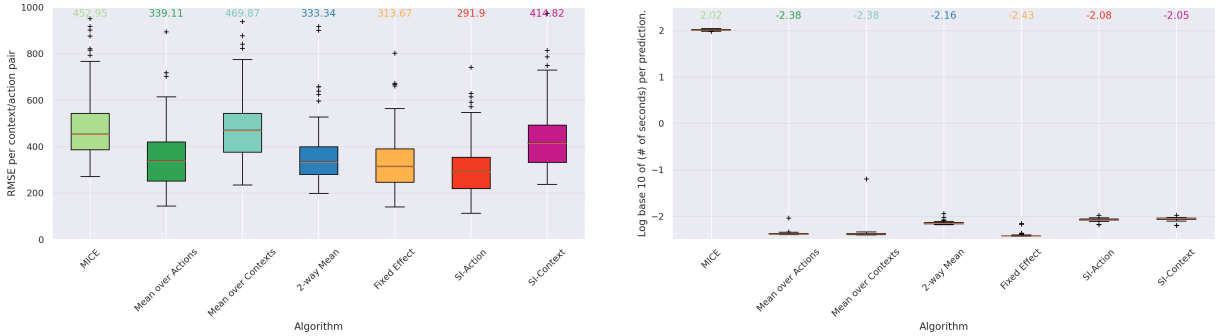
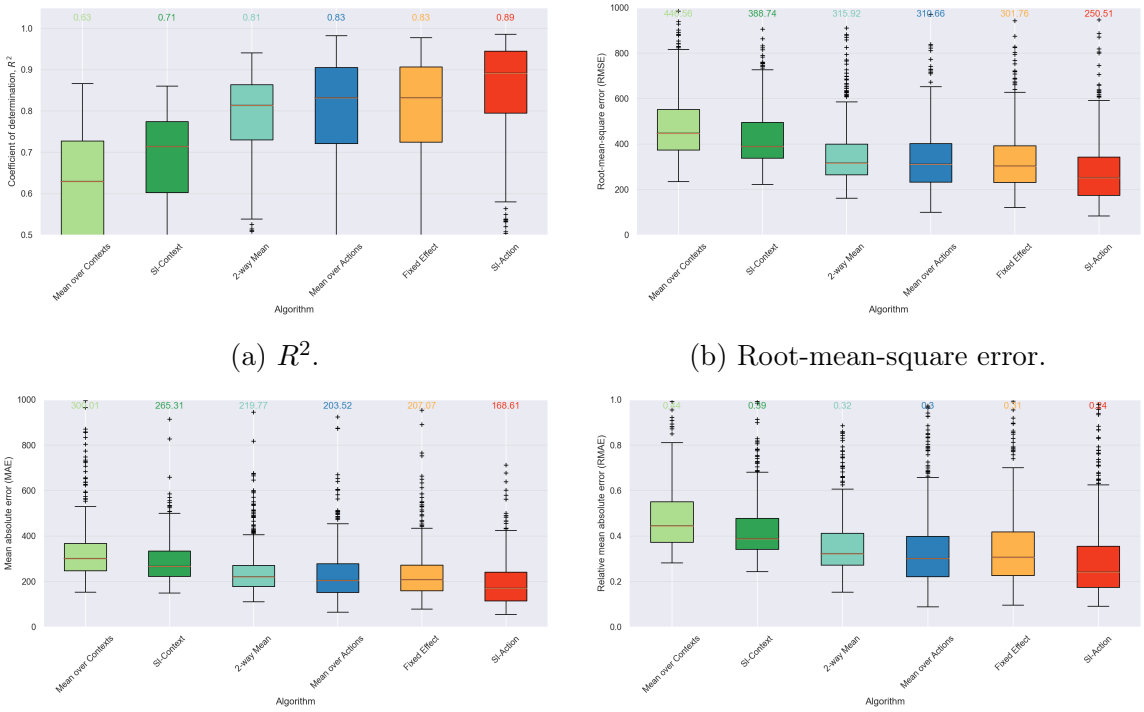



Figure B.4: MICE is roughly 4 orders of magnitude slower than `SI-A` and delivers has poor estimation performance compared to `SI-A` and the other baselines.

$R^2$ Figure B.5 demonstrates the performance on the leave-one-out (LOO) prediction task in Section 3.6 on various metrics. The relative ordering of method performance remains the same.

**Computation time** Figure B.6 demonstrates the time (on log scale) required for each leave-one-out (LOO) prediction task from Section 3.6. All methods are highly efficient, taking less than 0.1 seconds on almost all instances and roughly 0.01 seconds on most instances. The baselines are, on typical instances, roughly 2x faster than the methods based on synthetic interventions. The time required by synthetic interventions varies more due to the varying number of donor actions and training contexts.




(a) $R^2$.

(b) Root-mean-square error.

(c) Mean absolute error.

(d) Relative mean absolute error.

Figure B.5: The relative rankings of `SI-A` and baselines in terms of $R^2$ are similar across different metrics, with `SI-A` performing best, followed by mean-over-actions and the fixed effect estimators.

## B.7 Results on single samples

Figure B.7 shows the results of using `SI-A` on *unaveraged* data. In particular, for each cell type and compound, we select a single corresponding sample at random.

As described in Figure 3.4, we can apply some `ME` to de-noise our observations prior to regression. To this end, we consider the hard singular value thresholding (HSVT) estimator at energy level $\rho = 0.95$, that is, given the singular value decomposition $X = U\Sigma V^\top$, with singular values in decreasing order, we find that first $k$ such that $\sum_{i=1}^{k} \Sigma_{ii}^2 \geq \rho \|X\|_F^2$, and use $\text{HSVT}(X, k) = \sum_{i=1}^{k} \Sigma_{ii} U_i V_i^\top$.

We see that, as predicted by theory, matrix estimation improves the predictions on unaveraged data (`SI-A` vs. `SI-A`-HSVT). However, the results still do not match the performance of the simple mean-over-actions baseline.

Thus, prior to prediction, we perform the subspace inclusion hypothesis test described in Section 3.4. In particular, if $\widehat{\tau} \geq 0.1 \cdot \text{rank}(\mathbf{X}_{\text{test}})$, then we reject the hypothesis test, concluding that the SI method is unlikely to work. If the test passes, we use the `SI-A`-HSVT predictor; if it fails, we instead use the mean-across-actions predictor as a strong "fallback" option.

We see that adding this hypothesis test (`SI-A`-HSVT, +test) returns us to the performance level of the mean-across-actions baseline.

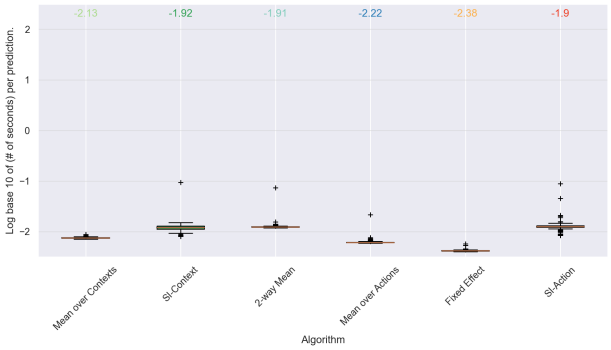



Figure B.6: Computation Times of `SI-A` and baselines. All methods are highly scalable, and `SI-A` is only about 2x slower than the simple baselines.

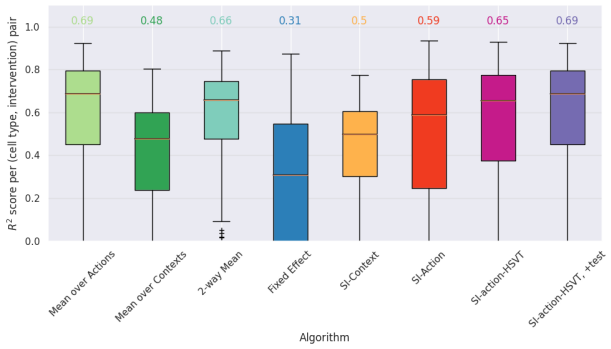


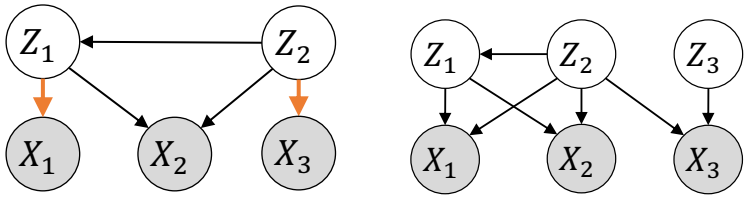

Figure B.7: Performance of causal imputation algorithms on recovering the effects of compounds in the CMAP dataset, using *unaveraged* data.

# Appendix C

# Appendix for Chapter 4

## C.1 Additional related work

Figure C.1 shows the two graphical conditions assumed in some prior works.




(a) Pure children (b) No twins

Figure C.1: **Graphical conditions assumed in prior works.** In **(a)**, the orange edges link pure children ($X_1$ and $X_3$) to their parents ($Z_1$ and $Z_3$, respectively). In **(b)**, the no twins assumption is satisfied since the observed children of $Z_1$, $Z_2$, and $Z_3$ are, respectively, $\{X_1, X_2\}$, $\{X_1, X_2, X_3\}$, and $\{X_3\}$, and these three sets are distinct.

## C.2 Non-generic soft interventions

We discuss the genericity condition in Assumption 7(b). We show that for soft interventions in which this genericity condition fails to hold, identifiability of the causal disentanglement problem as in Theorem 5 may fail. The following matrices satisfy all of Assumption 7, except for the genericity condition in Assumption 7(b), since $B_1^\top \mathbf{e}_1 = 2B_0^\top \mathbf{e}_1$:

$$B_0 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \qquad B_1 = \begin{bmatrix} 2 & 2 \\ 0 & 1 \end{bmatrix}, \qquad B_2 = \begin{bmatrix} 1 & 1 \\ 0 & 2 \end{bmatrix}, \qquad G = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Consider the alternative matrices

$$\widehat{B}_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad \widehat{B}_1 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}, \qquad \widehat{B}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}, \qquad \widehat{G} = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}.$$

These do not differ from the original tuple of matrices via a permutation. However, one can check that they are a valid solution, since

$$\Theta_0 = \widehat{\Theta}_0 = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}, \qquad \Theta_1 = \widehat{\Theta}_1 = \begin{bmatrix} 4 & 4 \\ 4 & 5 \end{bmatrix}, \qquad \Theta_1 = \widehat{\Theta}_1 = \begin{bmatrix} 1 & 1 \\ 1 & 5 \end{bmatrix}.$$

## C.3 Pseudoinverse of a covariance matrix

**Proposition 10.** *Let $X = GZ$ for $G \in \mathbb{R}^{p\times d}$ with full column rank. Assume $Cov(Z)$ is invertible and let $K := Cov(Z)^{-1}$. Then $Cov(X)^\dagger = H^\top K H$, where $H := G^\dagger$.*

*Proof.* The covariance matrices $\mathrm{Cov}(X)$ and $\mathrm{Cov}(Z)$ are related via $\mathrm{Cov}(X) = G\cdot\mathrm{Cov}(Z)\cdot G^\top$. The property $(UV)^\dagger = V^\dagger U^\dagger$ holds whenever $U$ has full column rank and $V$ has full row rank [Gre66]. The matrix $G$ has full column rank, $\mathrm{Cov}(Z)$ has full rank, and $G^\top$ has full row rank. Hence $\mathrm{Cov}(X)^\dagger = (G^\top)^\dagger \mathrm{Cov}(Z)^\dagger G^\dagger = H^\top K H$. □

## C.4 The partial order RQ decomposition

Recall the partial order RQ decomposition from Definition 4. We present Algorithm 10 to find the partial order RQ decomposition of a matrix. In Line 7, the normalize operator is $\mathrm{normalize}(\mathbf{v}) := \frac{\mathbf{v}}{\|\mathbf{v}\|_2}$. We let $\mathbf{0}_{d\times p}$ denote the $d \times p$ matrix of zeros and $Q_{j\succeq i}$ denote the submatrix of $Q$ on rows $j$ with $j \succeq i$. We say that a partial order $\prec$ is consistent with the total order $1, 2, \ldots, d$ if $i \prec j$ implies $i < j$. Any partial order can be put in this form by relabelling.

**Proposition 11.** *Let $H \in \mathbb{R}^{d\times p}$ be full rank and fix a partial order $\prec$ over $[d]$. Then there exists a unique partial order RQ decomposition of $H$. If $\prec$ is consistent with the total order $1, 2, \ldots, d$, the decomposition is returned by Algorithm 10.*

*Proof.* The matrix $Q_{j\succeq i}$ has fewer rows than columns and $\mathbf{h}_i \in \langle \mathbf{q}_j : j \succeq i\rangle$, by its construction in Algorithm 10. Hence the vector $\mathbf{r}$ of non-zero entries in the $i$-th row of $R$ is the unique solution to $Q_{j\succeq i}^\top \mathbf{r} = \mathbf{h}_i$. By construction, we see that $H = RQ$ and, moreover, $R_{ij} = 0$ if $j \not\succeq i$, and $\mathbf{q}_i$ is orthogonal to $\mathbf{q}_j$ for $i \prec j$. Furthermore, the entry $R_{ii}$ is positive, since $\mathbf{q}_i$ is the (normalized) projection of $\mathbf{h}_i$ onto $W_i$. □

## C.5 Further preliminaries for identifiability and reduction

We prove Fact 1 and discuss the reduction described in Section 4.3.1.

*Proof of Fact 1.* We have $B^\top B = B^\top\Big(\sum_{i=1}^d \mathbf{e}_i\mathbf{e}_i^\top\Big)B = \sum_{i=1}^d B^\top\big(\mathbf{e}_i\mathbf{e}_i^\top\big)B = \sum_{i=1}^d (B^\top\mathbf{e}_i)^{\otimes 2}$. □

**Algorithm 10** Partial Order RQ Decomposition

**Input:** Matrix $H \in \mathbb{R}^{d\times p}$, partial order $\prec$ over $[d]$ consistent with the total order $1, 2, \ldots, d$
**Output:** A partial order RQ decomposition $R$, $Q$
Let $R = \mathbf{0}_{d\times d}$, $Q = \mathbf{0}_{d\times p}$
**for** $i = 1, \ldots, d$ **do**
    Let $\mathbf{h}_i$ be the $i$-th row of $H$
    Let $W_i = \langle \mathbf{q}_j : j \succ i \rangle$
    Let $\mathbf{q}_i = \text{normalize}(\text{proj}_{W_i^\perp} \mathbf{h}_i)$ be the $i$-th row of $Q$
    Let $\mathbf{r} = (Q_{j\succeq i}^\top)^\dagger \mathbf{h}_i$
    For $i \preceq j$, let $R_{ij} = r_j$
**end for**
**return** $R$, $Q$

**Proposition 12.** *Consider the setup in Assumption 7. For $k, \ell \in [K]$, we have*

$$B_k^\top B_k - B_\ell^\top B_\ell = (B_k^\top \mathbf{e}_{i_k})^{\otimes 2} - (B_0^\top \mathbf{e}_{i_k})^{\otimes 2} - (B_\ell^\top \mathbf{e}_{i_\ell})^{\otimes 2} + (B_0^\top \mathbf{e}_{i_\ell})^{\otimes 2}, \tag{C.1}$$

*and thus*

$$\Theta_k - \Theta_\ell = (HB_k^\top \mathbf{e}_{i_k})^{\otimes 2} - (HB_0^\top \mathbf{e}_{i_k})^{\otimes 2} - (HB_\ell^\top \mathbf{e}_{i_\ell})^{\otimes 2} + (HB_0^\top \mathbf{e}_{i_\ell})^{\otimes 2}.$$

*Proof.* The proof follows the same steps as Proposition 2, together with the fact that

$$B_k^\top B_k - B_\ell^\top B_\ell = (B_k^\top B_k - B_0^\top B_0) - (B_\ell^\top B_\ell - B_0^\top B_0). \qquad \square$$

## C.6 Reduction

In this section, we show that a more general causal disentanglement problem can be simplified to one that satisfies our assumptions. We consider an unknown observational context, and multiple contexts with the same target. We focus here on the case of perfect interventions.

**Reducing to one intervention per node** We identify which contexts correspond to interventions on the same node. Thus, we can reduce to the case of one intervention per node by removing any redundant contexts. We do not use knowledge of which context is the observational context here.

**Proposition 13.** *Consider the setup in Assumptions 7 and 8. Assume generic parameters for $B_0$, $\lambda_k$, and $\lambda_\ell$. For $k, \ell \in [K]$, we have $r_{k,\ell} = 1$ if and only if $i_k = i_\ell$, where $r_{k,\ell} :=$ rank$(\Theta_k - \Theta_\ell)$.*

*Proof.* Since $H$ is full rank, we have $r_{k,\ell} = \text{rank}(B_k^\top B_k - B_\ell^\top B_\ell)$. Thus, we consider $\text{rank}(B_k^\top B_k - B_\ell^\top B_\ell)$. Suppose $i_k = i_\ell = i$. We have

$$B_k^\top B_k - B_\ell^\top B_\ell = \left(B_k^\top \mathbf{e}_i\right)^{\otimes 2} - \left(B_\ell^\top \mathbf{e}_i\right)^{\otimes 2},$$

by Equation (C.1) in Proposition 12. Both $B_k^\top \mathbf{e}_i$ and $B_\ell^\top \mathbf{e}_i$ have a single nonzero entry at the $i$-th coordinate, by Assumption 8. Thus $r_{k,\ell} = 1$.

Suppose $i_k \neq i_\ell$ and assume without loss of generality that $i_k < i_\ell$. Given a matrix $M$, let $M_U$ denote the submatrix of $M$ with rows and columns indexed by the elements of the set $U$. We have

$$\left(B_k^\top B_k - B_\ell^\top B_\ell\right)_{\{i_k,i_\ell\}} = \begin{bmatrix} \lambda_k^2 - (B_0)_{i_k,i_k}^2 & -(B_0)_{i_k,i_k}(B_0)_{i_k,i_\ell} \\ -(B_0)_{i_k,i_k}(B_0)_{i_k,i_\ell} & -\lambda_\ell^2 + (B_0)_{i_\ell,i_\ell}^2 - (B_0)_{i_k,i_\ell}^2 \end{bmatrix},$$

by Equation (C.1) in Proposition 12 and Assumption 8. For generic parameters, this submatrix has rank two, so the full matrix has rank at least two; i.e., $r_{k,\ell} \geq 2$. □

**Reducing to a known observational context** The previous section explains how to reduce to the case with one intervention per latent node. We may also reduce to the case with only one observational context: if more than one context is the observational context, they will all have the same inverse covariance matrix, so we may select only one of these contexts to serve as the observational context $k = 0$. Next we show that, with one intervention per node, and one observational context, we can identify the observational context. We show that the observational context has the "sparsest" changes from the other contexts. We formalize this intuition with the following definition.

**Definition 5.** *The* deviation score *of context $k$ is*

$$r_k := \sum_{\ell \in [K]\setminus\{k\}} r_{k,\ell},$$

*where $r_{k,\ell} := rank(\Theta_k - \Theta_\ell) for all\ k, \ell \in [K]$.*

**Proposition 14.** *Consider the setup in Assumption 7. Then $k^* \in \{0\} \cup [K]$ is an observational context if and only if $k^* = \arg\min_{k \in \{0\}\cup[K]} r_k$.*

*Proof.* Let source($\mathcal{G}$) denote the set of source nodes in $\mathcal{G}$. By Proposition 2, $r_{0,\ell} = 1 + \mathbb{1}_{\mathrm{pa}(i_\ell)\neq\varnothing}$ for all $\ell \in [K]$. Thus, $r_0 = 2K - |\mathrm{source}(\mathcal{G})|$.

For $k \neq 0$, we have $r_{k,\ell} \geq 2 + \mathbb{1}_{\mathrm{pa}(i_\ell)\neq\varnothing}$ for all $\ell \in [K] \setminus \{k\}$. Thus, $2K$. Since $\mathcal{G}$ must have at least one source node, we see that $r_k > r_0$ for all $k \neq 0$. □

## C.7 Hypothesis testing a necessary condition for model membership

We define the null hypothesis

$$H_0 : \mathrm{rank}(\Theta_k - \Theta_0) \leq 2 \qquad \forall\ k \in [K]$$

Assumption 7(b) implies that $H_0$ holds, by Proposition 2. The null hypothesis $H_0$ is a necessary condition for membership of $(\Theta_0, \Theta_1, \ldots, \Theta_K)$ in the model defined by Assumption 7. However, $H_0$ is not a sufficient condition for model membership: we may have $\mathrm{rank}(\Theta_k - \Theta_0) \leq 2$ for all $k \in [K]$, despite some interventions not targeting single nodes. For example, if $\mathcal{G}$ is

the empty graph, and all interventions have two targets, then $H_0$ holds. These cases may be ruled out with other conditions implied by model membership. We leave a membership test for our model to future work. Here, we focus on developing a test for $H_0$.

Prior work on testing latent variables models [DSS07; Squ+22b] use such rank constraints. To test whether a matrix $M \in \mathbb{R}^{p \times p}$ is rank $k$, one can test that all minors of size $k+1$ vanish; i.e., the collection of hypotheses

$$H_{A,B} : t_{A,B} = 0 \qquad A, B \subseteq [p], |A| = |B| = k+1$$

where $t_{A,B} := \det(M_{A,B})$.

For example, [Squ+22b] use this to test whether certain submatrices of a covariance matrix are rank one, as follows. Let $\widehat{M}$ be the sample covariance matrix computed from $n$ samples. If the underlying distribution is multivariate Gaussian, it is well-known that $\widehat{M}$ follows a Wishart distribution. Now, for each pair of subsets $A, B$, compute the empirical minor $\widehat{t}_{A,B} := \det(\widehat{M}_{A,B})$. Then, compute an estimate $\widehat{\texttt{Var}}(t_{A,B})$. Such an estimate can be obtained by evaluating the expression for $\texttt{Var}(t_{A,B})$ in [DMO08], which characterizes the moments of minors for Wishart matrices. Given this estimate, compute the z-score $z_{A,B} = \widehat{t}_{A,B} / \sqrt{\widehat{\texttt{Var}}(t_{A,B})}$. By typical asymptotic theory, $z_{A,B} \to \mathcal{N}(0,1)$ as $n \to \infty$, so we can use the z-score to compute an asymptotically correct p-value. Finally, the p-values for all pairs of subsets $A, B$ can be aggregated into a single p-value using a multiple hypothesis testing procedure such as Bonferroni correction or Sidak adjustment.

In principle, a similar procedure can be performed to test our null hypothesis $H_0$. However, under a Gaussianity assumption, $\Theta_k$ and $\Theta_0$ follow *inverse* Wishart distributions, rather than Wishart distribution. This would require expressions for the moments of minors for *inverse* Wishart matrices. We leave such a hypothesis test, which could give guarantees on false discovery rate control, to future work. Instead, we demonstrate the performance of a simple hypothesis test based on the singular values of the matrix $\widehat{\Theta}_k - \widehat{\Theta}_0$. Let

$$\rho_2(M) := \left(\sigma_1^2(M) + \sigma_2^2(M)\right) / \left(\sum_{i=1}^{p} \sigma_i^2(M)\right)$$

If $\text{rank}(M) \leq 2$, then $\rho_2(M) = 1$, otherwise, $\rho_2(M) < 1$. Thus, we may test $H_0$ by checking where $\rho_2(\widehat{\Theta}_k - \widehat{\Theta}) > \tau$ for some threshold $\tau$ near 1.

