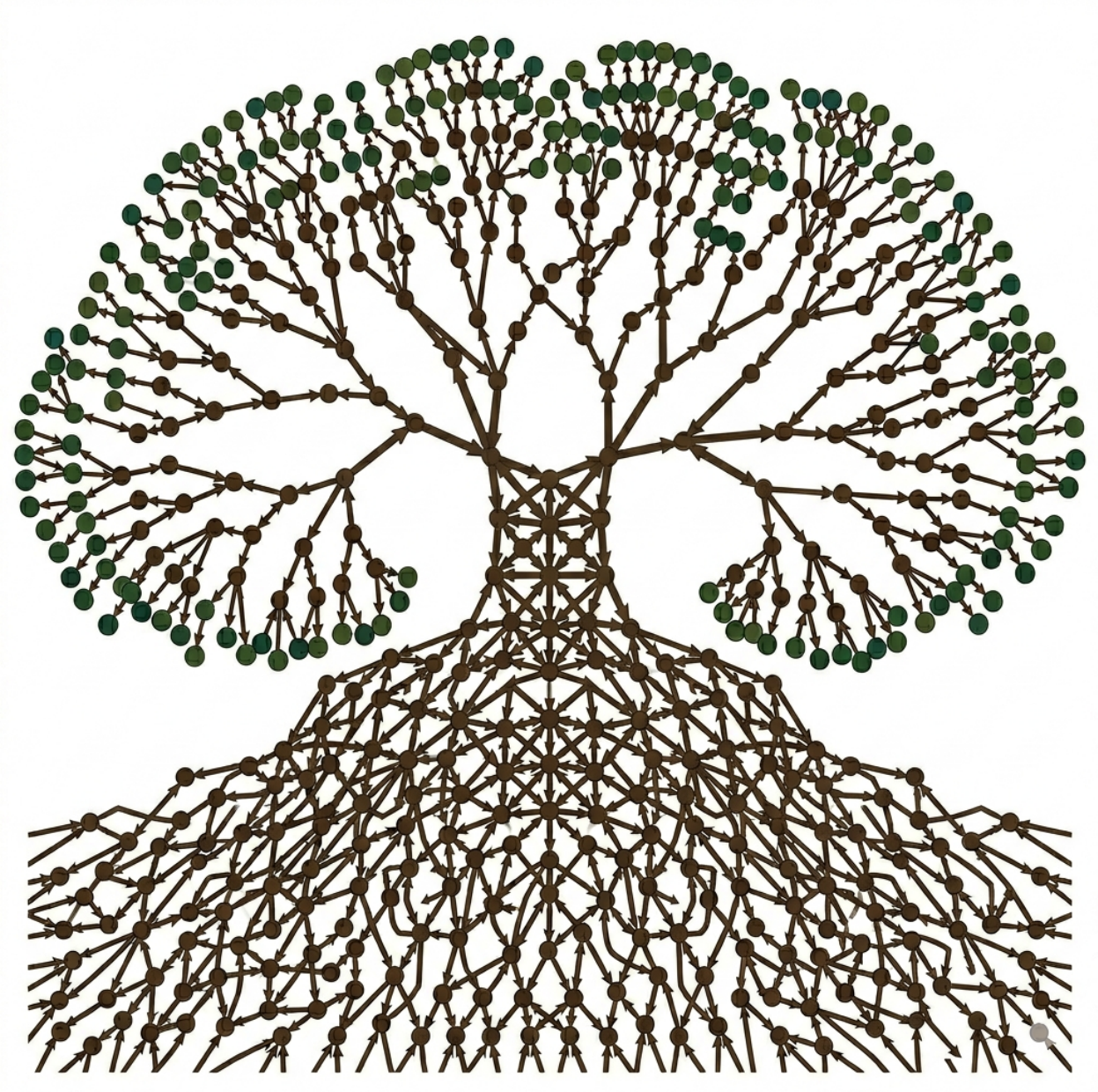
Principles of Causality

Bijan Mazaheri



Contents

Introduction

Helmets increase head injuries. Faster drivers arrive later. Hospital patients are less likely to have cancer when their bones are broken. But don't throw out your helmet, break your leg, or invest in transportation inefficiency just yet... Causality is the foundation of science and policy, but causal relationships can be obscured by a labyrinth of correlations.

This book establishes the principles that scientists and engineers can use to answer causal questions. Two principles carry most of the subject: exchangeability, which tells us when groups can be fairly compared, and the modification of data-generating processes, which tells us what an intervention actually does. Nearly every tool in this book, from randomized trials to do-calculus to synthetic controls, is an application of one or both.

From this study, a third principle emerges: causality is, at its core, about disentangling signals — causal from spurious, direct from indirect, and anomalies from different root causes. It is that last step, causal discovery, that ties the subject most directly to the frontier of modern AI.

Overview

This book will use basic probability, linear algebra, graph theory, and ordinary least squares (OLS), which you should review in Appendix A. The class is divided into four parts, with an optional prologue.

The History of Causality (Optional) The prologue covers how causality went from a matter for philosophers to a formal science, honored by a Nobel Prize, a Turing Award, and a Rousseeuw Prize.

Foundations of Causality This chapter builds intuition for the subject using paradoxes, causal diagrams, and potential outcomes. We establish the principle of exchangeability that makes randomized experiments work, and that the rest of the book learns to emulate. This is philosophical in flavor.

Structural Causal Models We then study how correlation propagates through data-generating processes. The unifying tool is a structural causal model, on which we build d-separation, which we use to study the do-calculus and potential outcome calculus as unifying theories for isolating causal signals. This portion is logical in flavor, like a discrete-math course.

Estimating Causal Effects With the first two principles of causality in place, we then turn to the techniques developed by statisticians, econometricians, and modern machine learning engineers to actually estimate causal quantities from noisy data. We will study outcome regression, inverse-propensity weighting, doubly robust methods, synthetic controls, and instrumental variables.

Causal Discovery Finally, we will study where graphical models come from in the first place. We will study a few famous approaches, learning the causal structure and data-generating process from the data itself. Throughout this study, we will uncover a third, more forward-looking principle about the disentanglement of signals. Though much of the content in this section was developed in philosophy departments, the flavor of this section is far closer to computer science.

Acknowledgments

This book is built on the foundation of a course taught by Professor Rohit Bhattacharya at Williams College, and I am grateful for his generosity in sharing the material. I also thank the students of ENGS 105.1 at Dartmouth, in both 2025 and 2026, who worked through early versions of these notes and helped iron out the mistakes. Any that remain are my own.

Chapter 1

The History and Philosophy of Causality

When starting a new subject, it is sometimes insightful to also understand its history, which can help reveal what motivated the various results we will learn about. This chapter briefly reviews this history for the study of causality.

For a long time, causality was only the subject of philosophy. Over the past 100 years, the field has rapidly formalized for economics and biostatistics. Over the past 20 years, computer scientists have also begun to formalize causality within computational and artificial intelligence. A Nobel Prize (economics), a Turing Award (computer science), and a Rousseeuw Prize (statistics) have all been awarded to researchers working on causality. The timeline in Table 1.1 traces the full arc, from Aristotle to the present day.

Three questions recur throughout this history, and the formal theory in the rest of this book is, in a sense, an extended answer to them:

1. What is a cause?
2. Can we learn causality?
3. How do we learn causality?

The first belongs to philosophy; the last will carry us into statistics, economics, and computer science.

1.1 What is a cause?

The question is ancient, and the earliest systematic answer is also the broadest.

Ancient and Classical Roots

> We do not have knowledge of a thing until we have grasped its why, that is to say, its cause. - Aristotle, 384-322 B.C.E.

In *Metaphysics*, Aristotle proposed four types of answers to the question "why?":

- *Material Cause*: What something is made of. E.g., the board floats because it is wood.
- *Formal Cause*: Its structure. E.g., the ratio $2:1$ causes an octave.
- *Efficient Cause*: The agent or process that brings it about. E.g., the ball fell because I dropped it.
- *Final Cause*: Its purpose or goal. E.g., the cause of a seed is the plant.

Medieval Contributions Building on Aristotle, Thomas Aquinas integrated the Aristotelian view of causality into Christian theology. He argued for a chain of causes leading back to a first cause (God), which became a key element in later discussions of God's existence and the nature of the universe.

TABLE 1.1 Timeline

384-322 BCE	Aristotle's *Metaphysics*
1265-1274	Thomas Aquinas, *Summa Theologiae*
1641	René Descartes, *Meditations on First Philosophy*
1739	David Hume, *A Treatise of Human Nature*
1747	James Lind performs the first clinical controlled trial
1781	Immanuel Kant, *Critique of Pure Reason*
1843	John Stuart Mill, *A System of Logic*
1923	Jerzy Neyman introduces potential outcomes for randomization
1924-1934	Sir Ronald Fisher formalizes randomized control trials
1959	Karl Popper formalizes experimentation and hypothesis falsifiability
1973	David Lewis publishes *Counterfactuals*
1974	Donald Rubin extends the potential outcomes framework outside of RCTs
1985	Judea Pearl begins to adapt his work on Bayesian networks to causality
1986	Jamie Robins introduces the first non-parametric structural equation model, referred to it as a finest causally interpreted structural tree graph (FCISTG)
1990	Peter Spirtes and Clark Glymour introduce the PC algorithm for learning causal networks (causal discovery)
1992	Jamie Robins introduces g-estimation
2012	Judea Pearl wins the Turing Award
2013	Thomas Richardson and Jamie Robins use Single World Intervention Graphs (SWIGs) to show that Rubin-Neyman's potential outcome framework and Pearl's graphical model framework are the same
2021	Joshua Angrist, David Card, and Guido Imbens win the Nobel Prize in Economics
2022	James Robins, Miguel Hernán, Thomas Richardson, Andrea Rotnitzky, and Eric Tchetgen Tchetgen win the Rousseeuw Prize in Statistics
2026	You're taking this class on causality!

Early Modern Roots Early modern philosophers began thinking about the nature of causation (as an action rather than a thing). René Descartes's dualism argues that ideas and the mind (or the "soul") exist separately from the physical world. The "causal adequacy principle" was motivated by a famous Roman saying from Lucretius: "Creatio ex Materia / Nothing comes from nothing." This principle is seen in many things—to heat something to 100 degrees, we need something at least 100 degrees. Descartes extended this idea to concepts: we cannot conceptualize infinity from only finite things.

He argued that the cause of something must be at least as "real" as the effect. This calls into question Aristotle's "formal cause" and "final cause," in which concepts and ideas are considered to cause real things. For Descartes, causation always flows from the physical to the conceptual and not the other way around. This is in alignment with the physics of the era, where the world was viewed as one big matter-moving system and causes were collisions of matter. It is also the natural precursor to the structural causal model, which will likewise treat the world as a data-generating process.

1.2 Can we learn causality?

Granting some notion of cause, a harder question follows: can causal facts ever be learned from observation alone?

Hume's Regularity Theory The dialogue on causal identifiability began with David Hume, who attributed causality to the regularity of co-occurring events.

> We may define a cause to be an object, followed by another, and where all objects similar to the first, are followed by objects similar to the second. - David Hume, *An Enquiry Concerning Human Understanding*, 1748

Problems with Regularity Theory There are a few notable problems with regularity theory:

1. **Imperfect Regularities**: Gambling doesn't always cause you to lose money, but it does cause you to lose money on average.
2. **Irrelevance**: Athletes' pre-game rituals always occur before games, but likely have no causal effect.
3. **Spurious Correlations**: Two events may regularly co-occur simply because they share a common cause, not because one causes the other.

Hume's Skepticism These problems led Hume (and many other philosophers) to conclude that "induction" is impossible. While it is easy to see how we can deduce things by applying a rule to a specific setting, it is not clear how we can go from many instances to a universal rule.

In *A Treatise of Human Nature*, Hume further expressed skepticism about whether causality (a form of induction) is a real thing at all. He suggested that causality is a habit of thought based on repeated associations of events, not an objective feature of the world.

Kant's Response

> "Give me matter, and I will construct a world out of it!" - Immanuel Kant

Immanuel Kant responded to Hume's skepticism by arguing that causality is a necessary condition for human experience. In his *Critique of Pure Reason*, he claimed that causality is not something we observe in the world but rather a fundamental structure of our minds, shaping how we perceive events. For Kant, causality is an *a priori* concept that allows us to make sense of the world. As we will see, foundational causal assumptions will in fact help us learn about causality, once we decide what it is.

Falsifiability

> "In so far as a scientific statement speaks about reality, it must be falsifiable: and in so far as it is not falsifiable, it does not speak about reality." - Karl R. Popper, *The Logic of Scientific Discovery*

Kant's strategy of starting with a model and refining it against experience is useless if the model cannot be proven wrong in the first place. In the 20th century, Karl Popper made this requirement precise under the name of falsifiability.

Probability Probability turns out to be remarkably helpful for induction. It handles imperfect regularities by letting a cause *increase the odds* of an effect rather than guarantee it. And Kant's idea of positing a model and backing it with data is precisely the spirit of Bayesian statistics. If you have taken a course on machine learning, you may have studied generalization bounds and regularization through VC-dimension; machine learning is itself a form of induction, and regularization is one way to quantify falsifiability.

1.3 How do we learn causality?

If the previous section asks whether causal knowledge is possible, this one asks how it is actually obtained. Two ideas organize nearly every method in this book. The first is to run an experiment and, when an experiment is impossible, to emulate one from observational data. The second, pursued largely on its own track, is to learn the causal structure itself from data. What follows traces who developed these ideas, and in which field.

Methods using Regularity Theory The earliest methods stayed close to Hume. John Stuart Mill took an empirical approach in *A System of Logic*, turning regularity into procedure: he outlined the method of agreement, the method of difference, and the method of concomitant variations, each a recipe for spotting causes in patterns of co-occurrence.

Biology and Experimentation James Lind is credited with performing the first clinical trial in 1747, randomizing six potential treatments for scurvy: cider, "elixir of vitriol" (dilute sulfuric acid), vinegar, seawater, a spice paste (made of garlic, mustard seed, horseradish, balsam of Peru, and gum myrrh), and oranges and lemons. Oranges and lemons were the winners.

Jerzy Neyman formalized the concept of a "potential outcome" for randomized trials in his master's thesis in 1923. Sir Ronald Fisher further formalized the role of randomization in experimentation [Fisher, 1934], which has since become the Randomized Controlled Trial, or RCT.

Formalizing Counterfactuals In 1973, David Lewis's *Counterfactuals* moved philosophy from regularity and falsifiability back toward counterfactuals. In the same year, Donald Rubin extended potential outcomes to more general settings that did not require an RCT [Rubin, 1974].

Epidemiology Other methods were developed to "simulate" an RCT from observational data, including propensity score matching. Jamie Robins pioneered many of these, including the g-methods, and, with Miguel Hernán, Thomas Richardson, Andrea Rotnitzky, and Eric Tchetgen Tchetgen, built much of the modern theory of causal inference for medicine and public health.

Econometrics Instrumental variables, synthetic controls, and natural experiments grew out of the effort to measure the causal effects of economic policy, in work by Joshua Angrist, David Card, Guido Imbens, and others.

Computer Science Judea Pearl and his collaborators formalized causal analysis using structural causal models, networks that encode "what causes what." Pearl's causal hierarchy distinguishes association from the outcomes of actions (seeing vs. doing), and actions from counterfactuals (doing vs. imagining or blaming). Considerable work has gone into understanding when and how the outcome of an action can be computed from data without running an experiment.

A separate line of work, known as causal discovery, asks how to learn these structural models in the first place. Many of its key algorithms came out of Carnegie Mellon's philosophy department, so computer science does not get all the credit here. Causal discovery has since seen success in domains such as learning gene regulatory networks in biology.

Machine Learning and Data Science Causal inference is now of growing interest in the machine learning community, both for classical problems such as treatment-effect estimation and for newer ones, like designing large batteries of experiments toward a scientific goal or finding the root cause of a system failure. A particularly active thread concerns learning latent causal concepts and disentangling causal relationships, under names like causal representation learning, causal disentanglement, and latent variable models.

Modern Philosophy More recently, Nancy Cartwright has developed more nuanced views that recognize the full complexity of causal relationships, though we will not pursue them here.

> **Main Idea 1**
>
> The definition of causality has changed over the years and is a subject of ongoing debate. As we mathematically formalize causality throughout this course, we need to be specific about what definition of causality we are addressing.

Chapter 2

Foundations of Causality

A university is sued for admitting men at a higher rate than women, yet within every single department it admits women at a higher rate than men. A safety device turns out to be associated with more injuries, not fewer. These are not data-entry errors; they are correlations deviating from causal intuition. This chapter lays the foundation for distinguishing the two. We start with paradoxes like these and the causal diagrams that dissolve them, then introduce potential-outcome random variables, which let us write down the thing we actually want — what would have happened to the same unit under a different treatment — as a precise mathematical object. With that language, we can describe the randomized controlled trial, the gold standard that makes causal questions answerable, and isolate the properties that give it its power: exchangeability and its conditional cousin. These concepts drive the rest of the book: once we know exactly what a randomized experiment buys us, we can look for the same things in observational settings without designed experiments. This chapter leans a little philosophical, but everything that follows rests on it.

2.1 Simpson's Paradox

Simpson's Paradox occurs when a trend that appears in different groups of data reverses or disappears when the groups are combined.

2.1.1 Case Study #1: Berkeley Admissions

In 1973, UC Berkeley was sued for sex discrimination [Bickel et al., 1975]. Aggregate data showed that 44% of male students were admitted, compared to only 35% of female students. However, breaking the data up by department revealed something unexpected: the vast majority of departments were admitting women at higher rates than men! The ones that were admitting more men were only barely out-admitting the women. How could this be?

To get some intuition for how this could be, take a look at Table 2.1, which provides a simplified illustration of this effect. The paradox arises because women were primarily applying to the more selective departments, whereas men were applying to different, less selective departments. This raises a crucial question: should we use per-department acceptance rates or overall acceptance rates when evaluating fairness? The answer is not obvious; perhaps women are being discouraged from applying to certain departments.

Department	Men	Women
Not-Selective	100 applied, 90% accepted	10 applied, 100% accepted
Selective	10 applied, 10% accepted	100 applied, 20% accepted

Table 2.1: A simplified specification of counts for the Berkeley admissions data demonstrating Simpson's Paradox.

2.1.2 Case Study #2: Cholesterol and Mortality

In standard machine learning, there is a pervasive narrative that "more data is always better." Causal inference challenges this assumption.

Consider a global health study looking at the relationship between cholesterol levels and mortality rates. If a hospital in a wealthy country analyzes its patients, it will find a positive association: higher cholesterol leads to higher mortality due to heart disease.

Suppose the researchers want to train a more robust model, so they aggregate data from dozens of other countries, including developing nations. Surprisingly, in the new massive global dataset, the association reverses: cholesterol is now *negatively* associated with death! How did adding more data make the model conclude that cholesterol saves lives?

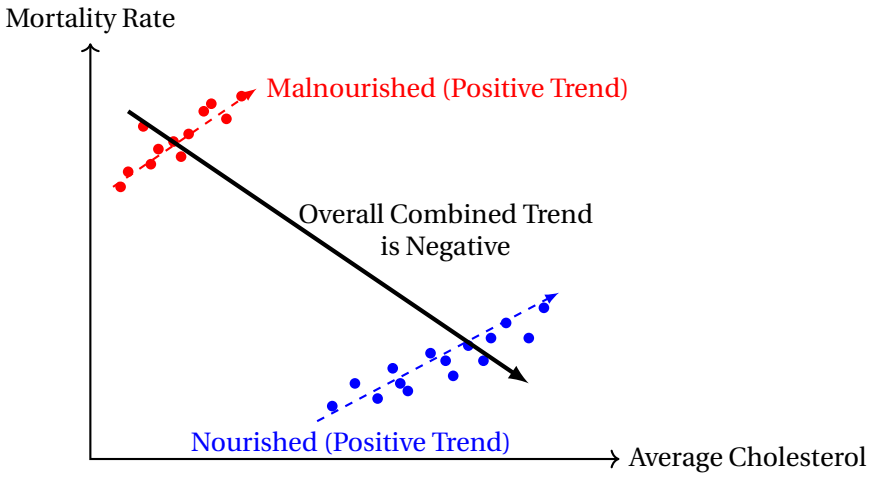


Figure 2.1: A scatter plot of Cholesterol vs. Mortality illustrating Simpson's Paradox. Both sub-groups (malnourished and nourished countries) show positive correlations. However, because malnutrition causes both low cholesterol and high mortality, the aggregated global trend appears negative.

As illustrated in Figure 2.1, adding data from more countries introduced a powerful new variable: malnutrition. In developing nations, extremely low average cholesterol is a biomarker for starvation and severe malnutrition, which inherently carries a larger mortality rate.

If we simply look at the aggregated data, our cholesterol measurement now contains hidden information about a patient's access to food. The model learns that low cholesterol means starvation, and therefore predicts higher mortality. By blindly adding more data without understanding the underlying causal structure, we introduced a paradox.

In the medical literature, this phenomenon—where higher cholesterol appears protective against mortality in certain populations—is known as the **lipid paradox** [Morris et al., 2021]. While our global example highlighted malnutrition, this paradox also frequently appears in clinical datasets when researchers aggregate generally healthy patients with those suffering from severe chronic illnesses or high systemic inflammation. Because advanced disease and inflammation simultaneously suppress cholesterol and drastically increase mortality risk, the aggregated data creates a spurious negative trend.

A nearly identical statistical illusion occurs in alcohol consumption studies, which famously produced a "J-shaped" mortality curve [Shaper et al., 1988]. For years, observational data suggested that moderate drinking was healthier than complete abstention. However, this was driven by an unobserved confounder: the "sick-quitter" effect. Many teetotalers abstain from alcohol precisely because they already have severe pre-existing health conditions. When these sick abstainers were aggregated with generally healthy individuals, it falsely made zero alcohol consumption appear dangerous.

Main Idea 2

Simpson's Paradox can form from latent subgroups in continuous data. Simply acquiring "more data" can actively harm your analysis if it introduces unobserved confounders.

2.1.3 Case Study #3: CO_2 and Climate

The Northern Hemisphere has most of the land and plant-life on our planet. Plants absorb CO_2 when they are active (during the summer) and release CO_2 when dormant (during the winter). Consequently, when the Northern Hemisphere is in winter, global temperatures drop and plants go dormant, causing CO_2 levels to rise. Conversely, in the summer, global temperatures rise and plants become active, causing CO_2 to fall.

This dynamic creates an out-of-phase oscillation over time, as shown in the left plot of Figure 2.2. Over decades, both background temperature and CO_2 increase, but within any single year, they move in opposite directions.

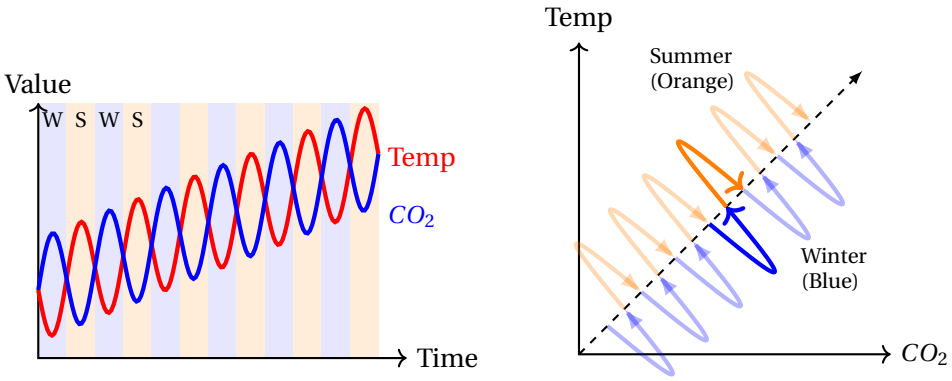


Figure 2.2: Left: Time series of Temperature and CO_2 oscillating out of phase. Right: The parametric plot of Temp vs CO_2. The highlighted arrows show a single year forming a negative "S" trend (moving top-left to bottom-right), while the overall multi-year trend moves bottom-left to top-right.

If we remove the time axis and simply plot Temperature against CO_2, we see a "snaking" trajectory. By coloring the seasons (blue for winter, orange for summer), we map the time series onto the parametric space. In the short term (within a single seasonal loop), the association between temperature and CO_2 is negative. However, over the course of many cycles, the overall long-term association is unmistakably positive [Keeling et al., 1976].

Main Idea 3

Simpson's Paradox is not just driven by sub-groups, but can also be an artifact of continuous latents (like dynamics over time).

2.2 Causal Diagrams

As we saw in the previous section, looking at data alone cannot tell us whether to trust the aggregated or disaggregated trend. To resolve these paradoxes and answer causal questions, we must understand the underlying data-generating process. We do this by drawing causal models.

2.2.1 The Asymmetry of Causation

Consider a simple two-node causal system representing the weather and pedestrian behavior: Raining → Umbrella. Statistically, these two variables are highly correlated. However, causality is fundamentally

asymmetric. If we intervene and change the weather (e.g., by seeding clouds to make it rain), umbrella usage will change. But if we intervene on the umbrella (e.g., by legally banning all umbrellas), the rain will not change! The causal direction informs us exactly how the joint probability distribution will shift when we intervene on the system.

2.2.2 Confounding: When to Condition

We can use causal diagrams to formalize the paradoxes we saw with both the global health data and the climate data. In both cases, the paradox is driven by a shared structural flaw: an unobserved common cause acting as a confounder. As shown in Figure 2.3, both systems share the exact same causal topology.

(a) Global Health Paradox (b) Climate Paradox

Figure 2.3: Causal DAGs revealing the identical structure behind two different manifestations of Simpson's Paradox. In (a), malnutrition acts as a confounder. In (b), the season acts as a temporal confounder.

The core causal questions here are: *What would have happened to mortality if lipid levels had been different?* and *What would have happened to the temperature if CO_2 had been different?*

Because of the causal direction, if we intervene to artificially lower CO_2 levels, there will be no shift in the Season. Similarly, intervening on a patient's Cholesterol will not change their underlying Malnutrition. Because these common causes remain fixed regardless of our intervention, we need to evaluate the causal effect by looking at the data "per season" and "per nutrition level." In systems with this structure, conditioning is good! By holding the confounder constant, we block the spurious correlation and isolate the true effect.

The Mathematics of Confounding

To mathematically prove how this paradox occurs, we can calculate the observational slope under a linear Structural Causal Model (SCM). Assume a standardized system where all variables have a variance of 1:

$$\begin{aligned} X &= N_X \\ A &= \alpha X + N_A \\ Y &= \beta A + \gamma X + N_Y \end{aligned}$$

The observational regression slope (predicting Y using A) is defined as $\hat{\beta}_{obs} = \frac{\text{Cov}(A,Y)}{\text{Var}(A)}$. Since we assume $\text{Var}(A) = 1$, this simplifies directly to the covariance:

$$\begin{aligned} \hat{\beta}_{obs} &= \text{Cov}(A, \beta A + \gamma X + N_Y) \\ &= \beta \text{Cov}(A, A) + \gamma \text{Cov}(A, X) + \text{Cov}(A, N_Y) \end{aligned}$$

Because A is independent of N_Y, the last term is 0. Since $\text{Cov}(A, A) = 1$, the first term is simply β. Evaluating $\text{Cov}(A, X)$ by substituting the structural equation for A yields exactly α. Therefore:

$$\hat{\beta}_{obs} = \beta + \gamma\alpha$$

This means that if we simply regress Y on A without adjusting for X, our naive best-fit prediction line will be:

$$\hat{Y} = (\beta + \gamma\alpha) A$$

The true causal effect is β, but the observational data yields an apparent slope of $\beta+\gamma\alpha$. This $\gamma\alpha$ term is the confounding bias. If the true causal effect is positive, but the bias is sufficiently negative, the observational trend will flip entirely, creating Simpson's Paradox.

2.2.3 Colliders: When NOT to Condition

Now let's examine a structure where conditioning goes wrong. Early in the pandemic, hospital records showed that smokers had significantly fewer instances of COVID-19. However, this data only looked at hospital records!

Figure 2.4: Causal DAG demonstrating collider bias. Admittance to the hospital is a collider. By only looking at hospital data, we condition on it (black node), creating a spurious negative association between Smoking and COVID-19.

As shown in Figure 2.4, Admittance to the hospital acts as a *collider*. Patients needed to be in the hospital for some reason; if you do not have COVID-19, it becomes more likely that there are some other reasons you were admitted, like smoking-related illnesses.

Unlike the confounding example, it is absolutely incorrect to condition on the hospital admittance. By only looking at admitted patients, the researchers are *creating* a spurious dependence between two previously independent variables. This is sometimes known as Berkson's paradox.

The Mathematics of Collider Bias

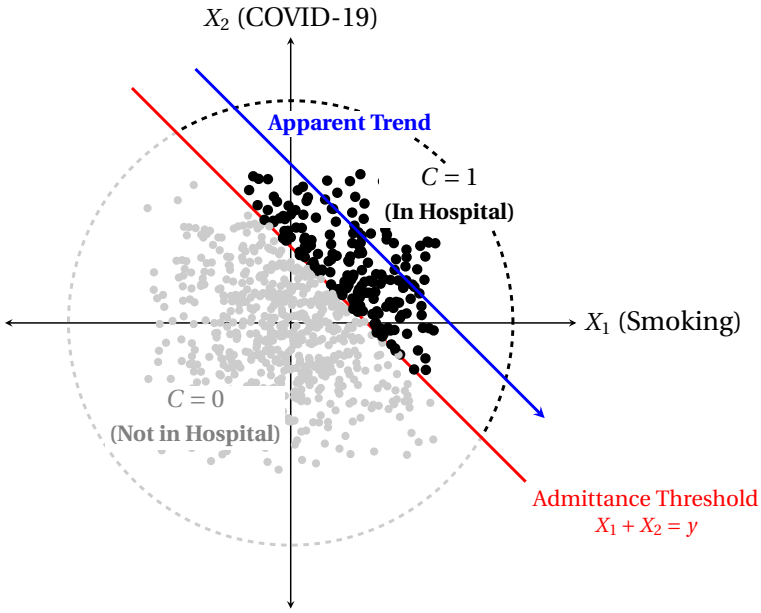


Figure 2.5: A geometric interpretation of collider bias. The red threshold line completely partitions the population. All data points above the threshold ($C = 1$) are observed, while the points below ($C = 0$) are hidden. Because the threshold chops off the distribution, it artificially constrains X_1 against X_2, creating a false negative correlation (blue line).

Collider bias occurs because conditioning on a common effect forces a spurious relationship between its causes. Imagine two completely independent traits, X_1 (Smoking) and X_2 (COVID-19), plotted on a scatter plot. Globally, they have absolutely no correlation.