We demonstrate the performance of this procedure for testing model membership. We generate 500 random models following Assumption 7, using the same hyperparameters as in Section 4.4. These models satisfy $H_0$. We also generate 500 random models where the interventions target two nodes instead of one. For each $k$, we pick intervention targets $I_k \subset [d]$ with $|I_k| = 2$, uniformly at random among all subsets of size two. We hold all other hyperparameters of the simulation fixed. We consider only $n = 2500$ samples, the smallest sample size used in Section 4.4, and vary the threshold $\tau$ from 0.97 to 0.999, linearly spaced over 20 values. The singular value based test is able to determine model membership at a rate well above random guessing, see Figure C.2.

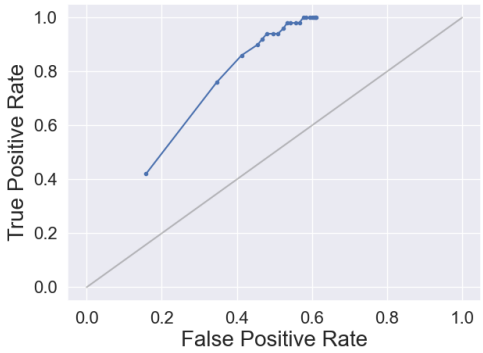



Figure C.2: **Performance of our singular-value-based hypothesis test for $H_0$, the hypothesis that all precision matrix differences are of rank at most two.** The gray line indicates the performance of randomly guessing. Our test performs thresholding on the value $\rho_2(\widehat{\Theta}_k - \widehat{\Theta}_0)$. Here, we vary the threshold from 0.97 to 0.999.

## C.8 Computational complexity

Algorithm 7 takes as input $K$ precision matrices in $\mathbb{R}^{p\times p}$. If these are each computed from at most $n$ samples, then the total cost is $\mathcal{O}(Kp^3 + Knp^2)$. Algorithm 6 runs for $K$ rounds and computes at most $2K$ projections per round. Each projection costs $\mathcal{O}(p^3)$, so the cost of this step is $\mathcal{O}(K^2p^3)$. In the remainder of Algorithm 7, we perform $\mathcal{O}(K)$ matrix multiplications and Cholesky decompositions. Each matrix multiplication costs $\mathcal{O}(p^2)$ and each Cholesky decomposition costs $\mathcal{O}(p^3)$. The other operations (e.g. selecting rows) in Algorithm 7 are negligible. Therefore, the overall runtime of Algorithm 7 is dominated by Algorithm 6, with a total cost of $\mathcal{O}(K^2p^3)$.

## C.9 Proofs

*Proof.* For (i), it suffices to find $\widehat{H}$ and $\{\widehat{B}_k\}_{k=1}^K$ such that $\widehat{B}_k\widehat{H} = B_kH$ for all $k \in [K]$, and such that there is no $\sigma \in S(\mathcal{G})$ satisfying $\widehat{B}_k = P_\sigma B_k P_\sigma^\top$, $H = P_\sigma H$, by Equation (4.2). Suppose $i_k \neq 1$ for any $k$. Let

$$\widehat{B}_k = \begin{bmatrix} \text{—} & \mathbf{e}_1 & \text{—} \\ & (B_k)_{2:d,1:d} & \end{bmatrix}, \qquad \widehat{H} = \begin{bmatrix} (B_0)_1^\top H \\ H_{2:d,1:p} \end{bmatrix}.$$

Then, for all $k$, we have $\widehat{B}_k\widehat{H} = B_kH$. Suppose that $Z_1$ has at least one parent, i.e., $(B_0)_{1j} \neq 0$ for at least one $j > 1$. Then the first row of $\widehat{H}$ is a nonzero combination of at least two rows of $H$. Hence it is not equal to a single row of $H$, since $H$ is full rank. Thus, $\widehat{H}$ is not equal to $H$ up to any permutation of rows.

For (ii), observe that for the stated example, the partial order $\prec$ given by $\widehat{B}_0$ differs in general from the partial order $\prec_{\mathcal{G}}$, since $Z_1$ has no predecessors in $\prec$. □

## C.10 Non-identifiability for imperfect interventions

**Parameter non-identifiability** In this section, we show that Assumption 8 is necessary to identify $H$. Let

$$B_0 = \begin{bmatrix} (B_0)_{11} & (B_0)_{12} \\ 0 & (B_0)_{22} \end{bmatrix}, \qquad B_1 = \begin{bmatrix} (B_1)_{11} & (B_1)_{12} \\ 0 & (B_0)_{22} \end{bmatrix},$$
$$B_2 = \begin{bmatrix} (B_0)_{11} & (B_0)_{12} \\ 0 & (B_2)_{22} \end{bmatrix}, \qquad H = \begin{bmatrix} 1 & H_{12} \\ 0 & 1 \end{bmatrix}.$$

Then, for any value $\widehat{H}_{12} \in \mathbb{R}$, we have $B_k H = \widehat{B}_k \widehat{H}$ for all $k$, where

$$\widehat{B}_0 = \begin{bmatrix} (B_0)_{11} & (B_0)_{12} + (B_0)_{11} H_{12} - (B_0)_{11} \widehat{H}_{12} \\ 0 & (B_0)_{22} \end{bmatrix}, \qquad \widehat{B}_1 = \begin{bmatrix} (B_1)_{11} & (B_1)_{12} + (B_1)_{11} H_{12} - (B_1)_{11} \widehat{H}_{12} \\ 0 & (B_0)_{22} \end{bmatrix},$$
$$\widehat{B}_2 = \begin{bmatrix} (B_0)_{11} & (B_0)_{12} + (B_0)_{11} H_{12} - (B_0)_{11} \widehat{H}_{12} \\ 0 & (B_2)_{22} \end{bmatrix}, \qquad \widehat{H} = \begin{bmatrix} 1 & \widehat{H}_{12} \\ 0 & 1 \end{bmatrix}.$$

**Graph non-identifiability** Suppose that $\mathcal{G}$ has edges $2 \to 1$, $3 \to 2$, and $3 \to 1$. Then the weight matrices have the form

$$B_0 = \begin{bmatrix} (B_0)_{11} & (B_0)_{12} & (B_0)_{13} \\ 0 & (B_0)_{22} & (B_0)_{23} \\ 0 & 0 & (B_0)_{33} \end{bmatrix}, \qquad B_1 = \begin{bmatrix} (B_0)_{11} & (B_0)_{12} & (B_0)_{13} \\ 0 & (B_1)_{22} & (B_1)_{23} \\ 0 & 0 & (B_0)_{33} \end{bmatrix},$$
$$B_2 = \begin{bmatrix} (B_0)_{11} & (B_0)_{12} & (B_0)_{13} \\ 0 & (B_2)_{22} & (B_2)_{23} \\ 0 & 0 & (B_0)_{33} \end{bmatrix}, \qquad B_3 = \begin{bmatrix} (B_0)_{11} & (B_0)_{12} & (B_0)_{13} \\ 0 & (B_0)_{22} & (B_0)_{23} \\ 0 & 0 & (B_3)_{33} \end{bmatrix},$$
$$H = \begin{bmatrix} 1 & H_{12} & H_{13} \\ 0 & 1 & H_{23} \\ 0 & 0 & 1 \end{bmatrix}.$$

Let $\widehat{\mathcal{G}}$ be the DAG with edges $2 \to 1$ and $3 \to 2$. We show that there exist $\widehat{H}$ and matrices $\widehat{B}_k$, following the support of $\widehat{\mathcal{G}}$, such that $B_k H = \widehat{B}_k \widehat{H}$ for all $k$. Note that $\mathcal{G}$ and $\widehat{\mathcal{G}}$ have the same transitive closure. Let $\widehat{H}_{12} \in \mathbb{R}$ and let $\widehat{H}_{13}, \widehat{H}_{23}$ be a solution to the system of equations:

$$\begin{bmatrix} (B_0)_{11} & (B_0)_{12} + (B_0)_{11}(H_{12} - \widehat{H}_{12}) \\ (B_1)_{11} & (B_1)_{12} + (B_1)_{11}(H_{12} - \widehat{H}_{12}) \end{bmatrix} \begin{bmatrix} \widehat{H}_{13} \\ \widehat{H}_{23} \end{bmatrix} = \begin{bmatrix} (B_0)_{11} H_{13} + (B_0)_{12} H_{23} + (B_0)_{13} \\ (B_1)_{11} H_{13} + (B_1)_{12} H_{23} + (B_1)_{13} \end{bmatrix}$$

This system generically has a solution, since for generic parameters the matrix on the left hand side is rank two. Then a solution, with all matrices $\widehat{B}_k$ have vanishing entry $(1, 3)$, is as

follows.

$$\widehat{B}_0 = \begin{bmatrix} (B_0)_{11} & (B_0)_{12} + (B_0)_{11}(H_{12} - \widehat{H}_{12}) & 0 \\ 0 & (B_0)_{22} & (B_0)_{23} + (B_0)_{22}(H_{23} - \widehat{H}_{23}) \\ 0 & 0 & (B_0)_{33} \end{bmatrix}$$

$$\widehat{B}_1 = \begin{bmatrix} (B_1)_{11} & (B_1)_{12} + (B_1)_{11}(H_{12} - \widehat{H}_{12}) & 0 \\ 0 & (B_0)_{22} & (B_0)_{23} + (B_0)_{22}(H_{23} - \widehat{H}_{23}) \\ 0 & 0 & (B_0)_{33} \end{bmatrix}$$

$$\widehat{B}_2 = \begin{bmatrix} (B_0)_{11} & (B_0)_{12} + (B_0)_{11}(H_{12} - \widehat{H}_{12}) & 0 \\ 0 & (B_2)_{22} & (B_2)_{23} + (B_2)_{22}(H_{23} - \widehat{H}_{23}) \\ 0 & 0 & (B_0)_{33} \end{bmatrix}$$

$$\widehat{B}_3 = \begin{bmatrix} (B_0)_{11} & (B_0)_{12} + (B_0)_{11}(H_{12} - \widehat{H}_{12}) & 0 \\ 0 & (B_0)_{22} & (B_0)_{23} + (B_0)_{22}(H_{23} - \widehat{H}_{23}) \\ 0 & 0 & (B_3)_{33} \end{bmatrix}$$

## C.11 Comparison to [Liu+22]

We compare Theorem 6 to [Liu+22]. We restate their main result for convenience and notation.

**Theorem** (Theorem 4.1 of [Liu+22]). *Let $Z = A_k Z + \varepsilon_k$ and $X = g(Z) + \varepsilon_x$. Let $\boldsymbol{\eta}_k$ be the sufficient statistic for the distribution of $Z$ in environment $k$. That is, $\boldsymbol{\eta}_k = \mathrm{vec}(\tilde{\Theta}_k)$, where $\tilde{\Theta}_k$ denotes the inverse covariance matrix of $Z$ in the $k$th setting and* vec *denotes the vectorization of a matrix. We assume that* vec *ignores zeros and repetitions. Assume that*

*(i)* $\{x \in \mathcal{X} \mid \varphi_{\varepsilon_x}(x) = 0\}$ *has measure zero, where $\varphi_{\varepsilon_x}$ is the characteristic function for $\varepsilon_x$,*

*(ii) $g$ is bijective, and*

*(iii) There exists $K + 1$ environments such that the following matrix is invertible:*

$$L = \begin{bmatrix} | & | & & | \\ \boldsymbol{\eta}_1 - \boldsymbol{\eta}_0 & \boldsymbol{\eta}_2 - \boldsymbol{\eta}_0 & \dots & \boldsymbol{\eta}_K - \boldsymbol{\eta}_0 \\ | & | & & | \end{bmatrix}.$$

*Then we can recover $g$ up to permutation and scaling.*

First, we show that (i) and (ii) hold in our setting. Our assumption that $X = GZ$ for $G$ invertible guarantees (ii). Our assumption that $X$ is a deterministic function of $Z$ corresponds to taking $\varepsilon_x \sim \delta_0$, i.e., $\varepsilon_x = 0$ with probability one. The characteristic function is $\varphi_\varepsilon(t) = 1$, thus satisfying (i).

We now show that (iii) is only satisfied in our setting when the number of edges in the latent graph is at most $d$. The vector $\mathrm{vec}(\tilde{\Theta}_k - \tilde{\Theta}_0)$ is of length $d + |E|$, where $|E|$ is the number of edges in the graphical model defined by $\Theta_0$. To be invertible, $L$ must be a square matrix, and hence we require $K \geq d + |E|$. If $|E| > d$, then $K > 2d$, and we must have an

**Algorithm 11** IdentifyPartialOrderFiniteSample

**Hyperparameters:** $\gamma$
**Input:** Precision matrices $(\Theta_0, \Theta_1, \ldots, \Theta_K)$
**Output:** Factor $\widehat{Q}$, partial order $\prec$
Let $\mathcal{I}_0 = \{\}$, $\widehat{Q} = \mathbf{0}_{d\times d}$
**for** $t = 1, \ldots K$ **do**
    Let $M_k = \text{proj}_{\widehat{Q}_{t-1}^{\perp}}(\Theta_k - \Theta_0)$ for each $k \notin \mathcal{I}_{t-1}$
    Let $\rho_k = \sigma_1^2(M_k)/(\sum_{i=1}^p \sigma_i^2(M_k))$ for each $k \notin \mathcal{I}_{t-1}$
    Pick $k \in \arg\max_{k\notin\mathcal{I}_{t-1}} \rho_k$
    Let $\widehat{\mathbf{q}}_k, \mathcal{A} =$ IdentifyAncestorsFiniteSample$(\Theta_k, \Theta_0, \{\widehat{\mathbf{q}}_i\}_{i\in\mathcal{I}_{t-1}}; \gamma)$
    Add $a' \succ k$ for any $a' \succeq a$, $a \in \mathcal{A}$
    Let $\mathcal{I}_t = \mathcal{I}_{t-1} \cup \{k\}$, $\widehat{Q}_t = [\widehat{\mathbf{q}}_k; \widehat{Q}_{t-1}]$
**end for**
**return** $\widehat{Q}$, $\prec$

intervention target $i$ such that $i = i_k$ for at least three values of $k$. We have $\tilde{\Theta}_k = B_k^\top B_k$, and thus

$$\tilde{\Theta}_k - \tilde{\Theta}_1 = (\lambda_k \mathbf{e}_{i_k})^{\otimes 2} - (B_0^\top \mathbf{e}_{i_k})^{\otimes 2} - (\lambda_\ell \mathbf{e}_{i_1})^{\otimes 2} + (B_0^\top \mathbf{e}_{i_1})^{\otimes 2}$$

Given $k_1$, $k_2$, and $k_3$ such that $i_{k_1} = i_{k_2} = i_{k_3} = i$, we see that $\boldsymbol{\eta}_{k_1}, \boldsymbol{\eta}_{k_2}$ and $\boldsymbol{\eta}_{k_3}$ differ only at position $(i, i)$. The space of vectors that differ in at most one entry is at most two-dimensional. Thus $\boldsymbol{\eta}_{k_1}$, $\boldsymbol{\eta}_{k_2}$, and $\boldsymbol{\eta}_{k_3}$ are not linearly independent, and $L$ is not invertible.

**Algorithm 12** IdentifyAncestorsFiniteSample

**Hyperparameters:** $\gamma$
**Input:** $\Theta_k$, $\Theta_0$, $\{\widehat{\mathbf{q}}_i\}_{i\in\mathcal{I}}$
**Output:** Vector $\widehat{\mathbf{q}}_k$, ancestor set $\mathcal{A}$
Let $\mathcal{A} = \mathcal{I}$
**for** $i \in \mathcal{I}$ **do**
    Let $W_{\neg i} = [\widehat{\mathbf{q}}_i : j \in \mathcal{I} \setminus \{i\}]$
    Let $M_{\neg i} = \text{proj}_{W_i^{\perp}}(\Theta_k - \Theta_0)$
    Let $\rho_k = \sigma_1^2(M_k)/(\sum_{i=1}^p \sigma_i^2(M_k))$ for each $k \notin \mathcal{I}_{t-1}$
    If $\rho_k \geq \gamma$, let $\mathcal{A} = \mathcal{A} \setminus \{i\}$
**end for**
Let $W = [\widehat{\mathbf{q}}_a : a \in \mathcal{A}]$
Let $M = \text{proj}_{W^{\perp}}(\Theta_k - \Theta_0)$
Let $\widehat{\mathbf{q}}_k$ be the (normalized) leading left singular vector of $M$
**return** $\widehat{\mathbf{q}}_k$, $\mathcal{A}$

## C.12 Finite-sample algorithms

**Matrix rank scoring.** In Line 7 of Algorithm 5 and Line 7 of Algorithm 6, we check whether a subspace is rank one. In the finite-sample setting, we represent these subspaces by matrices

**Algorithm 13** IterativeDifferenceProjectionFiniteSample

**Hyperparameters:** $\gamma$
**Input:** Precision matrices $(\Theta_0, \Theta_1, \ldots, \Theta_K)$
**Output:** $\widehat{H}$, $(\widehat{B}_0, \widehat{B}_1, \ldots, \widehat{B}_K)$
Let $d = K$
Let $\widehat{Q}, \prec =$ IdentifyPartialOrderFiniteSample$((\Theta_0, \Theta_1, \ldots, \Theta_K); \gamma)$
Let $\widehat{C}_k =$ cholesky$((\widehat{Q}^\dagger)^\top \Theta_k \widehat{Q}^\dagger)$ for $k = 0, \ldots, K$
Let $\mathcal{N} = [p]$
Let $\widehat{R} = I_d$
**for** $k = 1, \ldots, K$ **do**
    Let $\widehat{D}_k = \widehat{C}_k - \widehat{C}_0$
    Pick $\widehat{i}_k \in \arg\max_{i \in \mathcal{N}} \|(\widehat{D}_k)_i\|_2$
    Let $\mathcal{N} = \mathcal{N} \setminus \{\widehat{i}_k\}$
    Let $\widehat{R}_{\widehat{i}_k} = (\widehat{D}_k)_{\widehat{i}_k} + (\widehat{C}_0)_{\widehat{i}_k}$
**end for**
Let $\widehat{H}' = \widehat{R}\widehat{Q}$
Let $\widehat{H} = \Lambda \widehat{H}'$, for $\Lambda$ diagonal such that $\widehat{H}$ satisfies the conditions on $H$ in Assumption 7(c)
Let $\widehat{B}_0 =$ cholesky$((\widehat{H}^\dagger)^\top \Theta_0 \widehat{H}^\dagger)$
Let $\widehat{B}_k = \widehat{B}_0 + \mathbf{e}_{\widehat{i}_k} \left( |\widehat{\Lambda}_{\widehat{i}_k, \widehat{i}_k}| \mathbf{e}_{\widehat{i}_k} - \widehat{B}_0^\top \mathbf{e}_{\widehat{i}_k} \right)^\top$ for $k = 1, \ldots, K$
**return** $\widehat{H}$, $(\widehat{B}_0, \widehat{B}_1, \ldots, \widehat{B}_K)$

and measure how close the matrices are to rank one.

We use the score $\rho(M) := \sigma_1^2(M) / \left(\sum_{i=1}^p \sigma_i^2(M)\right)$, where $\sigma_i(M)$ is the $i$th largest singular value of $M$; $\rho(M)$ can be interpreted as the percentage of *spectral energy* associated to the largest singular value of $M$. Using this score to choose the next element of the partial order does not require hyperparameters, see Line 8 of Algorithm 11. In contrast, using this score to prune the set of ancestors requires a hyperparameter to determine whether a matrix is close enough to rank one, see $\gamma$ in Line 8 of Algorithm 12. Larger values of $\gamma$ (e.g., 0.999) result in a more conservative algorithm and will output a denser latent graph, while smaller values of $\gamma$ (e.g., 0.8) result in more aggressive pruning of the latent graph.

**Picking $\mathbf{q}_k$.** The matrix $M$ in Line 12 of Algorithm 12 is not guaranteed to be rank one in the finite-sample case. We instead select the leading left singular vector of $M$.

**Picking nonzero rows.** In the finite-sample case, the matrix $\widehat{D}_k$ will not usually have only one nonzero row, see Line 9 of Algorithm 13. We estimate the intervention target $i_k$ by picking the row of largest norm. Since we assume that $i_k$ is distinct for distinct $k$, we maintain a set $\mathcal{N}$ of candidate intervention targets and do not allow replicates.

## C.13 Code and Data

Our code can be found at

https://github.com/csquires/linear-causal-disentanglement-via-interventions.

**Optimizing over $S(\mathcal{G})$** Consider a partial order $\prec_{\mathcal{G}}$, a set of true intervention targets $i_1, \ldots, i_K$, and a set of estimated intervention targets $\widehat{i}_1, \ldots, \widehat{i}_K$. The integer linear program (C.2) computes the topological order $\pi^*$ consistent with $\prec_{\mathcal{G}}$ that maximizes the number of agreements between $i_k$ and $\widehat{i}_k$. The topological order $\pi^*$ can be recovered by letting $\pi^*(i) = j$ for the unique $j$ such that $A_{ij} = 1$. The first two lines of constraints ensure this uniqueness, and that $\pi^*(i) \neq \pi^*(i')$ for $i \neq i'$.

The final line of constraints ensures that $\pi^*$ is consistent with $\prec_{\mathcal{G}}$. If $\pi^*$ is not consistent with $\prec_{\mathcal{G}}$, then there exists $i, i', j$ such that $i \prec_{\mathcal{G}} i'$, $A_{i'j} = 1$, and $A_{ij'} = 0$ for all $j' \leq j$, which violates the constraint $\sum_{j' \leq j} (A_{ij'} - A_{i'j'}) \geq 0$.

$$\begin{aligned} \max_{A_{ij} \in \{0,1\}^{d \times d}} & \sum_{k=1}^{K} A_{i_k \widehat{i}_k} \\ \text{s.t.} & \sum_{i=1}^{d} A_{ij} = 1 \quad \forall\, j \in [d] \\ & \sum_{j=1}^{d} A_{ij} = 1 \quad \forall\, i \in [d] \\ & \sum_{j' \leq j} (A_{ij'} - A_{i'j'}) \geq 0 \quad \forall\, i \prec_{\mathcal{G}} i', \forall\, j \in [d] \end{aligned} \tag{C.2}$$

**Additional information on real data** The scRNA-seq dataset of [Urs+22] is available at:

https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE161824.

The TCGA dataset of [Liu+18] is available at:

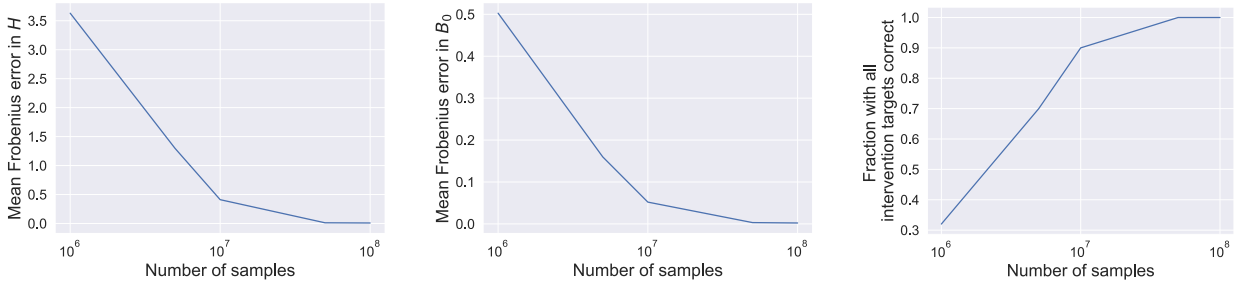
https://gdc-hub.s3.us-east-1.amazonaws.com/download/TCGA-LUAD.survival.tsv, and

https://gdc-hub.s3.us-east-1.amazonaws.com/download/TCGA-LUAD.htseq_fpkm.tsv.gz.