However, suppose admittance to the hospital ($C = 1$) only happens if a patient's combined severity of symptoms crosses a certain threshold, such as $X_1 + X_2 > y$. As shown in Figure 2.5, conditioning on $C = 1$ effectively chops off the bottom-left of the distribution. Within this restricted admittance region, if X_1 is low, X_2 *must* be high to cross the threshold, and vice versa. This geometrically induces an artificial negative correlation between two independent variables!

Main Idea 4

1) Associations can come from selection biases.
2) How you collected data is just as important as the system itself!

2.2.4 Revisiting Berkeley Admissions

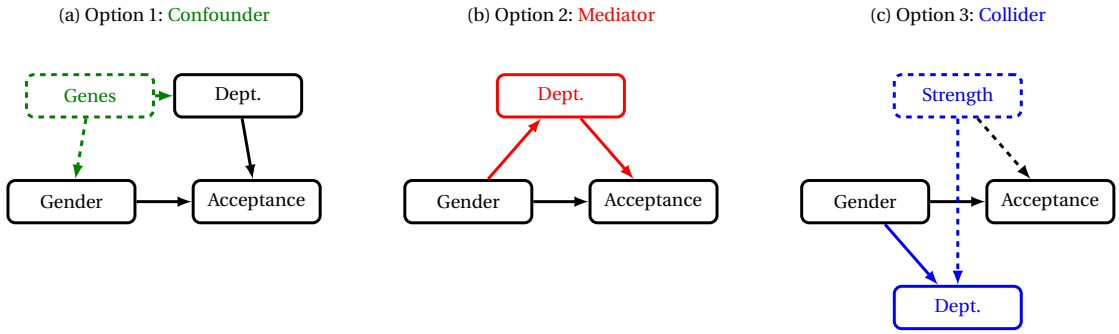


Figure 2.6: Three potential causal DAGs illustrating the Berkeley admissions paradox through the lens of a confounder (green structure), a mediator (red structure), and a collider (blue structure). The final model introduces a collider (bottom node), which creates a spurious association if we condition on the department.

Finally, let's return to the Berkeley admissions paradox. As we've just seen, our decision to aggregate data or condition on a variable depends entirely on whether that variable is a confounder or a collider. The problem is: what if the model is unknown?

The core causal question is: would the women have received a different decision if they had been men? To answer this, we must consider the causal model of the system, as shown in Figure 2.6. Different models yield entirely different counterfactual answers, demonstrating that conclusions are completely unclear without understanding the true data-generating process:

- **Option 1 (Confounder):** Suppose something like Genetics was a common cause, or *confounder*, for both Gender and the choice of Department. This probably means that the applicant would have still applied to the same departments in our counterfactual, since their interests are likely determined by other genetic factors. We should therefore hold the department constant. The dashed arrow in Figure 2.6 (a) indicates that the causal dependence flowing through Department is blocked. In this scenario, we might see a *lower* acceptance rate, as evidenced by the per-department differences.
- **Option 2 (Mediator):** What if the less-selective departments discouraged female applicants? In this case, gender acts as a *mediator* for the effect of gender on acceptance. In this case, if the female applicants had been men and hadn't been discouraged, then they would have had a *higher* acceptance rate. We still maintain a direct path from Gender to Acceptance here, as per-department rates for women are indeed higher. Still, this effect is overtaken, and the counterfactual answer corresponds to the aggregate difference: the women would have been accepted at a *higher* rate if they had been

men. Notice that mediators are yet another example of something you *should not* keep constant if you want to learn causal effects.

- **Option 3 (Collider):** What if an unobserved factor, like applicant strength, influences both the department applied to and the acceptance? In this model, the department acts as a *collider*. Looking at a specific department artificially skews the data because there must be some reason why the student applied. For example, a selective department that is considered a "male-dominated" field might have the full spectrum of men applying, but only the strongest and most determined women. In this case, we would be unsurprised by the higher acceptance rates in these departments. There is no real dependence between gender and capability or determination, but we artificially observe this dependence by looking at a specific department. For this reason, the correct approach would be to examine aggregate differences for fairness: the acceptance rates would have been *higher* if the women had been men.

2.2.5 Conclusion

Understanding causality is a critical component of decision-making in law, medicine, and science. In this class, we will learn how to formally think about causal systems, how to use them to answer causal questions, and how to discover what the systems are.

2.3 Counterfactuals and Pearl's Causal Ladder

You might have noticed that I was very careful to frame the "causal question" in the previous section to be about an aggregate group (e.g., "Would the women have been admitted at a higher rate if they had been men?"), rather than an individual (e.g., "Would I have gotten in if I had been a man?"). This is because it is fundamentally harder to answer deterministic questions about individuals than it is to answer questions about groups (which involve shifts in probability).

If we knew the exact, deterministic physical laws of the universe, then causality would be straightforward. We could ask a deterministic "what would have happened" question (a counterfactual) and use equations to simulate the exact answer. However, in fields like medicine, economics, and the social sciences, we lack omniscient knowledge of every microscopic variable.

To handle this uncertainty, we rely on probability, but standard probability only describes the world as it currently is. Probability and association cannot tell us what *would* happen if we changed the world, or what *would have* happened under different circumstances. This gives rise to two conceptual leaps:

1. Moving from probability to interventional changes in probability.
2. Moving from interventional changes in probability to deterministic counterfactuals.

We've already discussed the first of these leaps in detail using Simpson's paradox. To build intuition for the second leap, let's start with an example.

2.3.1 Effect Modification and Heterogeneous Responses

To see why moving from aggregate interventions to individual counterfactuals is so difficult, suppose we have developed a vaccine for a virus with 90% effectiveness. This means that the people who receive the vaccine are ten times less likely to develop COVID.

Let's imagine the population is split along two dimensions: genetic makeup (90% have Gene A, 10% have Gene B) and viral exposure (90% are exposed to the α variant, 10% to the δ variant). Consider two entirely different mechanical explanations for our 90% effectiveness rate:

1. **Model 1 (Genetic Response):** The vaccine's efficacy is entirely dependent on the host. Gene A individuals have an immune response that protects them completely from all strains. Gene B individuals lack this response and are completely unprotected.

2. **Model 2 (Viral Resistance):** The vaccine's efficacy is entirely dependent on the virus. All people respond to the vaccine, but it only neutralizes the dominant α variant. The δ variant completely bypasses the vaccine's protection.

Notice that the interventional effect of both scenarios is exactly the same—administering the vaccine to the population reduces the overall cases by a factor of 10. Despite this, the two explanations give rise to completely different counterfactual realities.

We can visualize this by scaling the rows and columns proportionally to the population splits. The area of each block represents the number of people in that subgroup. We will shade the regions green to indicate a healthy outcome (−) and orange to indicate a COVID-positive case (+).

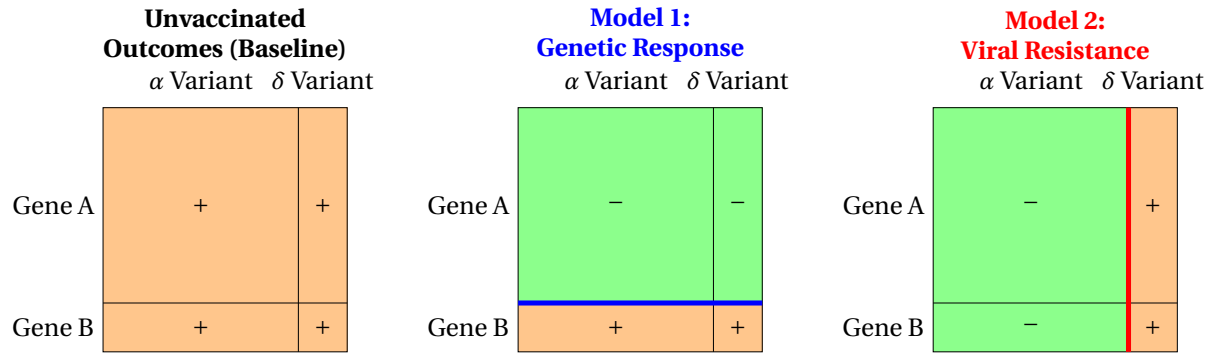


Figure 2.7: Comparison of individual responses to vaccination. The area of each box represents the population proportion. In Model 1, the outcome is determined entirely by the rows (indicated by the thick blue horizontal divider). In Model 2, the outcome is determined entirely by the columns (thick red vertical divider). Both yield 90% aggregate effectiveness (90% of the total area is green), but individual counterfactuals diverge drastically.

By comparing the areas in the plots, we can see exactly why the counterfactuals are different. Suppose an unvaccinated person with Gene A gets sick with the δ variant. In the unvaccinated baseline, they are in the top-right box (+).

If we want to know their counterfactual—what *would* have happened had they taken the vaccine—the answer depends entirely on the underlying model. In the first model, their counterfactual cell is green (−). They would not have gotten sick, and we can reasonably blame their illness on their refusal to be vaccinated. However, in the second model, their counterfactual cell is orange (+). They would have gotten sick from the resistant strain anyway, meaning the vaccine would have been useless for them. It is not really their "fault" that they got sick.

If we do not know the underlying mechanics connecting genetics, viral strains, and the immune system, we cannot distinguish between these two settings using only the 90% aggregate statistic.

> **Main Idea 5**
>
> There is no such thing as "the" causal effect of an action. Causal effects are specific to individuals and populations.

This example gets at the crux of the difference between interventional outcomes and counterfactuals. We can know the outcome of an intervention in aggregate (doing), but we cannot specify individual responses or counterfactuals (imagining) without a deeper model of the system's mechanics.

2.3.2 The Necessity of Structural Equations

When discussing Simpson's paradox, we showed the difference between seeing and doing by presenting two causal models with the same probability distribution but different interventional distributions. We

will now do the same for interventional distributions and (individual) counterfactual outcomes to show the need for "structural equations" that specify the functional dependencies among variables.

We will define two generating processes that use $X \to Y$ as the causal model between two Bernoulli random variables, and generate the same probability distribution.

$$\Pr(X=0) = \Pr(X=1) = \frac{1}{2}$$
$$\Pr(Y=0 \mid X=0) = \Pr(Y=1 \mid X=0) = \frac{1}{2}$$
$$\Pr(Y=1 \mid X=1) = \Pr(Y=2 \mid X=1) = \frac{1}{2}$$

That is, $X \sim U\{0,1\}$ (uniformly distributed between 0 and 1) and $Y \sim U\{X, X+1\}$.

Here is one model, using noise $N \sim U\{0,1\}$ as a Bernoulli random variable:

$$Y = X + N. \tag{2.1}$$

This is not the only model that gives us the conditional and interventional distributions for $X \to Y$. Here is another one:

$$Y = 2X + (1-2X)N. \tag{2.2}$$

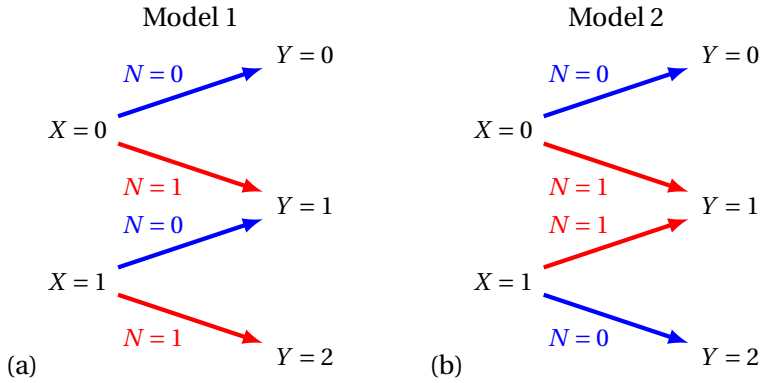


Figure 2.8: (a) Shows the mapping of inputs to outputs in the first model generated by Equation (2.1). (b) Shows the mapping of inputs to outputs in the second model generated by Equation (2.2). For both diagrams, blue arrows are used for $N=0$ and red arrows are used for $N=1$. A counterfactual corresponds to switching X while keeping the arrow color the same.

See Figure 2.8 for a diagram of how values of X map to values of Y. Now, let's consider $X=1, Y=1$. For the first model, changing X to 0 gives a counterfactual of $X=0, Y=0$ (N had to have been 0 and stays 0). However, for the second model, N had to have been 1. This means that changing X to 0 would lead to $X=0, Y=1$.

Exercise 1

If $X=1, Y=2$, then what would Y be if X had been 0?

Notice that it was the *equations* that let us compute these counterfactuals. Without them, the two models would look the same. We can also compute counterfactuals if we know the specific values of the noise variables, but knowing the equations is the fundamental requirement.

2.3.3 Parametric vs. Non-Parametric Models

In causal inference and statistics, we broadly categorize models into two types based on how much we assume about these structural equations:

- **Parametric Models:** A parametric model assumes a highly specific mathematical "form" for the relationships between variables, governed by a fixed, finite number of parameters.
  - *Example 1:* Linear structural equations with Gaussian additive noise (e.g., $Y = \beta_0 + \beta_1 X + N$).
  - *Example 2:* Logistic regression for binary outcomes.
  - *Example 3:* Generalized linear models like Poisson regression (used for modeling count data).
- **Non-Parametric Models:** Non-parametric models make no strict assumptions about the algebraic form of the equations or the distribution of the noise. They allow the data to dictate the shape of the relationship.
  - *Example 1:* Decision trees (which partition data rather than fitting a global algebraic equation).
  - *Example 2:* Bayesian networks over discrete variables (where we only specify probability tables, not algebraic equations).
  - *Example 3:* Completely arbitrary structural functions, often written generically as $Y = f(X, U)$, where f can be any unknown continuous or discontinuous mapping.

Main Idea 6

Interventions and experiments do not tell us individual counterfactuals unless we know the equations (often parametric) that govern the causal relationships.

It is worth noting that while the two math models from Figure 2.8 are different with respect to individual counterfactuals, they are the same with respect to the *average* counterfactual over a group of people. This is the crux of the difference between what Pearl distinguishes as "doing" and "imagining". Doing is the computation of an average response using a probability distribution and an assumed direction of data generation. Imagining is figuring out the counterfactual of an individual response.

If all we care about is the *average* treatment effect, then the role of the noise (or heterogeneous response) averages out over the group, becoming irrelevant. Therefore, we can think of the progression from observing data, to running interventions, to formulating structural equations as a process of continuous refinement toward the individual. We will eventually see that average treatment effects can be learned in purely non-parametric settings, even if individual counterfactuals cannot.

2.3.4 Pearl's Ladder of Causation

Judea Pearl formalized this exact progression into what he calls the "Ladder of Causation" [Pearl, 2009]. See Figure 2.9 for an illustration. This ladder divides queries into three distinct rungs, perfectly mirroring our progression from observing data, to asking about aggregate groups, to asking about individuals:

1. **Association (Seeing):** What does a survey tell us about the current world? This is standard probability, like $\Pr(Y \mid X)$. As Simpson's Paradox showed us, this rung is highly vulnerable to confounding.
2. **Intervention (Doing):** What *would* happen if we force a change in the world for a group of people? This is an interventional probability, denoted $\Pr(Y \mid \text{do}(X))$. Answering these questions requires a *causal diagram* to know what variables to hold constant.
3. **Counterfactuals (Imagining):** What *would have* happened to this specific individual if things had been different? Answering these questions requires knowing the underlying *structural equations* of the system.

Pearl's third level on the ladder of causality refers to individualized counterfactuals. Pearl just calls this rung "counterfactual," which can lead to some confusion in the potential outcome model we will study later. For this reason, we emphasize that this rung represents a counterfactual on an *individual*. Pearl considers this rung critical to human imagination, creativity, and our ability to take moral responsibility for our actions.

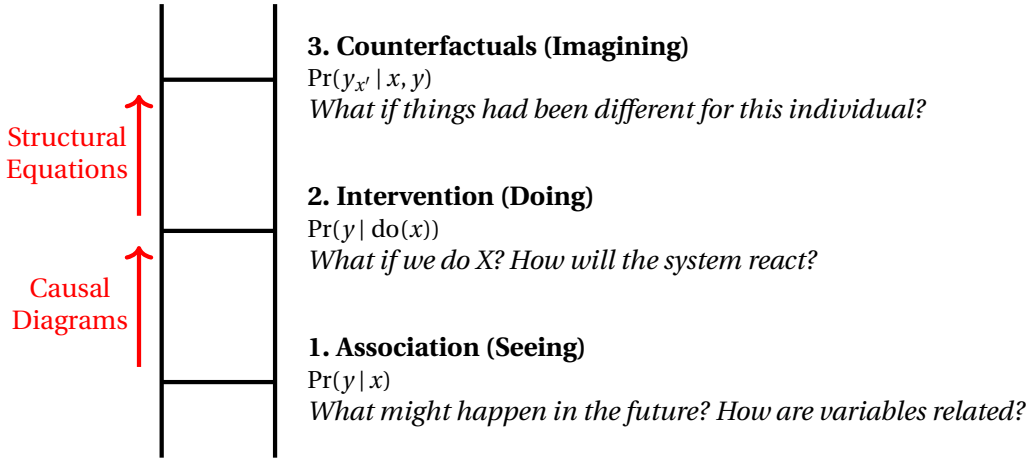


Figure 2.9: Pearl's Ladder of Causation. Standard statistics operates entirely on Rung 1. Moving to Rung 2 requires a causal diagram, and moving to Rung 3 requires structural equations.

2.4 Potential Outcomes and Randomized Controlled Trials

In the previous section, we established that answering individual counterfactual queries (Rung 3 of Pearl's ladder) requires a deep understanding of the system's structural equations. To formalize these "Imagining" queries mathematically, statisticians like Jerzy Neyman and Donald Rubin developed the *Potential Outcomes* framework [Imbens and Rubin, 2015].

2.4.1 Potential Outcomes and the Fundamental Problem

Suppose we want to understand the causal effect that receiving a medical treatment had on a specific patient's recovery. We will define two "potential outcomes" which represent the true outcome of a candidate if they were not treated ($Y^{(A=0)} = Y^{(0)}$) or treated ($Y^{(A=1)} = Y^{(1)}$). The individual causal effect (sometimes called the Individual Treatment Effect, or ITE, in epidemiology) is exactly the difference between these two parallel realities:

$$ITE = Y^{(1)} - Y^{(0)}. \tag{2.3}$$

Because the ITE requires us to know what *would have* happened to a specific person under an alternate scenario, it sits firmly on Rung 3 of Pearl's ladder. Of course, in practice, such a quantity cannot be directly computed. We either give a patient treatment or we do not—we cannot rewind time, change our course of action, and observe the alternate reality.

This dilemma is so absolute that it is known as "The Fundamental Problem of Causal Inference." The realized potential outcome is called the *factual*, and the unrealized one is called the *counterfactual*. Because we only ever get to see one of the two potential outcomes for any given individual, causal inference is sometimes interpreted as a "missing data problem." See Table 2.2 for an illustration.

2.4.2 Average Treatment Effects and Historical RCTs

Because Rung 3 is often inaccessible, we frequently step down to Rung 2 (Intervention / Doing). Instead of agonizing over the missing data for an *individual*, we can look at the expected value of the difference over a *population*. This aggregate metric is called the Average Treatment Effect (ATE):

$$ATE = \mathbb{E}[Y^{(1)} - Y^{(0)}]. \tag{2.4}$$

Linearity of Expectation allows us to distribute the expected value:

$$ATE = \mathbb{E}[Y^{(1)}] - \mathbb{E}[Y^{(0)}]. \tag{2.5}$$

Patient	Condition	Treatment (A)	$Y^{(0)}$	$Y^{(1)}$	ITE
웃_1	Mild	Not Treated	**1**	1	0
웃_2	Severe	Treated	0	**0**	0
웃_3	Severe	Treated	0	**1**	1
웃_4	Mild	Treated	1	**0**	-1
웃_5	Severe	Treated	0	**0**	0
웃_6	Mild	Not Treated	**1**	1	0
웃_7	Mild	Treated	0	**1**	1
웃_8	Severe	Not Treated	**0**	1	1

Table 2.2: A table showing factuals in bold and counterfactuals in gray. Causal inference is sometimes thought of as a missing data problem, where successfully filling in the gray values would give us direct access to the individual treatment effects.

This gives us a hint about what we need: if we could somehow replace the unobservable counterfactual expectation $\mathbb{E}[Y^{(1)}]$ with the observable conditional expectation $\mathbb{E}[Y \mid A = 1]$, we would be able to learn the ATE directly from our data.

However, as shown in Table 2.2, there is a major problem. The missingness in the table is *not* random. Figure 2.10 (a) shows a Directed Acyclic Graph explaining why: the patient's condition (severity of illness) confounds both the treatment assigned and the ultimate outcome. Doctors usually prescribe heavy treatment to the sickest patients, yielding treated groups that are composed of significantly more severe cases than untreated groups. This is the exact mechanism that drives Simpson's Paradox.

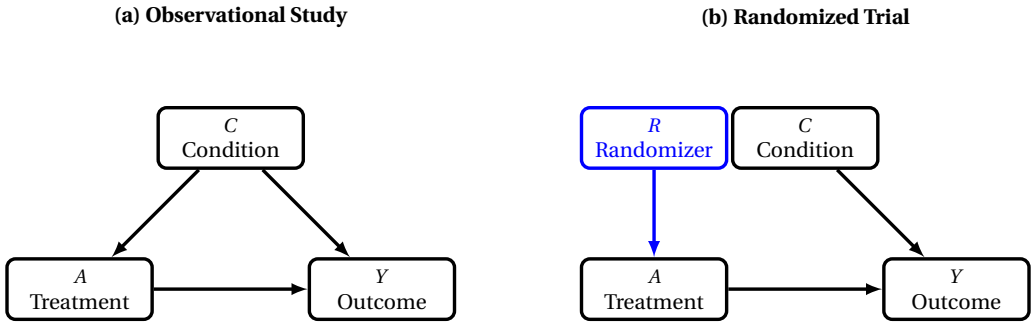


Figure 2.10: (a) Patient Condition (C) acts as a confounder. (b) A randomized coin flip (R) dictates treatment, completely overriding the doctor's choice and severing the arrow from Condition to Treatment (shown as a dashed gray line).

The intuition behind how to solve this confounding problem has deep historical roots. In 1747, Scottish physician James Lind conducted what is widely considered the first clinical trial to find a cure for scurvy. He took twelve sailors with similar cases of the disease and divided them into six pairs, giving each pair a different treatment—testing everything from cider, vinegar, and sulfuric acid to seawater, spices, and citrus fruits. By ensuring the sailors lived in the same quarters and ate the same baseline diet, Lind isolated the causal effect of the oranges and lemons.

In the 1920s, Sir Ronald Fisher mathematically formalized this approach in the context of agricultural experiments. Fisher was highly concerned with unobserved confounding. Imagine testing a new crop fertilizer on a large field. As seen in Figure 2.11, there may be hidden, unobserved bands of highly fertile soil running through the field. If a researcher systematically plants the treated seeds ($A = 1$) on the left side of the field and the control seeds ($A = 0$) on the right, they risk hopelessly confounding the effect of the fertilizer with the soil's natural fertility.

Fisher's insight was that by *randomly* assigning small plots to either receive or not receive the fertilizer, the laws of probability guarantee that both the treated and untreated plots will fall into the hidden fertile

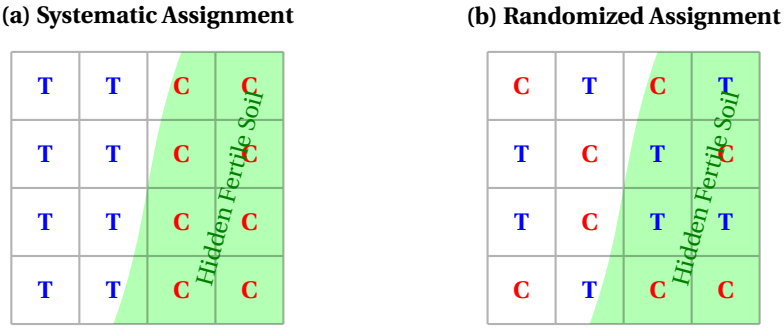


Figure 2.11: Fisher's insight on unobserved confounding in agricultural trials. (a) Systematic assignment perfectly confounds the Control (C) group with the hidden fertile soil. (b) Random assignment ensures that both treated and control plots land in the fertile band at approximately equal rates.

bands at perfectly equal rates. Randomization ensures that, in the limit of a large sample size, the *expected* baseline characteristics of both groups are perfectly identical.

By taking the decision of who gets the treatment away from the doctor (or the farmer) and handing it over to a randomizer (like a coin flip), we fundamentally alter the data-generating process. As shown in Figure 2.10(b), this mathematically severs the causal arrow from Condition to Treatment.

2.4.3 Exchangeability and Identifiability

Because the assignment of treatment was handed over to a purely random process, the missingness in Table 2.2 is now completely random. A patient's potential outcomes $Y^{(0)}$ and $Y^{(1)}$ exist independently of whether the coin landed on heads or tails.

The critical mathematical property being satisfied here is that the potential outcomes are statistically independent of the treatment assignment variable A. In other words, there is no "sampling bias" driving the visibility of each potential outcome. This is known as **Exchangeability**:

$$Y^{(a)} \perp\!\!\!\perp A. \tag{2.6}$$

Exchangeability essentially tells us that the group that received treatment and the group that did not receive treatment are "the same" and we can treat the two groups as valid counterfactuals for one another.

This independence assumption unlocks the first half of the ATE. It formally allows us to state that the expected counterfactual outcome of the entire population is exactly equal to the expected counterfactual outcome of the treated group:

$$\mathbb{E}[Y^{(a)}] = \mathbb{E}[Y^{(a)} \mid A = a] \tag{2.7}$$

However, to map this counterfactual quantity to our actual, observed data, we rely on a second logical step: if a patient is actually assigned treatment $A = a$, their observed outcome Y is exactly their potential outcome $Y^{(a)}$. This seemingly obvious property is called **Consistency**:

$$\mathbb{E}[Y^{(a)} \mid A = a] = \mathbb{E}[Y \mid A = a] \tag{2.8}$$

(We will rigorously formalize Consistency and its limitations later in this chapter).

By chaining these two steps together and plugging them into Equation 2.5, the Average Treatment Effect is fully **identified** using only observable data:

$$\mathbb{E}[Y^{(1)} - Y^{(0)}] = \mathbb{E}[Y^{(1)}] - \mathbb{E}[Y^{(0)}] = \mathbb{E}[Y \mid A = 1] - \mathbb{E}[Y \mid A = 0].$$

Main Idea 7

In a Randomized Controlled Trial, the physical act of randomization guarantees **Exchangeability** ($Y^{(a)} \perp\!\!\!\perp A$). When combined with **Consistency**, this mathematically unlocks the Average Treatment Effect, allowing us to identify causal effects directly from observable data.

Exercise 2

In COVID vaccine trials, age-related risks of COVID have led researchers to use a different randomization procedure (e.g., a coin flip with a different bias towards heads) for older patients than for younger ones. Can the old and young data from these trials be combined and still be treated as one large RCT? Why or why not?

2.4.4 A/B Testing in the Modern Tech Industry

While the terminology of Randomized Controlled Trials originates in medicine and agriculture, the exact same mathematical framework drives decision-making in the modern technology sector. In software engineering, marketing, and web development, an RCT is simply called an **A/B Test**.

When a tech company wants to know if a new recommendation algorithm, a changed user interface, or a different pricing model (Version B) increases user engagement compared to the current baseline (Version A), they cannot just look at observational data. Users who naturally navigate to a new feature might be systematically different (more tech-savvy, more engaged) than those who do not, which can confound the results. Simply alternating interfaces is insufficient as well, as different types of users log onto websites at different times, and households may visit websites in bursts.

Instead, the company relies on its routing software to act as the "coin flip," randomly assigning incoming web traffic to either the Control group (A) or the Treatment group (B). Because modern web platforms can run these experiments on millions of users simultaneously, the Law of Large Numbers practically guarantees perfect exchangeability between the two groups. Every unobserved confounder—from the user's geographical location to their current mood—is likely balanced across the two variants by randomization. As a result, any observed difference in the target metric (like click-through rate or revenue) can be confidently identified as the true causal Average Treatment Effect (ATE) of the new feature.

2.4.5 The Limits of RCTs: SUTVA and Interference

While Randomized Controlled Trials and A/B tests perfectly solve the problem of confounding by guaranteeing exchangeability, our mathematical formulation of potential outcomes relies on another foundational assumption. When we write $Y^{(1)}$ and $Y^{(0)}$ for a given patient, we are implicitly assuming that this notation is *well-defined*: that a patient's outcome only depends on *their own* treatment assignment, and that the treatment label itself refers to a single, unambiguous intervention. This is formalized as the **Stable Unit Treatment Value Assumption (SUTVA)**.

SUTVA consists of two core components:

1. **No Interference:** The treatment assignment of one unit does not affect the potential outcomes of any other unit.
2. **No Multiple Versions of Treatment:** The label $A = a$ refers to one well-defined intervention. There are no hidden variations of the treatment (e.g., one doctor giving a 50mg pill and another giving a 100mg pill under the same treatment label).

SUTVA and Consistency are deeply intertwined, but they play distinct roles.[1] SUTVA is what makes the *notation* $Y^{(a)}$ meaningful: if my outcome depends on my neighbor's treatment, or if "treatment" secretly means two different pills, then there is no single quantity that $Y^{(a)}$ can refer to. Consistency is then the

[1] Some authors fold the consistency equation into SUTVA itself, or use "consistency" as a label for the no-multiple-versions condition. We follow the convention that separates them; see [VanderWeele, 2009] for a careful treatment of the distinction.

linking equation that connects this well-defined notation to observed data: if a unit actually receives $A = a$, its observed outcome satisfies $Y = Y^{(a)}$. In other words, SUTVA secures the meaning of the potential outcome, and Consistency translates it into a mathematical rule. When SUTVA fails, Consistency does not merely become false—it becomes ill-posed, because the right-hand side of $Y = Y^{(a)}$ no longer picks out a unique quantity.

The "no multiple versions" condition is usually straightforward to maintain in a strictly controlled setting, but the "no interference" assumption is frequently violated in the real world, fundamentally undermining the results of an otherwise perfect RCT. For example, vaccines create herd immunity, meaning an untreated person's potential outcome $Y^{(0)}$ might change depending on whether their neighbors received $A = 1$. In the tech world, an A/B test for a new social media messaging feature will inevitably experience network effects; users in the control group might alter their behavior if all their friends are using the new feature in the treatment group.

Philosophical Aside: Is Consistency an Assumption or a Theorem?

The two languages of causal inference disagree about the status of Consistency. In the potential outcomes framework, $Y = Y^{(a)}$ when $A = a$ is an *axiom*—a primitive assumption we must adopt, since potential outcomes are themselves primitive objects. In the structural framework we develop in the next chapter, Consistency is a *theorem*: the observed outcome and the potential outcome are both computed by the same structural equation f_Y, so when the natural treatment happens to equal the intervened value, the two computations coincide automatically. This is why Consistency felt "seemingly obvious" when we first met it—the intuition that makes it obvious is secretly a structural one. The assumption's real content does not disappear, however; it reappears as the requirement that the intervention $\text{do}(A = a)$ corresponds to the *same mechanism* as the natural variation in A, which is exactly what SUTVA's "no multiple versions" condition guards.

Handling SUTVA Violations: Cluster Randomized Trials

When the "no interference" assumption of SUTVA inevitably breaks down, researchers often turn to *Cluster Randomized Trials* (CRTs).

For example, if we are testing a new educational intervention, randomly giving it to half the students in a classroom and withholding it from the other half might lead to students sharing resources (a phenomenon known as "spillover"). Instead of randomizing individuals, entire clusters—such as classrooms, hospitals, or entire villages—are assigned to the treatment or control group. By randomizing at the cluster level, we isolate the interference to *within* the cluster, preserving SUTVA and exchangeability between the treated clusters and the untreated clusters.

2.5 Identifying Causality from Observational Data

While Randomized Controlled Trials give us exchangeability cleanly, we often have to answer causal questions using purely observational data where treatment was not randomly assigned (e.g., we cannot ethically run an RCT forcing a random group of people to smoke). In the 1970s, Donald Rubin formalized a transformative conceptual leap: **we do not actually need physical randomization to identify causal effects; we only need exchangeability**.

Main Idea 8

Exchangeability ($Y^{(a)} \perp\!\!\!\perp A$) is the holy grail of causal inference. It is the key property that makes a causal effect identifiable. While a Randomized Controlled Trial is the gold-standard vehicle for achieving exchangeability, it is not the only way.

If we can find a way to make our treated and untreated groups exchangeable using purely observational data, the data will mathematically behave *as if* a coin had been flipped. The remainder of causal inference

is largely devoted to finding clever ways to achieve this without flipping a coin. To do so, the Rubin Causal Model leans heavily on a framework of formal assumptions.

2.5.1 Unconfoundedness and Conditional Exchangeability

Looking back at our missing data in Table 2.2, the data does *not* satisfy strict, population-level exchangeability. Severely ill patients are more likely to receive treatment and less likely to recover, heavily confounding the relationship. However, if we zoom in and look *only* within a band of similarly ill patients, the treatment assignment might look as good as random. This assumption—that we have exchangeability within specific strata of measured variables—is called **Unconfoundedness** (or **Conditional Exchangeability**).

Let C be a Bernoulli random variable representing the patient's condition, where $C = 0$ denotes a mild condition and $C = 1$ denotes a severe condition. The assumption of unconfoundedness is written mathematically as:

$$Y^{(a)} \perp\!\!\!\perp A \mid C \tag{2.9}$$

This means that if we condition on all relevant confounders C, the potential outcomes are independent of the treatment assignment. We consider different datapoints that are similar in their observed covariates to be valid counterfactuals for one another. We don't need perfect twins; we just need groups that have similar values of C.

2.5.2 The Conditional Average Treatment Effect (CATE) and Covariate Adjustment

Conditional exchangeability is incredibly powerful. It means that while we cannot compute an Individual Treatment Effect (ITE) because we lack a time machine, we *can* compute the interventional effect of the treatment for specific subgroups!

$$\begin{aligned}\mathbb{E}[Y^{(1)} - Y^{(0)} \mid C = c] &= \mathbb{E}[Y^{(1)} \mid C = c] - \mathbb{E}[Y^{(0)} \mid C = c] \\ &= \mathbb{E}[Y \mid C = c, A = 1] - \mathbb{E}[Y \mid C = c, A = 0]\end{aligned}$$

This estimand is called the **Conditional Average Treatment Effect (CATE)**. If the Average Treatment Effect (ATE) gives us a blunt, population-level summary, the CATE allows for personalized medicine. It lets us explicitly evaluate Heterogeneous Treatment Effects (HTEs) by asking: "Does this drug work differently for severe cases ($C = 1$) than it does for mild cases ($C = 0$)?"

To go back to an aggregate metric, we can use the Law of Total Expectation. The expected value of a single potential outcome for the entire population is recovered by marginalizing over our confounder C:

$$\mathbb{E}[Y^{(a)}] = \sum_{c \in \{0,1\}} \Pr(C = c)\mathbb{E}[Y^{(a)} \mid C = c] = \sum_{c \in \{0,1\}} \Pr(C = c)\mathbb{E}[Y \mid C = c, A = a] \tag{2.10}$$

By applying this logic to both potential outcomes, the Law of Total Expectation gives us the overall Average Treatment Effect:

$$\begin{aligned}\mathbb{E}[Y^{(1)} - Y^{(0)}] &= \sum_{c \in \{0,1\}} \Pr(C = c)\mathbb{E}[Y^{(1)} - Y^{(0)} \mid C = c] \\ &= \Pr(C = 0)\left(\mathbb{E}[Y \mid C = 0, A = 1] - \mathbb{E}[Y \mid C = 0, A = 0]\right) \\ &\quad + \Pr(C = 1)\left(\mathbb{E}[Y \mid C = 1, A = 1] - \mathbb{E}[Y \mid C = 1, A = 0]\right)\end{aligned}$$

This shows that the ATE is simply the weighted average of the CATEs for each group, weighted by that group's prevalence in the population. By identifying all confounders, breaking the population into unconfounded subgroups, computing the CATEs, and weighting them back together, Rubin proved we can estimate true causal effects without ever flipping a coin. This "functional" is known as the **covariate adjustment** or the **g-formula**, and it serves as the foundation for observational causal inference.

Mathematical Aside: Functions, Operators, and Functionals

From the perspective of pure mathematics, a functional *is* technically a function, as it maps an object from a domain to a codomain. However, in applied mathematics and statistics, we use a specific taxonomy to avoid confusion:

- A **function** maps numbers to numbers (e.g., $f(x) = x^2$).
- An **operator** maps functions to functions (e.g., the derivative $\frac{d}{dx}$).
- A **functional** maps an entire function or probability distribution down to a single scalar (e.g., a definite integral).

Calling the Average Treatment Effect (ATE) a "functional" explicitly reminds us that our mathematical input is an entire probability distribution. This cleanly separates our non-parametric target parameter from the specific parametric statistical models (functions) we might eventually use to estimate it from finite data.

"Fixing" Covariates. It is instructive to compare the g-formula to standard conditional probability: $\mathbb{E}[Y|A = a] = \sum_c \Pr(C = c|A = a)\mathbb{E}[Y|C = c, A = a]$. Intuitively, the g-formula calculates the expected outcome by holding the baseline distribution of C fixed, "not allowing" Bayes' rule to shift the distribution of C based on the treatment assignment A.

2.5.3 Computing the Adjustment Functional: Discrete and Continuous Examples

The covariate adjustment functional is *non-parametric*, meaning it is not limited by assumptions such as linearity or Gaussianity (this also means that it is limited to Rung 2 of the causal ladder). Nevertheless, the non-parametric functional can be computed in parametric settings. To see how covariate adjustment mechanically eliminates confounding, it is helpful to walk through how this functional operates on both discrete and continuous data.

Example 1: A Discrete Setting and Simpson's Paradox

Suppose a school district introduces an optional after-school tutoring program (A) and wants to measure its causal effect on a student's probability of passing the final exam (Y). The data is heavily confounded by a student's prior academic standing (C), where $C = 0$ means struggling and $C = 1$ means advanced. Naturally, struggling students are heavily pushed into the tutoring program, while advanced students skip it, creating a severe imbalance.

Imagine the district has 1,000 students. If an analyst simply looks at the observational data, they might construct something like Table 2.3. Looking at the bottom row, the raw conditional probability of passing given treatment is misleading:

- Pass rate for tutored students: $\Pr(Y = 1 \mid A = 1) = 71.7\%$
- Pass rate for untutored students: $\Pr(Y = 1 \mid A = 0) = 78.6\%$

The unadjusted difference is -6.9%. A naive reading of the conditional probabilities suggests the tutoring program is actively harming students!

	Tutored ($A = 1$)		**Not Tutored** ($A = 0$)	
Subgroup	Total	Passed	Total	Passed
Struggling ($C = 0$)	540	378 (70%)	60	24 (40%)
Advanced ($C = 1$)	40	38 (95%)	360	306 (85%)
Total	**580**	**416 (71.7%)**	**420**	**330 (78.6%)**

Table 2.3: Observational data for the after-school tutoring program.

However, this is a classic manifestation of **Simpson's Paradox**. The paradox occurs because the treated group ($A = 1$) is overwhelmingly composed of struggling students, who naturally have a lower baseline pass rate. Notice that when we look at the subgroups in Table 2.3, the tutoring helps *both* types of students:

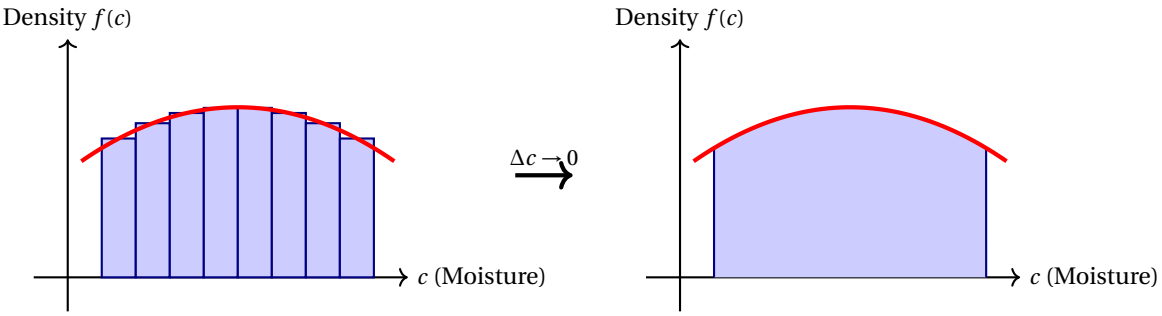
- **Struggling ($C = 0$):** 70% pass if tutored, 40% pass if not. (CATE = +0.30)
- **Advanced ($C = 1$):** 95% pass if tutored, 85% pass if not. (CATE = +0.10)

Because we assume unconfoundedness within these subgroups, these observed differences are the true CATEs. To find the population ATE, we apply the covariate adjustment functional (Equation 2.10). Instead of weighting by the proportion of students in the *treatment* groups (which caused the paradox), the functional weights each CATE by the true proportion of the population that group represents.

From our table, there are 600 struggling students ($\Pr(C = 0) = 0.6$) and 400 advanced students ($\Pr(C = 1) = 0.4$):

$$\begin{aligned} ATE &= \Pr(C = 0) \cdot \text{CATE}_{C=0} + \Pr(C = 1) \cdot \text{CATE}_{C=1} \\ &= (0.6 \times 0.30) + (0.4 \times 0.10) \\ &= 0.18 + 0.04 = \mathbf{0.22} \end{aligned}$$

The tutoring program actually increases the average student's probability of passing by 22 percentage points. By conditioning on C and marginalizing it out using the baseline population probabilities, the functional completely bypassed the biased treatment assignment mechanism and resolved Simpson's Paradox.



(a) Discrete Summation ($\sum$)
Bin width = Δc

(b) Continuous Integral ($\int$)
Infinitesimal width = dc

Figure 2.12: As the width of the covariate bins (Δc) becomes infinitesimally small (dc), the discrete summation of the g-formula transforms into a continuous integral.

Example 2: Transitioning to Continuous Confounders

Real-world confounders are rarely binary. Imagine an agricultural study evaluating a new fertilizer (A) on crop yield (Y). The primary confounder is soil moisture (C), measured as a continuous percentage from 0 to 100.

If we only had discrete math, we would have to chop the continuous moisture levels into discrete bins (e.g., 0-1%, 1-2%, etc.) and run the exact same summation we used in the tutoring example:

$$\sum_{bins} \Pr(C \in \text{bin}_i) \cdot \text{CATE}_{\text{bin}_i}.$$

However, as we make our bins infinitesimally small to capture the exact moisture level, the probability of landing in any specific bin $\Pr(C = c)$ transitions into a probability density function $f(c)dc$. Consequently, as illustrated in Figure 2.12, our discrete summation transforms into a continuous integral over the domain of C:

$$ATE = \int (\mathbb{E}[Y \mid C = c, A = 1] - \mathbb{E}[Y \mid C = c, A = 0])\, f(c)\, dc \tag{2.11}$$

To make the math easy, assume soil moisture is perfectly uniformly distributed across the farms, meaning $C \sim \text{Uniform}(0, 100)$. Because it is uniform, the density function is a constant: $f(c) = \frac{1}{100}$.

Suppose our observational data reveals that the expected crop yield follows these linear relationships based on moisture and fertilizer:

- Treated fields ($A = 1$): $\mathbb{E}[Y \mid C = c, A = 1] = 50 + 2c$
- Untreated fields ($A = 0$): $\mathbb{E}[Y \mid C = c, A = 0] = 40 + c$

The CATE at any exact moisture level c is simply the difference: $(50 + 2c) - (40 + c) = 10 + c$. Notice that the treatment effect is highly heterogeneous; the fertilizer works much better in wet soil.

To find the average causal effect across all farms, we plug this into our integral functional:

$$ATE = \int_0^{100} (10 + c)\left(\frac{1}{100}\right) dc = \frac{1}{100}\left[10c + \frac{c^2}{2}\right]_0^{100} = \frac{1}{100}(1000 + 5000) = \mathbf{60}$$

The average causal effect of the fertilizer is exactly 60 units of yield. Whether the confounder is a simple binary category or a complex continuous density, the covariate adjustment functional executes the exact same underlying logic: isolate the effect at every level of confounding, and take the weighted average over the population.

Example 3: Linear Regression as Covariate Adjustment

If we know a treatment effect is a constant homogeneous effect and that all the relationships are linear, we can drastically simplify covariate adjustment in continuous settings.

Recall our example from earlier lectures where Season (C) severely confounds the relationship between atmospheric CO_2 (A) and global temperatures (Y). In the winter, heating causes CO_2 to spike, but temperatures are naturally cold. In the summer, plants absorb CO_2, but temperatures are naturally hot. If we plot the raw 2D data (Figure 2.13a), the short-term seasonal cyclic trends imply a negative correlation (more CO_2 means colder weather!). However, over the course of decades, the global macro-trend is undeniably positive.

Suppose we select a set of variables including the confounders, and fit a single model containing both the treatment and our chosen confounder:

$$\mathbb{E}[Y \mid A = a, C = c] = \beta_0 + \beta_1 a + \beta_2 c \tag{2.12}$$

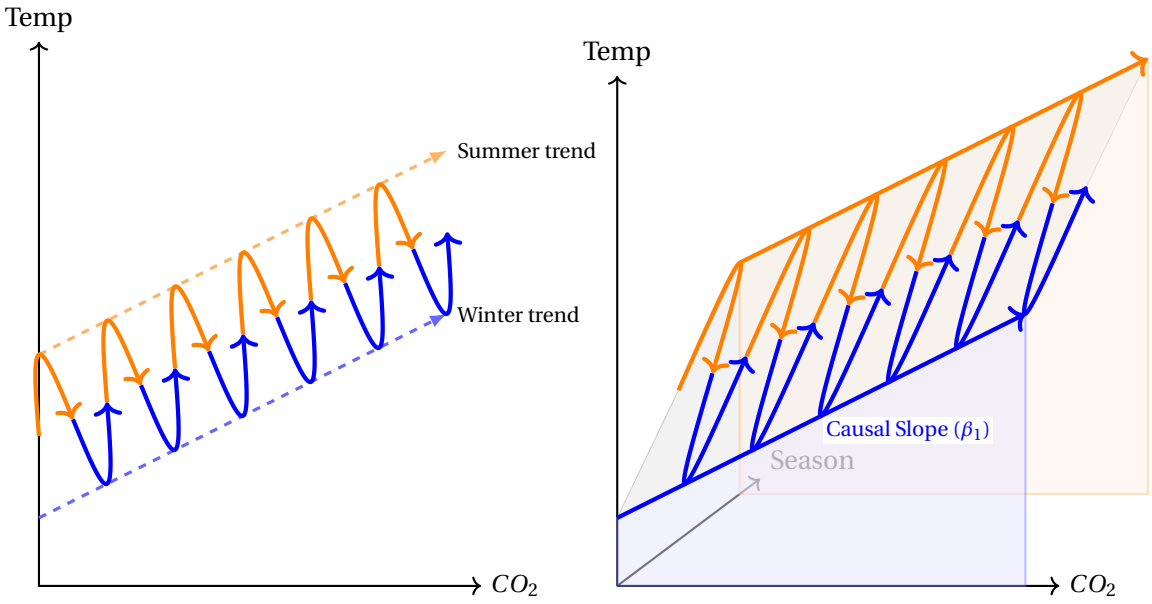
If we plug this regression equation directly into our formula for the CATE, notice that the β_0 and $\beta_2 c$ terms perfectly cancel out:

$$\begin{aligned}\text{CATE}_c &= \mathbb{E}[Y \mid A = a + 1, C = c] - \mathbb{E}[Y \mid A = a, C = c] \\ &= (\beta_0 + \beta_1(a + 1) + \beta_2 c) - (\beta_0 + \beta_1 a + \beta_2 c) \\ &= \beta_1\end{aligned}$$

Because the model is strictly linear and additive, the causal effect β_1 is a constant. The CATE is exactly equal to the ATE for every season, eliminating the need for integration and allowing the regression to pool outcomes across many values of C to estimate this coefficient.

Geometrically, as shown in Figure 2.13b, running a multiple regression lifts our confounded 2D data into a 3D space, fitting a flat plane to the points. Controlling for $C = c$ means taking a 2D slice through that plane at a specific season. Because the data structurally lives on this 3D plane, the slope of the line projected onto that single slice is β_1, completely cleansed of the confounding depth of the seasonal axis.

Thus, under the assumption of linearity, **the coefficient of the treatment variable in a multiple regression is the causal Average Treatment Effect.** Linear regression inherently performs covariate adjustment!



(a) 2D View (Confounded)

(b) 3D View (Stratified by Season)

Figure 2.13: Linear regression as covariate adjustment. In 2D space (a), the cyclic confounding of season obscures the true causal step from CO_2 to Temperature. By adding Season as a covariate in a multiple regression (b), we lift the data into 3D space. The regression coefficient (β_1) is geometrically equivalent to the slope of a 1D line projected onto a 2D slice where Season is held completely constant.

2.5.4 The Positivity Assumption (Overlap)

While unconfoundedness tells us that we *must* compare treated and untreated subjects within the same covariate strata, it implicitly assumes that such subjects actually exist. What if the school district from our first example had a strict policy that *every single* struggling student ($C = 0$) must attend tutoring?

If this happens, our data for struggling students will consist entirely of treated individuals. We will have no untreated struggling students to serve as a counterfactual baseline. Mathematically, attempting to calculate the CATE for $C = 0$ becomes impossible because $\mathbb{E}[Y \mid C = 0, A = 0]$ is undefined—it is an empty set.

To identify a causal effect, we require the **Positivity Assumption** (also known as **Overlap** or **Common Support**). Positivity dictates that for every possible subgroup defined by our confounders, there must be a non-zero probability of receiving the treatment and a non-zero probability of not receiving the treatment. Mathematically, for all values of c where $\Pr(C = c) > 0$:

$$0 < \Pr(A = 1 \mid C = c) < 1 \tag{2.13}$$

If $\Pr(A = 1 \mid C = c) = 1$ (everyone in the group is treated) or $\Pr(A = 1 \mid C = c) = 0$ (no one in the group is treated), we have a positivity violation. When positivity is violated in observational data, researchers are forced to either restrict their causal claims only to the subgroup where overlap exists or rely heavily on dangerous extrapolations.

2.5.5 The Identifiability Conditions: Putting it all Together

We can now summarize the trinity of assumptions required to estimate causal effects from purely observational data. If we want to move from an unobservable counterfactual quantity to an observable statistical quantity (like the g-formula), our data-generating process must satisfy:

1. **Conditional Exchangeability (Unconfoundedness):** $Y^{(a)} \perp\!\!\!\perp A \mid C$. There are no unmeasured confounders; within strata of C, treatment is essentially random.

2. **Positivity (Overlap):** $0 < \Pr(A = a \mid C = c) < 1$. Every type of person has some chance of receiving each treatment, ensuring we have data to compare.

3. **Consistency:** If $A = a$, then $Y = Y^{(a)}$. The observed outcome of a unit receiving treatment a is exactly its potential outcome under that treatment. (This presupposes SUTVA: the "no interference" and "no multiple versions" conditions are what make the notation $Y^{(a)}$ well-defined in the first place.)

When these three conditions hold, the Average Treatment Effect is fully **identified**. We can confidently apply the tools of observational causal inference—from simple covariate adjustment to complex machine learning—to estimate true causal effects without ever needing a Randomized Controlled Trial.

2.6 Chapter Summary: The Foundations of Causal Inference

We began this chapter with a warning: data, no matter how abundant, cannot speak for itself. As Simpson's Paradox perfectly illustrates, relying purely on statistical associations can lead us to conclusions that are not just mathematically inaccurate but also actively harmful in practice. To truly understand why things happen and what would happen if we intervened, we need a mathematical language for causality.

To build this foundation, we explored two of the most powerful paradigms in modern causal inference:

- **Pearl's Ladder of Causation:** We saw that moving from observation (Level 1) to intervention (Level 2) and ultimately to counterfactuals (Level 3) requires increasingly strong assumptions. We introduced the intuition behind causal diagrams and the do-operator to conceptualize the difference between "seeing" and "doing."

- **The Potential Outcomes Framework:** We reframed causal inference as a fundamental missing data problem. Because we can never observe both $Y^{(1)}$ and $Y^{(0)}$ for the same individual, we are forced to rely on assumptions about the comparability of different units to compute causal effects.

This brought us to the gold standard of causal inference: the Randomized Controlled Trial (RCT). By randomly assigning treatment, we break any links between unobserved confounders and the treatment itself. This ensures *exchangeability*, guaranteeing that our treated and untreated groups are apples-to-apples comparisons. We also saw that even when strict exchangeability fails globally, *conditional exchangeability* allows us to estimate causal effects by adjusting for covariates, bringing us one step closer to isolating true interventional signals from observational data.

However, RCTs are often expensive, unethical, or logistically impossible. In the real world, we are usually handed observational data and asked to make causal claims. We know from our identifiability conditions that we must achieve conditional exchangeability by controlling for the correct set of confounders. But in a complex system with dozens of interrelated variables, how do we know exactly which variables to condition on? And just as importantly, as we saw with colliders and mediators, which variables must we *avoid* conditioning on to prevent inducing new, catastrophic biases?

This is the exact challenge we will tackle in the next chapter. We will rigorously formalize the intuitive causal diagrams introduced here into the framework of **Structural Causal Models (SCMs)**. By learning the graphical rules of **d-separation**, we will develop a mathematically sound method for interrogating our assumptions. Before we ever run a regression or train a machine learning model, we will learn how to look at a system and definitively answer: *is this causal effect identifiable, and exactly which variables must we control for to isolate it?*

Chapter 3

Structural Causal Models

In the previous chapter, we established the fundamental rules of causal inference: to isolate a causal effect from observational data, we need exchangeability, positivity, and consistency. We learned that if we condition on the correct set of confounders, we can mathematically untangle the treatment from the noise. But a question remains: in a complex system with dozens of interacting variables, how do we know *which* variables to condition on, and which ones to avoid?

This chapter is the theory that comes before estimation: given a causal question and a picture of how the world is wired, we want to decide whether the effect of an intervention can be recovered from observational data, and if so, exactly which combination of observed quantities recovers it. Notice that this is not yet a statistical question. We are not estimating anything, and we have no data in hand. We will be reasoning about structure, and the work has the flavor of logic or discrete mathematics. We will manipulate graphs and probability expressions according to a small set of rules until a do-operator either disappears, leaving a functional we can estimate, or provably cannot.

The machinery is built in layers. We formalize how the world generates data as a structural causal model, and learn to read off when variables are associated from three atomic patterns — chains, forks, and colliders — that together yield properties of d-separation that explain how information moves throughout these structures. We then intervene, and the backdoor criterion frames "which variables to condition on" as an exercise in blocking information-flow. Single World Intervention Graphs let us draw a potential outcome on a graph, so that the exchangeability we demanded in the last chapter becomes something we can check by eye. When confounders go unobserved, and no adjustment set exists at all, the frontdoor criterion delivers the happy surprise that an effect can still sometimes be identified through a mediator rather than a control. The do-calculus closes the chapter with three rules that turn out to be not merely sound but complete: when they cannot eliminate a do-operator, the effect is genuinely unidentifiable, not simply beyond our ingenuity.

Underneath all of this runs one theme. The field speaks two dialects: the graphs of Pearl and the potential outcomes of Neyman and Rubin, which are too often taught as rival camps when they are not. Instead, we will see that these are two notations for one logic, and by the end of this chapter, you should be fluent enough in both to read whichever notation the problem in front of you happens to be written in.

3.1 Bayesian Causal Networks

3.1.1 Graph Theory Recap and Topological Sort

Before discussing how variables factorize, we must briefly recap the graph theory that underpins causal diagrams. A directed graph, denoted as $\mathcal{G} = (\mathbf{V}, \mathbf{E})$, is mathematically defined as a tuple of vertices and edges. In the context of causal inference, these components have specific meanings:

- **Vertices (V):** Represent random variables or events.
- **Edges (E):** Ordered pairs representing causal dependence, where a directed arrow indicates the flow of causality.

To navigate these graphs mathematically, we borrow standard genealogical terminology. For any given node $V_i \in \mathbf{V}$, we define the following sets:

- **Parents, PA(V_i):** The set of all nodes with a direct edge pointing *into* V_i.
- **Children, CH(V_i):** The set of all nodes with a direct edge pointing *out of* V_i.
- **Ancestors, AN(V_i):** The transitive closure of parents (i.e., parents, parents of parents, and so on).
- **Descendants, DE(V_i):** The transitive closure of children (i.e., children, children of children, and so on).

A foundational assumption in standard causal graphs is that there are no directed cycles (e.g., $A \to B \to A$ is not allowed). Throughout this text, we will restrict our focus entirely to Directed Acyclic Graphs (DAGs).

Mathematical Aside: Unrolling Cyclic Graphs

While standard DAGs do not allow for feedback loops, many real-world systems exhibit cyclical behavior (e.g., a dynamic where $A \to B \to C \to A$).
Recent research has explored handling cyclic graphs by "unrolling" them over time. As illustrated below, breaking the variables into distinct, time-indexed nodes instantly resolves the cycle and forms a valid DAG.

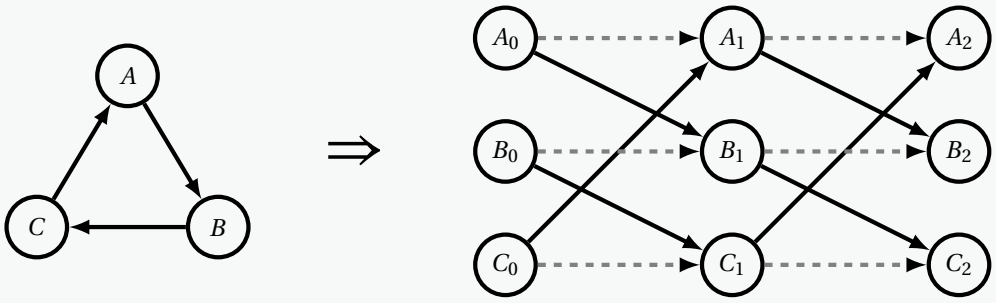


Figure Note: A cyclic causal relationship resolved by unrolling the system over discrete time steps to form a valid Directed Acyclic Graph (DAG). Diagonal lines represent the cyclic effects, while dashed horizontal lines represent temporal persistence.

Because a DAG contains no directed cycles, it naturally admits a **topological ordering** (or topological sort). A topological ordering is a linear sequence of the vertices such that for every directed edge $X \to Y$ in the graph, the parent X strictly precedes the child Y in the sequence. Depending on the graph's structure, there may be multiple valid topological orderings. For example, consider the graph in Figure 3.1 where $A \to B$, $A \to C$, $B \to D$, $C \to D$, and $D \to E$. Both (A, B, C, D, E) and (A, C, B, D, E) are valid topological sorts.

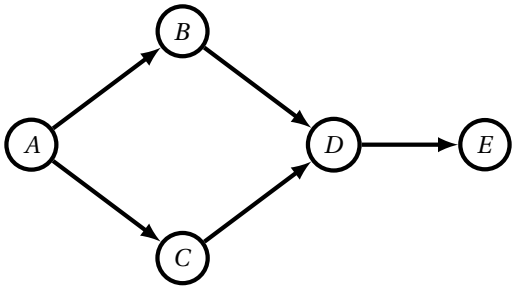


Figure 3.1: A DAG demonstrating multiple valid topological orderings, including (A, B, C, D, E) and (A, C, B, D, E).

For a step-by-step walkthrough of the computational algorithm used to find this ordering, please see the Graph Theory Preliminaries in the Appendix.

3.1.2 Generative Models and Independent Errors (NPSEM-IE)

If we think of the world as a data generator, we can interpret the topological ordering of DAGs as the chronological sequence in which data is generated. To truly treat a directed graph as a *causal* graph, we must make a foundational assumption about how this data generation occurs in the real world.

For the Causal Markov Condition (which we will formally define in the next section) to hold true in reality, we rely on what is called a **Non-Parametric Structural Equation Model with Independent Errors (NPSEM-IE)**.

In an NPSEM-IE, we assume that the value of every variable V_i in our DAG is determined by two things:

1. Its direct parents in the graph, $\mathbf{PA}(V_i)$.
2. A set of unobserved background factors unique to V_i, often denoted as N_i (or ϵ_i).

To visualize this, consider Figure 3.2. Every observed variable in the causal chain is pushed by its own invisible, latent error term.

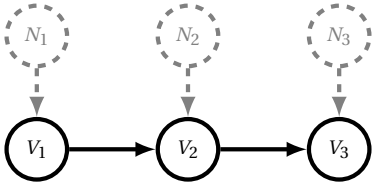


Figure 3.2: A causal graph where the observed variables (V_1, V_2, V_3) are generated by an NPSEM-IE. The dashed nodes represent the unobserved, independent background errors (N_1, N_2, N_3) that inject randomness into each deterministic structural equation.