*Processing.* We use `EnrichR` [Kul+16] to pick the $p = 100$ and $p = 83$ most variable genes in the proliferation regulation gene set from the Gene Ontology. We use the values from the processed dataset from [Urs+22]; the only additional processing removes cells which were assigned to synonymous mutations (i.e., those that do not change any amino acids and hence do not have structural effects).

*Semi-synthetic analysis.* Our algorithm recovers the problem parameters for the semi-synthetic data, see Figure C.3.

*Comparison to TCGA dataset.* Our survival analysis is performed using the Cox proportional hazards model from the `lifelines` package [Dav19]. To correct for multiple hypothesis testing, we use the Benjamini-Hochberg procedure from the `statsmodels` package [SP10].




(a) Error in estimating $H$ (b) Error in estimating $B_0$ (c) Intervention targets

Figure C.3: **(Semi-synthetic) The adapted version of Algorithm 7 is consistent for recovering $H$, $B_0$, and $\{i_k\}_{k=1}^K$ from semi-synthetic data.** At each sample size, we generate 50 datasets. Note the logarithmic scale on the x-axis. In **(a)**, we plot the mean of $\|\widehat{H} - H\|_2$, the error in Frobenius norm. In **(b)**, we plot the mean of $\|\widehat{B}_0 - B_0\|_2$. In **(c)**, we plot the fraction of models where all intervention targets were correctly estimated.

# Appendix D

# Appendix for Chapter 5

## D.1 Stronger results under faithfulness

The next lemma is useful in improving efficiency of conditional independence checks under the assumption of faithfulness.

**Lemma 13.** *Let* $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ *be disjoint subsets of variables. Under the causal faithfulness assumption, if* $\mathbf{A} \perp\!\!\!\perp \mathbf{B} \mid \mathbf{C}$ *and* $\mathbf{A} \perp\!\!\!\perp \mathbf{D} \mid \mathbf{C}$*, then* $\mathbf{A} \perp\!\!\!\perp (\mathbf{B} \cup \mathbf{D}) \mid \mathbf{C}$.

*Proof.* Suppose, for a contradiction, that $\mathbf{A} \not\perp\!\!\!\perp (\mathbf{B} \cup \mathbf{D}) \mid \mathbf{C}$. Under the causal faithfulness assumption, this means that there is a d-connected path $P$ from some $A \in \mathbf{A}$ to some $V \in \mathbf{B} \cup \mathbf{D}$ that is active with respect to $\mathbf{C}$. Without loss of generality, due to symmetry of the statement, suppose that $V \in \mathbf{B}$. That is, $P$ is a path from $A \in \mathbf{A}$ to some $V \in \mathbf{V}$ that is active with respect to $\mathbf{C}$. But such an active path $P$ contradicts the assumption that $\mathbf{A} \perp\!\!\!\perp \mathbf{B} \mid \mathbf{C}$. Contradiction. □

Note that Lemma 13 is *false* in general with respect to unfaithful distributions.

**Example 4.** *The simple 3-variable distribution* $Z = X \oplus Y$*, where* $X$ *and* $Y$ *are independent fair coin flips is unfaithful to any DAG on 3 nodes. To see why, observe that any two variables are unconditionally independent but completely dependent upon conditioning on the third. So, one would minimally have to use a v-structure, say* $X \to Z \leftarrow Y$ *to represent this. However,* $X \to Z$ *is an active path which implies* $X \not\perp\!\!\!\perp Z$ *under the causal faithfulness assumption, which is a contradiction.*

### D.1.1 Improving sample complexity via Lemma 13

Recall from Algorithm 14 that we need to perform conditional independence checks of the form $X \perp\!\!\!\perp \mathbf{N} \setminus \mathbf{S} \mid \mathbf{S}$, which could potentially involve up to $n$ variables. Meanwhile, we know from Section 5.6 that the required sample complexity typically increases as the number of variables involved increases. Under causal faithfulness, Lemma 13 allows us to perform checks with bounded number of variables: we can check $X \perp\!\!\!\perp V \mid \mathbf{S}$ for each $V \in \mathbf{N} \setminus \mathbf{S}$ and then apply Lemma 13 to conclude that $X \perp\!\!\!\perp \mathbf{N} \setminus \mathbf{S} \mid \mathbf{S}$.

## D.2 Adjusting via parents

**Lemma 2.** *If $\mathbb{P}$ is Markov with respect to a DAG $\mathcal{G}$ and $\mathbf{Z} = pa_{\mathcal{G}}(X) \subseteq \mathbf{V}$, then for any $\mathbf{S} \subseteq nd_{\mathcal{G}}(X)$, we have $\mathbb{P}_x(y) = \sum_{\mathbf{z} \cup \mathbf{s}} \mathbb{P}(y \mid x, \mathbf{z} \cup \mathbf{s}) \cdot \mathbb{P}(\mathbf{z} \cup \mathbf{s})$.*

*Proof.* Using the do-calculus (see Section 5.2.1),

$$\begin{aligned}
\mathbb{P}_x(y) &= \sum_{\mathbf{z} \cup \mathbf{s}} \mathbb{P}(y \mid \mathtt{do}(x), \mathbf{z} \cup \mathbf{s}) \cdot \mathbb{P}(\mathbf{z} \cup \mathbf{s} \mid \mathtt{do}(x)) && \text{(Law of total prob.)} \\
&= \sum_{\mathbf{z} \cup \mathbf{s}} \mathbb{P}(y \mid x, \mathbf{z} \cup \mathbf{s}) \cdot \mathbb{P}(\mathbf{z} \cup \mathbf{s} \mid \mathtt{do}(x)) && \text{(Rule 2)} \\
&= \sum_{\mathbf{z} \cup \mathbf{s}} \mathbb{P}(y \mid x, \mathbf{z} \cup \mathbf{s}) \cdot \mathbb{P}(\mathbf{z} \cup \mathbf{s}) && \text{(Rule 3)}
\end{aligned}$$

Recall that $\mathcal{G}_{\underline{X}}$ removes all outgoing edges from $X$. In the second line, we invoke Rule 2 of the do-calculus with $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{W}) = (Y, X, \mathbf{Z} \cup \mathbf{S}, \varnothing)$, using the fact that $Y$ and $X$ are d-separated by $\mathbf{Z} \cup \mathbf{S}$ in $\mathcal{G}_{\underline{X}}$, since $\mathrm{pa}_{\mathcal{G}}(X) = \mathbf{Z} \subseteq \mathbf{Z} \cup \mathbf{S} \subseteq \mathrm{ND}(X)$. In the third line, we invoke Rule 3 of the do-calculus with $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{W}) = (\mathbf{Z} \cup \mathbf{S}, X, \varnothing, \varnothing)$, using the fact that $\mathbf{Z} \cup \mathbf{S}$ and $X$ are d-separated in $\mathcal{G}_{\overline{X}}$ since $\mathbf{Z} \cup \mathbf{S} \subseteq \mathbf{N}$. □

**Lemma 3.** *Suppose $\mathbb{P}$ is Markov with respect to a DAG $\mathcal{G}$ and $\mathbf{Z} = pa_{\mathcal{G}}(X) \subseteq \mathbf{V}$, then for any $\mathbf{S} \subseteq ND_{\mathcal{G}}(X)$, we have*

$$\mathbb{P}_x(y) = \sum_{\mathbf{s}} \mathbb{P}(\mathbf{s}) \cdot \mathbb{P}(y \mid x, \mathbf{s}) \cdot \sum_{\mathbf{z} \setminus \mathbf{s}} \mathbb{P}(\mathbf{z} \setminus \mathbf{s} \mid x, y, \mathbf{s}) \cdot \frac{\mathbb{P}(x \mid \mathbf{s}) \cdot \mathbb{P}(\mathbf{z} \setminus \mathbf{s} \mid \mathbf{s})}{\mathbb{P}(x, \mathbf{z} \setminus \mathbf{s} \mid \mathbf{s})}$$

*Proof.* Continuing from Lemma 2, we see that

$$\begin{aligned}
&\mathbb{P}_x(y) \\
&= \sum_{\mathbf{z} \cup \mathbf{s}} \mathbb{P}(y \mid x, \mathbf{z} \cup \mathbf{s}) \cdot \mathbb{P}(\mathbf{z} \cup \mathbf{s}) && \text{(By Lemma 2)} \\
&= \sum_{\mathbf{z} \cup \mathbf{s}} \frac{\mathbb{P}(y, \mathbf{z} \setminus \mathbf{s} \mid x, \mathbf{s})}{\mathbb{P}(\mathbf{z} \setminus \mathbf{s} \mid x, \mathbf{s})} \cdot \mathbb{P}(\mathbf{z} \cup \mathbf{s}) && \text{(since } \mathbb{P}(y \mid x, \mathbf{z} \cup \mathbf{s}) = \tfrac{\mathbb{P}(y, \mathbf{z} \setminus \mathbf{s} \mid x, \mathbf{s})}{\mathbb{P}(\mathbf{z} \setminus \mathbf{s} \mid x, \mathbf{s})}\text{)} \\
&= \sum_{\mathbf{z} \cup \mathbf{s}} \frac{\mathbb{P}(y \mid x, \mathbf{s}) \cdot \mathbb{P}(\mathbf{z} \setminus \mathbf{s} \mid x, y, \mathbf{s})}{\mathbb{P}(\mathbf{z} \setminus \mathbf{s} \mid x, \mathbf{s})} \cdot \mathbb{P}(\mathbf{z} \cup \mathbf{s}) \\
&&& \text{(since } \mathbb{P}(y, \mathbf{z} \setminus \mathbf{s} \mid x, \mathbf{s}) = \mathbb{P}(y \mid x, \mathbf{s}) \cdot \mathbb{P}(\mathbf{z} \setminus \mathbf{s} \mid x, y, \mathbf{s})\text{)} \\
&= \sum_{\mathbf{z} \cup \mathbf{s}} \frac{\mathbb{P}(y \mid x, \mathbf{s}) \cdot \mathbb{P}(\mathbf{z} \setminus \mathbf{s} \mid x, y, \mathbf{s}) \cdot \mathbb{P}(x \mid \mathbf{s})}{\mathbb{P}(x, \mathbf{z} \setminus \mathbf{s} \mid \mathbf{s})} \cdot \mathbb{P}(\mathbf{z} \cup \mathbf{s}) && \text{(since } \mathbb{P}(\mathbf{z} \setminus \mathbf{s} \mid x, \mathbf{s}) = \tfrac{\mathbb{P}(x \mid \mathbf{s})}{\mathbb{P}(x, \mathbf{z} \setminus \mathbf{s} \mid \mathbf{s})}\text{)} \\
&= \sum_{\mathbf{z} \cup \mathbf{s}} \frac{\mathbb{P}(y \mid x, \mathbf{s}) \cdot \mathbb{P}(\mathbf{z} \setminus \mathbf{s} \mid x, y, \mathbf{s}) \cdot \mathbb{P}(x \mid \mathbf{s})}{\mathbb{P}(x, \mathbf{z} \setminus \mathbf{s} \mid \mathbf{s})} \cdot \mathbb{P}(\mathbf{z} \setminus \mathbf{s} \mid \mathbf{s}) \cdot \mathbb{P}(\mathbf{s}) \\
&&& \text{(since } \mathbb{P}(\mathbf{z} \cup \mathbf{s}) = \mathbb{P}(\mathbf{z} \setminus \mathbf{s} \mid \mathbf{s}) \cdot \mathbb{P}(\mathbf{s})\text{)} \\
&= \sum_{\mathbf{s}} \mathbb{P}(\mathbf{s}) \cdot \mathbb{P}(y \mid x, \mathbf{s}) \cdot \sum_{\mathbf{z} \setminus \mathbf{s}} \mathbb{P}(\mathbf{z} \setminus \mathbf{s} \mid x, y, \mathbf{s}) \cdot \frac{\mathbb{P}(x \mid \mathbf{s}) \cdot \mathbb{P}(\mathbf{z} \setminus \mathbf{s} \mid \mathbf{s})}{\mathbb{P}(x, \mathbf{z} \setminus \mathbf{s} \mid \mathbf{s})} \\
&&& \text{(Rearrange summation)}
\end{aligned}$$

□

**Lemma 4.** *Let $\widehat{T}_{\mathbf{S}}$ be the output of* `AdjustByS`*. Then, $|\widehat{T}_{\mathbf{S}} - T_{\mathbf{S}}| \leq \varepsilon_1/\alpha_{\mathbf{S}}$.*

*Proof.* We first separate the total error into the sum two terms: one term which only depends on the error in estimating $\mathbb{P}(\mathbf{s})$, and another term which depends only on the error in estimating $\mathbb{P}(y \mid x, \mathbf{s})$.

$$\begin{aligned}
\left|\widehat{T}_{\mathbf{S}} - T_{\mathbf{S}}\right| &= \left|\sum_{\mathbf{s}} R(\mathbf{s}) \cdot Q(y \mid x, \mathbf{s}) - \mathbb{P}(\mathbf{s}) \cdot \mathbb{P}(y \mid x, \mathbf{s})\right| \\
&= \left|\sum_{\mathbf{s}} R(\mathbf{s}) \cdot Q(y \mid x, \mathbf{s}) - \mathbb{P}(\mathbf{s}) \cdot Q(y \mid x, \mathbf{s}) + \mathbb{P}(\mathbf{s}) \cdot Q(y \mid x, \mathbf{s}) - \mathbb{P}(\mathbf{s}) \cdot \mathbb{P}(y \mid x, \mathbf{s})\right| \\
&\qquad \text{(Add 0)} \\
&= \left|\sum_{\mathbf{s}} (R(\mathbf{s}) - \mathbb{P}(\mathbf{s})) \cdot Q(y \mid x, \mathbf{s}) + \mathbb{P}(\mathbf{s}) \cdot (Q(y \mid x, \mathbf{s}) - \mathbb{P}(y \mid x, \mathbf{s}))\right| \\
&\leq \left|\sum_{\mathbf{s}} (R(\mathbf{s}) - \mathbb{P}(\mathbf{s})) \cdot Q(y \mid x, \mathbf{s})\right| + \left|\sum_{\mathbf{s}} \mathbb{P}(\mathbf{s}) \cdot (Q(y \mid x, \mathbf{s}) - \mathbb{P}(y \mid x, \mathbf{s}))\right| \\
&\qquad \text{(Triangle inequality)}
\end{aligned}$$

For the term depending on the error in estimating $\mathbb{P}(\mathbf{s})$, we have

$$\begin{aligned}
\left|\sum_{\mathbf{s}} (R(\mathbf{s}) - \mathbb{P}(\mathbf{s})) \cdot Q(y \mid x, \mathbf{s})\right| &\leq \sum_{\mathbf{s}} \left|R(\mathbf{s}) - \mathbb{P}(\mathbf{s})\right| \cdot Q(y \mid x, \mathbf{s}) \qquad \text{(Triangle inequality)} \\
&\leq \sum_{\mathbf{s}} \left|R(\mathbf{s}) - \mathbb{P}(\mathbf{s})\right| \quad \text{(Since } 0 \leq Q(y \mid x, \mathbf{s}) \leq 1 \text{ for all } \mathbf{s}) \\
&\leq \varepsilon_1/2 \qquad \text{(By the guarantee of } \texttt{EstimateMarginal})
\end{aligned}$$

For the term depending on the error in estimating $\mathbb{P}(y \mid x, \mathbf{s})$, we have

$$\begin{aligned}
\left|\sum_{\mathbf{s}} \mathbb{P}(\mathbf{s}) \cdot \Big(Q(y \mid x, \mathbf{s}) - \mathbb{P}(y \mid x, \mathbf{s})\Big)\right| &\leq \sum_{\mathbf{s}} \mathbb{P}(\mathbf{s}) \cdot \left|Q(y \mid x, \mathbf{s}) - \mathbb{P}(y \mid x, \mathbf{s})\right| \\
&\qquad \text{(Triangle inequality)} \\
&\leq \frac{\varepsilon_1}{2\alpha_{\mathbf{S}}} \\
&\qquad \text{(By the guarantee of } \texttt{EstimateConditional})
\end{aligned}$$

Thus, $|\widehat{T}_{\mathbf{S}} - T_{\mathbf{S}}| \leq \frac{\varepsilon_1}{2} + \frac{\varepsilon_1}{2\alpha_{\mathbf{S}}} \leq \frac{\varepsilon_1}{\alpha_{\mathbf{S}}}$, completing the proof. □

**Lemma 5.** *Let $\mathbf{S} \subseteq \mathbf{N}$ be such that $X \perp\!\!\!\perp_{\varepsilon_2} \mathbf{N} \setminus \mathbf{S} \mid \mathbf{S}$. Then, $|T_{\mathbf{S}} - \mathbb{P}_x(y)| \leq \frac{\varepsilon_2}{\alpha_{\mathbf{S}}}$.*

*Proof.* From Lemma 3, we have

$$\mathbb{P}_x(y) = \sum_{\mathbf{s}} \mathbb{P}(\mathbf{s}) \cdot \mathbb{P}(y \mid x, \mathbf{s}) \cdot \sum_{\mathbf{z} \setminus \mathbf{s}} \mathbb{P}(\mathbf{z} \setminus \mathbf{s} \mid x, y, \mathbf{s}) \cdot \frac{\mathbb{P}(x \mid \mathbf{s}) \cdot \mathbb{P}(\mathbf{z} \setminus \mathbf{s} \mid \mathbf{s})}{\mathbb{P}(x, \mathbf{z} \setminus \mathbf{s} \mid \mathbf{s})}$$

For notational simplicity, let us define

$$\gamma_{x,y,\mathbf{z} \cup \mathbf{s}} = \sum_{\mathbf{z} \setminus \mathbf{s}} \mathbb{P}(\mathbf{z} \setminus \mathbf{s} \mid x, y, \mathbf{s}) \cdot \frac{\mathbb{P}(x \mid \mathbf{s}) \cdot \mathbb{P}(\mathbf{z} \setminus \mathbf{s} \mid \mathbf{s})}{\mathbb{P}(x, \mathbf{z} \setminus \mathbf{s} \mid \mathbf{s})}$$

**Algorithm 14** `ApproxSearch` (Differences from `OracleSearch` highlighted in blue)

**Input:** Query $\mathbb{P}_x(y)$, non-descendants $\mathbf{N} \subseteq \mathbf{V}$, and error tolerances $\varepsilon_1 > 0$ and $\varepsilon_2 > 0$
**Output:** Estimate $\widehat{\mathbb{P}}_x(y)$ of $\mathbb{P}_x(y)$
**for** $k = 0, 1, 2, \ldots$ **do**
    Let $w_k^{-1} = |\mathbf{N}| \cdot \binom{|\mathbf{N}|}{k}$
    Let $\mathbf{C}_k$ be the set of size $k$ candidate sets,
    i.e. $\mathbf{C}_k = \{\mathbf{S} \subseteq \mathbf{N} : |\mathbf{S}| = k, \texttt{ApproxCondInd}(X, \mathbf{N} \setminus \mathbf{S}, \mathbf{S}, \varepsilon_2, w_k\delta)\}$
    **if** $|\mathbf{C}_k| > 0$ **then**
        Pick any $\mathbf{S} \in \mathbf{C}_k$
        **return** $\texttt{AdjustByS}(\mathbb{P}_x(y), \mathbf{S}, \varepsilon_1)$
    **end if**
**end for**

It remains to bound $\left| \sum_{\mathbf{s}} \mathbb{P}(\mathbf{s}) \cdot \mathbb{P}(y \mid x, \mathbf{s}) \cdot \left(1 - \gamma_{x,y,\mathbf{z} \cup \mathbf{s}}\right) \right|$. To do so, recall the definition of $\alpha_{\mathbf{S}} = \min_{\mathbf{s}} \mathbb{P}(x \mid \mathbf{s})$.

$$
\begin{aligned}
&\left|\sum_{\mathbf{s}} \mathbb{P}(\mathbf{s}) \cdot \mathbb{P}(y \mid x, \mathbf{s}) \cdot \left(1 - \gamma_{x,y,\mathbf{z}\cup\mathbf{s}}\right)\right| \\
&= \left|\sum_{\mathbf{s}} \mathbb{P}(\mathbf{s}) \cdot \mathbb{P}(y \mid x, \mathbf{s}) \cdot \left(1 - \sum_{\mathbf{z}\setminus\mathbf{s}} \mathbb{P}(\mathbf{z}\setminus\mathbf{s} \mid x, y, \mathbf{s}) \cdot \frac{\mathbb{P}(x \mid \mathbf{s}) \cdot \mathbb{P}(\mathbf{z}\setminus\mathbf{s} \mid \mathbf{s})}{\mathbb{P}(x, \mathbf{z}\setminus\mathbf{s} \mid \mathbf{s})}\right)\right| \\
&= \left|\sum_{\mathbf{z}\cup\mathbf{s}} \mathbb{P}(\mathbf{s}) \cdot \mathbb{P}(y \mid x, \mathbf{s}) \cdot \mathbb{P}(\mathbf{z}\setminus\mathbf{s} \mid x, y, \mathbf{s}) \cdot \left(1 - \frac{\mathbb{P}(x \mid \mathbf{s}) \cdot \mathbb{P}(\mathbf{z}\setminus\mathbf{s} \mid \mathbf{s})}{\mathbb{P}(x, \mathbf{z}\setminus\mathbf{s} \mid \mathbf{s})}\right)\right| \\
&\qquad\qquad (1 = \textstyle\sum_{\mathbf{z}\setminus\mathbf{s}} \mathbb{P}(\mathbf{z}\setminus\mathbf{s} \mid x, y, \mathbf{s})) \\
&= \left|\sum_{\mathbf{z}\cup\mathbf{s}} \mathbb{P}(\mathbf{s}) \cdot \frac{\mathbb{P}(y, \mathbf{z}\setminus\mathbf{s} \mid x, \mathbf{s})}{\mathbb{P}(x, \mathbf{z}\setminus\mathbf{s} \mid \mathbf{s})} \cdot \left(\mathbb{P}(x, \mathbf{z}\setminus\mathbf{s} \mid \mathbf{s}) - \mathbb{P}(x \mid \mathbf{s}) \cdot \mathbb{P}(\mathbf{z}\setminus\mathbf{s} \mid \mathbf{s})\right)\right| \\
&\leq \sum_{\mathbf{z}\cup\mathbf{s}} \mathbb{P}(\mathbf{s}) \cdot \frac{\mathbb{P}(y, \mathbf{z}\setminus\mathbf{s} \mid x, \mathbf{s})}{\mathbb{P}(x, \mathbf{z}\setminus\mathbf{s} \mid \mathbf{s})} \cdot \left|\mathbb{P}(x, \mathbf{z}\setminus\mathbf{s} \mid \mathbf{s}) - \mathbb{P}(x \mid \mathbf{s}) \cdot \mathbb{P}(\mathbf{z}\setminus\mathbf{s} \mid \mathbf{s})\right| \quad \text{(Triangle inequality)} \\
&= \sum_{\mathbf{z}\cup\mathbf{s}} \mathbb{P}(\mathbf{s}) \cdot \frac{\mathbb{P}(y, \mathbf{z}\setminus\mathbf{s} \mid x, \mathbf{s})}{\mathbb{P}(x, \mathbf{z}\setminus\mathbf{s} \mid \mathbf{s})} \cdot \frac{\mathbb{P}(\mathbf{z}\setminus\mathbf{s} \mid x, \mathbf{s})}{\mathbb{P}(\mathbf{z}\setminus\mathbf{s} \mid x, \mathbf{s})} \cdot \left|\mathbb{P}(x, \mathbf{z}\setminus\mathbf{s} \mid \mathbf{s}) - \mathbb{P}(x \mid \mathbf{s}) \cdot \mathbb{P}(\mathbf{z}\setminus\mathbf{s} \mid \mathbf{s})\right| \\
&\qquad\qquad \text{(Mutiply by 1)} \\
&= \sum_{\mathbf{z}\cup\mathbf{s}} \mathbb{P}(\mathbf{s}) \cdot \frac{\mathbb{P}(y \mid x, \mathbf{z}\cup\mathbf{s})}{\mathbb{P}(x \mid \mathbf{s})} \cdot \left|\mathbb{P}(x, \mathbf{z}\setminus\mathbf{s} \mid \mathbf{s}) - \mathbb{P}(x \mid \mathbf{s}) \cdot \mathbb{P}(\mathbf{z}\setminus\mathbf{s} \mid \mathbf{s})\right| \\
&\leq \sum_{\mathbf{z}\cup\mathbf{s}} \mathbb{P}(\mathbf{s}) \cdot \frac{1}{\mathbb{P}(x \mid \mathbf{s})} \cdot \left|\mathbb{P}(x, \mathbf{z}\setminus\mathbf{s} \mid \mathbf{s}) - \mathbb{P}(x \mid \mathbf{s}) \cdot \mathbb{P}(\mathbf{z}\setminus\mathbf{s} \mid \mathbf{s})\right| \qquad (\mathbb{P}(y \mid x, \mathbf{z}\cup\mathbf{s}) \leq 1) \\
&\leq \sum_{\mathbf{z}\cup\mathbf{s}} \mathbb{P}(\mathbf{s}) \cdot \frac{1}{\alpha_{\mathbf{s}}} \cdot \left|\mathbb{P}(x, \mathbf{z}\setminus\mathbf{s} \mid \mathbf{s}) - \mathbb{P}(x \mid \mathbf{s}) \cdot \mathbb{P}(\mathbf{z}\setminus\mathbf{s} \mid \mathbf{s})\right| \\
&\qquad\qquad (\text{By definition of } \alpha_{\mathbf{s}} = \min_{\mathbf{s}} \mathbb{P}(x \mid \mathbf{s})) \\
&= \frac{1}{\alpha_{\mathbf{s}}} \cdot \sum_{\mathbf{s}} \mathbb{P}(\mathbf{s}) \cdot \sum_{\mathbf{z}\setminus\mathbf{s}} \left|\mathbb{P}(x, \mathbf{z}\setminus\mathbf{s} \mid \mathbf{s}) - \mathbb{P}(x \mid \mathbf{s}) \cdot \mathbb{P}(\mathbf{z}\setminus\mathbf{s} \mid \mathbf{s})\right| \\
&\leq \frac{\varepsilon_2}{\alpha_{\mathbf{s}}} \qquad\qquad (\text{Since } X \perp\!\!\!\perp_{\varepsilon_2} \mathbf{Z}\setminus\mathbf{S} \mid \mathbf{S})
\end{aligned}
$$