To make this mechanism concrete, you are likely already familiar with an example of a Structural Equation Model (SEM) with additive noise (from the previous chapter). Such a model assigns explicit mathematical formulas to the graph:

$$\begin{aligned} V_1 &= N_1 \\ V_2 &= 2.5V_1 + N_2 \\ V_3 &= -1.2V_2 + N_3 \end{aligned}$$

In this parametric SEM, we strictly assume the relationships are linear and the unobserved errors are simply added to the result.

However, the "NP" in NPSEM stands for **non-parametric**. In causal graph theory, we do not assume to know anything about the underlying equations. We do not assume linearity, we do not assume additivity, and we do not assume Gaussian noise. We simply assume the *structural form*:

$$V_i = f_i(\mathbf{PA}(V_i), N_i) \tag{3.1}$$

where f_i can be any arbitrarily complex, non-linear, unknown function in the universe. This is why graphical models are so powerful—they allow us to prove causal identifiability without ever needing to know the true equations f_i.

As we discussed in the last chapter, knowing only the graph and this non-parametric form (but not the functions f_i themselves) is generally enough to identify interventional, "Rung 2" (doing) quantities like $\Pr(y \mid \text{do}(x))$ from observational data, though not to recover a specific individual's counterfactual. As we will see in Section 3.4, the NPSEM-IE itself does *define* those Rung 3 potential outcomes; what the graph alone limits is which of them we can identify from data.

A Structural Note on Positivity: Crucially, these unobserved error terms N_i are what make observational causal inference mathematically possible, as they provide the structural basis for the **Positivity (Overlap)** assumption discussed in the previous chapter. If we lacked an error term N_i for a treatment variable, the NPSEM dictates that treatment would be a purely deterministic function of its observed parents, $V_i = f_i(\mathbf{PA}(V_i))$. Under strict determinism, for any specific stratum of covariates, the probability of receiving the assigned treatment would be exactly 1, and the probability of receiving any other treatment would be 0—a direct violation of positivity. By injecting idiosyncratic "wobble" (e.g., a doctor's subjective preference or random traffic delays) into the equation, the background error N_i ensures that $0 < \Pr(V_i = v \mid \mathbf{PA}(V_i)) < 1$, granting us the overlap needed to compute counterfactuals.

The "Independent Errors" (-IE) part of the acronym is the crucial assumption that makes the graph work. We assume that the unobserved background noise N_i for each variable is completely independent of the background noise for any other variable ($N_i \perp\!\!\!\perp N_j$).[1]

If these background errors were *not* independent—for instance, if the unobserved noise affecting a patient's Age (N_A) was somehow correlated with the unobserved noise affecting their Education (N_E)—then we would have an unobserved confounder linking the two variables. We will discuss how to handle this later in the chapter. We will also formally connect these independent background errors (N_i) to the concept of potential outcomes.

3.1.3 Bayesian Networks and Factorization

To fully define a Bayesian Network, we must represent the joint probability distribution of all variables in the graph. In a DAG, we can achieve a massive reduction in complexity by leveraging the topological sort. This ordering is conceptually crucial because it aligns with the generative flow of the data (as defined by the NPSEM-IE), allowing us to systematically apply the chain rule of probability and factorize the joint distribution strictly in terms of each node's parents.

Because we assume the data is generated via an NPSEM-IE, we can formally apply the *Local Markov Condition*, which is also called the *Causal Markov Condition* in causal settings.[2] Using $\overline{\mathbf{DE}(V_i)} := \mathbf{V} \setminus \mathbf{DE}(V_i) \setminus \{V_i\} \setminus \mathbf{PA}(V_i)$ to represent the non-descendants of V_i (excluding its parents), this condition states:

$$V_i \perp\!\!\!\perp \overline{\mathbf{DE}(V_i)} \mid \mathbf{PA}(V_i) \tag{3.2}$$

This means that the only relevant information for determining the distribution of V_i is its direct parents. Let's take an arbitrary DAG $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ and assume the indices are labeled according to a topological ordering. Applying the chain rule of probability in the order of the topological sort gives (using an abbreviated lowercase v_i to denote $V_i = v_i$):

$$\Pr(\mathbf{v}) = \Pr(v_1) \cdot \Pr(v_2 \mid v_1) \cdot \Pr(v_3 \mid v_1, v_2) \dots \Pr(v_n \mid v_1, \dots v_{n-1}) = \prod_{i=1}^{n} \Pr(v_i \mid v_1, \dots, v_{i-1}) \tag{3.3}$$

Now, because of the topological ordering, all predecessors of V_i are mathematically guaranteed to be non-descendants. We can partition the set of predecessors $\{v_1, \dots v_{i-1}\} = \mathbf{pa}(v_i) \cup \overline{\mathbf{pa}(v_i)}$ and use the Local Markov Assumption to drop the conditioning on all non-parent predecessors, $\overline{\mathbf{pa}(v_i)}$. By evaluating the graph in topological order, we ensure that by the time we calculate the probability of any given node V_i, we have already fully accounted for the variables that generate it.

$$\Pr(\mathbf{v}) = \prod_{i=1}^{n} \Pr(v_i \mid \mathbf{pa}(v_i) \cup \overline{\mathbf{pa}(v_i)}) = \prod_{i=1}^{n} \Pr(v_i \mid \mathbf{pa}(v_i)) \tag{3.4}$$

We say that a probability distribution is "Markovian in $\mathcal{G}$" if it perfectly factorizes according to this equation.

[1] Independent errors is a *cross-world* (Rung 3) assumption: it makes the one-step-ahead potential outcomes mutually independent. This is stronger than the single-world condition (Robins' FFRCISTG) that SWIGs encode, which is why SWIGs can establish exchangeability without committing to full cross-world independence.

[2] Technically, these conditions are different since one the local markov condition applies to all Bayesian networks, and the Causal Markov Condition is the assumption that this condition applies to causal networks.

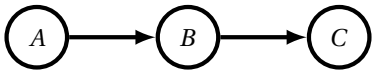
3.1.4 The Three Atomic Structures of Dependence Flow

Now that we have established that our data is generated by an NPSEM-IE, and that this structural assumption allows our joint probability distribution to factorize according to the DAG, we can answer our fundamental question: how does statistical association flow through a complex system?

Any complex causal graph, no matter how large, is built from three atomic, three-node structures: **Chains**, **Forks**, and **Colliders**. We can mathematically evaluate when variables are dependent by looking at their factorization, and we can easily prove when they are independent by simply applying our Local Markov Assumption ($V_i \perp\!\!\!\perp \overline{\mathbf{DE}(V_i)} \mid \mathbf{PA}(V_i)$).

The Chain (Mediation)

A chain represents a mechanism where A causes B, which in turn causes C.



According to our topological factorization rule, the joint distribution of this graph is:

$$\Pr(A,B,C) = \Pr(A)\Pr(B \mid A)\Pr(C \mid B) \tag{3.5}$$

Marginal Dependence ($A \not\perp\!\!\!\perp C$): If we want to know if A and C are associated in the overall population, we must marginalize out the intermediate variable B:

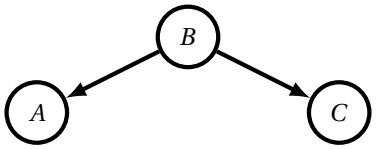
$$\Pr(A,C) = \sum_b \Pr(A,b,C) = \Pr(A)\sum_b \Pr(b \mid A)\Pr(C \mid b) \tag{3.6}$$

To see exactly why this is not equal to the independent product $\Pr(A)\Pr(C)$, recall the Law of Total Probability for the marginal distribution of C: $\Pr(C) = \sum_b \Pr(b)\Pr(C \mid b)$. Comparing the two sums, our joint equation contains the conditional term $\Pr(b|A)$, whereas the independent marginal contains $\Pr(b)$. Because A causes B, we know $\Pr(b \mid A) \neq \Pr(b)$. Therefore, $\Pr(A,C) \neq \Pr(A)\Pr(C)$, meaning A and C are **marginally dependent**. Association flows along the chain.

Conditional Independence ($A \perp\!\!\!\perp C|B$): What happens if we condition on the mediator B? We can skip the algebra and simply apply the Local Markov Assumption! In this graph, A is a non-descendant of C, and B is the parent of C. By definition, a node is independent of its non-descendants given its parents. Therefore, $C \perp\!\!\!\perp A \mid B$. Conditioning on the middle of a chain **blocks** the flow of association.

The Fork (Confounding)

A fork represents a common cause, where B causes both A and C.



The topological factorization for a fork is:

$$\Pr(A,B,C) = \Pr(B)\Pr(A \mid B)\Pr(C \mid B) \tag{3.7}$$

Marginal Dependence ($A \not\perp\!\!\!\perp C$): Marginalizing out the common cause B:

$$\Pr(A, C) = \sum_b \Pr(b) \Pr(A \mid b) \Pr(C \mid b) \tag{3.8}$$

To understand concretely why this does not equal $\Pr(A)\Pr(C)$, let's write out what the product of the independent marginals actually looks like:

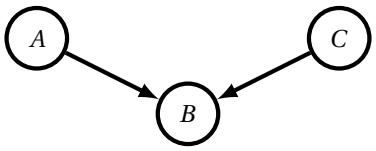
$$\Pr(A)\Pr(C) = \left(\sum_b \Pr(b) \Pr(A \mid b)\right)\left(\sum_b \Pr(b) \Pr(C \mid b)\right) \tag{3.9}$$

When comparing the joint marginalization to this expanded product, notice the missing cross-terms! The independent product computes sums independently, allowing A and C to vary across different hypothetical values of the common cause. But the joint distribution $\Pr(A, C)$ forces them to be evaluated at the *exact same* value of b simultaneously within a single summation. Thus, variables with a common cause are **marginally dependent**.

Conditional Independence ($A \perp\!\!\!\perp C \mid B$): If we condition on the common cause B, we again look to the Local Markov Assumption. A is a non-descendant of C, and B is the parent of C. The rule immediately gives us $C \perp\!\!\!\perp A \mid B$. Conditioning on a common cause **blocks** the spurious association.

The Collider (Selection)

A collider occurs when two independent variables, A and C, share a common effect B.



Notice the structural difference: A and C have no parents. The factorization is:

$$\Pr(A, B, C) = \Pr(A)\Pr(C)\Pr(B \mid A, C) \tag{3.10}$$

Marginal Independence ($A \perp\!\!\!\perp C$): Because C has no parents, its parent set is the empty set ($\emptyset$). Because A is a non-descendant of C, the Local Markov Assumption states that $C \perp\!\!\!\perp A \mid \emptyset$. This is a profound result! It means that without conditioning on anything, A and C are naturally **marginally independent**. The collision of two arrows inherently blocks the flow of association.

Conditional Dependence ($A \not\perp\!\!\!\perp C \mid B$): What happens if we do condition on the collider B?

$$\Pr(A, C \mid B) = \frac{\Pr(A)\Pr(C)\Pr(B \mid A, C)}{\Pr(B)} \tag{3.11}$$

Let's compare this to what the expression *would* look like if A and C were conditionally independent:

$$\Pr(A, C \mid B) = \Pr(A \mid B)\Pr(C \mid B) = \frac{\Pr(A, B)\Pr(C, B)}{\Pr(B)^2} \tag{3.12}$$

These two expressions are fundamentally different. The true conditional probability contains the term $\Pr(B \mid A, C)$, which intricately couples A and C together in the data-generating process. It cannot be split into separate pieces for A and C. By conditioning on a collider, we mathematically force a correlation between two variables that were previously independent. This is the source of selection bias.

Main Idea 9

Chains and forks carry dependence while colliders do not. When we condition on the middle vertex, the roles switch, and chains/forks do not carry dependence, while colliders do!

3.2 D-separation

How can we use causal DAGs to understand what is associated with what when we condition on a set of variables? In this section, we will establish the structural rules that answer this exact question. Because we are strictly looking at statistical association, everything in this chapter exists purely on Rung 1 of the Ladder of Causation.

Recall that a **directed path** is a sequence of adjacent edges where all arrows point in the same direction. When we look at the flow of statistical association, however, we can ignore the direction of the arrowheads. We proved mathematically in the previous section that statistical association flows freely through unconditioned chains and forks, but is naturally blocked by unconditioned colliders. Conversely, conditioning blocks chains and forks, but unblocks colliders.

This gives us four "atomic" structures describing the flow of dependence across a junction. Notice that we have introduced a fourth structure: a conditioned descendant of a collider. Because a descendant is essentially a noisy measurement of the collider, conditioning on it partially unblocks the collider, allowing some dependence to leak through. These four structures, under both active (open) and inactive (closed) conditions, are summarized in Figure 3.3.

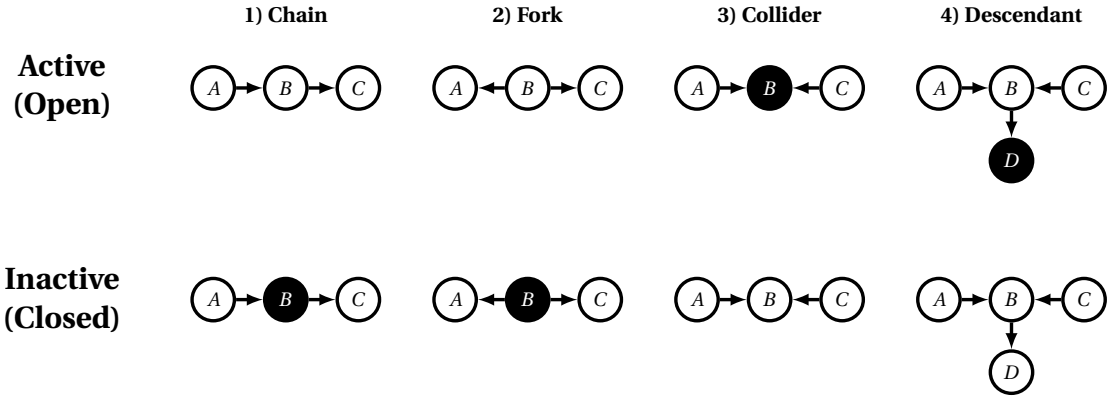


Figure 3.3: The four atomic structures governing the flow of statistical dependence. Paths are active (top row) when chains/forks are unconditioned and colliders/descendants are conditioned. They become inactive (bottom row) when the conditioning is reversed.

It turns out that we can compose these atomic structures together to understand how information flows across an arbitrarily long sequence of variables. This gives rise to the concept of **d-separation** (which stands for **directed separation** because it applies to directed graphs), providing a definitive algorithmic tool to determine if two variables are dependent or independent in a DAG.

To define d-separation, we must first formalize how atomic structures chain together into paths.

Definition 3.2.1 (Active Path)**.** Given a DAG $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ and a set of conditioning variables $\mathbf{Z}$, an undirected path between two nodes is an **active path** if for every intermediate node V_i along the route:

1. If V_i acts as a **Collider** ($V_{i-1} \to V_i \leftarrow V_{i+1}$), then $V_i \in \mathbf{Z}$ or a descendant of V_i is in $\mathbf{Z}$.
2. If V_i acts as a **Chain** ($V_{i-1} \to V_i \to V_{i+1}$) or a **Fork** ($V_{i-1} \leftarrow V_i \to V_{i+1}$), then $V_i \notin \mathbf{Z}$.

If a path fails to meet these conditions at even a single junction (e.g., an unconditioned collider blocks the way, or a conditioned fork blocks the way), the entire path becomes an **inactive path**.

Main Idea 10

Active paths represent channels that facilitate the "communication" of statistical dependence between variables in a network. Inactive paths carry no dependence.

To visualize how these atomic structures chain together to permit or restrict the communication of dependence, consider the undirected path given in Figure 3.4.

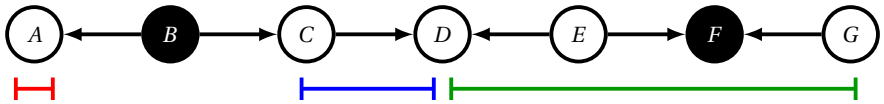


Figure 3.4: Chaining atomic structures along a path. We have conditioned on B and F. Because B is a conditioned fork, it blocks communication between A and C. Because D is an unconditioned collider, it blocks communication between C and E. Because F is a conditioned collider, it *unblocks* communication, allowing dependence to communicate freely between E and G. The three brackets identify the isolated chunks of the path that can internally communicate.

With the concept of active paths established, we can define our core structural property.

Definition 3.2.2 (D-separation and Separating Sets)**.** Given a DAG, two variables X and Y are **d-separated** given a conditioning set $\mathbf{Z}$ (denoted $X \perp\!\!\!\perp_d Y \mid \mathbf{Z}$) if there are no active paths between them. In this case, we call $\mathbf{Z}$ a **separating set**. If there is at least one active path relative to $\mathbf{Z}$, we say that X and Y are **d-connected**.

D-separation is powerful because it perfectly bridges the graphical structure of a causal DAG with the statistical properties of the probability distributions it can generate. If $X \perp\!\!\!\perp_d Y \mid \mathbf{Z}$ in the graph, then $X \perp\!\!\!\perp Y \mid \mathbf{Z}$ in the data.

Mathematical Aside: The Faithfulness Assumption

The rules of d-separation tell us that if two variables are d-separated in the graph, they are independent in the data. But what about the reverse? Under the **Faithfulness** assumption, we assume the converse is also true: if two variables are statistically independent in the data, we assume they are d-separated in the graph.
While this is usually a safe practical assumption, it is mathematically possible for two active paths to perfectly cancel each other out, yielding zero statistical association despite an active connection in the graph. Consider the following structural equations with independent noise terms (ε):

$$\begin{aligned} A &= \varepsilon_A \\ B &= A + \varepsilon_B \\ C &= A - B + \varepsilon_C \end{aligned}$$

The DAG for this system is a triangle: $A \to B$, $A \to C$, and $B \to C$. Because there are active paths between A and C, they are d-connected. However, if we substitute the equation for B into C, we get:

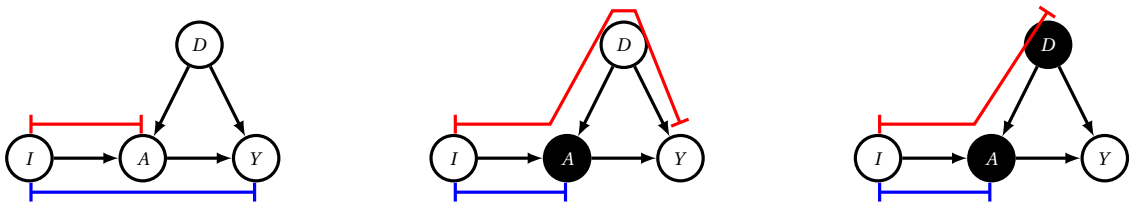
$$C = A - (A + \varepsilon_B) + \varepsilon_C = -\varepsilon_B + \varepsilon_C$$

Because the positive direct effect ($+A$) perfectly cancels out the negative indirect effect ($-A$ through B), C is entirely unaffected by A, meaning $A \perp\!\!\!\perp C$ in the data.
Such perfectly balanced equations are generally rare in nature (though this is a subject of much debate in the community). Fortunately, the faithfulness assumption is not needed for observational causal identification; it is primarily utilized later in Causal Discovery, where we attempt to learn the graph structure from the data.

3.2.1 Worked Example: Tracing Active Paths

To solidify these rules, let's trace paths through an Instrumental Variable (IV) setting, mapped in Figure 3.5 (we will discuss instrumental variables in depth later in the class). Suppose we are running an experiment where we want to measure the effect of **Treatment Compliance** (A) on an **Outcome** (Y). Patients have unmeasured **Demographics** (D) that confound the relationship. To help identify the effect, we randomize an **Intention to Treat** (I), such as an automated email reminder, which affects compliance but has no direct effect on the outcome.

Let's evaluate the relationship between Intention to Treat (I) and the Outcome (Y), using brackets to trace exactly how far dependence can communicate along the two possible routes.



(a) Unconditioned (b) Conditioned on A (c) Conditioned on A, D

Figure 3.5: Tracing active paths in an IV graph. The blue bracket traces the direct bottom route ($I \to A \to Y$), while the red bracket traces the top route via Demographics ($I \to A \leftarrow D \to Y$).

1. **Unconditioned Graph:** In Figure 3.5(a), there is an active path directly flowing via $I \to A \to Y$. A acts as an unconditioned chain along this blue route, so the path is open. I and Y are d-connected. Note that the red alternate path $I \to A \leftarrow D \to Y$ is currently blocked and stops at A, because A acts as an unconditioned collider on that specific route.

2. **Conditioning on a Collider:** In Figure 3.5(b), we condition on A. As expected, this blocks the blue direct chain, so the blue bracket stops at A. However, because A is a collider on the red route, conditioning on it inadvertently unblocks that segment. Since D acts as an unconditioned fork ($A \leftarrow D \to Y$), a brand new active path opens up, allowing the red bracket to span all the way from I to Y. I and Y remain d-connected!

3. **Achieving D-separation:** To completely d-separate I and Y, we must block all possible paths. In Figure 3.5(c), by adding D to our conditioning set, we ensure that the newly opened red path is blocked at the non-collider D. With neither bracket reaching Y, we can finally state $I \perp\!\!\!\perp_d Y \mid \{A, D\}$.

3.2.2 Feature Selection and the Markov Boundary

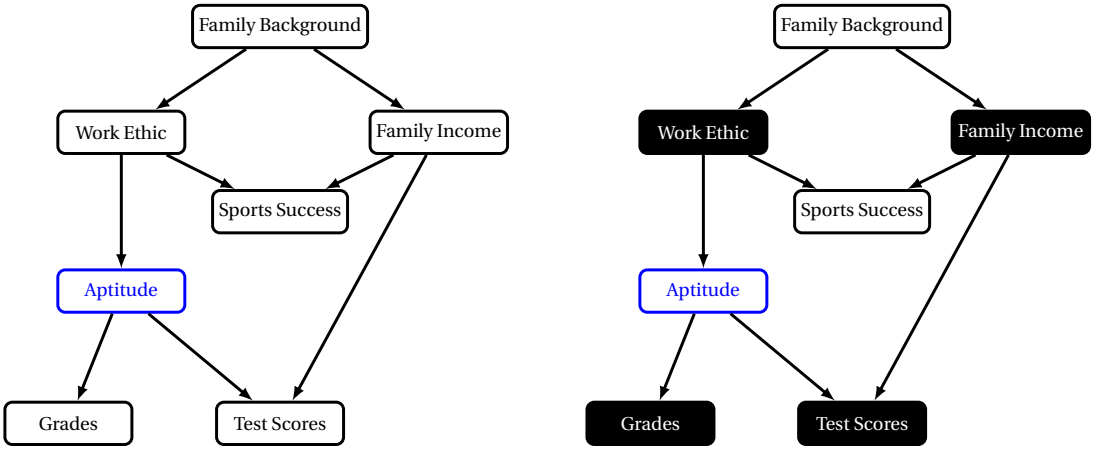
In machine learning and data science, predictive models frequently suffer from the *curse of dimensionality*. When we feed a model an excessive number of features, we give it more "chances" to improve its apparent performance on the training data by fitting to meaningless random variations. For example, in a 2-dimensional space, you cannot draw a straight 1D line perfectly through three arbitrary points. But if you introduce a third dimension (adding a covariate), you can perfectly fit a 2D plane to those three points, achieving zero error on your training data by exploiting variation that is utterly irrelevant to the true underlying phenomenon.

To combat this overfitting, we must practice feature selection: explicitly building models only on the most important variables while discarding superfluous ones. D-separation provides the perfect theoretical framework for this.

Consider a motivating example where we want to predict a Dartmouth applicant's latent **Aptitude** (Y) to make an admissions decision. We have access to several measured variables, mapped in Figure 3.6:

- **Work Ethic** (measured via a recommendation letter) is an upstream cause of Aptitude.
- **Grades** and **Test Scores** are downstream effects of Aptitude.

- **Family Income** is a cause of Test Scores (due to better access to tutoring), but does not intrinsically alter a student's true ability.
- **Family Background** is an upstream cause of both Family Income and Work Ethic.
- **Sports Success** is a downstream effect of both Work Ethic and Family Income.



(a) All Available Features

(b) The Markov Boundary for Aptitude

Figure 3.6: Causal DAG representing the Dartmouth admissions feature space. In (b), we condition on the Markov boundary (black nodes) of Aptitude, rendering the remaining variables (Family Background and Sports Success) structurally uninformative for prediction.

Obviously, we should include Work Ethic (a direct cause) and Test Scores (a direct effect) to predict Student Strength, as they provide strong, differing information about student aptitude. But what about Family Income? Interestingly, Test Scores become *even more informative* if we also know the Family Income. Since wealthy students frequently score highly regardless of intrinsic ability, accounting for Family Income allows us to "moderate" our understanding of the test score, isolating the variation that is actually driven by the student's true strength. In graphical terms, Aptitude and Family Income are d-connected if we condition on their shared child (Test Scores) in our model.

What about the remaining variables? A student's success in sports and their family background both have predictive power for aptitude initially. However, once we include Work Ethic and Family Income in our model, these variables are rendered entirely redundant. Family Background is d-separated from Aptitude by Work Ethic (which blocks the chain), and Sports Success is d-separated by Work Ethic and Family Income. Thus, they are superfluous and should be explicitly excluded from our regression.

The rules of d-separation formalize this optimal choice of features.

Definition 3.2.3 (Markov Boundary)**.** The **Markov Boundary** of a node A, denoted $\mathbf{MB}(A)$, is the minimal set of nodes such that A is d-separated from all other nodes in the network when conditioned on $\mathbf{MB}(A)$.

Once you know the values of the nodes in A's Markov boundary, no other variable in the graph can give you any more predictive information about A.

Theorem 3.2.4. *For any node A in a Bayesian network $\mathcal{G}$, $\mathbf{MB}(A)$ consists of exactly three sets of nodes:*

1. ***Its parents*** $\mathbf{PA}(A)$
2. ***Its children*** $\mathbf{CH}(A)$
3. ***Its co-parents*** $\mathbf{PA}(\mathbf{CH}(A))$

Conditioning on these nodes ensures $A \perp\!\!\!\perp_d B \mid \mathbf{MB}(A)$ for all $B \notin \{A\} \cup \mathbf{MB}(A)$.

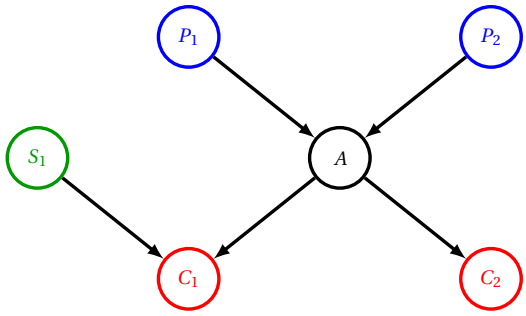


Figure 3.7: The Markov Boundary of node A (black). It consists of its parents **PA**(A), its children **CH**(A), and its co-parents **PA**(**CH**(A)). Conditioning on this specific set completely d-separates A from every other node in the graph.

Proof: Let B be an arbitrary node not in $\{A\} \cup \mathbf{MB}(A)$. We must show that every undirected path between A and B is blocked by $\mathbf{MB}(A)$. Consider the first edge on any path leaving A. This edge must either go to a parent or a child.

Case 1: The path leaves via a parent. The path begins as $A \leftarrow P \dots B$. Because $P \in \mathbf{MB}(A)$, P is conditioned on. Since arrows point away from P towards A, P must be a non-collider on this path (it acts as either a chain $A \leftarrow P \leftarrow \dots$ or a fork $A \leftarrow P \rightarrow \dots$). Conditioning on a non-collider blocks the path.

Case 2: The path leaves via a child. The path begins as $A \rightarrow C \dots B$. Because $C \in \mathbf{MB}(A)$, C is conditioned on. There are two sub-cases for C:

- **Case 2.1:** If C is a non-collider on the path (e.g., $A \rightarrow C \rightarrow \dots B$), conditioning on C immediately blocks the path.
- **Case 2.2:** If C is a collider on the path, the path must continue to one of C's parents, say D, forming $A \rightarrow C \leftarrow D \dots B$. Since D is a co-parent of A, $D \in \mathbf{MB}(A)$ and is thus conditioned on. Because the path enters D via an arrowhead (from C), it must leave D via an outgoing edge or another arrowhead, meaning D cannot be a collider for the remainder of the sequence. Therefore, D acts as a conditioned non-collider, blocking the path at D.

In all cases, every possible path from A to B is blocked by at least one node in $\mathbf{MB}(A)$. Thus, $A \perp\!\!\!\perp_d B \mid \mathbf{MB}(A)$. ■

The Markov boundary provides the ultimate mathematical tool for Rung 1 prediction, giving us the exact set of features needed to predict a variable without overfitting. However, as we know from the Ladder of Causation, predicting an outcome is fundamentally different from *changing* an outcome. A variable that is highly predictive might just be a powerful confounder or a downstream collider. To learn true causality, we must shift our objective: rather than capturing all possible associations, we need to use what we have learned to block dependence from communicating in spurious ways. In the next section, we will step up to Rung 2 and explore how to apply these exact same d-separation rules—not to maximize predictive power, but to systematically isolate pure causal effects.

3.3 Interventions and the Backdoor Criterion

Last time, we learned the graphical conditions under which variables exhibit dependence. An "active" path acts like an open channel that carries statistical dependence, and d-separation corresponds to the absence of that dependence.

It is crucial to remember that d-separation is a measure of *undirected* (or bidirectional) communication. Active paths carry statistical dependence in both directions, regardless of which way the arrowheads point. Causality, on the other hand, is strictly a *directional* flow (from cause to effect). We will now discuss how we can use our understanding of d-separation to block channels that carry non-causal dependence, thereby enabling causal conclusions from our statistics.

3.3.1 Causal vs. Backdoor Paths

When we observe dependence between a treatment A and an outcome Y, that dependence is typically a mixture of two things:

1. **Causal Pathways:** Directed paths flowing from A to Y (e.g., $A \to Y$).
2. **Backdoor Pathways:** Non-causal paths that have an arrow pointing *into* A (e.g., $A \leftarrow C \to Y$).

Notice that if we condition on the common cause C, the backdoor path is blocked! This should remind you of the concept of exchangeability from earlier lectures. We will formally connect these graphical structures to exchangeability soon, but informally, finding the true causal effect requires us to apply a covariate adjustment to a set of variables that successfully blocks all backdoor paths.

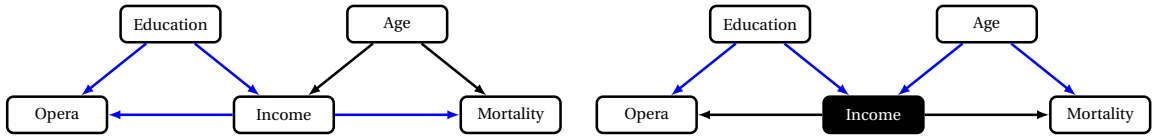
3.3.2 Motivating Causal Identification: The Opera Paradox

Before we dive into the mathematics of manipulating causal systems, we must understand *why* a rigorous graphical framework like d-separation is necessary to encourage appropriate skepticism of observational data when selecting these adjustment sets.

The New York Times recently reported on a UCL study that found an association between living longer and going to museums or the opera. At first glance, you might assume this is purely driven by an obvious common cause: income. Wealthier people partake in the arts more frequently, and they also have access to better healthcare and healthier food.

If your goal is to find the true causal effect of Opera on Mortality, a naive statistical approach would suggest simply controlling for the common cause (Income). But is controlling for income sufficient? Not quite.

Consider the possible DAG for our system given in Figure 3.8. Notice that Income is indeed a common cause of both Opera and Mortality. However, Income is also a *collider* between Education and Age (Education → Income ← Age).



(a) Confounding Paths (Unconditioned)

(b) Artificial Path (Condition on Income)

Figure 3.8: Tracing active paths in the opera DAG. In (a), Income is a common cause creating confounding paths (blue-edged forks), while Age is on a blocked path. (b) Adjusting for Income (now a conditioned black node) inadvertently opens a collider path (**Opera ← Education → Income ← Age → Mortality**, blue-edged path), creating an artificial association.

If we add Income to our conditioning set to block the obvious confounding path, we inadvertently *open* a new non-causal path! By fixing income in our data, we induce an artificial association between education and age. Suppose we restrict our data to only people with very high incomes. If a person in this bracket is very young, their high income is highly likely to be explained by education. Because these highly educated (and therefore young) people are going to the opera more often, our data will make it look like going to the opera makes you live longer!

Without a causal DAG, we might have thought that mathematically controlling for common causes would suffice to learn causality. We now know better: statistical conditioning allows dependence to communicate in non-causal, undirected ways. To extract pure causality, we need a mathematical operator that explicitly forces a directional change.

3.3.3 Do-Interventions

Now that we understand how Bayesian networks represent the natural data-generating process, we can explore what happens when we actively manipulate that process.

The most classic type of manipulation in causal inference is the **do-intervention** (often called a "hard" intervention). When we apply $\text{do}(A = a)$, we are reaching into the system and forcing the variable A to take on a specific, deterministic value a, regardless of whatever natural causes usually influence it.

Philosophical Aside: Do-Notation vs. Potential Outcomes

We now have two mathematical notations to describe the distribution yielded by an intervention:

$$\Pr(Y = y \mid \text{do}(A = a)) = \Pr(Y^{(a)} = y) \qquad \mathbb{E}[Y \mid \text{do}(A = a)] = \mathbb{E}[Y^{(a)}]$$

While they describe the exact same quantity, the underlying philosophical perspective is slightly different. The potential outcomes framework prefers to think of many possible counterfactual worlds, each corresponding to a different random potential outcome variable $Y^{(a)}$.
Pearl's do-framework, conversely, views the world as a singular system modeled using a causal graph. Interventions do not create different random variables; they simply manipulate the system's structural equations.

3.3.4 The Modularity Assumption

Recall that our Non-Parametric Structural Equation Models with Independent Errors (NPSEM-IE) are defined as a collection of autonomous structural equations: $V_i = f_i(\mathbf{PA}(V_i), N_i)$.

When we perform a hard intervention on A, we graph-mutilate the system by replacing its structural equation with the constant $A = a$. But how do we know that this surgical intervention doesn't inadvertently alter the mechanisms governing the rest of the network? This guarantee comes from the **modularity assumption** (sometimes called autonomy).

To see why modularity holds, consider the components that make up any other variable's generative mechanism, $\Pr(V_j \mid \mathbf{PA}(V_j))$ for $V_j \neq A$:

1. First, the structural equations f represent independent physical laws of nature, meaning changing f_A has no physical effect on the autonomous mechanism f_j.
2. Second, the independent background errors $N_1, \ldots, N_n$ are purely exogenous root nodes. Because causal effects only flow strictly downstream, an intervention on A cannot possibly alter the probability distribution of N_j.

Because neither the function f_j nor the error distribution of N_j is affected by the intervention on A, the conditional probability mechanism $\Pr(V_j \mid \mathbf{PA}(V_j))$ remains perfectly invariant.

3.3.5 The Truncated Factorization Formula

Because of the modularity assumption, the joint distribution of the mutilated graph—known as the interventional distribution—is identical to the original observational factorization, except the mechanism generating A is deleted.

To see exactly how an intervention differs from statistical conditioning, it is helpful to partition our variables (excluding A) into three sets relative to A:

- **N**: The **non-descendants** of A (ancestors, siblings, etc.).
- **C**: The **direct children** of A (where $A \in \mathbf{PA}(C_i)$).
- **D**: The **indirect descendants** of A (descendants further downstream, where $A \notin \mathbf{PA}(D_i)$).

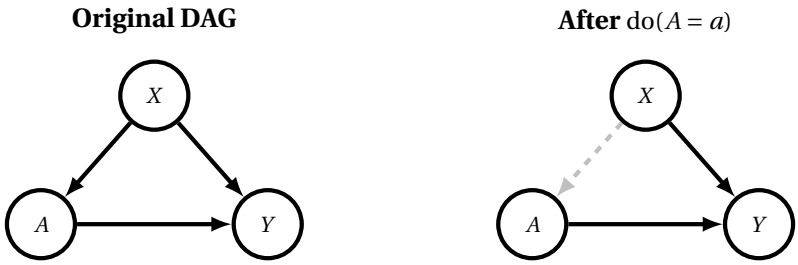


Figure 3.9: Graph mutilation under a do-intervention. **Left:** the original observational DAG, where the confounder X influences both A and Y. **Right:** after $\text{do}(A = a)$, the edge from $\mathbf{PA}(A) = \{X\}$ into A is severed (shown dashed). The variable A is now exogenously fixed at a, decoupled from its natural causes. All other structural mechanisms remain intact: X is generated by its original distribution, and $X \to Y$ and $A \to Y$ continue to operate as before. This severed-edge graph is the visual counterpart of the truncated factorization.

By the local Markov property, the standard observational factorization is:

$$\Pr(\mathbf{v}) = \underbrace{\prod_{N_i \in \mathbf{N}} \Pr(n_i \mid \mathbf{pa}(N_i))}_{\text{Non-descendants}} \times \Pr(a \mid \mathbf{pa}(A)) \times \underbrace{\prod_{C_i \in \mathbf{C}} \Pr(c_i \mid \mathbf{pa}(C_i))}_{\text{Direct Children}} \times \underbrace{\prod_{D_i \in \mathbf{D}} \Pr(d_i \mid \mathbf{pa}(D_i))}_{\text{Indirect Descendants}} \tag{3.13}$$

If we merely statistically *condition* on $A = a$, Bayes' rule dictates that the probability mass of the entire system shifts. Because dependence can communicate both directions through chains and forks, conditioning on A updates our beliefs about its ancestors $\mathbf{N}$.

$$\Pr(\mathbf{v} \mid A = a) = \prod_{N_i \in \mathbf{N}} \Pr(n_i \mid \mathbf{pa}(N_i), A = a) \times \prod_{C_i \in \mathbf{C}} \Pr(c_i \mid \mathbf{pa}(C_i))\Big|_{A=a} \times \prod_{D_i \in \mathbf{D}} \Pr(d_i \mid \mathbf{pa}(D_i)) \tag{3.14}$$

However, a **do-intervention** does not update our beliefs about the past; it creates a new physical reality going forward. By replacing $\Pr(A \mid \mathbf{PA}(A))$ with 1, the Truncated Factorization Formula yields:

$$\Pr(\mathbf{v} \mid \text{do}(a)) = \prod_{N_i \in \mathbf{N}} \Pr(n_i \mid \mathbf{pa}(N_i)) \times \prod_{C_i \in \mathbf{C}} \Pr(c_i \mid \mathbf{pa}(C_i))\Big|_{A=a} \times \prod_{D_i \in \mathbf{D}} \Pr(d_i \mid \mathbf{pa}(D_i)) \tag{3.15}$$

This partition makes the mechanism of causality explicitly clear. Because a node's structural equation relies *only* on its direct parents, the only mathematical terms in the entire system that are syntactically altered by the intervention are the direct children $\mathbf{C}$. The upstream variables $\mathbf{N}$ are physically generated before A and remain unchanged. The indirect descendants $\mathbf{D}$ only "feel" the intervention indirectly as the new values cascade down the causal chain through the children.

It is worth noting that this new truncated factorization is exactly the factorization we would see in a *new graph* where we have performed "mutilation" by removing the edges from $\mathbf{PA}(A)$ to A. See Figure 3.9 for an example. You might find this familiar, as the graphical impact of a do intervention is identical to that of randomization.

3.3.6 The Backdoor Criterion

Judea Pearl formalized this graphical intuition into the **Backdoor Criterion**.

As established earlier, to isolate pure causality, our goal is to systematically block all non-directed channels (spurious paths) so that the only dependence left to measure is the causal dependence flowing safely along the directed paths. We do this by blocking every **backdoor path**—paths that sneak in through the "back door" of the treatment node (pointing into A) to create undirected, spurious correlations.

Main Idea 11

A set of variables $\mathbf{X}$ satisfies the **backdoor criterion** with respect to a treatment A and an outcome Y in a DAG if:

1. No node in $\mathbf{X}$ is a descendant of A (we do not condition on effects of the treatment).
2. $\mathbf{X}$ blocks every backdoor path from A to Y (using standard d-separation rules).

If a set $\mathbf{X}$ satisfies the backdoor criterion, then conditioning on $\mathbf{X}$ successfully isolates the causal effect: $\Pr(y \mid \text{do}(a), \mathbf{x}) = \Pr(y \mid a, \mathbf{x})$.

However, our goal is usually to compute the global marginal interventional distribution, $\Pr(y \mid \text{do}(a))$. By the law of total probability, we must marginalize over the strata of $\mathbf{X}$ in the interventional world:

$$\Pr(y \mid \text{do}(a)) = \sum_{\mathbf{x}} \Pr(y \mid \text{do}(a), \mathbf{x}) \Pr(\mathbf{x} \mid \text{do}(a))$$

This is exactly where the truncated factorization comes to our rescue! Because the first rule of the backdoor criterion explicitly requires that $\mathbf{X}$ contains *no descendants* of A, the variables in $\mathbf{X}$ belong entirely to the non-descendant set $\mathbf{N}$. As we proved via the truncated factorization, the marginal probability distribution of non-descendants is perfectly invariant to the intervention because they are generated before the intervention occurs.

Therefore, $\Pr(\mathbf{x} \mid \text{do}(a)) = \Pr(\mathbf{x})$. Substituting this along with our isolated causal effect into the marginalization yields the famous **Backdoor Adjustment Formula**:

$$\Pr(y \mid \text{do}(a)) = \sum_{\mathbf{x}} \Pr(\mathbf{x}) \Pr(y \mid a, \mathbf{x})$$

This proves that if we can identify a valid graphical adjustment set, we can compute the exact effects of a counterfactual intervention using only observational data.

3.3.7 Back to Operas

Now that we have formally defined the Backdoor Criterion, let us return to the Opera Paradox from Figure 3.8. What sets of variables actually satisfy the Backdoor Criterion for A = Opera and Y = Mortality?

Testing several candidate adjustment sets:

- **{Education}:** Fails. Leaves the confounding path Opera ← Income → Mortality active.
- **{Age}:** Fails. Leaves the confounding path Opera ← Education → Income → Mortality active.
- **{Income}:** Fails. While it blocks the obvious confounding paths, Income is a collider on the path Opera ← Education → Income ← Age → Mortality. Conditioning on it opens this spurious path.
- **{Education, Income}:** Valid! Blocks all obvious confounding paths and successfully re-closes the artificial path opened by the Income collider.
- **{Age, Income}:** Valid! Similarly blocks all backdoor paths and successfully re-closes the artificial path opened by the Income collider.

Thus, an adjustment set does not have to be unique. Both of these valid sets will allow us to successfully compute the backdoor adjustment.

3.3.8 A Simple Adjustment Set

Finding a valid adjustment set in a massive graph might seem daunting, but there is a simple (though not necessarily minimal) adjustment set that is structurally guaranteed to work, provided it is fully observed.

Lemma 3.3.1. *For any DAG $\mathcal{G} = (\mathbf{V}, \mathbf{E})$, the set of direct parents of the treatment node, $\mathbf{PA}(A)$, always satisfies the backdoor criterion for any outcome Y.*

Proof: To prove this, we must verify the two conditions of the backdoor criterion:

1. **No Descendants:** By the definition of a Directed Acyclic Graph, a parent of A cannot also be a descendant of A, as this would create a cycle. Therefore, $\mathbf{PA}(A)$ contains no descendants of A.
2. **Blocks all backdoor paths:** By definition, every backdoor path from A to Y must begin with an arrow pointing into A (i.e., $A \leftarrow P \dots Y$). The node P emitting this arrow is, by definition, a parent of A ($P \in \mathbf{PA}(A)$). Because the path leaves P and goes to A, P acts as either a chain ($A \leftarrow P \leftarrow \dots$) or a fork ($A \leftarrow P \rightarrow \dots$) along this specific route. In both cases, P is a non-collider. Since we are conditioning on $\mathbf{PA}(A)$, we are conditioning on this non-collider, which immediately blocks the path at its very first node.

Because every possible backdoor path must pass through at least one parent, conditioning on all parents simultaneously blocks every backdoor path. ■

While $\mathbf{PA}(A)$ is a universally valid adjustment set, it is often impractical in real-world studies because we rarely observe *every* direct parent of a treatment variable. This is why the generalized backdoor criterion is so powerful: it allows us to find alternative sets of variables further up the causal chain that can still block all backdoor paths.

3.4 Single World Intervention Graphs (SWIGs)

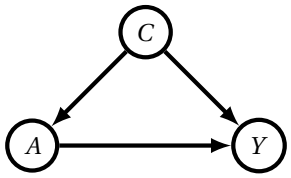
3.4.1 Bridging Graphs and Potential Outcomes

Up until now, we have largely used Judea Pearl's $\text{do}(\cdot)$ notation to represent interventions on a causal graph. However, earlier in this text, we introduced the concept of **potential outcomes** (e.g., $Y^{(a)}$), and we learned that the key to isolating causal effects was achieving **exchangeability**: $Y^{(a)} \perp\!\!\!\perp A | \mathbf{Z}$ for some set of covariates $\mathbf{Z}$. Can we evaluate exchangeability directly on a causal graph?

The "causal Markov condition" assumes that d-separation rules tell us when observed variables are independent. If we want to use d-separation to determine exchangeability, we need a way to physically represent potential outcome variables—like $Y^{(a)}$—on our DAGs.

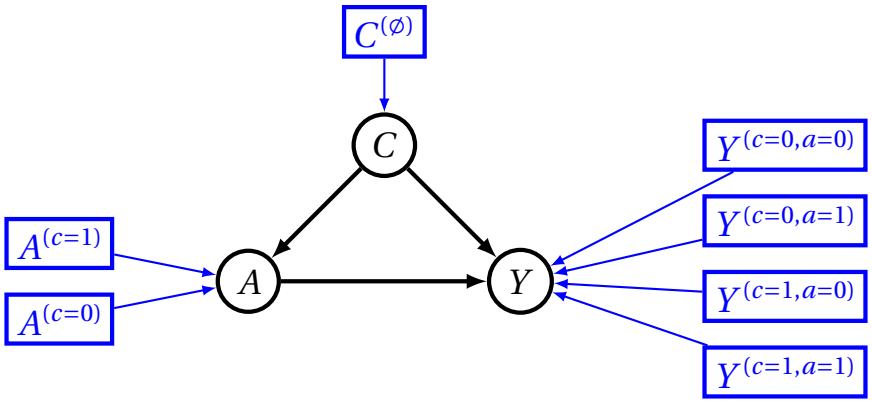
3.4.2 Recursive Substitution and NPSEM-IE

Let's start with a familiar DAG on binary variables: an action A, an outcome Y, and a confounder C.



If we assume a Non-Parametric Structural Equation Model with Independent Errors (NPSEM-IE), then each variable is determined by its parents and an independent source of background noise. This means the *one-step-ahead* potential outcomes—the value a variable would take if we intervened directly on its parents—are independent of each other.

For our binary graph, the one-step-ahead potential outcomes are $C^{(\emptyset)}$, $A^{(c=0)}$, $A^{(c=1)}$, $Y^{(c=0,a=0)}$, $Y^{(c=0,a=1)}$, $Y^{(c=1,a=0)}$, and $Y^{(c=1,a=1)}$. We could, technically, draw a massive graph with all of these independent background variables pointing into our observed variables:



But what if we only want to intervene on the action A, and not on the condition C? We want to find the potential outcome $Y^{(a)}$ (not $Y^{(c,a)}$). To find it, we must ask: **what are its parents?**

When an intervention does not include every parent of an outcome, we use **recursive substitution** to figure out exactly what inputs should be fed into the structural equation. Let's look at the natural structural equation for Y:

$$Y = f_Y(A, C, \varepsilon_Y)$$

When we intervene to set $A = a$, the inputs to this function change. A becomes the fixed constant a, and C becomes its own potential outcome under the intervention, $C^{(a)}$:

$$Y^{(a)} = f_Y(a, C^{(a)}, \varepsilon_Y)$$

By definition, the one-step-ahead potential outcome where we force the parents to specific values is written with those values in the superscript. Therefore, we can rewrite $f_Y(a, C^{(a)}, \varepsilon_Y)$ as $Y^{(a,C^{(a)})}$.

Now, we evaluate the upstream variables to resolve those parents. Because A is not an ancestor of C, causality dictates that intervening on A physically cannot change C. The counterfactual $C^{(a)}$ is exactly equal to the natural, observed C. By substituting the natural C into our equation, we arrive at:

$$Y^{(a)} = Y^{(a,C^{(a)})} = Y^{(a,C)} \tag{3.16}$$

But what does $Y^{(a,C)}$ actually mean? The superscript represents a forced intervention. Thus, $Y^{(a,C)}$ describes a world where we force $A = a$, and we simultaneously *force* C to take its naturally occurring value. Because forcing a variable to take the exact value it would have naturally taken anyway is physically indistinguishable from simply leaving it alone, $Y^{(a,C)}$ is mathematically equivalent to our target potential outcome, $Y^{(a)}$.

Thus, we have successfully determined the parents of $Y^{(a)}$: it is generated by first sampling the natural state of C, and then evaluating the one-step-ahead outcome under the intervention a and that natural state C.

3.4.3 Node Splitting

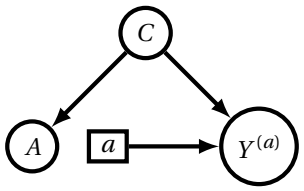
Notice what happened mathematically during recursive substitution: the mechanism generating the natural value of A (its dependency on C) was completely decoupled from the mechanism generating the outcome $Y^{(a)}$. The outcome Y only "sees" the forced intervention a. The natural, one-step-ahead potential outcome ($A^{(c)}$) still exists in the background—it represents what A *would have been*—but its downstream effects are severed.

Instead of drawing a massive graph of unobserved background noise or performing tedious algebraic substitution, Thomas Richardson and Jamie Robins developed a graphical tool that perfectly mirrors this decoupling: **node splitting** [Richardson and Robins, 2013].

When we intervene on a variable A, we literally split the node in two to physically represent its severed roles in the mutilated system:

1. **The receiving end (The Natural State):** A random variable representing what A *would have been* assigned by its parents. This captures the one-step-ahead potential outcome (e.g., $A^{(c)}$). Because C is unaffected by the intervention, this simplifies to just A.
2. **The transmitting end (The Intervened State):** The fixed, deterministic value (a) that we force upon the system. All downstream descendants now listen *only* to this fixed node.

By splitting the node on the graph, we create a **Single World Intervention Graph (SWIG)**.



In this SWIG, we can visually evaluate exchangeability! If we look at the graph, the only path between the natural action A and the potential outcome $Y^{(a)}$ is $A \leftarrow C \rightarrow Y^{(a)}$. If we condition on C, we block this path. Therefore, $Y^{(a)} \perp\!\!\!\perp_d A \mid C$. We have achieved conditional exchangeability!

Main Idea 12

Rules for constructing a SWIG for intervention a:

1. **Split:** Separate the action node A into a random node A and a fixed intervention node a.
2. **Rewire:** All incoming edges (parents) remain attached to the random node A. All outgoing edges (children) are moved to the fixed node a.
3. **Rename:** Any node that is a descendant of the fixed node a (including a itself) is now a potential outcome, so we append the superscript (a) to its name.

Once the SWIG $\mathcal{G}(a)$ is constructed, the standard rules of d-separation apply, giving us three important properties:

1. **Markov Property:** If Y is d-separated from Z given W in the original graph $\mathcal{G}$, then $\Pr(Y \mid Z, W) = \Pr(Y \mid W)$.
2. **Generalized Ignorability:** If $Y^{(a)}$ is d-separated from A given $\mathbf{W}^{(a)}$ in the SWIG $\mathcal{G}(a)$, then we have conditional exchangeability: $Y^{(a)} \perp\!\!\!\perp A | \mathbf{W}^{(a)}$. By the consistency assumption, this implies that $\Pr(Y^{(a)} | \mathbf{W}^{(a)}) = \Pr(Y \mid \mathbf{W}, A = a)$.
3. **Causal Irrelevance:** If a has no directed path to $Y^{(a)}$ in the SWIG $\mathcal{G}(a)$, then $\Pr(Y^{(a)}) = \Pr(Y)$.

Mathematical Note: Proving Generalized Ignorability

You might be wondering how the counterfactual variables in the SWIG ($Y^{(a)}$ and $\mathbf{W}^{(a)}$) suddenly transform into our standard observed variables (Y and $\mathbf{W}$) in the Generalized Ignorability property. This relies entirely on combining exchangeability with the **Consistency Assumption**.
If the SWIG tells us that $Y^{(a)} \perp\!\!\!\perp A \mid \mathbf{W}^{(a)}$, we can arbitrarily condition on $A = a$ without changing the probability (exchangeability):

$$\Pr(Y^{(a)} \mid \mathbf{W}^{(a)}) = \Pr(Y^{(a)} \mid \mathbf{W}^{(a)}, A = a)$$

Now that we are mathematically restricted to the subset of the population where the observed treatment A matches our intervention a, the consistency assumption activates. For these individuals, *all* potential outcomes under intervention a perfectly match their observed reality. This means $Y^{(a)} = Y$, and crucially, $\mathbf{W}^{(a)} = \mathbf{W}$ (even if $\mathbf{W}$ contains descendants of A!).

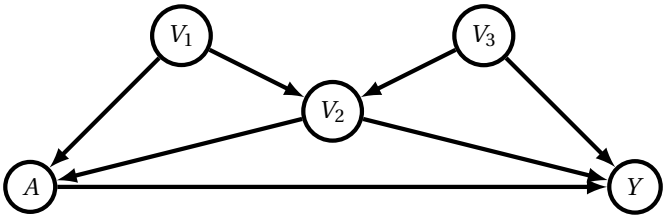
Therefore, we can drop the (a) superscripts entirely:

$$\Pr(Y^{(a)} \mid \mathbf{W}^{(a)}, A = a) = \Pr(Y \mid \mathbf{W}, A = a)$$

By stringing these equalities together, we successfully identify a counterfactual probability using purely observational data.

3.4.4 Back to Operas

Let's practice recursive substitution and node-splitting on a more complex system. In fact, this is exactly the opera example from earlier (Figure 3.8):



First, let's look at the recursive substitution for the potential outcome $Y^{(a)}$. To clean up the notation, we will denote the potential outcome inputs as function arguments (e.g., $Y^{(a)} = Y(a)$). In our DAG, the direct parents of Y are A, V_2, and V_3. Thus, the one-step-ahead potential outcome is:

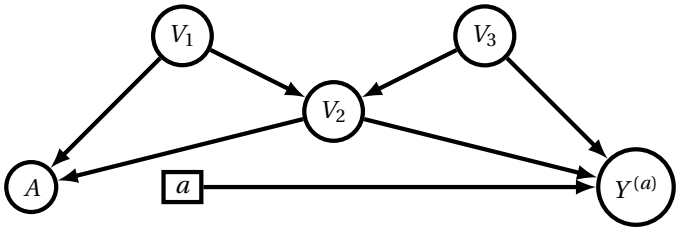
$$Y(a) = Y(a, V_2^{(a)}, V_3^{(a)})$$

Now we evaluate the upstream variables. Because V_2 and V_3 are not descendants of A, intervening on A physically cannot change them:

$$Y(a) = Y(a, V_2, V_3)$$

Notice that this algebra proves exactly what our SWIG rules tell us visually: because V_2 and V_3 (and V_1 further upstream) are not affected by the intervention on A, they do not get an (a) superscript. Thomas Richardson and James Robins invented SWIGs precisely so we can skip this tedious recursive substitution and jump straight to a graph!

Let's construct that SWIG by applying our node-splitting rules:



By evaluating d-separation on this SWIG, we can quickly check for exchangeability. For instance, is $Y^{(a)} \perp\!\!\!\perp_d A \mid V_2$? If we condition on V_2, we block the paths $A \leftarrow V_2 \rightarrow Y^{(a)}$ and $A \leftarrow V_1 \rightarrow V_2 \rightarrow Y^{(a)}$. However, conditioning on V_2 actually *opens* the path $A \leftarrow V_1 \rightarrow V_2 \leftarrow V_3 \rightarrow Y^{(a)}$ because V_2 acts as a collider on that specific route!

Therefore, in this SWIG, $Y^{(a)} \not\perp\!\!\!\perp_d A \mid V_2$. Conditioning on V_2 alone does not give us exchangeability. However, the sets $\{V_1, V_2\}$ and $\{V_2, V_3\}$ are both valid backdoor adjustment sets, and equivalently give us conditional exchangeability in the corresponding SWIG.

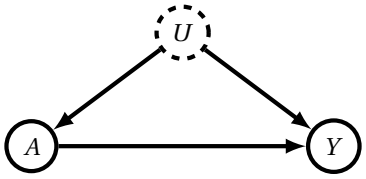
3.5 Unobserved Confounding and Latent Projections

3.5.1 The Problem of Unobserved Variables

Up until this point, we have operated under a highly optimistic assumption: *causal sufficiency*. Causal sufficiency assumes that every relevant variable in our system—particularly every common cause—has been measured and is included in our dataset. In the previous section, we learned that we can isolate causal effects by conditioning on a backdoor adjustment set. But what happens if a crucial variable in that set is unmeasured?

Consider one of the most famous causal debates of the 20th century: does smoking (A) cause lung cancer (Y)? For a long time, a common argument was that genetic factors might predispose certain people to both develop lung cancer *and* have a physiological desire to smoke.

This idea might sound ridiculous to us today, but Sir Ronald Fisher—the famous statistician who formalized randomized controlled trials—fervently believed this theory and was highly skeptical of the causal link between smoking and cancer. Because genome sequencing was impossible at the time, this genetic factor was a purely unobserved variable, which we will denote as U.



In this graph, the backdoor path $A \leftarrow U \rightarrow Y$ is completely open. Because U is unobserved, we cannot include it in a backdoor adjustment set. Consequently, if this is the true causal structure, the causal effect of A on Y is formally unidentifiable from observational data alone.

A causal graph describes a system, but it would be horribly inconvenient (and practically impossible) to model the entire world, including every unmeasured genetic or background factor. We need to understand when we can take a subset of observed variables and handle them in a vacuum, and how to represent unobserved variables when they cannot be ignored.

Ignoring Descendants We can always safely ignore the downstream effects (descendants) of our system of interest. Let's look at a simple example where we only wish to study a subset of variables. Assume a topological ordering of the entire system $W_1, \dots, W_n$ (meaning every parent comes before its children). We want to study the first m variables (where $m < n$).

Let $\mathbf{V} = \{W_1, \dots, W_m\}$ represent the **V**isible variables, and let $\mathbf{U} = \{W_{m+1}, \dots, W_n\}$ represent the **U**nobserved variables. Together, they form the full set of variables $\mathbf{W} = \mathbf{V} \cup \mathbf{U}$.

Because of the topological ordering, any variable in $\mathbf{V}$ can only have parents that appeared earlier in the sequence. Therefore, it is guaranteed that the parents of any visible variable are strictly contained within the visible set $\mathbf{V}$. The unobserved variables $\mathbf{U}$ can only be descendants (or irrelevant), never parents of $\mathbf{V}$.

Applying the law of total probability and the chain rule on the visible variables $\mathbf{v}$:

$$\Pr(\mathbf{v}) = \sum_{\mathbf{u}} \Pr(\mathbf{v}, \mathbf{u}) = \sum_{\mathbf{u}} \Pr(\mathbf{u} \mid \mathbf{v}) \prod_{i=1}^{m} \Pr(v_i \mid v_1, \dots, v_{i-1})$$

Because we established that the actual parents of v_i are already fully captured within $\{v_1 \dots, v_{i-1}\}$, we can simplify the conditional probability using the local Markov property:

$$\Pr(\mathbf{v}) = \sum_{\mathbf{u}} \Pr(\mathbf{u} \mid \mathbf{v}) \underbrace{\prod_{v_i \in \mathbf{v}} \Pr(v_i \mid \mathbf{PA}^{\mathcal{G}}(v_i))}_{\text{not dependent on } \mathbf{u}} = \prod_{v_i \in \mathbf{v}} \Pr(v_i \mid \mathbf{PA}^{\mathcal{G}}(v_i)) \underbrace{\sum_{\mathbf{u}} \Pr(\mathbf{u} \mid \mathbf{v})}_{=1} = \prod_{v_i \in \mathbf{v}} \Pr(v_i \mid \mathbf{PA}^{\mathcal{G}}(v_i))$$

Crucially, the factorization according to $\mathcal{G}$ on $V_1, \dots, V_m$ still perfectly applies! Unobserved descendants do not break our graphical assumptions.

When the NPSEM-IE Assumption Fails Conversely, we cannot willy-nilly ignore vertices that *precede* our system (like the genetic confounder in the smoking example), because they make up the parent sets of our generative distribution.

Recall from our earlier discussion on Bayesian Networks that we often rely on the Non-Parametric Structural Equation Model with Independent Errors (NPSEM-IE). The NPSEM-IE framework assumes that all unobserved background variables are independent and point to only a *single* observed variable, allowing us to treat them as independent noise: $A = f_A(\mathbf{PA}(A), \varepsilon_A)$.