□

**Theorem 12** (Formalization of Theorem 9)**.** *Fix error tolerances $\varepsilon_1 > 0$ and $\varepsilon_2 > 0$. Given a set of non-descendants $ND(\mathbf{X})$ of $\mathbf{X}$, $\varepsilon_1$-approximators* `EstimateMarginal` *and* `EstimateConditional`*, and an $\varepsilon_2$-approximate CI tester* `ApproxCondInd`*,* `ApproxSearch` *produces an estimate $\widehat{\mathbb{P}}_x(y)$ using an adjustment set $\mathbf{S}$ of size $|\mathbf{S}| \leq \min_{\mathcal{G}\in\mathcal{M}(\mathbb{P},\mathbf{N})} |pa_{\mathcal{G}}(X)|$ such that $|\mathbb{P}_x(y) - \widehat{\mathbb{P}}_x(y)| \leq \frac{\varepsilon_1+\varepsilon_2}{\alpha_{\mathbf{S}}}$, where $\alpha_{\mathbf{S}} = \min_{\mathbf{s}} \mathbb{P}(x \mid \mathbf{s})$.*

*Proof.* Let $\mathbf{Z} = \mathrm{pa}_{\mathcal{G}'}(X)$ for any $\mathcal{G}' \in \mathcal{M}(\mathbb{P}, \mathbf{N})$. As we output $\widehat{T}_{\mathbf{S}}$ as an estimate for $\mathbb{P}_x(y)$, we have

$$
\begin{aligned}
\left|\widehat{T}_{\mathbf{S}} - \mathbb{P}_x(y)\right| &= \left|\widehat{T}_{\mathbf{S}} - T_{\mathbf{S}} + T_{\mathbf{S}} - \mathbb{P}_x(y)\right| && \text{(Add 0)} \\
&\leq \left|\widehat{T}_{\mathbf{S}} - T_{\mathbf{S}}\right| + \left|T_{\mathbf{S}} - \mathbb{P}_x(y)\right| && \text{(Triangle inequality)}
\end{aligned}
$$

By Lemma 4, we have $|\widehat{T}_{\mathbf{S}} - T_{\mathbf{S}}| \leq \varepsilon_1/\alpha_{\mathbf{S}}$. By Lemma 5, we have $|\mathbb{P}_x(y) - T_{\mathbf{S}}| \leq \varepsilon_2/\alpha_{\mathbf{S}}$. □

## D.3 Minimal size adjustment sets

**Lemma 7.** *Given an arbitrary valid adjustment set* $\mathbf{Z} \subseteq \mathbf{N}$*, the subset* $\mathbf{S} \subseteq \mathbf{N}$ *is also a valid adjustment set if (i)* $Y \perp\!\!\!\perp \mathbf{Z} \setminus \mathbf{S} \mid X \cup \mathbf{S}$ *and (ii)* $X \perp\!\!\!\perp \mathbf{S} \setminus \mathbf{Z} \mid \mathbf{Z}$*.*

*Proof.* Suppose $\mathbf{Z}$ is some arbitrary valid adjustment set and $\mathbf{S}$ is another subset. Observe that

$$
\begin{aligned}
&\mathbb{P}(y \mid \mathtt{do}(x)) \\
&= \sum_{\mathbf{z}} \mathbb{P}(\mathbf{z}) \cdot \mathbb{P}(y \mid x, \mathbf{z}) && (\mathbf{Z} \text{ is valid adjset}) \\
&= \sum_{\mathbf{z}} \mathbb{P}(\mathbf{z}) \cdot \mathbb{P}(y \mid x, \mathbf{z}) \cdot \sum_{\mathbf{s} \setminus \mathbf{z}} \mathbb{P}(\mathbf{s} \setminus \mathbf{z} \mid y, x, \mathbf{z}) && (\text{Multiply by 1}) \\
&= \sum_{\mathbf{z} \cup \mathbf{s}} \mathbb{P}(\mathbf{z}) \cdot \mathbb{P}(y, \mathbf{s} \setminus \mathbf{z} \mid x, \mathbf{z}) \\
&= \sum_{\mathbf{z} \cup \mathbf{s}} \frac{\mathbb{P}(\mathbf{z})}{\mathbb{P}(\mathbf{z} \mid x)} \cdot \mathbb{P}(y, \mathbf{z} \cup \mathbf{s} \mid x) \\
&= \sum_{\mathbf{z} \cup \mathbf{s}} \mathbb{P}(y \mid x, \mathbf{z} \cup \mathbf{s}) \cdot \mathbb{P}(\mathbf{z} \cup \mathbf{s} \mid x) \cdot \frac{\mathbb{P}(\mathbf{z})}{\mathbb{P}(\mathbf{z} \mid x)} \\
&= \sum_{\mathbf{z} \cup \mathbf{s}} \mathbb{P}(y \mid x, \mathbf{z} \cup \mathbf{s}) \cdot \mathbb{P}(\mathbf{z} \cup \mathbf{s}) \cdot \frac{\mathbb{P}(\mathbf{z}) \cdot \mathbb{P}(x \mid \mathbf{z} \cup \mathbf{s})}{\mathbb{P}(x) \cdot \mathbb{P}(\mathbf{z} \mid x)} \\
&&& (\text{Since } \mathbb{P}(\mathbf{z} \cup \mathbf{s} \mid x) = \tfrac{\mathbb{P}(x, \mathbf{z} \cup \mathbf{s})}{\mathbb{P}(x)} = \tfrac{\mathbb{P}(\mathbf{z} \cup \mathbf{s}) \cdot \mathbb{P}(x \mid \mathbf{z} \cup \mathbf{s})}{\mathbb{P}(x)}) \\
&= \sum_{\mathbf{z} \cup \mathbf{s}} \mathbb{P}(\mathbf{z} \cup \mathbf{s}) \cdot \mathbb{P}(y \mid x, \mathbf{z} \cup \mathbf{s}) \cdot \frac{\mathbb{P}(x \mid \mathbf{z} \cup \mathbf{s})}{\mathbb{P}(x \mid \mathbf{z})} && (\text{Since } \mathbb{P}(x \mid \mathbf{z}) = \tfrac{\mathbb{P}(x) \cdot \mathbb{P}(\mathbf{z} \mid x)}{\mathbb{P}(\mathbf{z})})
\end{aligned}
$$

If (i) $Y \perp\!\!\!\perp \mathbf{Z} \setminus \mathbf{S} \mid X \cup \mathbf{S}$ and (ii) $X \perp\!\!\!\perp \mathbf{S} \setminus \mathbf{Z} \mid \mathbf{Z}$, then $\mathbb{P}(y \mid x, \mathbf{z} \cup \mathbf{s}) = \mathbb{P}(y \mid x, \mathbf{s})$ and $\frac{\mathbb{P}(x \mid \mathbf{z} \cup \mathbf{s})}{\mathbb{P}(x \mid \mathbf{z})} = 1$ respectively. Thus,

$$
\begin{aligned}
\mathbb{P}(y \mid \mathtt{do}(x)) &= \sum_{\mathbf{z} \cup \mathbf{s}} \mathbb{P}(\mathbf{z} \cup \mathbf{s}) \cdot \mathbb{P}(y \mid x, \mathbf{z} \cup \mathbf{s}) \cdot \frac{\mathbb{P}(x \mid \mathbf{z} \cup \mathbf{s})}{\mathbb{P}(x \mid \mathbf{z})} \\
&= \sum_{\mathbf{z} \cup \mathbf{s}} \mathbb{P}(\mathbf{z} \cup \mathbf{s}) \cdot \mathbb{P}(y \mid x, \mathbf{s}) \\
&= \sum_{\mathbf{s}} \mathbb{P}(\mathbf{s}) \cdot \mathbb{P}(y \mid x, \mathbf{s})
\end{aligned}
$$

That is, $\mathbf{S}$ is also a valid adjustment set. □

**Lemma 8.** *Suppose* $\mathbf{Z} = pa_{\mathcal{G}'}(X)$*. Then, for* any *valid adjustment set* $\mathbf{S} \subseteq \mathbf{N}$*, both (i)* $Y \perp\!\!\!\perp \mathbf{Z} \setminus \mathbf{S} \mid X \cup \mathbf{S}$ *and (ii)* $X \perp\!\!\!\perp \mathbf{S} \setminus \mathbf{Z} \mid \mathbf{Z}$ *hold.*

## D.4 Sample complexity: conditional independence testing

In the distribution testing literature, several methods have been developed which satisfy the requirements of the $\varepsilon_2$-approximate conditional independence tester `ApproxCondInd`.

[Can+18] proposes a test based on total variation distance that uses $\tilde{O}\left(\frac{|\Sigma_{\mathbf{A}}|^{1/2}\cdot|\Sigma_{\mathbf{B}}|^{1/2}\cdot|\Sigma_{\mathbf{C}}|^{1/2}}{\varepsilon_2^2}\right)$ samples from $\mathbb{P}$; see their Lemma 2.2.[1] There is also a simpler test based on the empirical mutual information, proposed by [Bha+21], that uses $\tilde{O}\left(\frac{|\Sigma_{\mathbf{A}}|\cdot|\Sigma_{\mathbf{B}}|\cdot|\Sigma_{\mathbf{C}}|}{\varepsilon_2^2}\right)$ samples from $\mathbb{P}$, though we use the former to obtain optimal dependence on the alphabet sizes.

**Lemma 9.** *Let $\varepsilon_2, \delta > 0$. Let $\mathbf{S}$ be the set picked by* `ApproxSearch` *in Line 7. Then, with probability $1-\delta$, we have $X \perp\!\!\!\perp_{\varepsilon_2} \mathbf{N} \setminus \mathbf{S} \mid \mathbf{S}$ as long as* `ApproxSearch` *takes $t_1$ samples for*

$$t_1 \in \tilde{O}\left(\frac{|\Sigma_X|^{1/2}\cdot|\Sigma_Y|^{1/2}\cdot|\Sigma_{\mathbf{N}}|^{1/2}\cdot k^*\cdot\log|\mathbf{N}||}{\varepsilon_2^2}\right),$$

*where $k^* := \min_{\mathcal{G}\in\mathcal{M}(\mathbb{P},\mathbf{N})}|pa_{\mathcal{G}}(X)|$.*

*Proof.* For any $k \in \{0,1,\ldots,|\mathbf{N}|\}$, there are $\binom{|\mathbf{N}|}{k}$ possible candidate sets in $\mathbf{C}_k$. Recall that $w_k^{-1} = |\mathbf{N}|\cdot\binom{|\mathbf{N}|}{k} \le |\mathbf{N}|\cdot\left(\frac{e\cdot|\mathbf{N}|}{k}\right)^k$. For each candidate set $\mathbf{S}\in\mathbf{C}_k$, we know that $\mathbf{S}\subseteq\mathbf{N}$, so testing $\texttt{ApproxCondInd}(X, \mathbf{N}\setminus\mathbf{S}, \mathbf{S}, \varepsilon_2, w_k\delta)$ succeeds with probability at least $w_k\delta$ while using

$$\tilde{O}\left(\frac{|\Sigma_X|^{1/2}\cdot|\Sigma_{\mathbf{N}\setminus\mathbf{S}}|^{1/2}\cdot|\Sigma_{\mathbf{S}}|^{1/2}\cdot\log(w_k)}{\varepsilon_2^2}\right) \subseteq \tilde{O}\left(\frac{|\Sigma_X|^{1/2}\cdot|\Sigma_{\mathbf{N}})|^{1/2}\cdot k\cdot\log(|\mathbf{N}|)}{\varepsilon_2^2}\right)$$

samples from $\mathbb{P}$.

Suppose `ApproxSearch` adjusts using $\mathbf{S}$ of size $|\mathbf{S}| = k^*$, and let $t_1$ be as stated. By a union bound, the probability that any tests fails is at most

$$\sum_{k=0}^{k^*} w_k\delta\cdot\binom{|\mathbf{N}|}{k} = \delta\cdot\sum_{k=0}^{k^*}\frac{1}{|\mathbf{N}|} \le \delta,$$

i.e., all tests succeed with probability $1-\delta$, and thus the returned $\mathbf{S}$ satisfies $X \perp\!\!\!\perp_{\varepsilon_2} \mathbf{N}\setminus\mathbf{S} \mid \mathbf{S}$ with probability $1-\delta$, as desired. □

## D.5 Sample complexity: estimation

In this section, our goal is to prove the following result:

**Lemma 10.** *Let $\varepsilon_1, \delta > 0$, $\mathbf{S}\subseteq\mathbf{V}$, $K = |\Sigma_{\mathbf{S}}|$, and $\alpha_{\mathbf{S}} = \min_{\mathbf{s}}\mathbb{P}(x \mid \mathbf{S}=\mathbf{s})$. Let $\widehat{T}_{\mathbf{S}}$ be the estimate returned by* `AdjustByS`*. Then, with probability $1-\delta$, we have $|\widehat{T}_{\mathbf{S}} - T_{\mathbf{S}}| \le \varepsilon_1$ as long as* `AdjustByS` *takes $t_2$ samples for*

$$t_2 \in \tilde{O}\left(\frac{1}{\varepsilon_1^2\alpha_{\mathbf{s}}} + \frac{K}{\varepsilon_1\alpha_{\mathbf{S}}}\log(K)\right).$$

In particular, we considered the *Poissonized* setting, where `AdjustByS` takes $N\sim\mathsf{Pois}(t_2)$ samples; i.e., the number of samples is randomized rather than deterministic. To begin, we recall basic facts about sub-Gaussian distributions and Poisson distributions in Appendix D.5.1 and Appendix D.5.2.

[1] Here, we assume that $\varepsilon_2^{-1}$ is sufficiently large compared to $|\Sigma_{\mathbf{A}}|$, $|\Sigma_{\mathbf{B}}|$, and $|\Sigma_{\mathbf{C}}|$.

### D.5.1 Preliminaries: sub-Gaussian distributions

First, we recall the definition of a sub-Gaussian distribution, as a distribution whose moment-generating function (MGF) has a Gaussian-like upper bound.

**Definition 6.** $\mathsf{V}$ *is* sub-Gaussian *with* variance proxy $\sigma^2$ *if* $\mathbb{E}[\mathsf{V}] = 0$ *and*

$$M_{\mathsf{V}}(t) := \mathbb{E}[\exp(t\mathsf{V})] \leq \exp\left(\frac{\sigma^2 t^2}{2}\right)$$

*We write* $V \in \mathsf{subG}(\sigma^2)$.[2]

In turn, it is a standard fact that sub-Gaussian distributions possess Gaussian-like tail bounds (see e.g. [RH23, Lemma 1.3]):

**Claim 1.** *If* $\mathsf{V} \in \mathsf{subG}(\sigma^2)$*, then* $\mathbb{P}(|\mathsf{V}| > \varepsilon) \leq 2\exp\left(-\frac{\varepsilon^2}{2\sigma^2}\right)$

It follows directly from the definition of the MGF that sub-Gaussian distributions have simple behavior under addition and scaling; we restate and prove this fact for the sake of completion.

**Claim 2.** *Let* $\mathsf{V}_k \in \mathsf{subG}(\sigma_k^2)$ *for* $k = 1, \ldots, K$*, all independent. Let* $\mathsf{J} = \sum_{k=1}^K a_k \mathsf{V}_k$ *for* $a_k \geq 0$*. Then* $\mathsf{J} \in \mathsf{subG}\left(\sum_{k=1}^K a_k^2 \sigma_k^2\right)$.

*Proof.* Using basic properties of the moment-generating function, we have

$$\begin{aligned} M_{\mathsf{J}}(t) &= \prod_{k=1}^K M_{\mathsf{V}_k}(a_k t) && \text{(By independence of } \mathsf{V}_k\text{'s)} \\ &\leq \prod_{k=1}^K \exp\left(\frac{a_k^2 \sigma_k^2 t^2}{2}\right) \\ &= \exp\left(\frac{\left(\sum_{k=1}^K a_k^2 \sigma_k^2\right) t^2}{2}\right) \end{aligned}$$

□

In turn, this can be used to show that sample averages of a Bernoulli distribution are sub-Gaussian, with variance proxy inversely proportional to the number of samples.

**Claim 3.** *Let* $\mathsf{B}^{(i)} \sim \mathsf{Bern}(p)$ *be independent samples for* $i = 1, 2, \ldots, n$*, and let*

$$\mathsf{P} = \frac{1}{n}\sum_{i=1}^n B^{(i)}.$$

*Then* $\mathsf{P} - p \in \mathsf{subG}\left(\frac{1}{4n}\right)$*. Thus,* $\mathbb{P}(|\mathsf{P} - p| > t) \leq 2\exp(-2nt^2)$.

[2] It is more common to write $\mathsf{V} \sim \mathsf{subG}(\sigma^2)$, but this practice is an easily avoidable abuse of notation. Here we use $\mathsf{subG}(\sigma^2)$ to denote the set of all distributions which are sub-Gaussian with variance proxy $\sigma^2$, this notation reflects that sub-Gaussianity is a property of a distribution.

Finally, we note that a mixture of countably many conditionally independent sub-Gaussian distributions is a sub-Gaussian distribution.

**Claim 4.** *Let* $\mathsf{U} \sim \mathbb{P}(\mathsf{U})$ *be a discrete random variable taking values in* $\{1, 2, \ldots\}$*. Suppose that, conditioned on* $\mathsf{U} = k$*, the random variable* $\mathsf{V}$ *has MGF* $M_k(t)$*. Then*

$$M_{\mathsf{V}}(t) = \sum_{k=1}^{\infty} \mathbb{P}(\mathsf{U} = k) \cdot M_k(t).$$

*Proof.*

$$\begin{aligned} M_{\mathsf{V}}(t) &= \mathbb{E}\left[\exp\left(t\mathsf{V}\right)\right] \\ &= \mathbb{E}\left[\mathbb{E}\left[\exp\left(t\mathsf{V}\right) \mid \mathsf{U}\right]\right] \\ &= \mathbb{E}\left[M_{\mathsf{U}}(t)\right] \\ &= \sum_{k=1}^{\infty} \mathbb{P}(\mathsf{U} = k) \cdot M_{\mathsf{V}_k}(t) \end{aligned}$$

□

In particular, if $\mathsf{V}$ is conditionally sub-Gaussian for each value of $U$, then $\mathsf{V}$ itself is sub-Gaussian, as follows.

**Claim 5.** *Let* $\mathsf{U} \sim \mathbb{P}(\mathsf{U})$ *be a discrete random variable taking values in* $\{1, 2, \ldots\}$*. Suppose that, conditioned on* $U = k$*,* $\mathsf{V} \in \mathit{subG}(\sigma_k^2)$*. Then* $\mathsf{V} \in \mathit{subG}(\sigma_{max}^2)$ *where* $\sigma_{max} = \sup_{k=1}^{\infty} \sigma_k^2$*.*

## D.5.2 Preliminaries: Poisson distributions

To streamline the presentation of our Poissonized analysis, it will be helpful in this section to define new distributions and state some of their basic properties. These new distributions will be defined by the following three-step process: (1) pick the sample size $\mathsf{N}$ from a Poisson distribution, (2) draw $\mathsf{N}$ samples from some "base" distribution, and (3) define a random vector in terms of those samples and take the induced distribution of that random variable.

For our first definition, step (2) draws samples from a base distribution $\mathbb{P}(\mathsf{V})$ over a discrete random variable $\mathsf{V}$ with alphabet $\Sigma_{\mathsf{V}}$, and step (3) defines the *signature* (counts) of these samples.

**Definition 7.** *Fix a distribution* $\mathbb{P}(\mathsf{V})$ *and an integer* $n > 0$*. We define the* Poisson counts *distribution* $\mathsf{PoisCounts}(n, \mathbb{P}(\mathsf{V}))$ *as follows. First, let* $\mathsf{N} \sim \mathsf{Pois}(n)$*. Then, let* $\mathsf{V}^{(i)} \sim \mathbb{P}(\mathsf{V})$ *be independent samples for* $i = 1, 2, \ldots, \mathsf{N}$*. Let*

$$\mathit{Sig} = \{\mathsf{N}_v\}_{v \in \Sigma_{\mathsf{V}}} \quad \textit{where} \quad \left\{ \mathsf{N}_v = \sum_{i=1}^{\mathsf{N}} \mathbb{1}_{\mathsf{V}^{(i)} = v} \right\}_{v \in \Sigma_V}$$

*We define* $\mathsf{PoisCounts}(n, \mathbb{P}(\mathsf{V}))$ *as the induced distribution of* $\mathit{Sig}$*.*

One of the important reasons for working in the Poissonized setting is the independence of the random variables $\mathsf{N}_v$ for $v \in \Sigma_{\mathsf{V}}$, and the fact that their conditional distributions are binomial (see e.g. [Can+22, Fact C.1]).