But what happens when U causes both A and Y? If we attempt to restrict our equations only to the visible variables, U gets fully absorbed into the background noise terms:

$$A = f_A(\varepsilon_A, U) \quad \Longrightarrow \quad A = f_A^*(\varepsilon_A^*)$$
$$Y = f_Y(A, \varepsilon_Y, U) \quad \Longrightarrow \quad Y = f_Y^*(A, \varepsilon_Y^*)$$

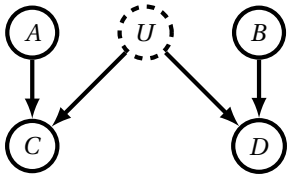
Because U is hiding inside both ε_A^* and ε_Y^*, our new background error terms are no longer independent! They will fluctuate together. When the NPSEM-IE assumption fails due to unmeasured common causes, we say our system has **correlated errors**, and a standard DAG is no longer mathematically sufficient to describe it.

3.5.2 Acyclic Directed Mixed Graphs (ADMGs)

Suppose we want to model a causal system that is *not* an NPSEM-IE. We want a way to go from a graph with unobserved confounding nodes, $\mathcal{G}(\mathbf{V}, \mathbf{U})$, to a new graph defined strictly on the observed nodes, $\mathcal{G}(\mathbf{V})$, with the following properties:

1. **Preserve causal information:** V_i causes V_j in $\mathcal{G}(\mathbf{V})$ if and only if V_i causes V_j in $\mathcal{G}(\mathbf{V}, \mathbf{U})$.
2. **Preserve statistical information:** d-separation holds in $\mathcal{G}(\mathbf{V})$ if and only if it holds in $\mathcal{G}(\mathbf{V}, \mathbf{U})$.

To do this, we must make our graph "mixed" by introducing bidirected edges to represent the correlated errors. Consider the following graph:



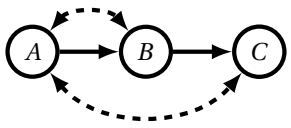
Can you draw a standard DAG on just the observed variables A, B, C, D that satisfies all d-separation properties of the original? The answer is no. We clearly need to connect C and D to account for the unobserved confounder U. But if we add a directed edge ($C \to D$ or $D \to C$), we alter the d-separation properties and introduce a false causal dependence.

Main Idea 13

An **Acyclic Directed Mixed Graph (ADMG)** $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ is a graph where:

- $\mathbf{E}$ contains only directed ($\to$) and bidirected ($\leftrightarrow$) edges.
- $\mathcal{G}$ has at most two edges between two vertices: one directed and one bidirected.
- There are no directed cycles.

The following is a valid ADMG:



3.5.3 Latent Projection

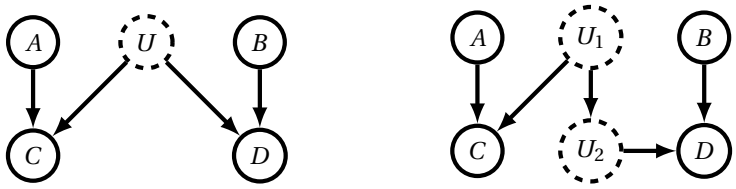
We can systematically construct an ADMG $\mathcal{G}(\mathbf{V})$ from a DAG with unobserved variables $\mathcal{G}(\mathbf{V},\mathbf{U})$ ($\mathbf{V}$ for "visible" and $\mathbf{U}$ for "unobserved") using a process called **latent projection**. The rules are simple:

1. Start with a base subgraph containing only the observed nodes $\mathbf{V}$.
2. If there is a directed path $V_i \to \ldots \to V_j$ in $\mathcal{G}(\mathbf{V},\mathbf{U})$ where all intermediary vertices are in $\mathbf{U}$, add a directed edge $V_i \to V_j$ to $\mathcal{G}(\mathbf{V})$.
3. If there is a diverging path $V_i \leftarrow \ldots \to V_j$ in $\mathcal{G}(\mathbf{V},\mathbf{U})$ where all intermediary vertices are in $\mathbf{U}$ and there are no colliders along the path, add a bidirected edge $V_i \leftrightarrow V_j$ to $\mathcal{G}(\mathbf{V})$.

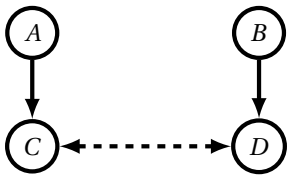
Notice the critical caveat in Rule 3: *no colliders along the path.* If two observed variables are connected entirely through unobserved variables, but one of those unobserved variables is a collider (e.g., $V_i \leftarrow U_1 \to U_c \leftarrow U_2 \to V_j$), no bidirected edge is drawn directly between V_i and V_j. Why? Because colliders naturally block the flow of dependence! If the unobserved collider U_c is strictly downstream of our observed variables and has no observed descendants, it falls under our earlier rule for "ignoring descendants" and disappears entirely.

However, if that unobserved collider U_c *does* have an observed descendant V_{desc} (i.e., $U_c \to V_{desc}$), we cannot simply ignore it. The paths from U_1 and U_2 flow directly down into V_{desc}. By applying our latent projection rules, this creates two separate bidirected edges: $V_i \leftrightarrow V_{desc}$ and $V_{desc} \leftrightarrow V_j$. Notice what just happened: V_{desc} now has two arrowheads pointing into it. The latent projection mathematically ensures that two directed paths to an observed descendant naturally form a collider in the visible graph!

It is also worth noting that different DAGs may have the same latent projection. Consider these two DAGs:



These two DAGs both project to the exact same ADMG, perfectly capturing the unobserved confounding without cluttering the graph:



3.5.4 M-Separation

M-separation in an ADMG is exactly analogous to d-separation in a standard DAG. You can simply treat the bidirected edge $V_i \leftrightarrow V_j$ as a hidden common cause $V_i \leftarrow U \to V_j$.

Active paths between A and B are defined by the exact same structural rules as d-separation, but chains and colliders can now include bidirected arrows:

- **Chains** ($\to V \to$, $\leftrightarrow V \to$, $\leftarrow V \leftrightarrow$): Open by default, closed by conditioning on V.
- **Forks** ($\leftarrow V \to$): Open by default, closed by conditioning on V.
- **Colliders** ($\to V \leftarrow$, $\leftrightarrow V \leftarrow$, $\leftrightarrow V \leftrightarrow$): Closed by default, opened by conditioning on V or any of its descendants.

Notice that a collider happens anytime two arrowheads meet at a node, regardless of whether the edge is directed or bidirected.

Main Idea 14

Latent projections preserve the conditional independencies of the underlying system. For variables $A, B \in \mathbf{V}$ and $\mathbf{C} \subset \mathbf{V} \setminus \{A, B\}$:

$$A \perp\!\!\!\perp_d^{\mathcal{G}(\mathbf{V},\mathbf{U})} B \mid \mathbf{C} \iff A \perp\!\!\!\perp_m^{\mathcal{G}(\mathbf{V})} B \mid \mathbf{C}$$

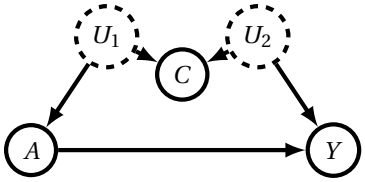
Just as d-separation stands for *directed* separation, m-separation stands for *mixed* separation because it applies to mixed graphs.

3.5.5 M-Bias

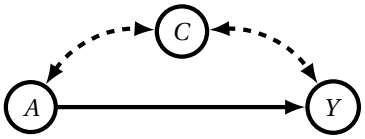
While m-separation is a general rule for evaluating paths in mixed graphs, the concept is most famous for its role in a classic causal structure known as the **M-Graph** (named because the nodes and edges visually form the letter "M").

Consider a scenario where we want to find the effect of an Action (A) on an Outcome (Y). Both A and Y are influenced by different, completely unobserved confounders (U_1 and U_2). However, both of these unobserved confounders also happen to cause a third observed variable, C.

If this structure sounds familiar, it should! This is the exact same topological path that caused our artificial association in the Opera Paradox (Figure 3.8). In that example, Education (U_1) and Age (U_2) were the diverging causes, while Income (C) was the central collider. The only difference here is that U_1 and U_2 are completely unobserved.



Underlying DAG



Latent Projection (ADMG)

In the underlying DAG, is the backdoor path from A to Y open? Tracing the path, we find $A \leftarrow U_1 \rightarrow C \leftarrow U_2 \rightarrow Y$. Because C is a collider on this path, the path is naturally blocked! A and Y are independent of confounding without adjusting for anything.

If we look at the ADMG, M-separation gives us the exact same result: $A \leftrightarrow C \leftrightarrow Y$. Because two arrowheads meet at C, it is a collider, and the path is closed. This brings us to a critical realization. In observational disciplines, researchers often operate under a widespread fallacy: *"If you gather data from before treatment, nothing is a descendant of A, so you don't need to worry about colliders."*

The logic dictates that controlling for pre-treatment baseline covariates is always a safe way to "reduce variance" or "compare apples to apples." Yes, measuring a variable before treatment guarantees that it is not a descendant of A or Y, meaning it cannot be a direct collider *between* the treatment and the outcome. However, this does not preclude M-Bias! As we see in the M-Graph, a variable measured before the treatment can still easily act as a collider between two *unobserved* upstream causes.

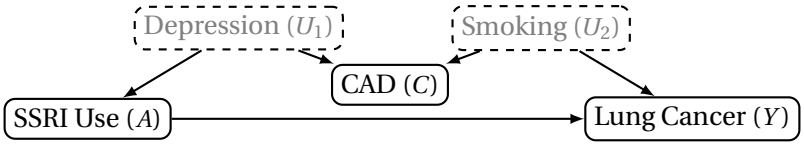
If we assumed causal sufficiency (meaning we had actually measured U_1 and U_2), this heuristic would be mostly harmless. Throwing the full set of variables $\{C, U_1, U_2\}$ into a regression model would be perfectly fine—even though conditioning on C opens the path, simultaneously conditioning on U_1 and U_2 blocks it again.

However, when dealing with unobserved variables, this pre-treatment heuristic is dangerously wrong. If a researcher sees that a pre-treatment variable C is correlated with both A and Y and mistakenly assumes it is a standard confounder, they will blindly *condition* on it. Doing so inadvertently opens the collider path, engineering a spurious non-causal dependence between A and Y!

Because U_1 and U_2 are missing, the researcher cannot condition on them to close the path back up. Along with standard post-treatment colliders, M-Bias proves that blindly conditioning willy-nilly on every variable you measure is a dangerous practice that can actively manufacture bias where none previously existed.

Example 1: The Clinical Trap (Antidepressants and Lung Cancer)

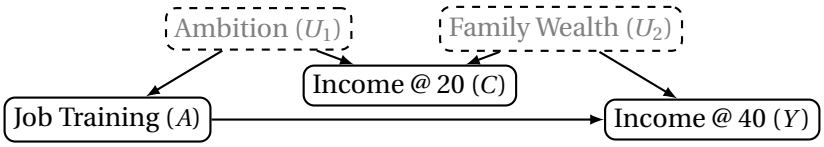
M-bias is a massive threat to retrospective medical studies using Electronic Health Records. Imagine researchers want to test if SSRIs (A) cause Lung Cancer (Y). To "compare apples to apples," they adjust for Coronary Artery Disease (C), a very common pre-treatment baseline condition.



Because "Depression severity" and "Smoking history" are notoriously underreported or missing in medical databases, they act as unobserved variables. In the real world, SSRIs and Lung Cancer are entirely independent. But by controlling for the collider (CAD), the researchers mathematically force a negative correlation between depression and smoking, artificially making it look like SSRI use drives Lung Cancer!

Example 2: The Policy Trap (Job Training and Future Earnings)

Economists evaluating policy face the same trap. Suppose we want to measure the effect of an optional Job Training program (A) on Future Earnings at age 40 (Y). Econometric tradition insists on controlling for Baseline Income at age 20 (C) to ensure we aren't just measuring pre-existing wealth.



Income at age 20 is clearly recorded before the job training program (A) and before the outcome (Y), making it a textbook pre-treatment covariate. However, it still falls prey to this exact M-bias issue. "Personal Ambition" (U_1) and "Inherited Family Wealth" (U_2) are unobserved, but both drive a 20-year-old's Baseline Income. By controlling for Baseline Income, researchers open the collider. They accidentally create an inverse correlation between ambition and family wealth for a given income bracket, which completely biases the estimated effect of the job training program.

3.6 Frontdoor Adjustment

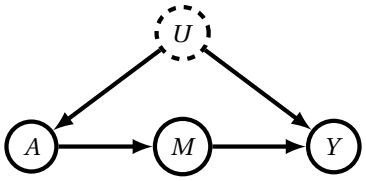
3.6.1 Defeating the Unobserved Confounder

In the previous section, we discussed the problem of unobserved confounding, famously illustrated by Sir Ronald Fisher's skepticism regarding smoking (A) and lung cancer (Y). Recall that Fisher hypothesized an unobserved genetic factor (U) that predisposed people to both smoke and develop cancer. Because we cannot observe U, the backdoor path $A \leftarrow U \rightarrow Y$ remains wide open. In our ADMG, this is represented by a bidirected edge $A \leftrightarrow Y$.

If this is our entire model, the causal effect of smoking on cancer is formally *unidentifiable* from observational data. We cannot use the Backdoor Criterion.

However, the answer to the smoking debate came through the isolation of a causal mechanism: tar buildup in the lungs. Tar became the mediator (M) that carried the effect of smoking to lung cancer. Tar is clearly caused by smoking and is very unlikely to be caused by underlying genetic factors.

By identifying a mechanism that the unobserved confounder does *not* touch, we get the classic **Frontdoor Graph**:



Judea Pearl discovered that in this specific scenario, we can identify the causal effect of A on Y without observing U. Instead of blocking the backdoor, we pass through the "front door" (M).

3.6.2 Deriving the Frontdoor Adjustment

The intuition behind the Frontdoor Adjustment is that we can sequentially compose the isolated causal effect of A on M with the isolated causal effect of M on Y. It is essentially the backdoor adjustment applied twice!

1. **Effect of A on M:** Because there are no open backdoor paths from A to M, the observational probability is identically the causal probability:

$$\Pr(m \mid \text{do}(a)) = \Pr(m \mid a)$$

2. **Effect of M on Y:** To find the causal effect of M on Y, we must block the backdoor path $M \leftarrow A \leftarrow U \rightarrow Y$. We can do this by using A as a backdoor adjustment set! Summing over all possible values of A (denoted as a' to distinguish it from our specific intervention a), we get:

$$\Pr(y \mid \text{do}(m)) = \sum_{a'} \Pr(y \mid m, a') \Pr(a')$$

3. **Combining them:** We piece these two mechanisms together by summing over all possible states of the mediator m, multiplying the chance that our intervention a causes m by the chance that m causes y.

This gives us the **Frontdoor Adjustment Formula**:

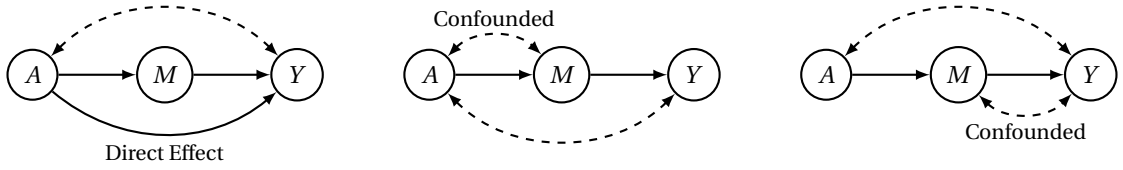
$$\Pr(y \mid \text{do}(a)) = \sum_{m} \Pr(m \mid a) \sum_{a'} \Pr(y \mid m, a') \Pr(a')$$

3.6.3 The Frontdoor Criterion

Generalizing from the smoking example, a specific set of rigorous graphical rules dictates exactly when this formula can be safely applied to a system.

> **Main Idea 15**
>
> A set of variables **M** satisfies the **frontdoor criterion** relative to treatment A and outcome Y if:
>
> 1. **Complete Mediation:** **M** intercepts all directed paths from A to Y.
> 2. **Unconfounded Treatment-Mediator Link:** There are no unblocked backdoor paths from A to **M**.
> 3. **Blocked Mediator-Outcome Paths:** All backdoor paths from **M** to Y are blocked by A.

Let's look at some systems where these rules fail. Figure 3.10 demonstrates three graphs where the frontdoor adjustment cannot be used to identify the causal effect of A on Y.



(a) Rule 1 Violated

M does not completely intercept the effect of A.

(b) Rule 2 Violated

The $A \to M$ link has an open backdoor path.

(c) Rule 3 Violated

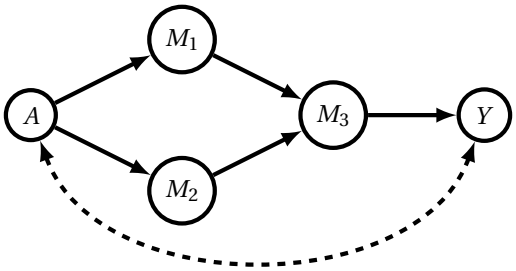
The $M \to Y$ backdoor path is not blocked by A.

Figure 3.10: Examples of causal graphs that fail the Frontdoor Criterion. In all three cases, the frontdoor adjustment formula will yield biased results.

3.6.4 Choice of Mediators

In more complex systems with multiple pathways, we often have flexibility in choosing our mediating set $\mathbf{M}$, as well as the adjustment set used to block the backdoor paths from $\mathbf{M}$ to Y.

Consider a system where the effect of A on Y branches through two parallel mechanisms, M_1 and M_2, before converging on a final mediator M_3:



To use the frontdoor adjustment, we need a valid set $\mathbf{M}$ that intercepts all directed paths from A to Y without introducing confounding. In this graph, we have two perfectly valid choices for our frontdoor mediator:

- **Option 1 ($\mathbf{M} = \{M_1, M_2\}$):** This joint set successfully intercepts all directed paths. There are no unblocked backdoor paths from A to $\{M_1, M_2\}$. To compute the second half of the frontdoor formula (the effect of $\{M_1, M_2\}$ on Y), we use A as our backdoor adjustment set to block the $M_i \leftarrow A \leftrightarrow Y$ paths.
- **Option 2 ($\mathbf{M} = \{M_3\}$):** The single node M_3 also intercepts all directed paths. There are no unblocked backdoor paths from A to M_3. However, when it comes time to compute the effect of M_3 on Y, we must block the backdoor paths flowing backwards through M_1 and M_2: $M_3 \leftarrow M_1 \leftarrow A \leftrightarrow Y$ and $M_3 \leftarrow M_2 \leftarrow A \leftrightarrow Y$.

If we choose Option 2, notice that we actually have *two* different valid choices for the backdoor adjustment set required by the internal frontdoor formula!

1. We can adjust for $\{A\}$ (the standard frontdoor approach).
2. We can adjust for $\{M_1, M_2\}$, which perfectly blocks the flow of confounding before it ever reaches A.

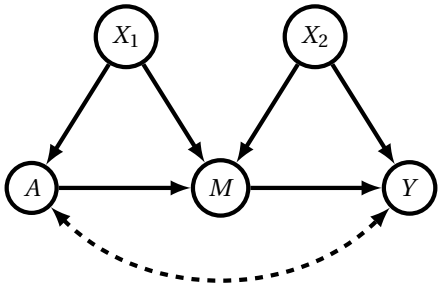
This demonstrates that the Frontdoor Criterion is not a rigid, single-variable formula, but a flexible structural strategy. As long as the three rules are satisfied, the choice of mediators and adjustment sets is up to you (or determined by which variables you actually have data for)!

3.6.5 Generalizations and Limitations

The standard frontdoor graph is just one specific topography. The logic of composing mechanisms allows us to generalize the frontdoor adjustment to many other settings, provided we respect the graphical rules. Below are examples of how these building blocks can be chained together.

Example 1: Composing Multiple Backdoor Adjustments (Computable)

Suppose we have a mediating variable M between our treatment A and outcome Y. However, unlike the strict frontdoor graph, the $A \to M$ link is confounded by an observed variable X_1, and the $M \to Y$ link is confounded by another observed variable X_2. We still have unobserved confounding between the treatment and outcome ($A \leftrightarrow Y$).



This graph violates the strict Frontdoor Criterion because both Rule 2 and Rule 3 fail. However, the compositional logic of the frontdoor adjustment still saves us! We can isolate the two mechanistic steps by performing two separate backdoor adjustments *before* multiplying them together:

1. **Effect of A on M:** We block the open backdoor path by adjusting for X_1:
$$\Pr(m \mid \text{do}(a)) = \sum_{x_1} \Pr(m \mid a, x_1) \Pr(x_1)$$
2. **Effect of M on Y:** We block the unobserved $A \leftrightarrow Y$ path by adjusting for A, and we block the new X_2 backdoor path by adjusting for X_2:
$$\Pr(y \mid \text{do}(m)) = \sum_{a', x_2} \Pr(y \mid m, a', x_2) \Pr(a', x_2)$$

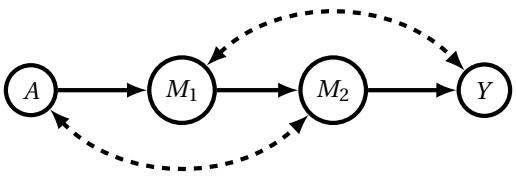
By substituting these local, backdoor-adjusted probabilities back into the core frontdoor composition formula ($\sum_m \Pr(m \mid \text{do}(a)) \Pr(y \mid \text{do}(m))$), we get:

$$\Pr(y \mid \text{do}(a)) = \sum_m \left[\sum_{x_1} \Pr(m \mid a, x_1) \Pr(x_1) \right] \left[\sum_{a', x_2} \Pr(y \mid m, a', x_2) \Pr(a', x_2) \right]$$

This identifies the full causal effect without needing a single, system-wide adjustment set!

Example 2: The Double Frontdoor (Computable)

We can push this compositional logic even further. What if our sequential mediators both suffer from distinct unobserved confounding? Consider the **Double Frontdoor**: $A \to M_1 \to M_2 \to Y$, where $A \leftrightarrow M_2$ and $M_1 \leftrightarrow Y$.



No single variable blocks all the confounding here. However, because M_2 completely intercepts the directed path from A to Y, we can break the problem into two isolated chunks and apply our adjustment formulas recursively!

1. **Effect of A on M_2:** If we temporarily ignore Y, the subgraph $A \to M_1 \to M_2$ with unobserved confounding $A \leftrightarrow M_2$ is a textbook Frontdoor graph! We can identify this effect using M_1 as the frontdoor mediator:
$$\Pr(m_2 \mid \text{do}(a)) = \sum_{m_1} \Pr(m_1 \mid a) \sum_{a'} \Pr(m_2 \mid m_1, a') \Pr(a')$$
2. **Effect of M_2 on Y:** If we look at the effect of M_2 on Y, we have an open backdoor path: $M_2 \leftarrow M_1 \leftrightarrow Y$. We can close this path by performing a standard Backdoor adjustment, using M_1 as our adjustment set:
$$\Pr(y \mid \text{do}(m_2)) = \sum_{m_1'} \Pr(y \mid m_2, m_1') \Pr(m_1')$$

To find the total effect of A on Y, we simply compose these two isolated mechanisms by summing over all possible states of the intercepting node M_2:

$$\Pr(y \mid \text{do}(a)) = \sum_{m_2} \Pr(m_2 \mid \text{do}(a)) \Pr(y \mid \text{do}(m_2))$$

Substituting our two formulas in, we get the massive **Double Frontdoor Functional**:

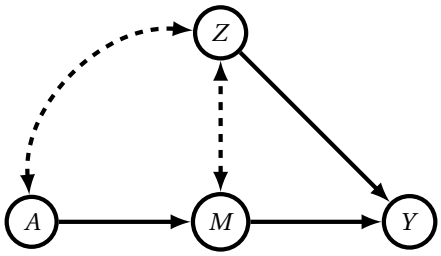
$$\Pr(y \mid \text{do}(a)) = \sum_{m_2} \underbrace{\left[\sum_{m_1} \Pr(m_1 \mid a) \sum_{a'} \Pr(m_2 \mid m_1, a') \Pr(a')\right]}_{\text{Frontdoor effect of } A \text{ on } M_2} \underbrace{\left[\sum_{m_1'} \Pr(y \mid m_2, m_1') \Pr(m_1')\right]}_{\text{Backdoor effect of } M_2 \text{ on } Y}$$

By chaining these tools together, we successfully identify the causal effect of A on Y entirely from observational data, perfectly bypassing the two overlapping arcs of unobserved confounding!

Example 3: When Causal Effects Don't Compose (Not Computable)

The Frontdoor derivation suggests a tempting intuition: if we can cleanly identify the causal effect of A on M, and we can cleanly identify the causal effect of M on Y, can we always just multiply them together to find the effect of A on Y?

Unfortunately, no. Consider the following ADMG where A and M share unobserved confounding with a third variable Z.



You might notice that we *can* identify the two individual mechanistic links:

- $\Pr(m \mid \text{do}(a))$ is identifiable (the path $A \leftrightarrow Z \leftrightarrow M$ contains a collider at Z and is therefore inherently blocked, meaning $\Pr(m \mid \text{do}(a)) = \Pr(m \mid a)$).
- $\Pr(y \mid \text{do}(m))$ is identifiable (we can simply use Z as a backdoor adjustment set to block the $M \leftrightarrow Z \to Y$ path).

However, it turns out that we **cannot** compose these two to get $\Pr(y \mid \text{do}(a))$. The confounding between Z and M makes it mathematically impossible to disentangle how their joint probability distribution shifts when we intervene on A. If M and Z interact to cause y, the unobserved background factors connecting A, Z, and M ruin our ability to just stitch the local formulas together.

This counter-example highlights a profound realization: relying on visual intuition and ad-hoc algebraic substitutions is dangerous. What we need is a rigid, mathematical calculus that tells us exactly when and how we can translate interventional probabilities into observational ones. Once we develop this calculus in the next lecture, you will return to this exact problem in your homework and prove that the causal effect is structurally unidentifiable.

3.7 Causal (Do and PO) Calculus

3.7.1 The Need for a Causal Algebra

At the end of the last section, we hit a wall. We saw that even when individual causal links (like $A \to M$ and $M \to Y$) are identifiable, we cannot always simply compose them to find the total effect of A on Y due to complex, unobserved confounding.

Our visual intuition failed us. We need a formal, symbolic algebra that allows us to manipulate probabilities involving the $\text{do}(\cdot)$ operator or potential outcomes $y^{(a)}$. Just like standard algebra allows us to isolate a variable x, we want a "causal algebra" that allows us to systematically transform expressions containing interventional probabilities ($\Pr(y \mid \text{do}(a))$) into purely observational probabilities ($\Pr(y \mid a)$) that we can estimate from data.

Historically, this calculus was developed in two parallel languages: the Potential Outcomes (PO) calculus using SWIGs, and the Do-Calculus developed by Judea Pearl. Both approaches yield the exact same results, and we will explore both by deriving the Frontdoor Adjustment.

3.7.2 Deriving the Frontdoor Adjustment with SWIGs

Because we are already familiar with Single World Intervention Graphs (SWIGs) from previous lectures, let's start there. SWIGs provide a graphical calculus through node-splitting and three core rules mapping the graph to potential outcomes:

1. If Y is d-separated from Z given W in $\mathcal{G}$, then

$$\Pr(y \mid z, w) = \Pr(y \mid w) \quad \text{(Markov property)}.$$

2. If $Y^{(z)}$ is d-separated from $Z^{(z)}$ given $W^{(z)}$ in the SWIG $\mathcal{G}(z)$, then

$$\Pr(y^{(z)} \mid w^{(z)}) = \Pr(y \mid w, z) \quad \text{(Generalized Ignorability)}.$$

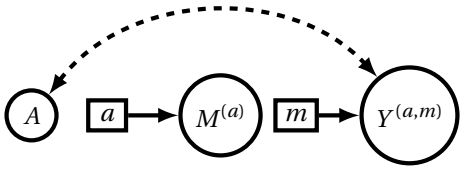
3. If z has no directed path to $Y^{(z)}$ in the SWIG $\mathcal{G}(z)$, then

$$\Pr(y^{(z)}) = \Pr(y) \quad \text{(Causal Irrelevance)}.$$

Our goal is to compute the total causal effect of A on Y, which is $\Pr(y^{(a)})$, in the Frontdoor graph. We start by applying the Law of Total Probability over the potential outcomes of the mediator, $M^{(a)}$:

$$\Pr(y^{(a)}) = \sum_m \Pr(y^{(a)} \mid M^{(a)} = m) \Pr(M^{(a)} = m) \tag{3.17}$$

To solve this, let's look at the SWIG where we simultaneously split on both interventions, $A = a$ and $M^{(a)} = m$.



In this graph, $M^{(a)}$ has no incoming arrows except from the fixed intervention a. It is completely d-separated from $Y^{(a,m)}$. Therefore, $Y^{(a,m)} \perp\!\!\!\perp M^{(a)}$, and we can drop the conditioning:

$$\Pr(y^{(a)}) = \sum_m \Pr(y^{(a,m)}) \Pr(M^{(a)} = m)$$

Next, we apply **Causal Irrelevance**. Because intervening on M blocks the only directed path from a to the outcome, the intervention $A = a$ is now completely irrelevant to Y. Thus, $\Pr(y^{(a,m)}) = \Pr(y^{(m)})$:

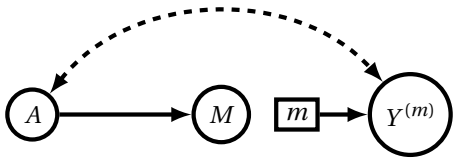
$$\Pr(y^{(a)}) = \sum_m \Pr(y^{(m)}) \Pr(m^{(a)}) \tag{3.18}$$

This is a formal expression of our intuition from earlier: we can compose the causal effect of A on M with the causal effect of M on Y. Let's solve these two terms.

First, look at the effect of A on M. If we only split on A, the SWIG shows $M^{(a)} \perp\!\!\!\perp A$. This gives us Generalized Ignorability, allowing us to swap the potential outcome for the observed conditional probability:

$$\Pr(m^{(a)}) = \Pr(m \mid a) \tag{3.19}$$

Second, let's look at the effect of M on Y. Here is the SWIG split only on $M = m$:



Notice that the only open path between M and $Y^{(m)}$ goes through A. If we condition on A, we block this backdoor path, meaning $Y^{(m)} \perp\!\!\!\perp M \mid A$. We apply the Law of Total Probability to introduce A (using a' to represent its states):

$$\Pr(y^{(m)}) = \sum_{a'} \Pr(y^{(m)} \mid a') \Pr(a')$$

Because $Y^{(m)} \perp\!\!\!\perp M \mid A$, we can freely introduce m into the condition without changing the probability. Once we have $\Pr(y^{(m)} \mid m, a')$, we can apply **Generalized Ignorability** to drop the potential outcome superscript:

$$\Pr(y^{(m)}) = \sum_{a'} \Pr(y \mid m, a') \Pr(a') \tag{3.20}$$

Finally, we substitute (3.19) and (3.20) back into (3.18) to get our final result:

$$\Pr(y^{(a)}) = \sum_m \Pr(m \mid a) \sum_{a'} \Pr(y \mid m, a') \Pr(a') \tag{3.21}$$

3.7.3 Pearl's Do-Calculus and Graph Mutilations

While SWIGs are highly intuitive because they explicitly show the new "potential outcome" nodes, drawing a new graph for every combination of interventions can become tedious. Judea Pearl developed an equivalent, purely symbolic algebra called **Do-Calculus**.