**Claim 6.** *Let* $\mathit{Sig} \sim \mathsf{PoisCounts}(n, \mathbb{P}(\mathsf{V}))$. *Then the random variables* $\{\mathsf{N}_v\}_{v \in \Sigma_{\mathsf{V}}}$ *are mutually independent, with marginal distributions*

$$\mathsf{N}_v \sim \mathsf{Pois}(n \cdot \mathbb{P}(v)), \tag{D.1}$$

*and with conditional distributions*

$$(\mathsf{N}_v \mid \mathsf{N} = n') \sim \mathsf{Bin}(n', \mathbb{P}(v)) \tag{D.2}$$

*for any integer* $n' > 0$.

For our second definition, step (2) draws samples from a base distribution $\mathbb{P}(\mathsf{V}, \mathsf{W})$ over discrete random variables $\mathsf{V}$ and $\mathsf{W}$ with alphabets $\Sigma_{\mathsf{V}}$ and $\Sigma_{\mathsf{W}}$, and step (3) defines the conditional *frequencies* for each value $v \in \Sigma_{\mathsf{V}}$ given each value $w \in \Sigma_{\mathsf{W}}$.

**Definition 8.** *Fix a distribution* $\mathbb{P}(\mathsf{V}, \mathsf{W})$ *and an integer* $n > 0$. *We define the distribution* $\mathsf{PoisFreqs}(n, \mathbb{P}(\mathsf{V}, \mathsf{W}))$ *as follows. First, let* $\mathit{Sig} \sim \mathsf{PoisCounts}(n, \mathbb{P}(\mathsf{V}, \mathsf{W}))$, *recalling that* $\mathit{Sig} = \{\mathsf{N}_{vw}\}_{(v,w) \in \Sigma_{\mathsf{V}} \times \Sigma_{\mathsf{W}}}$. *For each* $w$, *let* $\mathsf{N}_w = \sum_{v \in \Sigma_V} \mathsf{N}_{vw}$. *Then, let*

$$\mathit{Freqs} = \{\mathsf{P}_{v|w}\}_{v \in \Sigma_v, w \in \Sigma_w} \quad \textit{where} \quad \left\{\mathsf{P}_{v|w} = \frac{\mathsf{N}_{vw}}{\mathsf{N}_w}\right\}_{v \in \Sigma_V, w \in \Sigma_W}$$

*with the convention* $0/0 = 0$. *We define* $\mathsf{PoisFreqs}(n, \mathbb{P}(V, W))$ *as the induced distribution of* $\mathit{Freqs}$.

The following independence and expectation properties of this distribution are essential to showing (conditional) sub-Gaussianity of a random variable that is central to our analysis.

**Claim 7.** *Let* $\mathit{Freqs} \sim \mathsf{PoisFreqs}(n, \mathbb{P}(\mathsf{V}, \mathsf{W}))$. *Then, for any* $v, v' \in \Sigma_{\mathsf{V}}$ *and* $w, w' \in \Sigma_{\mathsf{W}}$ *with* $w \neq w'$, *the random variables* $\mathsf{P}_{v|w}$ *and* $\mathsf{P}_{v'|w'}$ *are independent.*

*Proof.* The independence of $\mathsf{P}_{v|w}$ and $\mathsf{P}_{v'|w'}$ follows from independence of the pairs $(\mathsf{N}_{vw}, \mathsf{N}_w)$ and $(\mathsf{N}_{v'w'}, \mathsf{N}_{w'})$; a consequence of Claim 6. □

Now, we are ready to combine our results on Poisson distributions with our discussion of sub-Gaussian distributions. In particular, *conditioned on the count of* $\mathsf{N}_w$ *being greater than* $n'$, Poisson frequencies are sub-Gaussian with variance proxy inversely proportional to $n'$.

**Claim 8.** *Let* $\mathit{Freqs} \sim \mathsf{PoisFreqs}(n, \mathbb{P}(\mathsf{V}, \mathsf{W}))$. *For any* $n_w > 0$, *let* $E_{n_w}$ *be the event that* $\mathsf{N}_w > n_w$. *Then, conditioned on* $E_{n_w}$, *we have* $\mathsf{P}_{v|w} \in \mathit{subG}\left(\frac{1}{4n_w}\right)$.

*Proof.* By (D.2) in Claim 6, for any $n' > 0$, we have

$$\mathsf{N}_{vw} \mid \mathsf{N}_w = n' \sim \mathsf{Bin}(n', \mathbb{P}(v \mid w))$$

Thus, by Claim 3 on sub-Gaussianity of empirical frequencies, $\left(\mathsf{P}_{v|w} - \mathbb{P}(v \mid w) \mid \mathsf{N}_w = n'\right) \in \mathsf{subG}\left(\frac{1}{4n'}\right)$. Thus, by Claim 5, conditioned on $E_{n_w}$, $\mathsf{P}_{v|w} - \mathbb{P}(v \mid w) \in \mathsf{subG}(\frac{1}{4n_w})$. □

As we will see, Claim 8 is crucial to controlling the estimation error under a certain event. We use the following standard bound on the concentration of Poisson distributions to establish that this event happens with high probability (see [Can+22, Theorem A.8]).

**Claim 9.** *Let* $\mathsf{N} \sim \mathsf{Pois}(n)$. *Then, for any* $0 < t < n$,

$$\mathbb{P}(\mathsf{N} \leq n - t) \leq \exp\left(\frac{-t^2}{2(n+t)}\right)$$

*In particular, for* $b \in [0, 1]$, *taking* $t = (1 - b) \cdot n$, *we have*

$$\mathbb{P}(\mathsf{N} \leq b \cdot n) \leq \exp\left(-\frac{(1-b)^2}{4-2b} n\right)$$

### D.5.3 Notation

Now, we specialize the results above to our setting. In particular, we use the following induced Poisson counts, for a sample drawn from $\mathsf{PoisCounts}(n, \mathbb{P}(\mathsf{V}, \mathsf{W}), n)$ where $\mathsf{V} = Y$ and $\mathsf{W} = (\mathbf{S}, X)$. For fixed values $x \in \Sigma_x$, $y \in \Sigma_y$, and for each $\mathbf{s} \in [K]$, let

$$\mathsf{N}_{\mathbf{s}} := \sum_{i=1}^{\mathsf{N}} \mathbb{1}_{\mathbf{S}^{(i)}=\mathbf{s}}, \qquad \mathsf{N}_{\mathbf{s},x} := \sum_{i=1}^{\mathsf{N}} \mathbb{1}_{\mathbf{S}^{(i)}=\mathbf{s}, X^{(i)}=x}, \qquad \mathsf{N}_{\mathbf{s},x,y} := \sum_{i=1}^{\mathsf{N}} \mathbb{1}_{\mathbf{S}^{(i)}=\mathbf{s}, X^{(i)}=x, Y=y^{(i)}},$$

and the induced Poisson frequencies

$$\mathsf{P}_{y|\mathbf{s},x} := \frac{\mathsf{N}_{\mathbf{s},x,y}}{\mathsf{N}_{\mathbf{s},x}}.$$

Now, we define a random variable that corresponds to the estimate $\widehat{T}_{\mathbf{S}}$, in particular,

$$\mathsf{J} := \sum_{\mathbf{s}} \mathbb{P}(\mathbf{s}) \cdot \left(\mathsf{P}_{y|\mathbf{s},x} - \mathbb{P}(y \mid \mathbf{s}, x)\right). \tag{D.3}$$

Next, we introduce a threshold $a > 0$, which partitions the values of $\mathbf{s}$ into two cases. In particular, let

$$\mathcal{S}_{\geq a} := \{\mathbf{s} \in \Sigma_{\mathbf{S}} : \mathbb{P}(\mathbf{s}, x) \geq a\},$$

and note that $\mathbb{P}(\mathbf{s}) \leq \frac{a}{\alpha_{\mathbf{S}}}$ for $\mathbf{s} \notin \mathcal{S}_{\geq a}$, by definition of $\mathcal{S}_{\geq a}$ and $\alpha_{\mathbf{S}}$. In our analysis, we "give up" on $\mathbf{s} \notin \mathcal{S}_{\geq a}$, essentially clipping the values of $\mathbb{P}(\mathbf{s})$ to zero. In particular, our analysis primarily focuses on the random variable $\mathsf{J}_{\geq a}$, and on the event $E^{\text{conc}}_{\geq a}$:

$$\mathsf{J}_{\geq a} := \sum_{\mathbf{s} \in \mathcal{S}_{\geq a}} \mathbb{P}(\mathbf{s}) \cdot \left(\mathsf{P}_{y|\mathbf{s},x} - \mathbb{P}(y \mid \mathbf{s}, x)\right), \text{and}$$

$$E^{\text{conc}}_{\geq a} := \bigcap_{\mathbf{s} \in \mathcal{S}_{\geq a}} \left\{ N_{\mathbf{s},x} > \frac{n \cdot \mathbb{P}(\mathbf{s}, x)}{2} \right\}.$$

### D.5.4 Analysis

First, we transform the bound on $|\mathsf{J}|$ into a bound on $|\mathsf{J}_{\geq a}|$, and decompose into two terms.

**Lemma 14.** *Let $\varepsilon > \frac{aK}{\alpha_{\mathbf{S}}}$ and $t = \varepsilon - \frac{aK}{\alpha_{\mathbf{S}}}$. Then*

$$\mathbb{P}(|\mathsf{J}| > \varepsilon) \leq \mathbb{P}\left(|\mathsf{J}_{\geq a}| > t \mid E^{conc}_{\geq a}\right) + (1 - \mathbb{P}(E^{conc}_{\geq a})).$$

*Proof.* First, we upper bound $|J|$ as follows.

$$\begin{aligned}
|\mathsf{J}| &\leq |\mathsf{J}_{\geq a}| + \left|\sum_{\mathbf{s} \notin \mathcal{S}_{\geq a}} \mathbb{P}(\mathbf{s}) \cdot (\mathsf{P}_{y|\mathbf{s},x} - \mathbb{P}(y \mid \mathbf{s}, x))\right| && \text{(Triangle inequality)} \\
&\leq |\mathsf{J}_{\geq a}| + \sum_{\mathbf{s} \notin \mathcal{S}_{\geq a}} \mathbb{P}(\mathbf{s}) \\
&\leq |\mathsf{J}_{\geq a}| + \frac{aK}{\alpha_{\mathbf{S}}} && \text{(Definition of } \mathcal{S}_{\geq a} \text{ and using } |\mathcal{S}_{\geq a}| \leq K)
\end{aligned}$$

Thus, if $|\mathsf{J}_{\geq a}| < \varepsilon - \frac{aK}{\alpha_{\mathbf{S}}}$, then $|\mathsf{J}| < \varepsilon$. Now, for any $t$, we have by the law of total probability

$$\begin{aligned}
\mathbb{P}(|\mathsf{J}_{\geq a}| > t) &= \mathbb{P}(|\mathsf{J}_{\geq a}| > t \mid E^{\text{conc}}_{\geq a}) \cdot \mathbb{P}(E^{\text{conc}}_{\geq a}) + \mathbb{P}(|\mathsf{J}_{\geq a}| > t \mid \neg E^{\text{conc}}_{\geq a}) \cdot (1 - \mathbb{P}(E^{\text{conc}}_{\geq a})) \\
&\leq \mathbb{P}(|\mathsf{J}_{\geq a}| > t \mid E^{\text{conc}}_{\geq a}) + (1 - \mathbb{P}(E^{\text{conc}}_{\geq a}))
\end{aligned}$$

□

The first term can be bounded using that $\mathsf{J}_{\geq a}$ is sub-Gaussian conditional on $E^{\text{conc}}_{\geq a}$, which follows from conditional sub-Gaussianity of its summands and additivity of (conditionally) independent sub-Gaussian distributions (Claim 8 and Claim 2). The second term can be bounded using tail concentration bounds for Poisson random variables (Claim 9).

**Lemma 15** (Concentration under the good event)**.** *Let $t > 0$. Then*

$$\mathbb{P}(|\mathsf{J}_{\geq a}| > t \mid E^{conc}_{\geq a}) \leq 2\exp\left(-n\alpha_{\mathbf{S}} t^2\right).$$

*Proof.* Let $E^{\text{conc}}_{\geq a}$ hold. By Claim 8, for any $\mathbf{s} \in \mathcal{S}_{\geq a}$, we have

$$\mathsf{P}_{y|\mathbf{s},x} - \mathbb{P}(y \mid \mathbf{s}, x) \in \mathsf{subG}\left(\frac{1}{2 \cdot n \cdot \mathbb{P}(\mathbf{s}, x)}\right)$$

By Claim 7, $\mathsf{P}_{y|\mathbf{s},x}$ are conditionally independent given $N_{\mathbf{s},x}$. Thus, conditioned on $E^{\text{conc}}_{\geq a}$, we have

$$\mathsf{J}_{\geq a} \in \mathsf{subG}\left(\sum_{\mathbf{s} \in \mathcal{S}_{\geq a}} \frac{\mathbb{P}(\mathbf{s})^2}{2 \cdot n \cdot \mathbb{P}(\mathbf{s}, x)}\right)$$

by Claim 2. Noting that $\frac{\mathbb{P}(\mathbf{s},x)}{\mathbb{P}(\mathbf{s})} = \mathbb{P}(x \mid \mathbf{s}) \geq \alpha_{\mathbf{S}}$, we obtain that

$$\mathsf{J}_{\geq a} \in \mathsf{subG}\left(\frac{1}{2n\alpha_{\mathbf{s}}}\right)$$

Finally, the bound holds by Claim 1. □

**Lemma 16** (Probability of the bad event)**.** $1 - \mathbb{P}(E^{conc}_{\geq a}) \leq K \exp\left(\frac{-na}{12}\right)$

*Proof.* This result follows directly a union bound and Poisson concentration bounds, as follows. In particular, recall that $N_{\mathbf{s},x} \sim \mathsf{Pois}(n\mathbb{P}(\mathbf{s},\mathbf{x}))$, so

$$\begin{aligned}
1 - \mathbb{P}(E^{\text{conc}}_{\geq a}) &\leq \sum_{\mathbf{s}\in\mathcal{S}_{\geq a}} \mathbb{P}\left(N_{\mathbf{s},x} \leq \frac{n\cdot\mathbb{P}(\mathbf{s},x)}{2}\right) && \text{(Union bound)} \\
&\leq \sum_{\mathbf{s}\in\mathcal{S}_{\geq a}} \exp\left(\frac{-n\cdot\mathbb{P}(\mathbf{s},x)}{12}\right) && \text{(Claim 9 with } b = 1/2) \\
&\leq K\cdot\exp\left(\frac{-na}{12}\right) && \text{(Definition of } \mathcal{S}_{\geq a} \text{ and } |\mathcal{S}_{\geq a}| \leq K)
\end{aligned}$$

□

Putting the previous three results together, we obtain the following bound as a function of $a$.

**Lemma 17.** *Let* $t = \varepsilon - \frac{aK}{\alpha_{\mathbf{S}}}$*, and* $n \geq \frac{1}{\alpha_{\mathbf{S}}t^2}\log\left(\frac{4}{\delta}\right) + \frac{12}{a}\log\left(\frac{2K}{\delta}\right)$*. Then* $\mathbb{P}(|J| > \varepsilon) \leq \delta$*.*

*Proof.* By re-arranging Lemma 15 and Lemma 16, we have

1. If $n \geq \frac{1}{\alpha_{\mathbf{S}}t^2}\log\left(\frac{4}{\delta}\right)$, then $\mathbb{P}(|\mathsf{J}_{\geq a}| > t \mid E^{\text{conc}}_{\geq a}) \leq \frac{\delta}{2}$, and
2. If $n \geq \frac{12}{a}\log\left(\frac{2K}{\delta}\right)$, then $1 - \mathbb{P}(E^{\text{conc}}_{\geq a}) \leq \frac{\delta}{2}$.

The bound follows by Lemma 14. □

We can now prove our main bound by picking $a$. One route is to proceed by optimizing the bound; the same result is achieved by requiring that $\frac{aK}{\alpha_{\mathbf{S}}}$ has the same "units" as $\varepsilon$, i.e., picking $a = \lambda\frac{\varepsilon\alpha_{\mathbf{S}}}{K}$ for some constant $\lambda \in (0,1)$, as follows.

**Theorem 14.** *Fix* $\varepsilon > 0$ *and let* $\alpha_{\mathbf{S}} := \min_{\mathbf{s}} \mathbb{P}(x \mid \mathbf{s})$*. Let* $n = \mathcal{O}\left(\frac{1}{\varepsilon^2\alpha_{\mathbf{s}}}\log\left(\frac{1}{\delta}\right) + \frac{K}{\varepsilon\alpha_{\mathbf{S}}}\log\left(\frac{K}{\delta}\right)\right)$*. Then* $\mathbb{P}(|J| < \varepsilon) \leq 1 - \delta$*.*

*Proof.* Let $a = \lambda\frac{\varepsilon\alpha_{\mathbf{S}}}{K}$ for some constant $\lambda \in (0,1)$. Then $t = \varepsilon - \frac{aK}{\alpha_{\mathbf{S}}} = (1-\lambda)\varepsilon$, and by Lemma 17, we have

$$n \geq \frac{1}{\alpha_{\mathbf{S}}(1-\lambda)^2\varepsilon^2}\log\left(\frac{4}{\delta}\right) + \frac{12K}{\lambda\varepsilon\alpha_{\mathbf{S}}}\log\left(\frac{2K}{\delta}\right)$$

suffices to ensure $\mathbb{P}(|J| > \varepsilon) \leq \delta$. Using $\lambda = 1/2$, we obtain the result. □

## D.6 Hardness results

Here, we argue for the necessity of knowledge about a valid topological ordering in order to correctly adjust, even in the oracle setting.

**Lemma 11** (Necessity of knowing a non-descendant set $\mathrm{nd}_{\mathcal{G}'}(\mathbf{X})$)**.** *There exist two probability distributions $\mathbb{P}$ and $\mathbb{P}'$ on 3 binary variables $\{X, Y, Z\}$ that are generated according to topological orderings $\sigma : X \prec Z \prec Y$ and $\sigma' : Z \prec X \prec Y$ respectively such that $\mathbb{P}$ is indistinguishable from $\mathbb{P}'$ and adjusting for $\mathbb{P}_x(y)$ using $\sigma'$ yields an approximation error of 1/2 in the worst case. This holds even when we have a conditional independence oracle for $\mathbb{P}$ and we are given the Markov equivalence class of a graph $\mathcal{G}$ which $\mathbb{P}$ is Markov with respect to.*

*Proof.* Let us two probability distributions $\mathbb{P}$ and $\mathbb{P}'$ on three binary variables $\{X, Y, Z\}$ that are generated according to topological orderings $\sigma : X \prec Z \prec Y$ and $\sigma' : Z \prec X \prec Y$ respectively, where:

- $\mathbb{P}$ is defined as flipping a fair coin for $X$, $Z$ copies $X$, and $Y$ is the XOR of $X$ and $Z$ (i.e. $Y = 0$ always).
- $\mathbb{P}'$ is defined as flipping a fair coin for $Z$, $X$ copies $Z$, and $Y$ is the XOR of $X$ and $Z$ (i.e. $Y = 0$ always).

For both probability distributions, the graph skeleton is an undirected triangle on the 3 vertices and samples from the joint distribution on $(X, Y, Z)$ have the form like $(B, 0, B)$, where $B \sim \mathsf{Bern}(1/2)$. So, *without* knowing whether $X$ is upstream of $Z$ or $Z$ is upstream of $X$, we cannot distinguish between $\mathbb{P}$ and $\mathbb{P}'$ even with infinite samples from the joint distribution $(X, Y, Z)$ and knowledge of the true causal graph skeleton. See Figure D.1 for an illustration.

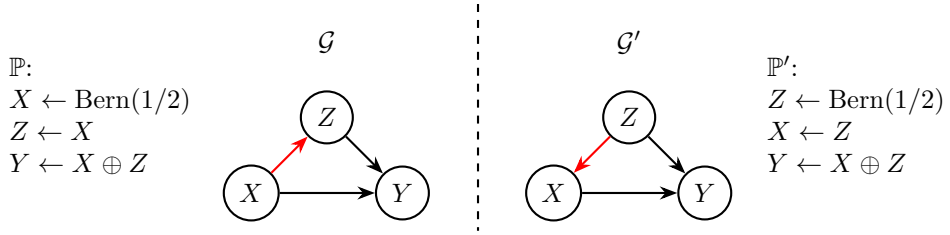



Figure D.1: Two possible causal systems with their corresponding causal graphs. One can verify that $\mathbb{P}$ and $\mathbb{P}'$ are indistinguishable from observations and that $\mathcal{G}$ and $\mathcal{G}'$ belong to the same Markov equivalence class.

Suppose we intervene $X = 0$ and wish to determine $\mathbb{P}(Y = 0 \mid \mathtt{do}(X = 0))$.
Under $\mathcal{G}$, since $X$ has no incoming arcs, the correct adjustment is

$$\mathbb{P}(Y = 0 \mid \mathtt{do}(X = 0)) = \mathbb{P}(Y = 0 \mid X = 0) = 1 \tag{D.4}$$

Meanwhile, under $\mathcal{G}'$, we have to use the backdoor adjustment formula, yielding

$$\mathbb{P}'(Y = 0 \mid \mathtt{do}(X = 0)) = \sum_{z \in \{0,1\}} \mathbb{P}'(Y = 0 \mid Z = z, X = 0)\mathbb{P}'(Z = z) = 1/2 \tag{D.5}$$

Now, consider intervening $X = 0$.
Under $\mathbb{P}$, we now have

- $X \leftarrow 0$
- $Z \leftarrow X$
- $Y \leftarrow X \oplus Z = 0$

That is, $\mathbb{P}(Y = 0 \mid \texttt{do}(X = 0)) = 1$. If we use (D.4), then we correctly conclude $\mathbb{P}(Y = 0 \mid \texttt{do}(X = 0)) = 1$. However, if we use (D.5), then we wrongly conclude $\mathbb{P}(Y = 0 \mid \texttt{do}(X = 0)) = 1/2$.

Under $\mathbb{P}'$, we now have

- $Z \leftarrow \mathsf{Bern}(1/2)$
- $X \leftarrow 0$
- $Y \leftarrow X \oplus Z = Z$

That is, $\mathbb{P}'(Y = 0 \mid \texttt{do}(X = 0)) = 1/2$. If we use (D.5), then we correctly conclude $\mathbb{P}'(Y = 0 \mid \texttt{do}(X = 0)) = 1/2$. However, if we use (D.4), then we wrongly conclude $\mathbb{P}'(Y = 0 \mid \texttt{do}(X = 0)) = 1$.

So, one would incur an error of $1/2$ in the approximation in the worst case. □

Note that an error of $1/2$ is not hard to get in expectation: flip a fair coin and then just always conclude that $\mathbb{P}(y \mid \texttt{do}(x))$ is the outcome of the coin toss. If $\mathbb{P}(y \mid \texttt{do}(x)) = \alpha$, then the expected error is $(1/2) \cdot (1 - \alpha) + (1/2) \cdot \alpha = 1/2$.

**Lemma 12** (Necessity of error dependence on $\Omega(\varepsilon_2/\alpha)$)**.** *Suppose $0 < \sqrt{\varepsilon} \leq \alpha \leq 1/2$. There exists a probability distribution $\mathbb{P}$ defined over 4 binary variables $\{A, B, X, Y\}$ in a clique with a topological ordering of $A \prec B \prec X \prec Y$ such that: $X \perp\!\!\!\perp_{\varepsilon_2} B \mid A$, $\min_a \mathbb{P}(x \mid a) \geq \frac{\alpha}{2}$, $|\widehat{\mathbb{P}}_x(y) - \mathbb{P}_x(y)| \geq \frac{\varepsilon_2}{8\alpha}$, where $\{A, B\}$ is the correct adjustment set for $\mathbb{P}_x(y)$ but $\hat{\mathbb{P}}_x(y)$ is computed using $\{A\}$ as the adjustment set. This holds even if all conditional probabilities of $\mathbb{P}$ are given.*

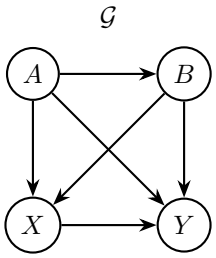
*Proof.* For notational simplicity, we write $\varepsilon$ to refer to $\varepsilon_2$ in this proof. Consider the following probability distribution $\mathbb{P}$ defined over 4 binary variables $\{A, B, X, Y\}$ in a topological ordering of $A \prec B \prec X \prec Y$: see Figure D.2.

We first check that all the (conditional) probabilities of $\mathbb{P}$ are well-defined. Since $0 < \sqrt{\varepsilon} \leq \alpha \leq 1/2$, the only non-straightforward term to verify is $\mathbb{P}(A = 1)$. Observe that

$$\frac{\varepsilon}{4\alpha} \cdot \frac{\alpha - \varepsilon/4}{1 - \sqrt{\varepsilon}/2} \leq 1 \iff \varepsilon \cdot (\alpha - \varepsilon/4) \leq 4\alpha \cdot (1 - \sqrt{\varepsilon}/2) \iff 2\alpha\sqrt{\varepsilon} + \alpha\varepsilon - \varepsilon^2/4 \leq 4\alpha$$

which is true as $0 < \varepsilon < \sqrt{\varepsilon} < \alpha \leq 1$ implies $2\alpha\sqrt{\varepsilon} + \alpha\varepsilon - \varepsilon^2/4 \leq 3\alpha\sqrt{\varepsilon} \leq 3\alpha \leq 4\alpha$. Therefore, $0 \leq \mathbb{P}(A = 1) \leq 1$.

One can verify the following conditional probabilities with respect to $\mathbb{P}$:




$$A = \begin{cases} 1 & \text{w.p. } \frac{\varepsilon}{4\alpha} \cdot \frac{\alpha - \varepsilon/4}{1 - \sqrt{\varepsilon}/2} \\ 0 & \text{else} \end{cases} \qquad X = \begin{cases} A & \text{w.p. } 1 - \alpha \\ 1 - A & \text{w.p. } \alpha - \sqrt{\varepsilon}/2 \\ B & \text{w.p. } \sqrt{\varepsilon}/2 \end{cases}$$

$$B = \begin{cases} 1 - A & \text{w.p. } 1 - \sqrt{\varepsilon} \\ 0 & \text{w.p. } \sqrt{\varepsilon}/2 \\ 1 & \text{w.p. } \sqrt{\varepsilon}/2 \end{cases} \qquad Y = \begin{cases} 1 & \text{if } X = 0, A = 1, B = 0 \\ 0 & \text{else} \end{cases}$$

Figure D.2: Probability distribution $\mathbb{P}$ defined over 4 binary variables $\{A, B, X, Y\}$ in a topological ordering of $A \prec B \prec X \prec Y$ with parameters $\varepsilon$ and $\alpha$, where $0 < \sqrt{\varepsilon} \leq \alpha \leq 1/2$.

| $a$ | $b$ | $\mathbb{P}(b \mid a)$ | $\mathbb{P}(X = 0 \mid a, b)$ | $\mathbb{P}(X = 0 \mid a)$ | $\sum_x \lvert\mathbb{P}(x \mid a, b) - \mathbb{P}(x \mid a)\rvert$ |
|---|---|---|---|---|---|
| 0 | 0 | $\sqrt{\varepsilon}/2$ | $1 - \alpha + \sqrt{\varepsilon}/2$ | $1 - \alpha + \varepsilon/4$ | $\sqrt{\varepsilon} - \varepsilon/2$ |
| 0 | 1 | $1 - \sqrt{\varepsilon}/2$ | $1 - \alpha$ | $1 - \alpha + \varepsilon/4$ | $\varepsilon/2$ |
| 1 | 0 | $1 - \sqrt{\varepsilon}/2$ | $\alpha$ | $\alpha - \varepsilon/4$ | $\varepsilon/2$ |
| 1 | 1 | $\sqrt{\varepsilon}/2$ | $\alpha - \sqrt{\varepsilon}/2$ | $\alpha - \varepsilon/4$ | $\sqrt{\varepsilon} - \varepsilon/2$ |

For instance,

$$\begin{aligned} &\mathbb{P}(X = 0 \mid A = 0) \\ &= \mathbb{P}(B = 0 \mid A = 0) \cdot \mathbb{P}(X = 0 \mid A = 0, B = 0) \\ &\quad + \mathbb{P}(B = 1 \mid A = 0) \cdot \mathbb{P}(X = 0 \mid A = 0, B = 1) \\ &= (\sqrt{\varepsilon}/2) \cdot (1 - \alpha + \sqrt{\varepsilon}/2) + (1 - \sqrt{\varepsilon}/2) \cdot (1 - \alpha) \\ &= 1 - \alpha + \varepsilon/4 \end{aligned}$$

and

$$\begin{aligned} &\mathbb{P}(X = 0 \mid A = 1) \\ &= \mathbb{P}(B = 0 \mid A = 1) \cdot \mathbb{P}(X = 0 \mid A = 1, B = 0) \\ &\quad + \mathbb{P}(B = 1 \mid A = 1) \cdot \mathbb{P}(X = 0 \mid A = 1, B = 1) \\ &= (1 - \sqrt{\varepsilon}/2) \cdot \alpha + (\sqrt{\varepsilon}/2) \cdot (\alpha - \sqrt{\varepsilon}/2) \\ &= \alpha - \varepsilon/4 \\ &= 1 - \mathbb{P}(X = 0 \mid A = 0) \end{aligned}$$

The detailed workings for $\sum_x \lvert\mathbb{P}(x \mid a, b) - \mathbb{P}(x \mid a)\rvert$ for different values of $a, b \in \{0, 1\}$ are as follows:

**When $A = 0$ and $B = 0$:**

$$
\begin{aligned}
&\sum_x |\mathbb{P}(x \mid a, b) - \mathbb{P}(x \mid a)| \\
&= |\mathbb{P}(X = 0 \mid A = 0, B = 0) - \mathbb{P}(X = 0 \mid A = 0)| \\
&\quad + |\mathbb{P}(X = 1 \mid A = 0, B = 0) - \mathbb{P}(X = 1 \mid A = 0)| \\
&= \left|(1 - \alpha + \sqrt{\varepsilon}/2) - (1 - \alpha + \varepsilon/4)\right| + \left|(\alpha - \sqrt{\varepsilon}/2) - (\alpha - \varepsilon/4)\right| \\
&= 2\left(\sqrt{\varepsilon}/2 - \varepsilon/4\right) \\
&= \sqrt{\varepsilon} - \varepsilon/2
\end{aligned}
$$

**When $A = 0$ and $B = 1$:**

$$
\begin{aligned}
&\sum_x |\mathbb{P}(x \mid a, b) - \mathbb{P}(x \mid a)| \\
&= |\mathbb{P}(X = 0 \mid A = 0, B = 1) - \mathbb{P}(X = 0 \mid A = 0)| \\
&\quad + |\mathbb{P}(X = 1 \mid A = 0, B = 1) - \mathbb{P}(X = 1 \mid A = 0)| \\
&= |(1 - \alpha) - (1 - \alpha + \varepsilon/4)| + |(\alpha) - (\alpha - \varepsilon/4)| \\
&= 2\,(\varepsilon/4) \\
&= \varepsilon/2
\end{aligned}
$$

**When $A = 1$ and $B = 0$:**

$$
\begin{aligned}
&\sum_x |\mathbb{P}(x \mid a, b) - \mathbb{P}(x \mid a)| \\
&= |\mathbb{P}(X = 0 \mid A = 1, B = 0) - \mathbb{P}(X = 0 \mid A = 1)| \\
&\quad + |\mathbb{P}(X = 1 \mid A = 1, B = 0) - \mathbb{P}(X = 1 \mid A = 1)| \\
&= |(\alpha) - (\alpha - \varepsilon/4)| + |(1 - \alpha) - (1 - \alpha + \varepsilon/4)| \\
&= 2\,(\varepsilon/4) \\
&= \varepsilon/2
\end{aligned}
$$

**When $A = 1$ and $B = 1$:**

$$
\begin{aligned}
&\sum_x |\mathbb{P}(x \mid a, b) - \mathbb{P}(x \mid a)| \\
&= |\mathbb{P}(X = 0 \mid A = 1, B = 1) - \mathbb{P}(X = 0 \mid A = 1)| \\
&\quad + |\mathbb{P}(X = 1 \mid A = 1, B = 1) - \mathbb{P}(X = 1 \mid A = 1)| \\
&= \left|(\alpha - \sqrt{\varepsilon}/2) - (\alpha - \varepsilon/4)\right| + \left|(1 - \alpha + \sqrt{\varepsilon}/2) - (1 - \alpha + \varepsilon/4)\right| \\
&= 2\left(\sqrt{\varepsilon}/2 - \varepsilon/4\right) \\
&= \sqrt{\varepsilon} - \varepsilon/2
\end{aligned}
$$

We are now ready to verify the required properties about $\mathbb{P}$:

1. $X \perp\!\!\!\perp_\varepsilon B \mid A$
2. $\min_{x,a} \mathbb{P}(x \mid a) \geq \alpha/2$
3. $|\hat{\mathbb{P}}_x(y) - \mathbb{P}_x(y)| \geq \frac{\varepsilon}{8\alpha}$

We can see that

$$\begin{aligned}
&\sum_{x,a,b} \mathbb{P}(a) \cdot |\mathbb{P}(x, b \mid a) - \mathbb{P}(x \mid a) \cdot \mathbb{P}(b \mid a)| \\
&= \sum_{a,b} \mathbb{P}(a) \cdot \mathbb{P}(b \mid a) \cdot \sum_x |\mathbb{P}(x \mid a, b) - \mathbb{P}(x \mid a)| \\
&= \mathbb{P}(A=0) \cdot \mathbb{P}(B=0 \mid A=0) \cdot (\sqrt{\varepsilon} - \varepsilon/2) + \mathbb{P}(A=0) \cdot \mathbb{P}(B=1 \mid A=0) \cdot (\varepsilon/2) \\
&\quad + \mathbb{P}(A=1) \cdot \mathbb{P}(B=0 \mid A=1) \cdot (\varepsilon/2) + \mathbb{P}(A=1) \cdot \mathbb{P}(B=1 \mid A=1) \cdot (\sqrt{\varepsilon} - \varepsilon/2) \\
&= \mathbb{P}(A=0) \cdot (\sqrt{\varepsilon}/2) \cdot (\sqrt{\varepsilon} - \varepsilon/2) + \mathbb{P}(A=0) \cdot (1 - \sqrt{\varepsilon}/2) \cdot (\varepsilon/2) \\
&\quad + \mathbb{P}(A=1) \cdot (1 - \sqrt{\varepsilon}/2) \cdot (\varepsilon/2) + \mathbb{P}(A=1) \cdot (\sqrt{\varepsilon}/2) \cdot (\sqrt{\varepsilon} - \varepsilon/2) \\
&= (\sqrt{\varepsilon}/2) \cdot (\sqrt{\varepsilon} - \varepsilon/2) + (1 - \sqrt{\varepsilon}/2) \cdot \varepsilon/2 \\
&= \varepsilon
\end{aligned}$$

Therefore, $X \perp\!\!\!\perp_\varepsilon B \mid A$.

Since $\varepsilon \leq \alpha/2$, we also have $\min_a \mathbb{P}(x \mid a) = \alpha - \varepsilon/4 \geq \alpha/2$.

It remains to argue that $|\hat{p}_x(y) - p_x(y)| > \varepsilon/\alpha$.

Let us now consider $\mathbb{P}_x(y)$. Since $X \perp\!\!\!\perp_\varepsilon B \mid A$, Algorithm 14 may terminate with $\{A\}$ as the adjustment set and output the estimate $\hat{\mathbb{P}}_x(y) = \sum_a \mathbb{P}(a) \cdot \mathbb{P}(y \mid x, a)$. By construction, $Y$ is an indicator variable for whether $(A, B, X) = (1, 0, 0)$ and $\{A, B\}$ is the only valid adjustment set for $\mathbb{P}_x(y)$. So,

$$\begin{aligned}
&\left| \sum_a \mathbb{P}(a) \cdot \mathbb{P}(y \mid x, a) - \sum_{a,b} \mathbb{P}(a, b) \cdot \mathbb{P}(y \mid x, a, b) \right| \\
&= \left| \sum_{a,b} \mathbb{P}(a) \cdot \mathbb{P}(y \mid x, a, b) \cdot \Big[ \mathbb{P}(b \mid a) - \mathbb{P}(b \mid x, a) \Big] \right| \\
&= \left| \sum_{a,b} \mathbb{P}(y \mid x, a, b) \cdot \frac{\mathbb{P}(a) \cdot \mathbb{P}(b \mid a)}{\mathbb{P}(x \mid a)} \cdot \Big[ \mathbb{P}(x \mid a) - \mathbb{P}(x \mid a, b) \Big] \right| \\
&= \mathbb{P}(A=1) \cdot \frac{\mathbb{P}(B=0 \mid A=1)}{\mathbb{P}(X=0 \mid A=1)} \cdot \Big| \mathbb{P}(X=0 \mid A=1) - \mathbb{P}(X=0 \mid A=1, B=0) \Big| \\
&= \frac{\varepsilon}{4\alpha} \cdot \frac{\alpha - \varepsilon/4}{1 - \sqrt{\varepsilon}/2} \cdot \frac{1 - \sqrt{\varepsilon}/2}{\alpha - \varepsilon/4} \cdot (\varepsilon/2) \\
&= \frac{\varepsilon}{8\alpha}
\end{aligned}$$

Therefore, $|\hat{\mathbb{P}}_x(y) - \mathbb{P}_x(y)| \geq \frac{\varepsilon}{8\alpha}$ even if we know the exact conditional probabilities of $\mathbb{P}$. □

## D.7 Experimental details

All experiments are run locally on a Macbook Pro (2019) and everything ran under a minute. We provide the code used for reproducing our experiments in the following anonymous repository: https://anonymous.4open.science/r/causal-effect-estimation-without-graph-8013/README.md.

### D.7.1 Synthetic causal DAG generation.

We generate 100 ten-node causal DAGs using the `causaldag` package. Denoting the nodes as $V_1, \dots V_{10}$, we take $X = V_9$ and $Y = V_{10}$. For $d \in [100]$, the basic structure of DAG $\mathcal{G}_d$ is drawn from a directed Erdös-Rényi model with density 0.5, with additional processing to ensure that $X \to Y$ and that $|\text{pa}(X)|$ and $|\text{pa}(Y)|$ are uniformly distributed between 1 and 5. Then, for $i \in [10]$, the causal mechanism $\mathbb{P}(V_i \mid \text{pa}_{\mathcal{G}_d}(V_i))$ is drawn from a product of $\text{Beta}(1,1)$ distributions.[3] For $\mathbb{P}(Y \mid \text{pa}_{\mathcal{G}_d}(V_i))$, we *tilt* these original distributions towards one by setting $\mathbb{P}_{new}(Y = y \mid \text{pa}_{\mathcal{G}_d}(V_i)) \propto \mathbb{P}(Y = y \mid \text{pa}_{\mathcal{G}_d}(V_i)) \cdot e^{2y}$, ensuring that the baselines cannot perform just by increasing smoothing.

### D.7.2 Implementation details.

For the subroutines `EstimateMarginal` and `EstimateConditional` , we fit the causal mechanisms $\mathbb{P}(V_i \mid \text{pa}_{\mathcal{G}_d}(V_i))$ separately for each value of $\mathbb{P}_{\mathcal{G}_d}(V_i)$ using a smoothed empirical counts estimator (also called the add-$n$ estimator, see e.g., [Bha+22]). For the `ApproxCondInd` subroutine, we compared the U-statistic based test proposed by [Kim+23] against a simple thresholding test on the total variation distance between $\widehat{\mathbb{P}}(\mathbf{C}, \mathbf{A}, \mathbf{B})$ and $\widehat{\mathbb{P}}(\mathbf{C}) \cdot \widehat{\mathbb{P}}(\mathbf{A} \mid \mathbf{C}) \cdot \widehat{\mathbb{P}}(\mathbf{B} \mid \mathbf{C})$, and found the thresholding test achieved better performance.

### D.7.3 Hyperparameter selection.

Let $\alpha_{grid} = (0.1, 0.25, 0.5, 1, 2)$ and $\varepsilon_{grid} = (0.05, 0.75, 0.1, 0.2)$. We run all three methods with each value $\alpha \in \alpha_{grid}$ used for smoothing, and we run `ApproxSearch` also with all values of $\varepsilon \in \varepsilon_{grid}$ used for conditional independence thresholding, over a training set of 10 DAGs with 5 runs for each DAG. For each method, we select the hyperparameters which perform give the best mean squared error of the produced estimate, and use those for our main experiments. For all methods, we have $\alpha = 0.1$, and for `ApproxSearch`, $\varepsilon = 0.2$.

[3] i.e., for $\mathbf{w} \in \{0,1\}^{|\text{pa}_{\mathcal{G}_d}(V_i)|}$, we draw $p_{\mathbf{w}}$ independently from $\text{Beta}(1,1)$, and set $\mathbb{P}(V_i = 1 \mid \text{pa}_{\mathcal{G}_d}(V_i) = \mathbf{w}) = p_{\mathbf{w}}$.

# References


[ADH10] Alberto Abadie, Alexis Diamond, and Jens Hainmueller. "Synthetic control methods for comparative case studies: Estimating the effect of California's tobacco control program". In: *Journal of the American statistical Association* 105.490 (2010), pp. 493–505 (cit. on pp. 52, 80, 82).

[AHZ21] Jeffrey Adams, Niels Hansen, and Kun Zhang. "Identification of partially observed linear causal models: graphical conditions for the non-Gaussian and heterogeneous cases". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 22822–22833 (cit. on p. 92).

[Aga+18] Anish Agarwal, Muhammad Jehangir Amjad, Devavrat Shah, and Dennis Shen. "Model agnostic time series analysis via matrix estimation". In: *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 2.3 (2018), pp. 1–39 (cit. on p. 77).

[ASS20] Anish Agarwal, Devavrat Shah, and Dennis Shen. "Synthetic interventions". In: *arXiv:2006.07691* (2020) (cit. on pp. 52, 72, 74, 78, 80–82, 86, 133).

[Aga+19] Anish Agarwal, Devavrat Shah, Dennis Shen, and Dogyoon Song. "On robustness of principal component regression". In: *Advances in Neural Information Processing Systems* 32 (2019) (cit. on p. 80).

[Agr+23] Raj Agrawal, Chandler Squires, Neha Prasad, and Caroline Uhler. "The DeCAMFounder: nonlinear causal discovery in the presence of hidden variables". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 85.5 (2023), pp. 1639–1658 (cit. on pp. 60, 62, 72).

[AHB21] Kartik Ahuja, Jason Hartford, and Yoshua Bengio. "Properties from mechanisms: an equivariance perspective on identifiable representation learning". In: *International Conference on Learning Representations*. 2021 (cit. on pp. 90, 91).

[AHB22] Kartik Ahuja, Jason Hartford, and Yoshua Bengio. "Weakly supervised representation learning with sparse perturbations". In: *Advances in Neural Information Processing Systems*. Ed. by Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho. 2022. URL: https://openreview.net/forum?id=6ZI4iF_T7t (cit. on p. 92).

[Ahu+23] Kartik Ahuja, Divyat Mahajan, Yixin Wang, and Yoshua Bengio. "Interventional causal representation learning". In: *International conference on machine learning*. PMLR. 2023, pp. 372–407 (cit. on pp. 92, 93, 104).

[AMP97] Steen A. Andersson, David Madigan, and Michael D. Perlman. "A characterization of Markov equivalence classes for acyclic digraphs". In: *The Annals of Statistics* 25.2 (1997), pp. 505–541 (cit. on p. 112).

[Ark+21] Dmitry Arkhangelsky, Susan Athey, David A Hirshberg, Guido W Imbens, and Stefan Wager. "Synthetic difference-in-differences". In: *American Economic Review* 111.12 (2021), pp. 4088–4118 (cit. on p. 82).

[Ars+18] Salim Arslan, Sofia Ira Ktena, Antonios Makropoulos, Emma C Robinson, Daniel Rueckert, and Sarah Parisot. "Human brain mapping: A systematic comparison of parcellation methods for the human cerebral cortex". In: *NeuroImage* 170 (2018), pp. 5–30 (cit. on p. 61).

[Art11] M. Artin. *Algebra*. Pearson Education, 2011 (cit. on p. 23).

[Ash+00] Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. "Gene ontology: tool for the unification of biology". In: *Nature Genetics* 25.1 (2000), pp. 25–29 (cit. on p. 102).

[Ath+21] Susan Athey, Mohsen Bayati, Nikolay Doudchenko, Guido Imbens, and Khashayar Khosravi. "Matrix completion methods for causal panel data models". In: *Journal of the American Statistical Association* 116.536 (2021), pp. 1716–1730 (cit. on pp. 80, 82).

[AIW18] Susan Athey, Guido W Imbens, and Stefan Wager. "Approximate residual balancing: debiased inference of average treatment effects in high dimensions". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 80.4 (2018), pp. 597–623 (cit. on p. 115).

[BF+19] Md Badsha, Audrey Qiuyan Fu, et al. "Learning causal biological networks with the principle of Mendelian randomization". In: *Frontiers in Genetics* 10 (2019), p. 460 (cit. on p. 82).

[Bai09] Jushan Bai. "Panel data models with interactive fixed effects". In: *Econometrica* 77.4 (2009), pp. 1229–1279. URL: http://www.jstor.org/stable/40263859 (cit. on pp. 80, 82).

[BW16] Jushan Bai and Peng Wang. "Econometric analysis of large factor models". In: *Annual Review of Economics* 8 (2016), pp. 53–80 (cit. on p. 61).

[Bar+22] Elias Bareinboim, Juan Correa, Duligur Ibeling, and Thomas Icard. "On Pearl's Hierarchy and the Foundations of Causal Inference (1st edition)". In: *Probabilistic and Causal Inference: the Works of Judea Pearl*. Ed. by Hector Geffner, Rita Dechter, and Joseph Halpern. ACM Books, 2022, pp. 507–556 (cit. on p. 30).

[BP12] Elias Bareinboim and Judea Pearl. "Causal inference by surrogate experiments: z-identifiability". In: *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*. 2012, pp. 113–120 (cit. on p. 113).