To use Do-Calculus, we evaluate d-separation (or m-separation for ADMGs) not on node-split SWIGs, but on "mutilated" versions of our original causal graph $\mathcal{G}$. These graph modifications represent what happens when we physically intervene in a system.

Let $\mathbf{X}$ be some set of nodes in our graph. We define two specific graph modifications:

1. $\mathcal{G}_{\overline{\mathbf{X}}}$ **(Incoming arrows deleted):** This is the graph $\mathcal{G}$ where all arrows pointing *into* nodes in $\mathbf{X}$ have been removed. This represents the post-intervention graph. If we force $\mathbf{X}$ to a specific value $\text{do}(\mathbf{x})$, it no longer listens to its natural parents.
2. $\mathcal{G}_{\underline{\mathbf{X}}}$ **(Outgoing arrows deleted):** This is the graph $\mathcal{G}$ where all arrows pointing *out* of $\mathbf{X}$ have been removed. We use this graph to check for backdoor paths. By deleting the direct causal effect of $\mathbf{X}$, any remaining active paths must be non-causal (backdoor) paths.

Naturally, we can combine these. $\mathcal{G}_{\overline{\mathbf{X}}\underline{\mathbf{Z}}}$ is the graph where incoming arrows to $\mathbf{X}$ and outgoing arrows from $\mathbf{Z}$ are simultaneously removed.

3.7.4 The Three Rules of Do-Calculus

Do-Calculus consists of three rules that allow us to add or remove variables from the observational condition, add or remove variables from the interventional condition, and swap interventions for observations.

Let $\mathcal{G}$ be a causal DAG or ADMG, and let $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$, and $\mathbf{W}$ be disjoint sets of variables. The following three rules hold:

Rule 1: Insertion/Deletion of Observations

$$\Pr(\mathbf{y} \mid \text{do}(\mathbf{x}), \mathbf{z}, \mathbf{w}) = \Pr(\mathbf{y} \mid \text{do}(\mathbf{x}), \mathbf{w})$$

if $\mathbf{Y} \perp\!\!\!\perp_m \mathbf{Z} \mid \mathbf{X}, \mathbf{W}$ in the mutilated graph $\mathcal{G}_{\overline{\mathbf{X}}}$.
Intuition: This is simply the standard Markov property (m-separation) applied to the post-intervention world. If we intervene on $\mathbf{X}$, and in this new intervened reality, $\mathbf{Z}$ gives us no new information about $\mathbf{Y}$ (given $\mathbf{W}$), we can safely drop $\mathbf{Z}$ from our conditioning set.

Rule 2: Action/Observation Exchange

$$\Pr(\mathbf{y} \mid \text{do}(\mathbf{x}), \text{do}(\mathbf{z}), \mathbf{w}) = \Pr(\mathbf{y} \mid \text{do}(\mathbf{x}), \mathbf{z}, \mathbf{w})$$

if $\mathbf{Y} \perp\!\!\!\perp_m \mathbf{Z} \mid \mathbf{X}, \mathbf{W}$ in the mutilated graph $\mathcal{G}_{\overline{\mathbf{X}}\underline{\mathbf{Z}}}$.
Intuition: This is the generalized Backdoor Criterion. If we delete all the outgoing causal arrows from $\mathbf{Z}$ ($\underline{\mathbf{Z}}$), and find that $\mathbf{Z}$ is completely separated from $\mathbf{Y}$, it means there are no open backdoor paths from $\mathbf{Z}$ to $\mathbf{Y}$. If there is no confounding between $\mathbf{Z}$ and $\mathbf{Y}$, then *observing* $\mathbf{Z}$ is mathematically equivalent to *intervening* on $\mathbf{Z}$.

Rule 3: Insertion/Deletion of Actions

$$\Pr(\mathbf{y} \mid \text{do}(\mathbf{x}), \text{do}(\mathbf{z}), \mathbf{w}) = \Pr(\mathbf{y} \mid \text{do}(\mathbf{x}), \mathbf{w})$$

if $\mathbf{Y} \perp\!\!\!\perp_m \mathbf{Z} \mid \mathbf{X}, \mathbf{W}$ in the mutilated graph $\mathcal{G}_{\overline{\mathbf{X}\mathbf{Z}(\mathbf{W})}}$, where $\mathbf{Z}(\mathbf{W})$ is the set of $\mathbf{Z}$-nodes that are not ancestors of any $\mathbf{W}$-node in $\mathcal{G}_{\overline{\mathbf{X}}}$.
Intuition: (For simplicity, assume $\mathbf{W}$ is empty so we look at $\mathcal{G}_{\overline{\mathbf{X}\mathbf{Z}}}$). If we sever all incoming arrows to $\mathbf{Z}$, and $\mathbf{Z}$ is separated from $\mathbf{Y}$, it means there is no directed causal path from $\mathbf{Z}$ to $\mathbf{Y}$. If $\mathbf{Z}$ cannot physically cause $\mathbf{Y}$, then intervening on $\mathbf{Z}$ will have absolutely no effect on $\mathbf{Y}$. Thus, we can drop the $\text{do}(\mathbf{z})$ entirely.

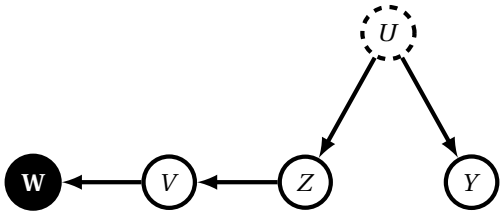
3.7.5 Intervening in Conditioned Settings

The condition $\mathbf{Z}(\mathbf{W})$ in Rule 3 is the most awkward piece of bookkeeping in the do-calculus. The cleanest way to motivate it is to ask why the rule cannot simply use the more natural-looking mutilation $\mathcal{G}_{\overline{\mathbf{Z}}}$ — severing $\mathbf{Z}$'s incoming arrows wholesale.

The reasoning behind $\mathcal{G}_{\overline{\mathbf{Z}}}$ is straightforward: $\text{do}(\mathbf{z})$ replaces $\mathbf{Z}$'s structural equation with a constant, so $\mathbf{Z}$'s parents stop influencing it. Severing the incoming arrows captures this for d-separation purposes. The

problem arises when $\mathbf{Z}$ is an ancestor of $\mathbf{W}$. Conditioning on $\mathbf{W}$ — a descendant of $\mathbf{Z}$ — carries information about $\mathbf{Z}$'s value, which carries information about $\mathbf{Z}$'s parents, which can carry information about $\mathbf{Y}$ via backdoor paths. If we sever $\mathbf{Z}$'s incoming arrows, those backdoors disappear from the mutilated graph, and the m-separation check falsely declares $\mathbf{Y} \perp\!\!\!\perp_m \mathbf{Z} | \mathbf{W}$. The caveat fixes this by saying: leave the incoming arrows in place for $\mathbf{Z}$-nodes that are ancestors of $\mathbf{W}$.

Motivating example: A hidden backdoor. Consider the following structure, where we ask whether Rule 3 lets us simplify $\Pr(y \mid \text{do}(z), w)$ to $\Pr(y \mid w)$:



Concretely: U is unobserved health-consciousness (dashed because it's unmeasured), Z is whether the patient enrolled in a cancer screening program, V is a detected tumor, W is enrollment in an oncology clinic, and Y is cardiac health. Health-conscious patients are more likely to screen ($U \to Z$) and to have good cardiac outcomes ($U \to Y$). Screening leads to detection, which leads to clinic enrollment. Crucially, screening does *not* directly cause cardiac outcomes — there is no directed path from Z to Y.

Naively, then, intervening on Z should not affect Y.

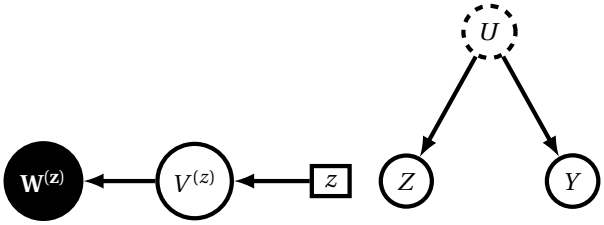
But in the observational world, conditioning on clinic enrollment ($W = 1$) selects for people who screened, which selects for the health-conscious, which biases their cardiac-health rate upward relative to the general population.

Under $\text{do}(z = 1)$, everyone screens, so clinic enrollees are now just "people with tumors" — no longer enriched for health-consciousness. Their cardiac-health rate drops to the population baseline. Even though Z does not cause Y, the conditional distribution of Y given W differs across the two regimes.

The do-calculus resolution. Because Z is an ancestor of W (via V), $\mathbf{Z}(\mathbf{W}) = \emptyset$, and the mutilated graph $\mathcal{G}_{\overline{\mathbf{Z}(\mathbf{W})}}$ is just $\mathcal{G}$. The backdoor $Z \leftarrow U \to Y$ is active (we are not conditioning on U), so the m-separation check correctly concludes $Y \not\perp\!\!\!\perp_m Z \mid W$ and Rule 3 forbids deleting $\text{do}(z)$. We can verify this algebraically: under $\text{do}(z)$, the variable W depends on z and noise but not on U, so $\Pr(y \mid \text{do}(z), w) = \Pr(y)$. Observationally, W does carry information about U (via $W \leftarrow V \leftarrow Z \leftarrow U$), so $\Pr(y|w) \neq \Pr(y)$. The two quantities differ exactly because the backdoor through U is real.

What goes wrong without the caveat. Suppose we ignored the caveat and used $\mathcal{G}_{\overline{Z}}$ instead — severing Z's incoming arrows. Then $U \to Z$ vanishes from the mutilated graph, the backdoor $Z \leftarrow U \to Y$ is broken, no path from Z to Y remains, and the m-separation check (incorrectly) succeeds. Rule 3 would license dropping $\text{do}(z)$, and we would wrongly conclude that the screening program has no effect on the observed cardiac-health rate among clinic enrollees. The $\mathbf{Z}(\mathbf{W})$ caveat is the bookkeeping that prevents this: when $\mathbf{Z}$ is an ancestor of $\mathbf{W}$, we keep its incoming arrows so the m-separation check can still see the relevant backdoors.

The SWIG interpretation. The same caveat falls out naturally in the SWIG framework. Splitting Z into the natural value Z and the intervention z gives:



The natural Z retains its parent U; the intervention z inherits the descendants $(V^{(z)}, W^{(z)})$. The variable Y takes no superscript, since it is not a descendant of Z — there is no directed path from the intervention z to Y, which appears to satisfy the precondition for Causal Irrelevance.

But Causal Irrelevance fails here, and the picture shows why. The natural Z is connected to Y through U, even though the intervention z is not. When we condition on factual W (the W we observe, which corresponds to the natural-regime W ultimately driven by Z), we condition on a variable whose distribution depends on the natural Z, and therefore on U, and therefore on Y. The conditioning ties the natural-regime distribution of Y to the value of W, in a way the intervention regime does not. Pearl's $\mathbf{Z}(\mathbf{W})$ caveat is the do-calculus way of bookkeeping the fact the SWIG makes visually obvious: Z keeps its parent in the split graph precisely because that parent is the bridge from the conditioning variable back to the outcome.

3.7.6 The Power of Completeness

Using these three rules, you can derive the Backdoor Adjustment. You can derive the Frontdoor Adjustment. In fact, you can derive *any* identifiable causal effect. In 2006, two groups independently proved that the **Do-Calculus is complete** [Shpitser and Pearl, 2006, Huang and Valtorta, 2006].

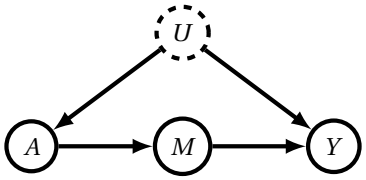
This means that if a causal effect $\Pr(\mathbf{y} \mid \text{do}(\mathbf{x}))$ can be identified from observational data in a given graph, there exists a sequence of applications of Rule 1, Rule 2, and Rule 3 that will successfully reduce the expression into pure observational probabilities. Conversely, if you exhaust all possible applications of Do-Calculus and still cannot remove all the $\text{do}(\cdot)$ operators, it is mathematically proven that the causal effect is structurally unidentifiable.

> **Main Idea 16**
>
> The Do-Calculus is complete and can identify all identifiable causal effects. With Do-Calculus, causal inference transitions from an art of "finding the right intuition" into a rigorous, solvable algorithmic science.

3.7.7 Deriving the Frontdoor Adjustment with Do-Calculus

Let's see Do-Calculus in action. We are going to mathematically derive the exact same Frontdoor Adjustment formula, but this time using graph mutilations instead of SWIGs.

Recall the frontdoor causal graph $\mathcal{G}$:



Our goal is to compute the causal effect of A on Y, which is $\Pr(y \mid \text{do}(a))$. We want to systematically apply the three rules until all $\text{do}(\cdot)$ operators are gone.

Step 1: Introduce the Mediator By the standard Law of Total Probability, we can introduce the mediator M into our expression:

$$\Pr(y \mid \text{do}(a)) = \sum_m \Pr(y \mid m, \text{do}(a)) \Pr(m \mid \text{do}(a)) \tag{3.22}$$

Now we have two interventional terms to solve: $\Pr(m \mid \text{do}(a))$ and $\Pr(y \mid m, \text{do}(a))$.

Step 2: Isolate the effect of A on M Let's look at the second term, $\Pr(m \mid \text{do}(a))$. We want to turn this into an observational probability using **Rule 2** (Action/Observation Exchange). To change $\text{do}(a)$ to a, we need to check if $M \perp\!\!\!\perp_m A$ in the mutilated graph $\mathcal{G}_{\underline{A}}$.

- In $\mathcal{G}_{\underline{A}}$, we delete the outgoing arrow $A \to M$.

- The only remaining path between A and M is the backdoor path $A \leftarrow U \rightarrow Y \leftarrow M$.
- This path contains a collider at Y, meaning it is already naturally blocked!

Because $M \perp\!\!\!\perp_m A$ in $\mathcal{G}_{\underline{A}}$, Rule 2 allows us to write:

$$\Pr(m \mid \text{do}(a)) = \Pr(m \mid a) \tag{3.23}$$

Step 3: Isolate the effect of M on Y Now let's tackle the first term from Step 1: $\Pr(y \mid m, \text{do}(a))$. This one is trickier.

First, we want to upgrade m to an intervention $\text{do}(m)$ using **Rule 2**. We must check if $Y \perp\!\!\!\perp_m M \mid A$ in the graph $\mathcal{G}_{\overline{A}\underline{M}}$.

- In $\mathcal{G}_{\overline{A}\underline{M}}$, we delete incoming arrows to A ($U \rightarrow A$) and outgoing arrows from M ($M \rightarrow Y$).
- The only remaining path between M and Y is $M \leftarrow A \leftarrow U \rightarrow Y$.
- However, because $U \rightarrow A$ is deleted, this path is severed. Even if it weren't, we are conditioning on A, which blocks it.

Because $Y \perp\!\!\!\perp_m M \mid A$ holds, we can swap m for $\text{do}(m)$:

$$\Pr(y \mid m, \text{do}(a)) = \Pr(y \mid \text{do}(m), \text{do}(a))$$

Next, we want to drop $\text{do}(a)$ entirely using **Rule 3** (Insertion/Deletion of Actions). We check if $Y \perp\!\!\!\perp_m A$ in the graph $\mathcal{G}_{\overline{MA}}$.

- In $\mathcal{G}_{\overline{MA}}$, we delete incoming arrows to M ($A \rightarrow M$) and to A ($U \rightarrow A$).
- With these arrows severed, there are absolutely no paths connecting A and Y.

Because $Y \perp\!\!\!\perp_m A$ holds, intervening on A has no effect if we are already intervening on M. Rule 3 allows us to drop $\text{do}(a)$:

$$\Pr(y \mid \text{do}(m), \text{do}(a)) = \Pr(y \mid \text{do}(m)) \tag{3.24}$$

Step 4: Compute the interventional effect of M We have reduced our target to $\Pr(y \mid \text{do}(m))$. To solve this, we introduce A back into the equation (using the dummy variable a' to represent its states) via the Law of Total Probability:

$$\Pr(y \mid \text{do}(m)) = \sum_{a'} \Pr(y \mid \text{do}(m), a') \Pr(a' \mid \text{do}(m))$$

We reduce these final two terms using Do-Calculus:

1. **Reduce** $\Pr(a' \mid \text{do}(m))$: We want to drop $\text{do}(m)$. By **Rule 3**, we check $\mathcal{G}_{\overline{M}}$. The arrow $A \rightarrow M$ is deleted. The only path is $M \rightarrow Y \leftarrow U \rightarrow A$, which is blocked by the collider at Y. Thus, $\Pr(a' \mid \text{do}(m)) = \Pr(a')$.
2. **Reduce** $\Pr(y \mid \text{do}(m), a')$: We want to change $\text{do}(m)$ to m. By **Rule 2**, we check $\mathcal{G}_{\underline{M}}$. The arrow $M \rightarrow Y$ is deleted. The backdoor path $M \leftarrow A \leftarrow U \rightarrow Y$ is blocked because we are conditioning on a'. Thus, $\Pr(y \mid \text{do}(m), a') = \Pr(y \mid m, a')$.

Substituting these back in gives us:

$$\Pr(y \mid \text{do}(m)) = \sum_{a'} \Pr(y \mid m, a') \Pr(a') \tag{3.25}$$

Step 5: The Final Assembly We have successfully eliminated every single $\text{do}(\cdot)$ operator. If we take our results from (3.23) and (3.25) and plug them back into our starting point in (3.22), we get:

$$\Pr(y \mid \text{do}(a)) = \sum_{m} \Pr(m \mid a) \sum_{a'} \Pr(y \mid m, a') \Pr(a')$$

This is the exact Frontdoor Adjustment formula! What previously required a leap of intuition was derived purely mechanically by checking path separations on mutilated graphs.

3.8 Conclusion: Two Languages, One Logic

Throughout this chapter, we have bridged the two dominant paradigms of causal inference: Judea Pearl's graphical causal models and the Rubin-Robins potential outcomes framework. Historically, these two camps were often seen as distinct, with researchers adhering rigidly to one or the other. However, as we have demonstrated, they are fundamentally describing the exact same mathematical reality.

Here are the core takeaways from our journey into Structural Causal Models and potential outcomes:

- **Causal Effects Require Exchangeability:** To isolate a causal effect, we need the potential outcome to be independent of the actual action taken, conditional on some set of covariates ($Y^{(a)} \perp\!\!\!\perp A | \mathbf{Z}$).
- **SWIGs Make Exchangeability Visual:** By applying node-splitting to a causal DAG, we generate a Single World Intervention Graph (SWIG). This brilliant tool allows us to use the familiar rules of d-separation to test for exchangeability directly on the graph.
- **Graphs and Algebra are Complementary:** While recursive substitution gives us the rigorous, algebraic definition of a potential outcome under an NPSEM-IE, SWIGs provide a rapid visual shortcut. As we saw with the Frontdoor Adjustment, we can derive complex causal estimands using either Pearl's Do-Calculus or the potential outcome rules of generalized ignorability and causal irrelevance. The result is always the same.

Throughout this chapter, you have learned to be a causal "chef." By formalizing your assumptions into a structural graph, you now possess the mathematical toolkit to rigorously prove *when* and *how* a causal effect can be identified. You know how to use the backdoor criterion to find the exact set of variables required to achieve conditional exchangeability, and you know exactly which variables (like colliders) to avoid.

However, **identification is only half the battle**. Do-calculus tells us *what* mathematical expression we need to compute (for example, the g-formula or the inverse probability weights), but it assumes we have access to infinite data and true probability distributions. It does not tell us *how* to actually compute those quantities from a finite, noisy, real-world dataset.

In the next chapter, we will shift from causal identification to **Causal Estimation**. We will put on our "cook" hats and learn the statistical recipes required to turn our identified graphs into actual numbers. We will explore a taxonomy of estimation strategies—ranging from classical econometric tools like Ordinary Least Squares (OLS) regression to modern, doubly robust machine learning algorithms—that allow us to confidently extract causal estimates from observational data.

Chapter 4

Estimating Causal Effects

Given a system and a question, the previous chapter taught us exactly which functional of the observed distribution to compute. But quantities we derived are statements about distributions we never see directly; in practice, we have a finite, noisy sample, and we have to *estimate* our functional.

This chapter is about that estimation, and it is organized around a single idea: nearly every estimator we meet is an instance of a problem you have already seen in machine learning. Inverse propensity weighting is not a new statistical trick but a correction for distribution shift — the treated units are a biased sample of the population, and we reweight them to make them resemble the whole population. Synthetic controls are a matrix completion problem: the potential outcome we never observed is a missing entry in a units-by-time matrix, recoverable when that matrix is low-rank, exactly as in the Netflix problem. Even the most classical methods line up as a progression: outcome regression, partialling out, and inverse weighting each trade away functional-form assumptions for a potentially non-parametric machine learning model until the doubly robust estimator asks only that we get one of two models right. We close with a new idea: rather than adjusting for confounding, instrumental variables hunt for a source of natural randomization already present in the data. In the homework, we will study how this is, in fact, just the other side of the same coin.

The thread throughout is that estimating causal effects is not a separate craft from the statistics and machine learning you may know — it is exactly machine learning and statistics, but guided by the principles of causal models that we have established. While the previous chapter felt like graph theory and discrete logic, this part of the class will feel heavily grounded in statistics, econometrics, and predictive modeling.

4.1 Outcome Regression

To compute causal effects from observational data, we must isolate the true effect of a treatment or action on an outcome from the noise of confounding variables.

Remember the confounded setting from our earlier causal diagrams. We saw that Season acts as a confounder for CO_2 and Temperature, while Nutrition acts as a confounder for Lipids and Mortality. If we want to estimate the true causal effect of our treatment A on our outcome Y, we determined that we need to look at the data while holding the confounder X constant.

If we have *conditional exchangeability* — meaning the potential outcomes are independent of the treatment assignment given the covariates ($Y^{(a)} \perp\!\!\!\perp A \mid X$) — then controlling for X allows us to extract the true causal effect.

But what does it computationally mean to "control for" X? When X is categorical, it is clear: we simply stratify the data into subsets with constant values (e.g., restricting the analysis to a specific country or department). If X is continuous, we need to compute an integral on a probability density function that is not so easy to estimate. By making parametric assumptions about the functional form of the relationships, we can use **regression** to bypass this difficulty.

4.1.1 Regression as Conditioning

The most natural way to think about a causal coefficient in a parametric model is as a *partial derivative*: the change in Y per unit change in A, holding all other causes of Y fixed. Multiple regression makes this precise. Suppose everything in our system is perfectly linear; using a Structural Causal Model (SCM) from earlier, let:

$$X = \epsilon_X$$
$$A = \alpha X + \epsilon_A$$
$$Y = \beta_A A + \beta_X X + \epsilon_Y$$

The structural coefficient $\beta_A = \partial Y / \partial A$ is exactly the causal effect we want.

A naive simple regression of Y on A alone is biased by the confounder:

$$\hat{Y} = (\beta_A + \beta_X \alpha) A.$$

But running a multiple regression of Y on *both* A and X asks OLS to find the best-fit coefficients for both variables simultaneously, and it perfectly recovers the structural equation:

$$\hat{Y} = \hat{\beta}_A A + \hat{\beta}_X X.$$

Here $\hat{\beta}_A$ recovers exactly the causal coefficient β_A. **Including a confounder X in your linear regression is computationally equivalent to conditioning on it** — the partial derivative interpretation is the causal interpretation.

> **Main Idea 17**
>
> Including a backdoor adjustment set **X** in a regression is equivalent to conditioning on it.

4.1.2 The Problem with Nonlinear Confounding

This story is clean as long as the confounder really does enter linearly. In practice it almost never does. Recall the relationship between CO_2 and Temperature, which is confounded by the climate (the month/season). If we plot CO_2 and Temperature over time, we see a "snaking" oscillation because the month strongly predicts local peaks and valleys.

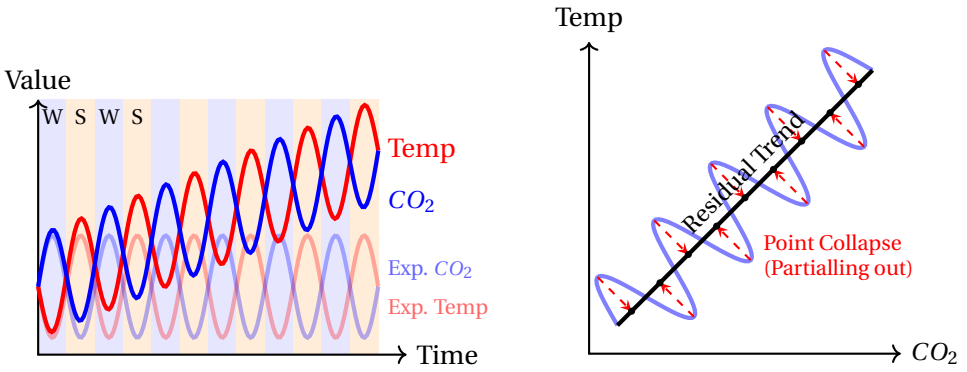


Figure 4.1: Left: Time series of Temperature and CO_2, overlaid with their expected seasonal baseline (lighter opacity). Because the actual values pull away from the flat expectations, the residuals (the gap between them) are steadily increasing. Right: The parametric plot of Temp vs CO_2. By "partialling out" the expected season, we collapse the snaking loops onto the true underlying causal Residual Trend line (red arrows).

The treatment A (atmospheric CO_2) plausibly does affect outcome Y (temperature) approximately linearly — a degree of CO_2 buys you some fraction of a degree of warming. But the confounder X (month) emphatically does not enter linearly: its effect on both A and Y is sinusoidal. If we plug month-as-a-number into a multiple regression alongside CO_2, OLS does its best to fit a line through an oscillation, fails, and the $\hat{\beta}_A$ coefficient picks up whatever bias is needed to compensate. Worse, on a short time window the seasonal slope can dominate the true causal trend entirely, and we recover a coefficient with the wrong sign.

The conceptual problem is that we're asking one regression to do two jobs: model the linear treatment effect of interest *and* model the gnarly nonlinear confounding. A linear regression simply isn't expressive enough to do the second job, and forcing it to try contaminates the first.

Theorem 4.1.1 (Partially Linear Model / Population Partialling Out)**.** *Consider the partially linear model*

$$\begin{aligned} A &= h(X) + \eta, \qquad \mathbb{E}[\eta \mid X] = 0 \\ Y &= A\beta_A + g(X) + \epsilon, \qquad \mathbb{E}[\epsilon \mid A, X] = 0, \end{aligned}$$

where g and h are arbitrary unknown functions. Let $\tilde{Y} = Y - \mathbb{E}[Y \mid X]$ and $\tilde{A} = A - \mathbb{E}[A \mid X] = \eta$. Then

$$\beta_A = \left(\mathbb{E}[\tilde{A}\tilde{A}^\top]\right)^{-1}\mathbb{E}[\tilde{A}\tilde{Y}].$$

Proof Sketch: *Notation.* Throughout this proof, A, X, Y, ϵ, η are random variables, and expectations are taken over their joint distribution. The residuals $\tilde{A}$ and $\tilde{Y}$ are random variables, not vectors of fitted residuals from a finite sample.

The proof has two moves: *cancel g via residualization*, then *isolate β_A*.

Cancel g via residualization. The nuisance function $g(X)$ is the obstacle: it can be arbitrarily complicated, and we have no parametric handle on it. The trick is that $g(X)$ is completely determined by X, so it lives entirely inside the conditional expectation $\mathbb{E}[\cdot \mid X]$. Take the conditional expectation of the model given X:

$$\mathbb{E}[Y \mid X] = \mathbb{E}[A \mid X]\,\beta_A + g(X),$$

where the structural error vanishes because $\mathbb{E}[\epsilon \mid X] = 0$, and $g(X)$ passes through the conditional expectation untouched. Now form the residual $\tilde{Y} = Y - \mathbb{E}[Y \mid X]$:

$$\tilde{Y} = \big(A\beta_A + g(X) + \epsilon\big) - \big(\mathbb{E}[A \mid X]\,\beta_A + g(X)\big) = \tilde{A}\beta_A + \epsilon,$$

where $\tilde{A} = A - \mathbb{E}[A \mid X]$. The arbitrary nuisance g has cancelled — regardless of what g actually is — and the semiparametric problem has been reduced to an honest linear regression of $\tilde{Y}$ on $\tilde{A}$.

Isolate β_A. The final step is the standard OLS move. Multiply both sides by $\tilde{A}$ and take expectations:

$$\mathbb{E}[\tilde{A}\tilde{Y}] = \mathbb{E}[\tilde{A}\tilde{A}^\top]\,\beta_A + \mathbb{E}[\tilde{A}\epsilon].$$

The last term vanishes because $\tilde{A}$ is a function of A and X, and $\mathbb{E}[\epsilon \mid A, X] = 0$ implies $\mathbb{E}[\tilde{A}\epsilon] = 0$. Solving for β_A gives the claimed formula. ∎

From population to sample. In a finite sample of N i.i.d. observations, the population expectations get replaced by sample averages: $\hat{\beta}_A = \left(\sum_i \tilde{a}_i \tilde{a}_i^\top\right)^{-1} \sum_i \tilde{a}_i \tilde{y}_i$, where $\tilde{a}_i$ and $\tilde{y}_i$ are the residuals of the i-th observation. The next subsection makes this finite-sample version exact in the linear-nuisance case.

4.1.3 The Frisch-Waugh-Lovell Theorem

The population result above is the conceptual point, but it raises an immediate practical question: in a finite sample, with linear nuisance models, does this two-stage residualization actually give the same answer as just running one big multiple regression of Y on A and X? The **Frisch-Waugh-Lovell (FWL) Theorem** says yes, exactly — the two procedures are algebraically identical.

This is the linear, finite-sample anchor for the partialling-out logic. It is reassuring: even before invoking the semiparametric machinery of the previous subsection, we already know that residualization recovers the OLS coefficient *mechanically*, as a property of the matrix algebra.