[BP14] Elias Bareinboim and Judea Pearl. "Transportability from multiple environments with limited experiments: completeness results". In: *Advances in Neural Information Processing Systems*. 2014, pp. 280–288 (cit. on p. 76).

[BP16] Elias Bareinboim and Judea Pearl. "Causal inference and the data-fusion problem". In: *Proceedings of the National Academy of Sciences* 113.27 (2016), pp. 7345–7352 (cit. on p. 76).

[BK07] Robert M Bell and Yehuda Koren. "Lessons from the Netflix prize challenge". In: *Acm Sigkdd Explorations Newsletter* 9.2 (2007), pp. 75–79 (cit. on p. 76).

[BCH14] Alexandre Belloni, Victor Chernozhukov, and Christian Hansen. "Inference on treatment effects after selection among high-dimensional controls". In: *Review of Economic Studies* 81.2 (2014), pp. 608–650 (cit. on p. 115).

[Bel+21] Anastasiya Belyaeva, Louis Cammarata, Adityanarayanan Radhakrishnan, Chandler Squires, Karren Dai Yang, GV Shivashankar, and Caroline Uhler. "Causal network models of SARS-CoV-2 expression and aging to identify candidates for drug repurposing". In: *Nature communications* 12.1 (2021), p. 1024 (cit. on p. 76).

[Bel+17] Anastasiya Belyaeva, Saradha Venkatachalapathy, Mallika Nagarajan, G.V. Shivashankar, and Caroline Uhler. "Network analysis identifies chromosome intermingling regions as regulatory hotspots for transcription". In: *PNAS* 114.52 (2017), pp. 13714–13719 (cit. on p. 60).

[BCV13] Yoshua Bengio, Aaron Courville, and Pascal Vincent. "Representation learning: a review and new perspectives". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8 (2013), pp. 1798–1828 (cit. on p. 90).

[Ber+20] Daniel Bernstein, Basil Saeed, Chandler Squires, and Caroline Uhler. "Ordering-based causal structure learning in the presence of latent variables". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, pp. 4098–4108 (cit. on p. 60).

[BPZ17] Dimitris Bertsimas, Colin Pawlowski, and Ying Daisy Zhuo. "From predictive methods to missing data imputation: an optimization approach". In: *The Journal of Machine Learning Research* 18.1 (2017), pp. 7133–7171 (cit. on p. 76).

[Bha+22] Arnab Bhattacharyya, Sutanu Gayen, Saravanan Kandasamy, Vedant Raval, and Vinodchandran N Variyam. "Efficient interventional distribution learning in the PAC framework". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2022, pp. 7531–7549 (cit. on p. 171).

[Bha+21] Arnab Bhattacharyya, Sutanu Gayen, Eric Price, and NV Vinodchandran. "Near-optimal learning of tree-structured distributions by Chow-Liu". In: *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*. 2021, pp. 147–160 (cit. on p. 159).

[Bla17] Jr Blalock. *Causal Models in the Social Sciences*. Routledge, 2017 (cit. on p. 60).

[BWZ19] Jelena Bradic, Stefan Wager, and Yinchu Zhu. "Sparsity double robust inference of average treatment effects". In: *arXiv:1905.00744* (2019) (cit. on p. 115).

[Bre+22] Johann Brehmer, Pim De Haan, Phillip Lippe, and Taco S Cohen. "Weakly supervised causal representation learning". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 38319–38331 (cit. on p. 92).

[Bre01] Leo Breiman. "Statistical modeling: The two cultures (with comments and a rejoinder by the author)". In: *Statistical science* 16.3 (2001), pp. 199–231 (cit. on p. 21).

[Bro97] Rasmus Bro. "PARAFAC. Tutorial and applications". In: *Chemometrics and Intelligent Laboratory Systems* 38.2 (1997), pp. 149–171 (cit. on p. 77).

[Cai+19] Ruichu Cai, Feng Xie, Clark Glymour, Zhifeng Hao, and Kun Zhang. "Triad constraints for learning causal structure of latent variables". In: *Advances in Neural Information Processing Systems* 32 (2019), pp. 12883–12892 (cit. on pp. 60, 62, 72, 92).

[Can20] Clément L Canonne. "A survey on distribution testing: your data is big. But is it blue?" In: *Theory of Computing* (2020), pp. 1–100 (cit. on p. 115).

[Can+22] Clément L Canonne et al. "Topics and techniques in distribution testing: A biased but representative sample". In: *Foundations and Trends® in Communications and Information Theory* 19.6 (2022), pp. 1032–1198 (cit. on pp. 162, 163).

[Can+18] Clément L Canonne, Ilias Diakonikolas, Daniel M Kane, and Alistair Stewart. "Testing conditional independence of discrete distributions". In: *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*. 2018, pp. 735–748 (cit. on p. 159).

[CP22] Federico Castelletti and Stefano Peluso. "Network structure learning under uncertain interventions". In: *Journal of the American Statistical Association* (2022), pp. 1–12 (cit. on p. 100).

[CGM24] Ting-Hsuan Chang, Zijian Guo, and Daniel Malinsky. "Post-selection inference for causal effects after causal discovery". In: *arXiv:2405.06763* (2024) (cit. on p. 114).

[CSS19] Moses Charikar, Kirankumar Shiragur, and Aaron Sidford. "A general framework for symmetric property estimation". In: *Advances in Neural Information Processing Systems* 32 (2019) (cit. on p. 115).

[Cha15] Sourav Chatterjee. "Matrix estimation by universal singular value thresholding". In: *The Annals of Statistics* 43.1 (2015), pp. 177–214 (cit. on pp. 77, 80).

[CZ23] Yiqun Chen and James Zou. "GenePT: A Simple But Effective Foundation Model for Genes and Cells Built From ChatGPT". In: *bioRxiv* (2023) (cit. on p. 50).

[Che+22] Debo Cheng, Jiuyong Li, Lin Liu, Kui Yu, Thuc Duy Le, and Jixue Liu. "Toward unique and unbiased causal effect estimation from data with hidden variables". In: *IEEE Transactions on Neural Networks and Learning Systems* (2022) (cit. on p. 114).

[Che+18] Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. "Double/debiased machine learning for treatment and structural parameters". In: *The Econometrics Journal* 21.1 (2018), pp. C1–C68 (cit. on p. 115).

[CCK13] Victor Chernozhukov, Denis Chetverikov, and Kengo Kato. "Gaussian approximations and multiplier bootstrap for maxima of sums of high-dimensional random vectors". In: *The Annals of Statistics* 41.6 (2013), pp. 2786–2819 (cit. on p. 68).

[Chi95] David Maxwell Chickering. "A transformational characterization of equivalent Bayesian network structures". In: *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*. UAI'95. Montréal, Qué, Canada: Morgan Kaufmann Publishers Inc., 1995, pp. 87–98 (cit. on p. 108).

[Chi02] David Maxwell Chickering. "Optimal structure identification with greedy search". In: *Journal of Machine Learning Research* 3.Nov (2002), pp. 507–554 (cit. on pp. 60, 62, 63, 67, 113).

[Chi20] Max Chickering. "Statistically efficient greedy equivalence search". In: *Conference on Uncertainty in Artificial Intelligence*. PMLR. 2020, pp. 241–249 (cit. on p. 67).

[CHM04] Max Chickering, David Heckerman, and Chris Meek. "Large-sample learning of Bayesian networks is NP-hard". In: *Journal of Machine Learning Research* 5 (2004), pp. 1287–1330 (cit. on p. 67).

[Cho+11] Myung Jin Choi, Vincent YF Tan, Animashree Anandkumar, and Alan S Willsky. "Learning latent tree graphical models". In: *Journal of Machine Learning Research* 12 (2011), pp. 1771–1812 (cit. on p. 61).

[CGB23] Davin Choo, Themistoklis Gouleakis, and Arnab Bhattacharyya. "Active causal structure learning with advice". In: *International Conference on Machine Learning*. PMLR. 2023, pp. 5838–5867 (cit. on p. 113).

[CS23a] Davin Choo and Kirankumar Shiragur. "Adaptivity complexity for causal graph discovery". In: *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*. UAI '23. 2023 (cit. on p. 113).

[CS23b] Davin Choo and Kirankumar Shiragur. "New metrics and search algorithms for weighted causal DAGs". In: *International Conference on Machine Learning*. PMLR. 2023, pp. 5868–5903 (cit. on p. 113).

[CS23c] Davin Choo and Kirankumar Shiragur. "Subset verification and search algorithms for causal DAGs". In: *International Conference on Artificial Intelligence and Statistics*. 2023 (cit. on p. 113).

[CSB22] Davin Choo, Kirankumar Shiragur, and Arnab Bhattacharyya. "Verification and search algorithms for causal DAGs". In: *Advances in Neural Information Processing Systems* 35 (2022) (cit. on p. 113).

[CH20] Carlos Cinelli and Chad Hazlett. "Making sense of sensitivity: extending omitted variable bias". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 82.1 (2020), pp. 39–67 (cit. on p. 114).

[Col+12] Diego Colombo, Marloes H Maathuis, Markus Kalisch, and Thomas S Richardson. "Learning high-dimensional directed acyclic graphs with latent and selection variables". In: *The Annals of Statistics* (2012), pp. 294–321 (cit. on pp. 61, 113).

[Com94] Pierre Comon. "Independent component analysis, a new concept?" In: *Signal Processing* 36.3 (1994), pp. 287–314 (cit. on pp. 90, 91).

[CR70] Roger C Conant and W Ross Ashby. "Every good regulator of a system must be a model of that system". In: *International Journal of Systems Science* 1.2 (1970), pp. 89–97 (cit. on p. 30).

[DAm+21] Alexander D'Amour, Peng Ding, Avi Feller, Lihua Lei, and Jasjeet Sekhon. "Overlap in observational studies with high-dimensional covariates". In: *Journal of Econometrics* 221.2 (2021), pp. 644–654 (cit. on pp. 110, 115).

[Dav19] Cameron Davidson-Pilon. "lifelines: survival analysis in Python". In: *Journal of Open Source Software* 4.40 (2019), p. 1317 (cit. on p. 151).

[Dix+16] 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". In: *Cell* 167.7 (2016), pp. 1853–1866 (cit. on p. 76).

[Do 92] Manfredo Perdigao Do Carmo. *Riemannian Geometry*. Trans. by J Flaherty Francis. Vol. 2. Springer, 1992 (cit. on p. 23).

[Drt+17] Mathias Drton, Shaowei Lin, Luca Weihs, and Piotr Zwiernik. "Marginal likelihood and model selection for Gaussian latent tree and forest models". In: *Bernoulli* 23.2 (2017), pp. 1202–1232 (cit. on p. 61).

[DMO08] Mathias Drton, Hélène Massam, and Ingram Olkin. "Moments of minors of Wishart matrices". In: *The Annals of Statistics* 36.5 (2008), pp. 2261–2283 (cit. on pp. 68, 145).

[DP07] Mathias Drton and Michael D Perlman. "Multiple testing and error control in Gaussian graphical model selection". In: *Statistical Science* 22.3 (2007), pp. 430–449 (cit. on p. 68).

[DSS07] Mathias Drton, Bernd Sturmfels, and Seth Sullivant. "Algebraic factor analysis: tetrads, pentads and beyond". In: *Probability Theory and Related Fields* 138.3-4 (2007), pp. 463–493 (cit. on pp. 61, 72, 145).

[Ebe07] Frederick Eberhardt. "Causation and intervention". In: *Unpublished doctoral dissertation, Carnegie Mellon University* (2007), p. 93 (cit. on p. 113).

[EGS05] Frederick Eberhardt, Clark Glymour, and Richard Scheines. "On the number of experiments sufficient and in the worst case necessary to identify all causal relations among N variables". In: *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*. 2005, pp. 178–184 (cit. on pp. 91, 100, 112, 113).

[EGS06] Frederick Eberhardt, Clark Glymour, and Richard Scheines. "N-1 experiments suffice to determine the causal relations among N variables". In: *Innovations in machine learning*. Springer, 2006, pp. 97–112 (cit. on p. 113).

[Eis+98] Michael B Eisen, Paul T Spellman, Patrick O Brown, and David Botstein. "Cluster analysis and display of genome-wide expression patterns". In: *Proceedings of the National Academy of Sciences* 95.25 (1998), pp. 14863–14868 (cit. on p. 61).

[EG22] Adam N Elmachtoub and Paul Grigas. "Smart "predict, then optimize"". In: *Management Science* 68.1 (2022), pp. 9–26 (cit. on p. 25).

[Ena+19] Oana M Enache, David L Lahr, Ted E Natoli, Lev Litichevskiy, David Wadden, Corey Flynn, Joshua Gould, Jacob K Asiedu, Rajiv Narayan, and Aravind Subramanian. "The GCTx format and cmap {Py, R, M, J} packages: resources for optimized storage and integrated traversal of annotated dense matrices". In: *Bioinformatics* 35.8 (2019), pp. 1427–1429 (cit. on p. 136).

[EHS13] Doris Entner, Patrik Hoyer, and Peter Spirtes. "Data-driven covariate selection for nonparametric estimation of causal effects". In: *Artificial intelligence and statistics*. PMLR. 2013, pp. 256–264 (cit. on p. 114).

[ES02] Steven N Evans and Philip B Stark. "Inverse problems as statistics". In: *Inverse problems* 18.4 (2002), R55 (cit. on p. 35).

[FLM21] Max H Farrell, Tengyuan Liang, and Sanjog Misra. "Deep neural networks for estimation and inference". In: *Econometrica* 89.1 (2021), pp. 181–213 (cit. on p. 115).

[Fri+00] Nir Friedman, Michal Linial, Iftach Nachman, and Dana Pe'er. "Using Bayesian networks to analyze expression data". In: *Journal of Computational Biology* 7.3-4 (2000), pp. 601–620 (cit. on pp. 60, 82).

[FNM19] Benjamin Frot, Preetam Nandy, and Marloes H Maathuis. "Robust causal structure learning with some hidden variables". In: *Journal of the Royal Statistical Society* (2019) (cit. on pp. 60, 62, 72).

[GA21] Ming Gao and Bryon Aragam. "Efficient Bayesian network structure learning via local Markov boundary search". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 4301–4313 (cit. on p. 122).

[Ger+23] Jan E Gerken, Jimmy Aronsson, Oscar Carlsson, Hampus Linander, Fredrik Ohlsson, Christoffer Petersson, and Daniel Persson. "Geometric deep learning and equivariant neural networks". In: *Artificial Intelligence Review* 56.12 (2023), pp. 14605–14662 (cit. on p. 41).

[GST23] AmirEmad Ghassami, Ilya Shpitser, and Eric Tchetgen Tchetgen. "Partial identification of causal effects using proxy variables". In: *arXiv preprint arXiv:2304.04374* (2023) (cit. on p. 39).

[GN21] Evarist Giné and Richard Nickl. *Mathematical Foundations of Infinite-dimensional Statistical Models*. Cambridge University Press, 2021 (cit. on p. 23).

[GZS19] Clark Glymour, Kun Zhang, and Peter Spirtes. "Review of causal discovery methods based on graphical models". In: *Frontiers in Genetics* 10 (2019), p. 524 (cit. on p. 113).

[Gre+19] Kristjan Greenewald, Dmitriy Katz, Karthikeyan Shanmugam, Sara Magliacane, Murat Kocaoglu, Enric Boix-Adserà, and Guy Bresler. "Sample efficient active learning of causal trees". In: *Advances in Neural Information Processing Systems* 32 (2019) (cit. on p. 113).

[GSK21] Kristjan Greenewald, Karthikeyan Shanmugam, and Dmitriy Katz. "High-dimensional feature selection for sample efficient treatment effect estimation". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2021, pp. 2224–2232 (cit. on p. 115).

[Gre+07] Arthur Gretton, Kenji Fukumizu, Choon Hui Teo, Le Song, Bernhard Schölkopf, Alexander J Smola, et al. "A kernel statistical test of independence." In: *Nips*. Vol. 20. Citeseer. 2007, pp. 585–592 (cit. on p. 72).

[Gre66] Thomas Nall Eden Greville. "Note on the generalized inverse of a matrix product". In: *Siam Review* 8.4 (1966), pp. 518–521 (cit. on p. 142).

[GP21] Richard Guo and Emilija Perkovic. "Minimal enumeration of all possible total effects in a Markov equivalence class". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2021, pp. 2395–2403 (cit. on p. 39).

[Gyö+02] László Györfi, Michael Kohler, Adam Krzyzak, Harro Walk, et al. *A Distribution-free Theory of Nonparametric Regression*. Vol. 1. Springer, 2002 (cit. on p. 23).

[HHS15] Yoni Halpern, Steven Horng, and David Sontag. "Anchored discrete factor analysis". In: *arXiv:1511.03299* (2015) (cit. on p. 92).

[Har+70] Richard A Harshman et al. "Foundations of the PARAFAC procedure: models and conditions for an "explanatory" multimodal factor analysis". In: (1970) (cit. on p. 77).

[HB12] Alain Hauser and Peter Bühlmann. "Characterization and greedy learning of interventional Markov equivalence classes of directed acyclic graphs". In: *The Journal of Machine Learning Research* 13.1 (2012), pp. 2409–2464 (cit. on pp. 63, 91, 113).

[HMM18] Christina Heinze-Deml, Marloes H Maathuis, and Nicolai Meinshausen. "Causal structure learning". In: *Annual Review of Statistics and Its Application* 5 (2018), pp. 371–391 (cit. on p. 60).

[HKK07] Desmond J Higham, Gabriela Kalna, and Milla Kibble. "Spectral clustering and its use in bioinformatics". In: *Journal of computational and applied mathematics* 204.1 (2007), pp. 25–37 (cit. on p. 70).

[Hof+19] Markus Hofmarcher, Elisabeth Rumetshofer, Djork-Arne Clevert, Sepp Hochreiter, and Günter Klambauer. "Accurate prediction of biological assays with high-throughput microscopy images and convolutional networks". In: *Journal of Chemical Information and Modeling* 59.3 (2019), pp. 1163–1171 (cit. on p. 77).

[HL16] Mohammad Javad Hosseini and Su-In Lee. "Learning sparse Gaussian graphical models with overlapping blocks". In: *Advances in Neural Information Processing Systems*. 2016, pp. 3808–3816 (cit. on p. 61).

[HLV14] Huining Hu, Zhentao Li, and Adrian Vetta. "Randomized experimental design for causal graph discovery". In: *Advances in Neural Information Processing Systems* 27 (2014) (cit. on p. 113).

[Hua+21] Lamei Huang, Zhixing Guo, Fang Wang, and Liwu Fu. "KRAS mutation: from undruggable to druggable in cancer". In: *Signal Transduction and Targeted Therapy* 6.1 (2021), p. 386 (cit. on p. 103).

[HEH13] Antti Hyttinen, Frederick Eberhardt, and Patrik O Hoyer. "Experiment selection for causal discovery". In: *Journal of Machine Learning Research* 14 (2013), pp. 3041–3071 (cit. on p. 91).

[HST19] Aapo Hyvarinen, Hiroaki Sasaki, and Richard Turner. "Nonlinear ICA using auxiliary variables and generalized contrastive learning". In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR. 2019, pp. 859–868 (cit. on pp. 91, 125).

[HP99] Aapo Hyvärinen and Petteri Pajunen. "Nonlinear independent component analysis: Existence and uniqueness results". In: *Neural networks* 12.3 (1999), pp. 429–439 (cit. on p. 37).

[IA21] Guido Imbens and Susan Athey. "Breiman's two cultures: A perspective from econometrics". In: *Observational Studies* 7.1 (2021), pp. 127–133 (cit. on p. 21).

[Jab+20] Amin Jaber, Murat Kocaoglu, Karthikeyan Shanmugam, and Elias Bareinboim. "Causal discovery from soft interventions with unknown targets: characterization and learning". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 9551–9561 (cit. on pp. 91, 113).

[Jac+93] Dean A Jackson, A Bassim Hassan, Rachel J Errington, and Peter R Cook. "Visualization of focal sites of transcription within human nuclei." In: *The EMBO journal* 12.3 (1993), pp. 1059–1065 (cit. on p. 48).

[JO14] Prateek Jain and Sewoong Oh. "Provable tensor factorization with missing data". In: *Advances in Neural Information Processing Systems*. 2014, pp. 1431–1439 (cit. on p. 77).

[Jin+23] Zhijing Jin, Yuen Chen, Felix Leeb, Luigi Gresele, Ojasv Kamal, LYU Zhiheng, Kevin Blin, Fernando Gonzalez Adauto, Max Kleiman-Weiner, Mrinmaya Sachan, et al. "Cladder: Assessing causal reasoning in language models". In: *Thirty-seventh Conference on Neural Information Processing Systems*. 2023 (cit. on p. 19).

[KS22] Marcus Kaiser and Maksim Sipos. "Unsuitability of NOTEARS for causal graph discovery when dealing with dimensional quantities". In: *Neural Processing Letters* 54.3 (2022), pp. 1587–1595 (cit. on p. 114).

[Kea+94] Michael Kearns, Yishay Mansour, Dana Ron, Ronitt Rubinfeld, Robert E Schapire, and Linda Sellie. "On the learnability of discrete distributions". In: *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing*. 1994, pp. 273–282 (cit. on p. 110).

[Khe+20] Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvarinen. "Variational autoencoders and nonlinear ICA: a unifying framework". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, pp. 2207–2217 (cit. on pp. 90, 91).

[Kid+16] Brian A Kidd, Aleksandra Wroblewska, Mary R Boland, Judith Agudo, Miriam Merad, Nicholas P Tatonetti, Brian D Brown, and Joel T Dudley. "Mapping the effects of drugs on the immune system". In: *Nature Biotechnology* 34.1 (2016), p. 47 (cit. on p. 76).

[Kim+23] Ilmun Kim, Matey Neykov, Sivaraman Balakrishnan, and Larry Wasserman. "Conditional independence testing for discrete distributions: beyond chi-squared and G-tests". In: *arXiv:2308.05373* (2023) (cit. on p. 171).

[Kiv+21] Bohdan Kivva, Goutham Rajendran, Pradeep Ravikumar, and Bryon Aragam. "Learning latent causal graphs via mixture oracles". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 18087–18101 (cit. on p. 92).

[Kiv+22] Bohdan Kivva, Goutham Rajendran, Pradeep Ravikumar, and Bryon Aragam. "Identifiability of deep generative models without auxiliary information". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 15687–15701 (cit. on p. 125).

[KDV17] Murat Kocaoglu, Alex Dimakis, and Sriram Vishwanath. "Cost-optimal learning of causal graphs". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 1875–1884 (cit. on p. 113).

[KB09] Tamara G Kolda and Brett W Bader. "Tensor decompositions and applications". In: *SIAM review* 51.3 (2009), pp. 455–500 (cit. on p. 80).

[Kul+16] Maxim V Kuleshov, Matthew R Jones, Andrew D Rouillard, Nicolas F Fernandez, Qiaonan Duan, Zichen Wang, Simon Koplev, Sherry L Jenkins, Kathleen M Jagodnik, Alexander Lachmann, et al. "Enrichr: a comprehensive gene set enrichment analysis web server 2016 update". In: *Nucleic Acids Research* 44.W1 (2016), W90–W97 (cit. on p. 151).

[KR16] Erich Kummerfeld and Joseph Ramsey. "Causal clustering for 1-factor measurement models". In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2016, pp. 1655–1664 (cit. on pp. 60–62, 68, 70, 72).

[Laa+23] Lars van der Laan, Marco Carone, Alex Luedtke, and Mark van der Laan. "Adaptive debiased machine learning using data-driven model selection techniques". In: *arXiv preprint arXiv:2307.12544* (2023) (cit. on p. 127).

[Lam23] Wai-yin Lam. "Causal razors". In: *arXiv:2302.10331* (2023) (cit. on pp. 39, 109).

[LB20] Sanghack Lee and Elias Bareinboim. "Causal effect identifiability under partial-observability". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 5692–5701 (cit. on p. 113).

[LCB20a] Sanghack Lee, Juan D Correa, and Elias Bareinboim. "General identifiability with arbitrary surrogate experiments". In: *Uncertainty in Artificial Intelligence*. PMLR. 2020, pp. 389–398 (cit. on pp. 79, 113).

[LCB20b] Sanghack Lee, Juan D Correa, and Elias Bareinboim. "General transportability: synthesizing observations and experiments from heterogeneous domains". In: 2020 (cit. on p. 76).

[LD18] Dennis Leung and Mathias Drton. "Algebraic tests of general Gaussian latent tree models". In: *Advances in Neural Information Processing Systems* 31 (2018) (cit. on p. 61).

[Lie+09] Erez Lieberman-Aiden et al. "Comprehensive mapping of long-range interactions reveals folding principles of the human genome". In: *Science* 326 (2009), pp. 289–293 (cit. on p. 72).

[LH13] Wim J van der Linden and Ronald K Hambleton. *Handbook of Modern Item Response Theory*. Springer Science & Business Media, 2013 (cit. on p. 61).

[Lin+18] Erik M. Lindgren, Murat Kocaoglu, Alexandros G. Dimakis, and Sriram Vishwanath. "Experimental design for cost-aware learning of causal graphs". In: *Advances in Neural Information Processing Systems* 31 (2018) (cit. on p. 113).

[Liu+18] Jianfang Liu, Tara Lichtenberg, Katherine A Hoadley, Laila M Poisson, Alexander J Lazar, Andrew D Cherniack, Albert J Kovatich, Christopher C Benz, Douglas A Levine, Adrian V Lee, et al. "An integrated TCGA pan-cancer clinical data resource to drive high-quality survival outcome analytics". In: *Cell* 173.2 (2018), pp. 400–416 (cit. on pp. 103, 151).

[Liu+22] Yuhang Liu, Zhen Zhang, Dong Gong, Mingming Gong, Biwei Huang, Anton van den Hengel, Kun Zhang, and Javen Qinfeng Shi. "Weight-variant latent causal models". In: *arXiv:2208.14153* (2022) (cit. on pp. 92, 93, 100, 148).

[LWT19] Mohammad Lotfollahi, Alexander F Wolf, and Fabian J Theis. "scGen predicts single-cell perturbation responses". In: *Nature Methods* 16.8 (2019), p. 715 (cit. on p. 76).

[MC15] Marloes H Maathuis and Diego Colombo. "A generalized back-door criterion". In: *The Annals of Statistics* 43.3 (2015), pp. 1060–1088 (cit. on p. 110).

[MN19] Oliver J Maclaren and Ruanui Nicholson. "What can be estimated? Identifiability, estimability, causal inference and ill-posed inverse problems". In: *arXiv preprint arXiv:1904.02826* (2019) (cit. on pp. 34, 35).

[Mal24] Daniel Malinsky. "A cautious approach to constraint-based causal model selection". In: *arXiv:2404.18232* (2024) (cit. on p. 114).

[McI+18] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. "UMAP: uniform manifold approximation and projection". In: *Journal of Open Source Software* 3.29 (2018) (cit. on p. 83).

[MFF21] Andrew C Miller, Nicholas J Foti, and Emily B Fox. "Breiman's two cultures: You don't have to choose sides". In: *Observational Studies* 7.1 (2021), pp. 161–169 (cit. on p. 21).

[Mor+22] Gemma Elyse Moran, Dhanya Sridhar, Yixin Wang, and David Blei. "Identifiable deep generative models via sparse decoding". In: *Transactions on Machine Learning Research* (2022) (cit. on p. 90).

[NHM18] Preetam Nandy, Alain Hauser, and Marloes H Maathuis. "High-dimensional consistency in score-based and hybrid structure learning". In: *The Annals of Statistics* 46.6A (2018), pp. 3151–3183 (cit. on p. 60).

[NHZ24] Ignavier Ng, Biwei Huang, and Kun Zhang. "Structure learning with continuous optimization: A sober look and beyond". In: *Causal Learning and Reasoning*. PMLR. 2024, pp. 71–105 (cit. on p. 114).

[Pea95] Judea Pearl. "Causal diagrams for empirical research". In: *Biometrika* 82.4 (1995), pp. 669–688 (cit. on p. 113).

[Pea09] Judea Pearl. *Causality: Models, Reasoning and Inference*. 2nd. USA: Cambridge University Press, 2009 (cit. on pp. 106, 112).

[PTK18] Emilija Perković, Johannes Textor, and Markus Kalisch. "Complete graphical characterization and construction of adjustment sets in Markov equivalence classes of ancestral graphs". In: *Journal of Machine Learning Research* 18 (2018), pp. 1–62 (cit. on pp. 106, 109, 110).

[PB14] Jonas Peters and Peter Bühlmann. "Identifiability of Gaussian structural equation models with equal error variances". In: *Biometrika* 101.1 (2014), pp. 219–228 (cit. on p. 60).

[PC20] Eugen Pircalabelu and Gerda Claeskens. "Community-based group graphical lasso". In: *J. Mach. Learn. Res.* 21 (2020), pp. 64–1 (cit. on p. 61).

[PM18] Scott D Pope and Ruslan Medzhitov. "Emerging principles of gene expression programs and their regulation". In: *Molecular cell* 71.3 (2018), pp. 389–397 (cit. on p. 82).

[RSZ06] Joseph Ramsey, Peter Spirtes, and Jiji Zhang. "Adjacency-faithfulness and conservative causal inference". In: *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*. 2006, pp. 401–408 (cit. on p. 109).

[RSW21] Alexander Gilbert Reisach, Christof Seiler, and Sebastian Weichwald. "Beware of the Simulated DAG! Causal Discovery Benchmarks May Be Easy to Game". In: *Advances in Neural Information Processing Systems*. Ed. by A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan. 2021 (cit. on p. 69).

[RS02] Thomas Richardson and Peter Spirtes. "Ancestral graph Markov models". In: *The Annals of Statistics* 30.4 (2002), pp. 962–1030 (cit. on p. 60).

[RH23] Philippe Rigollet and Jan-Christian Hütter. "High-dimensional statistics". In: *arXiv:2310.19244* (2023) (cit. on p. 160).

[RHB00] JM Robins, MA Hernán, and B Brumback. "Marginal structural models and causal inference in epidemiology." In: *Epidemiology* 11.5 (2000), pp. 550–560 (cit. on p. 60).

[Roj+18] Mateo Rojas-Carulla, Bernhard Schölkopf, Richard Turner, and Jonas Peters. "Invariant models for causal transfer learning". In: *Journal of Machine Learning Research* 19.36 (2018), pp. 1–34 (cit. on p. 30).

[Ros52] Murray Rosenblatt. "Remarks on a multivariate transformation". In: *The annals of mathematical statistics* 23.3 (1952), pp. 470–472 (cit. on p. 37).

[Rub76] Donald B Rubin. "Inference and missing data". In: *Biometrika* 63.3 (1976), pp. 581–592 (cit. on p. 77).

[RWC03] David Ruppert, Matt P Wand, and Raymond J Carroll. *Semiparametric Regression*. 12. Cambridge University Press, 2003 (cit. on p. 24).

[Sac+05] Karen Sachs, Omar Perez, Dana Pe'er, Douglas A Lauffenburger, and Garry P Nolan. "Causal protein-signaling networks derived from multiparameter single-cell data". In: *Science* 308.5721 (2005), pp. 523–529 (cit. on p. 71).

[Sae+20] Basil Saeed, Anastasiya Belyaeva, Yuhao Wang, and Caroline Uhler. "Anchored causal inference in the presence of measurement error". In: *Conference on uncertainty in artificial intelligence*. PMLR. 2020, pp. 619–628 (cit. on p. 92).

[SP10] Skipper Seabold and Josef Perktold. "statsmodels: econometric and statistical modeling with python". In: *9th Python in Science Conference*. 2010 (cit. on p. 151).

[SSA22] Abhin Shah, Karthikeyan Shanmugam, and Kartik Ahuja. "Finding valid adjustments under non-ignorability with minimal DAG knowledge". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2022, pp. 5538–5562 (cit. on p. 114).

[SSK23] Abhin Shah, Karthikeyan Shanmugam, and Murat Kocaoglu. "Front-door adjustment beyond Markov equivalence with limited graph knowledge". In: *Advances in Neural Information Processing Systems* (2023) (cit. on p. 114).

[Sha+15] Karthikeyan Shanmugam, Murat Kocaoglu, Alexandros G. Dimakis, and Sriram Vishwanath. "Learning causal graphs with small interventions". In: *Advances in Neural Information Processing Systems* 28 (2015) (cit. on p. 113).

[Shi+16] Nathaniel Shiers, Piotr Zwiernik, John AD Aston, and James Q Smith. "The correlation space of Gaussian latent tree models and model selection without fitting". In: *Biometrika* 103.3 (2016), pp. 531–545 (cit. on p. 61).

[SHH09] Shohei Shimizu, Patrik O Hoyer, and Aapo Hyvärinen. "Estimation of linear non-Gaussian acyclic models for latent factors". In: *Neurocomputing* 72.7-9 (2009), pp. 2024–2027 (cit. on p. 62).

[Shi+06] Shohei Shimizu, Patrik O Hoyer, Aapo Hyvärinen, Antti Kerminen, and Michael Jordan. "A linear non-Gaussian acyclic model for causal discovery". In: *Journal of Machine Learning Research* 7.10 (2006) (cit. on p. 60).

[SE17] Susan M Shortreed and Ashkan Ertefaie. "Outcome-adaptive lasso: variable selection for causal inference". In: *Biometrics* 73.4 (2017), pp. 1111–1122 (cit. on p. 115).

[SP08] Ilya Shpitser and Judea Pearl. "Complete identification methods for the causal hierarchy". In: *Journal of Machine Learning Research* 9 (2008), pp. 1941–1979 (cit. on p. 79).

[SVR10] Ilya Shpitser, Tyler VanderWeele, and James M Robins. "On the validity of covariate adjustment for estimating causal effects". In: *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*. 2010, pp. 527–536 (cit. on p. 110).

[Sil+06] Ricardo Silva, Richard Scheines, Clark Glymour, Peter Spirtes, and David Maxwell Chickering. "Learning the structure of linear latent variable models". In: *Journal of Machine Learning Research* 7.2 (2006) (cit. on pp. 60, 62, 92).

[SWU21] Liam Solus, Yuhao Wang, and Caroline Uhler. "Consistency guarantees for greedy permutation-based causal inference algorithms". In: *Biometrika* 108.4 (2021), pp. 795–814 (cit. on pp. 60, 63, 113).

[Spi01] Peter Spirtes. "An anytime algorithm for causal inference". In: *International Workshop on Artificial Intelligence and Statistics*. PMLR. 2001, pp. 278–285 (cit. on p. 61).

[Spi+00] Peter Spirtes, Clark N Glymour, Richard Scheines, and David Heckerman. *Causation, Prediction, and Search*. MIT Press, 2000 (cit. on pp. 61, 67, 107, 109, 113, 129).

[Spi13] Peter L Spirtes. "Calculation of entailed rank constraints in partially non-linear and cyclic models". In: *arXiv:1309.7004* (2013) (cit. on pp. 60, 64).

[SBJ20] Aude Sportisse, Claire Boyer, and Julie Josses. "Estimation and imputation in probabilistic principal component analysis with missing not at random data". In: *Advances in Neural Information Processing Systems* 33 (2020) (cit. on p. 77).

[Squ+20] Chandler Squires, Sara Magliacane, Kristjan Greenewald, Dmitriy Katz, Murat Kocaoglu, and Karthikeyan Shanmugam. "Active structure learning of causal DAGs via directed clique trees". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 21500–21511 (cit. on p. 113).

[Squ+23] Chandler Squires, Anna Seigal, Salil S Bhate, and Caroline Uhler. "Linear causal disentanglement via interventions". In: *International Conference on Machine Learning*. PMLR. 2023, pp. 32540–32560 (cit. on p. 89).

[Squ+22a] Chandler Squires, Dennis Shen, Anish Agarwal, Devavrat Shah, and Caroline Uhler. "Causal imputation via synthetic interventions". In: *Conference on Causal Learning and Reasoning*. PMLR. 2022, pp. 688–711 (cit. on p. 73).

[SU22] Chandler Squires and Caroline Uhler. "Causal structure learning: a combinatorial perspective". In: *Foundations of Computational Mathematics* 23.5 (2022), pp. 1781–1815 (cit. on pp. 90, 91, 106).

[SWU20] Chandler Squires, Yuhao Wang, and Caroline Uhler. "Permutation-based causal structure learning with unknown intervention targets". In: *Conference on Uncertainty in Artificial Intelligence*. PMLR. 2020, pp. 1039–1048 (cit. on p. 91).

[Squ+22b] Chandler Squires, Annie Yun, Eshaan Nichani, Raj Agrawal, and Caroline Uhler. "Causal structure discovery between clusters of nodes induced by latent factors". In: *Conference on Causal Learning and Reasoning*. PMLR. 2022, pp. 669–687 (cit. on pp. 59, 145).

[SFG19] Ralph Stadhouders, Guillaume J Filion, and Thomas Graf. "Transcription factors and 3D genome conformation in cell-fate decisions". In: *Nature* 569.7756 (2019), pp. 345–354 (cit. on pp. 48, 60).

[Sta+20] Stefan G Stark, Joanna Ficek, Francesco Locatello, Ximena Bonilla, Stéphane Chevrier, Franziska Singer, Gunnar Rätsch, and Kjong-Van Lehmann. "SCIM: universal single-cell matching with unpaired feature sets". In: *Bioinformatics* (2020) (cit. on p. 93).

[SB12] Daniel J Stekhoven and Peter Bühlmann. "MissForest—non-parametric missing value imputation for mixed-type data". In: *Bioinformatics* 28.1 (2012), pp. 112–118 (cit. on pp. 52, 77).

[Ste00] Matthew Stephens. "Dealing with label switching in mixture models". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 62.4 (2000), pp. 795–809 (cit. on p. 39).

[Sto+20] Jonathan M Stokes, Kevin Yang, Kyle Swanson, Wengong Jin, Andres Cubillos-Ruiz, Nina M Donghia, Craig R MacNair, Shawn French, Lindsey A Carfrae, Zohar Bloom-Ackerman, et al. "A deep learning approach to antibiotic discovery". In: *Cell* 180.4 (2020), pp. 688–702 (cit. on p. 77).

[Sub+17] Aravind Subramanian, Rajiv Narayan, Steven M Corsello, David D Peck, Ted E Natoli, Xiaodong Lu, Joshua Gould, John F Davis, Andrew A Tubelli, Jacob K Asiedu, et al. "A next generation connectivity map: L1000 platform and the first 1,000,000 profiles". In: *Cell* 171.6 (2017), pp. 1437–1452 (cit. on pp. 74, 83).

[STD10] Seth Sullivant, Kelli Talaska, and Jan Draisma. "Trek separation for Gaussian graphical models". In: *The Annals of Statistics* 38.3 (2010), pp. 1665–1685 (cit. on p. 64).

[Sut19] Richard Sutton. "The bitter lesson". In: *Incomplete Ideas (blog)* 13.1 (2019), p. 38 (cit. on p. 31).

[TWS15] Kean Ming Tan, Daniela Witten, and Ali Shojaie. "The cluster graphical lasso for improved estimation of Gaussian graphical models". In: *Computational statistics & data analysis* 85 (2015), pp. 23–36 (cit. on p. 61).

[Tan+17] Meng How Tan, Qin Li, Raghuvaran Shanmugam, Robert Piskol, Jennefer Kohler, Amy N Young, Kaiwen Ivy Liu, Rui Zhang, Gokul Ramaswami, Kentaro Ariyoshi, et al. "Dynamic landscape and regulation of RNA editing in mammals". In: *Nature* 550.7675 (2017), pp. 249–254 (cit. on p. 76).

[TP02] Jin Tian and Judea Pearl. "A general identification condition for causal effects". In: *Eighteenth National Conference on Artificial intelligence*. 2002, pp. 567–573 (cit. on p. 113).

[TB97] Lloyd N Trefethen and David Bau III. *Numerical Linear Algebra*. Vol. 50. Siam, 1997 (cit. on p. 95).

[Tro+01] Olga Troyanskaya, Michael Cantor, Gavin Sherlock, Pat Brown, Trevor Hastie, Robert Tibshirani, David Botstein, and Russ B Altman. "Missing value estimation methods for DNA microarrays". In: *Bioinformatics* 17.6 (2001), pp. 520–525 (cit. on p. 76).

[Tsi06] Anastasios A Tsiatis. *Semiparametric Theory and Missing Data*. Vol. 4. Springer, 2006 (cit. on p. 24).

[UT19] Madeleine Udell and Alex Townsend. "Why are big data matrices approximately low rank?" In: *SIAM Journal on Mathematics of Data Science* 1.1 (2019), pp. 144–160 (cit. on p. 80).

[Uhl+13] Caroline Uhler, Garvesh Raskutti, Peter Bühlmann, and Bin Yu. "Geometry of the faithfulness assumption in causal inference". In: *The Annals of Statistics* (2013), pp. 436–463 (cit. on p. 65).

[US17] Caroline Uhler and G.V. Shivashankar. "Regulation of genome organization and gene expression by nuclear mechanotransduction". In: *Nature Reviews Molecular Cell Biology* 18 (2017), pp. 717–727 (cit. on pp. 60, 72).

[Urs+22] Oana Ursu, James T Neal, Emily Shea, Pratiksha I Thakore, Livnat Jerby-Arnon, Lan Nguyen, Danielle Dionne, Celeste Diaz, Julia Bauman, Mariam Mounir Mosaad, et al. "Massively parallel phenotyping of coding variants in cancer with Perturb-seq". In: *Nature Biotechnology* (2022), pp. 1–10 (cit. on pp. 102, 151).

[Val84] Leslie G Valiant. "A theory of the learnable". In: *Communications of the ACM* 27.11 (1984), pp. 1134–1142 (cit. on p. 110).

[VO99] Stef Van Buuren and Karin Oudshoorn. *Flexible multivariate imputation by MICE*. Leiden: TNO, 1999 (cit. on p. 77).

[Vap98] Vladimir N Vapnik. "Statistical learning theory". In: (1998) (cit. on p. 24).

[Var+21] Burak Varici, Karthikeyan Shanmugam, Prasanna Sattigeri, and Ali Tajer. "Scalable intervention target estimation in linear models". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 1494–1505 (cit. on p. 104).

[VP90] Thomas Verma and Judea Pearl. "Equivalence and synthesis of causal models". In: *Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence*. 1990, pp. 255–270 (cit. on pp. 91, 112).

[VCB22] Matthew J Vowels, Necati Cihan Camgoz, and Richard Bowden. "D'ya like DAGs? A survey on structure learning and causal discovery". In: *ACM Computing Surveys* 55.4 (2022), pp. 1–36 (cit. on p. 113).

[Wal+13] Akbar K Waljee, Ashin Mukherjee, Amit G Singal, Yiwei Zhang, Jeffrey Warren, Ulysses Balis, Jorge Marrero, Ji Zhu, and Peter DR Higgins. "Comparison of imputation methods for missing laboratory data in medicine". In: *BMJ open* 3.8 (2013) (cit. on p. 76).

[WS20] Yuhao Wang and Rajen D Shah. "Debiased inverse propensity score weighting for estimation of average treatment effects with high-dimensional confounders". In: *arXiv:2011.08661* (2020) (cit. on p. 115).

[Wan+17] Yuhao Wang, Liam Solus, Karren D. Yang, and Caroline Uhler. "Permutation-based causal inference algorithms with interventions". In: *Neural Information Processing Systems*. Vol. 31. 2017 (cit. on pp. 71, 113).

[Wis28] John Wishart. "Sampling errors in the theory of two factors". In: *British Journal of Psychology* 19.2 (1928), p. 180 (cit. on p. 68).

[Xie+20] Feng Xie, Ruichu Cai, Biwei Huang, Clark Glymour, Zhifeng Hao, and Kun Zhang. "Generalized independent noise condition for estimating latent variable causal graphs". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 14891–14902 (cit. on pp. 60, 62, 72, 92).

[Xie+22] Feng Xie, Biwei Huang, Zhengming Chen, Yangbo He, Zhi Geng, and Kun Zhang. "Identification of linear non-Gaussian latent hierarchical structure". In: *International Conference on Machine Learning*. PMLR. 2022, pp. 24370–24387 (cit. on p. 92).

[Yad+22] Steve Yadlowsky, Hongseok Namkoong, Sanjay Basu, John Duchi, and Lu Tian. "Bounds on the conditional and average treatment effect with unobserved confounding factors". In: *The Annals of Statistics* 50.5 (2022), pp. 2587–2615 (cit. on p. 115).

[YKU18] Karren Yang, Abigail Katcoff, and Caroline Uhler. "Characterizing and learning equivalence classes of causal DAGs under interventions". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 5541–5550 (cit. on pp. 77, 91).

[Yan+21a] Karren Dai Yang, Anastasiya Belyaeva, Saradha Venkatachalapathy, Karthik Damodaran, Abigail Katcoff, Adityanarayanan Radhakrishnan, GV Shivashankar, and Caroline Uhler. "Multi-domain translation between single-cell imaging and sequencing data using autoencoders". In: *Nature Communications* 12.1 (2021), pp. 1–10 (cit. on p. 77).

[Yan+21b] Mengyue Yang, Furui Liu, Zhitang Chen, Xinwei Shen, Jianye Hao, and Jun Wang. "CausalVAE: disentangled representation learning via neural structural causal models". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 9593–9602 (cit. on p. 90).

[ZLT14] Benito van der Zander, Maciej Liśkiewicz, and Johannes Textor. "Constructing separators and adjustment sets in ancestral graphs". In: *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*. UAI '14. 2014 (cit. on p. 110).

[Zeč+23] Matej Zečević, Moritz Willig, Devendra Singh Dhami, and Kristian Kersting. "Causal parrots: Large language models may talk causality but are not causal". In: *arXiv preprint arXiv:2308.13067* (2023) (cit. on p. 31).

[Zen+24] Zhenghao Zeng, Sivaraman Balakrishnan, Yanjun Han, and Edward H Kennedy. "Causal inference with high-dimensional discrete covariates". In: *arXiv:2405.00118* (2024) (cit. on p. 122).

[Zha13] Jiji Zhang. "A comparison of three occam's razors for markovian causal models". In: *The British journal for the philosophy of science* (2013) (cit. on p. 39).

[Zha+11] Kun Zhang, Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. "Kernel-based conditional independence test and application in causal discovery". In: *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*. 2011, pp. 804–813 (cit. on p. 72).

[Zhe+18] Xun Zheng, Bryon Aragam, Pradeep K Ravikumar, and Eric P Xing. "DAGs with NO TEARS: Continuous optimization for structure learning". In: *Advances in Neural Information Processing Systems* 31 (2018) (cit. on p. 114).

[Zim+21] Roland S Zimmermann, Yash Sharma, Steffen Schneider, Matthias Bethge, and Wieland Brendel. "Contrastive learning inverts the data generating process". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 12979–12990 (cit. on p. 91).