Geometrically, OLS is an orthogonal projection. As shown in Figure 4.2, regressing Y on A and X projects the Y vector onto the plane spanned by A and X. The FWL theorem mathematically guarantees that projecting Y onto the X space first, and then projecting the remaining residual onto the A space, is functionally identical to projecting onto both simultaneously.

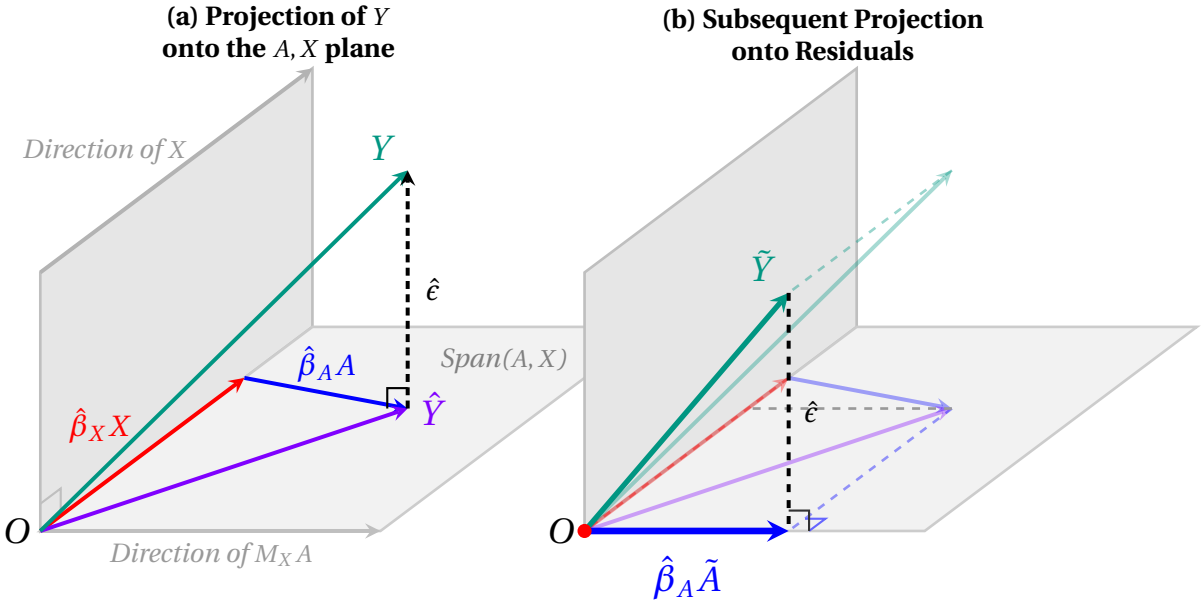


Figure 4.2: Geometric intuition of the Frisch-Waugh-Lovell Theorem.

Let's prove this using linear algebra. Let X, Y, and A be N-dimensional vectors of data. We can define a projection matrix $P_X = X(X^\top X)^{-1}X^\top$ that projects any vector onto the column space of X.[1] We also define the "residual-maker" matrix $M_X = I - P_X$, which projects onto the orthogonal complement of X.

Applying M_X to Y gives us the residual vector $\tilde{Y} = Y - P_X Y$, and applying it to A gives us $\tilde{A}$.

Theorem 4.1.2 (Frisch-Waugh-Lovell)**.** *Consider the linear regression model $Y = A\beta_A + X\beta_X + \epsilon$. The OLS estimate $\hat{\beta}_A$ is identical to the estimate obtained by regressing the residuals of Y on X ($\tilde{Y}$) onto the residuals of A on X ($\tilde{A}$).*

Proof: The full OLS regression decomposes the outcome vector Y into fitted values and the orthogonal residual $\hat{\epsilon}$, as visually established in Panel (a) of Figure 4.2:

$$Y = \hat{\beta}_A A + \hat{\beta}_X X + \hat{\epsilon}$$

Geometrically, we can isolate the component of Y that is orthogonal to X by multiplying the entire equation by the residual-maker matrix M_X:

$$M_X Y = M_X(\hat{\beta}_A A + \hat{\beta}_X X + \hat{\epsilon})$$

Because matrix multiplication is linear, we can distribute M_X:

$$M_X Y = \hat{\beta}_A M_X A + \hat{\beta}_X M_X X + M_X \hat{\epsilon}$$

Now, we evaluate the terms on the right side. This step directly corresponds to the stacked projection layers shown in Panel (b) of Figure 4.2:

1. $M_X X = 0$, because projecting X onto its orthogonal complement leaves nothing. The $\hat{\beta}_X$ term completely vanishes (represented by the red vector collapsing to the origin in Panel (b)).

[1] If you want to know why the projection matrix is formulated this way, I recommend taking a look at Appendix A and reviewing the Moore-Penrose Pseudo-inverse. Generally, this involves writing down the squared errors (which are quadratic), taking a derivative, and setting it equal to zero.

2. $M_X \hat{\epsilon} = \hat{\epsilon}$, because the OLS residuals from the full regression are already perfectly orthogonal to both A and X. This explains why the projection of Y (teal layer) and the projection of $\hat{Y}$ (violet layer) align perfectly in the diagram; the orthogonal error term drops out of the projection.

Substituting these back into the equation leaves us with:

$$M_X Y = \hat{\beta}_A M_X A + \hat{\epsilon}$$

By definition, $M_X Y$ is the vector of residuals from regressing Y on X (which we call $\tilde{Y}$), and $M_X A$ is the vector of residuals from regressing A on X (which we call $\tilde{A}$).

$$\tilde{Y} = \hat{\beta}_A \tilde{A} + \hat{\epsilon}$$

This is precisely a simple linear regression of $\tilde{Y}$ on $\tilde{A}$ without an intercept. Geometrically, projecting $\tilde{Y}$ onto $\tilde{A}$ extracts the blue vector $\hat{\beta}_A \tilde{A}$ from Panel (b). Because $\hat{\epsilon}$ is orthogonal to $\tilde{A}$ (since it is orthogonal to the entirety of A and X), $\hat{\beta}_A$ perfectly satisfies the OLS normal equation for this residualized model:

$$\hat{\beta}_A = (\tilde{A}^\top \tilde{A})^{-1} \tilde{A}^\top \tilde{Y}$$

■

So FWL is the linear special case that anchors the broader population partialling-out result: in the linear, finite-sample world, the two-stage residualization isn't an approximation or a clever heuristic — it gives *exactly* the same number as full multiple regression, on the nose. The population/partially-linear theorem then extends the same idea beyond linearity, by replacing the projection matrix M_X with the conditional expectation operator $\mathbb{E}[\cdot \mid X]$ as the "residual-maker."

One assumption is worth flagging explicitly before we move on, because it motivates much of what follows. The partial linear model writes $Y = A\beta_A + g(X) + \epsilon$, with β_A a single scalar — the same partial derivative for every unit. This is a **homogeneity** assumption: every unit responds identically to a one-unit change in treatment, regardless of X. When it holds, the conditional average treatment effect (CATE) at every X equals the average treatment effect (ATE), and the recovered $\hat{\beta}_A$ unambiguously estimates both. When it fails — when the true effect $\beta_A(X)$ genuinely varies with covariates — partialling out instead recovers a weighted average of CATEs with weights proportional to $\text{Var}(A \mid X)$, which is rarely the weighting we wanted. The next section develops a method that targets the ATE directly without requiring this kind of structural assumption on the outcome surface.

4.2 Inverse Propensity Weighting

In the previous section, we built up two approaches to estimating the causal coefficient β_A, each loosening assumptions about the functional forms in the SCM. The fully linear outcome regression assumed every edge in the graph was linear: $X \to A$, $X \to Y$, and $A \to Y$. Partialling out then relaxed this — by introducing nuisance functions that absorb arbitrary nonlinearity in how X influences A and Y, the partial linear model lets us estimate β_A even under highly nonlinear confounding. But one assumption remained untouched: the $A \to Y$ relationship was still linear, with β_A a single scalar — the same partial derivative for every unit.

That last linearity assumption is doing real work, and it fails in at least two important settings. The first is **heterogeneous effects**: when a drug helps young patients more than old ones, or a policy affects rich and poor households differently, the partial-linear coefficient becomes a weighted average of the unit-level CATEs rather than the ATE itself. The second is **dose-response curves**: when the question of interest is the *shape* of Y's response to A — does the second cigarette of the day do as much damage as the twentieth? — a single slope is the wrong object to estimate in the first place.

Inverse Propensity Weighting (IPW) takes a fundamentally different approach: rather than modeling the outcome at all, it models the *treatment assignment mechanism* — the conditional probability $\Pr(A \mid X)$ of receiving each treatment given covariates — and re-weights the observed sample to simulate a population in which A no longer depends on X. Once the data has been re-weighted to break the confounding link, we can estimate the $A \to Y$ relationship using whatever flexible nonparametric regression we like, with no homogeneity or linearity assumption to violate. The natural setting for IPW is discrete treatments, where estimating $\Pr(A \mid X)$ is a standard classification problem; for continuous A the propensity becomes a conditional density, which is considerably harder to estimate well.

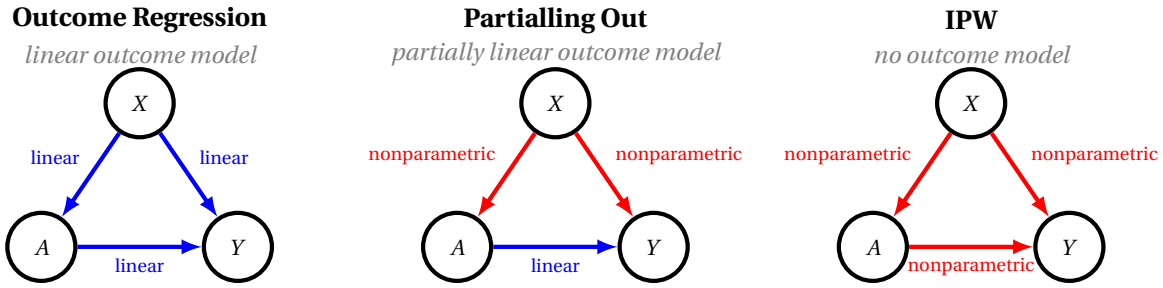


Figure 4.3: The progression of modeling choices across our three approaches. Outcome regression imposes a linear functional form on every relationship in the SCM. Partialling out absorbs nonlinear $X \to A$ and $X \to Y$ relationships into nuisance functions but keeps $A \to Y$ linear (and hence assumes homogeneous effects). IPW imposes no functional-form restrictions at all — every edge can be arbitrary. The modeling burden is relocated to the propensity $\Pr(A \mid X)$, which is the one quantity that must be estimated for re-weighting to work.

In its most general form, the IPW estimator for the expected potential outcome (also known as the **Horvitz-Thompson estimator** in the statistical survey literature) is:

$$\mathbb{E}[Y^{(a)}] = \mathbb{E}\left[\frac{\mathbb{1}[A=a]\,Y}{\Pr(A=a \mid X)}\right] \tag{4.1}$$

The re-weighting principle applies universally across data types, but we will start our derivation in a discrete, categorical setting to make the math transparent. By the end of this section, we will see how the same logic extends to continuous outcomes and regression settings.

4.2.1 Motivating Example: Simpson's Paradox

To build intuition for the weighting idea, let's revisit a simple table demonstrating Simpson's paradox.

Severity	All		Untreated		Treated	
	Patients	Survived	Patients	Survived	Patients	Survived
Severe	110	≈ 14%	10	10 %	100	14 %
Minor	110	≈ 76%	100	76%	10	80 %
Total	**220**	**≈ 45 %**	**110**	**70 %**	**110**	**20 %**

Marginally, the data make treatment look terrible: 70% of untreated patients survive vs. only 20% of treated patients. But within each severity stratum, treatment helps slightly — 14% vs. 10% in severe cases, 80% vs. 76% in minor cases. The marginal numbers mislead because the treated and untreated groups are not comparable populations.

The reason they are not comparable is structural: treated patients are *overrepresented* in the severe stratum and *underrepresented* in the minor stratum. Untreated patients are the mirror image — overrepresented in minor cases, underrepresented in severe ones. The treatment groups look different not because treatment causes harm, but because the conditions under which treatment was assigned mean we are comparing apples to oranges.

If we give 10x the weight to the underrepresented data points (treated minor cases and untreated severe cases), the weighted distribution becomes balanced — the treated and untreated groups now have the same severity composition, and the marginal comparison becomes meaningful. We can then apply any regular regression to this weighted data without explicitly conditioning on severity, because the act of re-weighting has already broken the confounding link. The next two subsections formalize this idea: first via a quick detour through machine learning, then by deriving the explicit weights.

4.2.2 Domain Adaptation and Weighting

We are going to take a brief detour. Much of machine learning can be summarized in the following optimization problem:

$$f = \text{argmin}_{f \in \mathscr{F}} \sum_{x,y \in \text{Data}} \text{Loss}(f(x), y). \tag{4.2}$$

This is called empirical risk (loss) minimization. We pick some function class $\mathscr{F}$ (e.g., linear models where $f(x) = mx+b$), and then we pick some loss function (e.g., $(y-f(x))^2$, the squared loss), and we find a function in that function class that minimizes the expected loss across the data.

The hope is that the data is representative of the eventual use case. If there is a shift in the distribution of X ("covariate shift"), then the ideal model will change, because more frequent $X = x$ events become more important to optimize! For example, training a self-driving car in California will over-emphasize performance on sunny days, meaning its driving may not be optimal for New Hampshire. Sunny days are more likely in California, but snowy days are 10x more likely in NH, so the loss on snowy days is 10x as important in NH.

If we *know* what the covariate shift will be, we can adjust our empirical loss by re-weighting the data. Say the data is drawn from a probability distribution $p(x)$ in California, and we want to go to a distribution $q(x)$ in NH. The adjusted optimization is:

$$f = \text{argmin}_{f \in \mathscr{F}} \sum_{x,y \in \text{Data}} \frac{q(x)}{p(x)} \text{Loss}(f(x), y). \tag{4.3}$$

Here, $\frac{q(x)}{p(x)}$ is an importance weight. This minimizes the expected loss in NH using data from CA by weighting up the points that are underrepresented in p relative to q. The same reweighting idea, applied to the joint distribution over (A, X, Y), will give us IPW.

4.2.3 Deriving the IPW Estimator

Let's apply the reweighting logic to the Simpson's paradox setup. We use $A \in \{0,1\}$ for treatment, $X \in \{0,1\}$ for the condition/severity, and $Y \in \{0,1\}$ for survival.

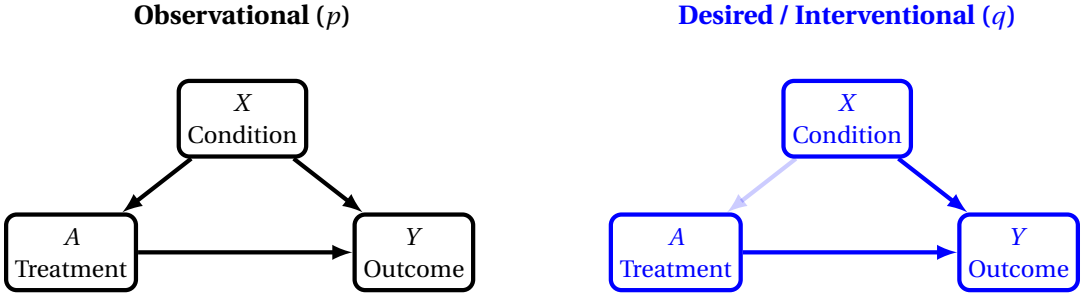


Figure 4.4: Left: The observational distribution p where the condition X confounds the treatment and outcome. Right: The desired weighted distribution q where the causal arrow from X to A is effectively removed, guaranteeing exchangeability.

As shown in the left panel of Figure 4.4, the observational distribution we *have* contains a confounding path:

$$\Pr(A, X, Y) = \Pr(X) \Pr(A \mid X) \Pr(Y \mid A, X). \tag{4.4}$$

The interventional distribution we *want* is one where $\Pr(A) = \Pr(A \mid X)$, meaning the treatment assignment is independent of the covariates. As shown in the right panel of Figure 4.4, this represents a system where the arrow from X to A has been removed:

$$q(A, X, Y) = \Pr(X) \Pr(A) \Pr(Y \mid A, X). \tag{4.5}$$

This factorization means $A \perp\!\!\!\perp X$ in our desired distribution q, which aligns with isolating the causal effect (exchangeability holds). The corresponding weight for each data point is:

$$w(A,X,Y) \coloneqq \frac{q(A,X,Y)}{\Pr(A,X,Y)} = \frac{\Pr(A)}{\Pr(A \mid X)}. \tag{4.6}$$

For each column A, we re-weight each row according to this ratio. This is called "inverse propensity weighting" because we are giving each datapoint a weight corresponding to the inverse of its propensity to get the treatment it ended up receiving. People who get treatment and were likely to get treatment (severe cases that were treated) are weighted down, and people who were unlikely to get treatment but did get it (mild cases that were treated) are weighted up.

Exercise 3

Another way of thinking of the problem is an overrepresentation of the severe cases in the treated population and an overrepresentation of the mild cases in the untreated population. What weights would this correspond to? Are they the same?

The resulting IPW estimator is given by reweighting the estimator for the conditional probability. Sums run over the N data points indexed by i, with realized values (a_i, x_i, y_i). Weights are given in blue.

$$\begin{aligned} \Pr(Y = y \mid A = a) &\coloneqq \frac{\sum_{i=1}^{N} \mathbb{1}[a_i = a, y_i = y]}{\sum_{i=1}^{N} \mathbb{1}[a_i = a]} \\ \Pr(Y^{(a)} = y) &\coloneqq \frac{\sum_{i=1}^{N} w(a_i,x_i,y_i)\,\mathbb{1}[a_i = a, y_i = y]}{\sum_{i=1}^{N} w(a_i,x_i,y_i)\,\mathbb{1}[a_i = a]} \end{aligned} \tag{4.7}$$

Substituting the weight $w(a_i,x_i,y_i) = \Pr(A = a_i)/\Pr(A = a_i \mid X = x_i)$ for each data point gives:

$$\Pr(Y^{(a)} = y) = \frac{\sum_{i=1}^{N} \Pr(A = a_i)/\Pr(A = a_i \mid X = x_i)\,\mathbb{1}[a_i = a, y_i = y]}{\sum_{i=1}^{N} \Pr(A = a_i)/\Pr(A = a_i \mid X = x_i)\,\mathbb{1}[a_i = a]} \tag{4.8}$$

The indicator $\mathbb{1}[a_i = a]$ zeros out every term where $a_i \neq a$, so within the surviving terms we can replace a_i with a in the weight:

$$\Pr(Y^{(a)} = y) = \frac{\sum_{i=1}^{N} \Pr(A = a)/\Pr(A = a \mid X = x_i)\,\mathbb{1}[a_i = a, y_i = y]}{\sum_{i=1}^{N} \Pr(A = a)/\Pr(A = a \mid X = x_i)\,\mathbb{1}[a_i = a]} \tag{4.9}$$

Now $\Pr(A = a)$ is a constant that doesn't depend on i, so it factors out and cancels between the numerator and denominator:

$$\Pr(Y^{(a)} = y) = \frac{\sum_{i=1}^{N} 1/\Pr(A = a \mid X = x_i)\,\mathbb{1}[a_i = a, y_i = y]}{\sum_{i=1}^{N} 1/\Pr(A = a \mid X = x_i)\,\mathbb{1}[a_i = a]} \tag{4.10}$$

This is the simpler weight $1/\Pr(A = a|X = x_i)$ that gives the method its name — the inverse of unit i's propensity to receive treatment a.

4.2.4 IPW and Backdoor Equivalence

Now that we have derived the practical inverse propensity weights, we can connect this approach to the graphical methods we learned earlier. If you remember the Backdoor Adjustment formula, the concept of standardizing over covariates should feel familiar. In fact, we can algebraically prove that they are perfectly identical operations in the discrete case.

The IPW estimator computes the interventional distribution by re-weighting the observational joint distribution. Assuming a set of covariates $\mathbf{x}$, the weight applied to the specific treatment group a is the inverse of the propensity score: $w(a,\mathbf{x}) = \frac{1}{\Pr(a|\mathbf{x})}$. Thus, we can express the interventional probability (using Pearl's do-notation) as the sum over the re-weighted joint distribution:

$$\Pr(y \mid \mathrm{do}(a)) = \sum_{\mathbf{x}} \Pr(\mathbf{x}, a, y) \cdot \frac{1}{\Pr(a \mid \mathbf{x})} \tag{4.11}$$

Using the chain rule of probability, we can factor the observational joint distribution $\Pr(\mathbf{x}, a, y)$ into:

$$\Pr(\mathbf{x}, a, y) = \Pr(\mathbf{x}) \Pr(a \mid \mathbf{x}) \Pr(y \mid a, \mathbf{x}) \tag{4.12}$$

Substituting this expansion back into the IPW formula:

$$\Pr(y \mid \text{do}(a)) = \sum_{\mathbf{x}} \frac{\Pr(\mathbf{x}) \Pr(a \mid \mathbf{x}) \Pr(y \mid a, \mathbf{x})}{\Pr(a \mid \mathbf{x})}$$

The propensity score $\Pr(a \mid \mathbf{x})$ appears in both the numerator and the denominator, so they cancel:

$$\Pr(y \mid \text{do}(a)) = \sum_{\mathbf{x}} \Pr(\mathbf{x}) \Pr(y \mid a, \mathbf{x}) \tag{4.13}$$

This is exactly the Backdoor Adjustment formula (or the g-formula for $\Pr(Y^{(a)} = y)$). IPW (re-weighting the population) and Backdoor Adjustment (standardizing over the covariates) are mathematically identical ways to isolate the causal signal.

One additional way to interpret this functional is to observe how it differs from the regular conditional probability. By the Law of Total Probability, the purely observational probability of the outcome is:

$$\Pr(y \mid a) = \sum_{\mathbf{x}} \Pr(\mathbf{x} \mid a) \Pr(y \mid a, \mathbf{x}) \tag{4.14}$$

The IPW functional in Eq. (4.13) "forces" the distribution of $\mathbf{X}$ to simply be its marginal $\Pr(\mathbf{x})$, regardless of the treatment group. The ratio required to turn Eq. (4.14) into Eq. (4.13), which is $\Pr(\mathbf{x}) / \Pr(\mathbf{x} \mid a)$, is mathematically equivalent to $\Pr(a) / \Pr(a \mid \mathbf{x})$ by Bayes' theorem — exactly our inverse propensity weight.

This expression also delivers on the heterogeneity motivation from the start of the section. The functional $\sum_{\mathbf{x}} \Pr(\mathbf{x}) \Pr(y \mid a, \mathbf{x})$ places no constraint on how $\Pr(y \mid a, \mathbf{x})$ varies across $\mathbf{x}$ — the conditional outcome distribution can differ arbitrarily from one stratum to the next, and the formula simply averages over the marginal distribution of covariates. Equivalently, IPW computes $\mathbb{E}[Y^{(a)}]$ separately for each treatment level a, with no parameters shared across levels. There is no single "treatment effect slope" that the method is forced to commit to, and consequently, no homogeneity assumption to violate.

> **Main Idea 18**
>
> IPW allows us to recover the interventional distribution from observational data by appropriately re-weighting the data points to simulate an exchangeable, randomized trial.

4.2.5 Positivity

The IPW estimator divides by $\Pr(A = a \mid X)$, which means it implicitly requires this quantity to be strictly positive for every value of X in the support of the data. If there are values of X where some treatment level a is never observed — so $\Pr(A = a \mid X) = 0$ — the weight is undefined and the estimator breaks down. This requirement is called **positivity** (sometimes **overlap** or **common support**).

Positivity is a substantive assumption about the data, not just a technical convenience. If certain patient profiles are simply never given a particular treatment in your data — say, a drug is never prescribed to children, or a policy is never applied in rural areas — IPW cannot extrapolate to estimate the counterfactual effect of giving them that treatment. No re-weighting trick recovers information that isn't in the sample. Conceptually, positivity says that every region of covariate space contains both treated and untreated units, so there is enough variation in A at every X to identify the effect.

Even when positivity holds in principle, it can fail in practice when $\Pr(A = a \mid X)$ is technically positive but very small. A propensity of 0.01 produces a weight of 100, and a single such observation can dominate the estimator's variance. The IPW estimator becomes unstable, with effective sample sizes far smaller than N. Practical diagnostics include plotting the histogram of estimated propensity scores by treatment group and looking for regions where one group has very few observations. Common (lossy) remedies include trimming extreme weights, restricting analysis to the region of common support, or fitting more flexible propensity models.

4.2.6 Generalizing to Continuous and Regression Settings

While our mathematical derivation focused on discrete, categorical data to explicitly show how IPW recovers the backdoor adjustment functional, the re-weighting concept is agnostic to your data type. The exact same inverse propensity weights can be applied to continuous outcomes, continuous covariates, and (with some care) continuous treatments.

If you are working with continuous outcomes and want to fit a regression, you do not need to manually calculate conditional probabilities. Instead, you compute the propensity scores, calculate the weight $w_i = 1/\Pr(A = a_i \mid X = x_i)$ for each individual — the inverse of the probability that unit i received its observed treatment given its covariates — and pass those weights directly into a Weighted Least Squares (WLS) regression:

$$\hat{\theta} = \text{argmin}_\theta \sum_{i=1}^{N} w_i \left(y_i - f_\theta(x_i, a_i)\right)^2 \tag{4.15}$$

By applying these weights to your loss function, the regression algorithm optimizes over the pseudo-population where confounding has been eliminated, granting your fitted parameters a valid causal interpretation.

Pitfall: WLS Can Quietly Re-introduce Homogeneity

The WLS recipe above is a valid IPW procedure, but the heterogeneity advantage from the start of the section depends on what you put inside $f_\theta(X, A)$. If you fit the simple linear model $f_\theta(X, A) = \theta_A A + \theta_X^\top X$, then even with IPW weights you have committed to a single scalar θ_A that doesn't vary with X — you are back to assuming homogeneous effects, just in a re-weighted sample. The propensity weights correct for confounding bias, but they cannot rescue a misspecified outcome model.
Two ways to genuinely preserve the heterogeneity property:

1. **Estimate each potential outcome separately.** Compute $\hat{\mathbb{E}}[Y^{(a)}] = \sum_i w_i \mathbb{1}[A_i = a] Y_i / \sum_i w_i \mathbb{1}[A_i = a]$ for each treatment level a, then take differences. No parameters are shared across treatment levels, so unit-level heterogeneity is preserved by construction.
2. **Include A-by-X interactions in f_θ.** For example, $f_\theta(X, A) = \theta_A A + \theta_X^\top X + \theta_{AX}^\top (A \cdot X)$. This lets the treatment effect vary with X, recovering a CATE function rather than collapsing it to a single slope.

The general takeaway: IPW removes the confounding-bias modeling burden, but if you reintroduce a parametric outcome model on top of the weights, that model still has to be flexible enough not to fight the heterogeneity you cared about in the first place.

This pitfall reveals a tension. IPW alone protects against confounding bias by avoiding outcome modeling entirely, but as soon as we want to estimate continuous outcomes through regression, an outcome model finds its way back in. A natural question is whether we can combine the two approaches — IPW for re-weighting *and* an outcome model for capturing structure — in a way that is robust to misspecification of either component. This is exactly the goal of the next section, on **doubly robust** estimation.

4.3 Doubly Robust Estimation

While outcome regression and inverse probability weighting (IPW) offer distinct ways to adjust for confounding, they are not mutually exclusive. In the 1990s, while exploring the mathematical limits of causal estimators and semi-parametric theory, statisticians Jamie Robins, Andrea Rotnitzky, and Lue Zhao discovered a beautiful property: They realized that by combining a model for the treatment assignment with a model for the outcome, they could create an estimator with a built-in statistical safety net [Robins et al., 1994].

$$\mathbb{E}[Y^{(a)}] = \mathbb{E}\left[\frac{\mathbb{1}[A = a]}{\Pr(A = a \mid \mathbf{X})} \times (Y - \mathbb{E}[Y \mid A, \mathbf{X}]) + \mathbb{E}[Y \mid A = a, \mathbf{X}]\right] \tag{4.16}$$

This estimator will yield a consistent and unbiased estimate of the causal effect as long as at least *one* of the underlying models—either the propensity score model or the outcome regression model—is correctly specified, even if the other is entirely wrong. This property became known as "double robustness."

4.3.1 Three Perspectives on AIPW

To intuitively understand how this safety net operates, it helps to view the Augmented IPW (AIPW) estimator from different angles. We can make sense of the exact same doubly robust formula from three distinct conceptual starting points:

Perspective 1: The Bias-Corrected Estimator

A natural starting point for estimating the expected potential outcome $\mathbb{E}[Y^{(a)}]$ is to train a machine learning model, $\hat{\mu}_a(\mathbf{X})$, to predict the outcome Y from covariates $\mathbf{X}$ among the treated population ($A = a$). We could then estimate the population-level effect by averaging our model's predictions over everyone: $\mathbb{E}[\hat{\mu}_a(\mathbf{X})]$.

However, our model is likely imperfect. To arrive at the true expected potential outcome, we need to correct our initial guess by adding the population-wide expected error, or the "counterfactual residual":

$$\mathbb{E}[Y^{(a)}] = \mathbb{E}[\hat{\mu}_a(\mathbf{X})] + \mathbb{E}[Y^{(a)} - \hat{\mu}_a(\mathbf{X})] \tag{4.17}$$

The first term is our model's guess, and the second term is the ideal bias correction. If we could compute this exactly, the equation would balance perfectly.

The problem is that we cannot observe the potential outcome $Y^{(a)}$ for everyone. We only have the actual outcome Y for the specific subgroup of individuals who actually received the treatment ($A = a$). If we try to estimate our model's bias by simply averaging the residuals of the people we *did* observe, we are computing a conditional expectation:

$$\mathbb{E}[Y - \hat{\mu}_a(\mathbf{X}) \mid A = a] \tag{4.18}$$

This residual is **confounded**. Because the treatment assignment depends on $\mathbf{X}$, the subgroup that received $A = a$ does not represent the general population. For example, if a treatment was primarily given to older patients, averaging the observed residuals only tells us about our model's error on older patients. We learn nothing about its error on the rest of the population.

To fix our biased estimator, we need to convert this confounded expectation back into a marginal expectation over the whole population. We accomplish this using Inverse Propensity Weighting (IPW). By multiplying the observed residuals by the inverse of the propensity score, we up-weight the underrepresented groups and mathematically force the covariates back to their marginal distribution:

$$\mathbb{E}\left[\frac{\mathbb{1}[A = a]}{\Pr(A = a \mid \mathbf{X})}(Y - \hat{\mu}_a(\mathbf{X}))\right] = \mathbb{E}[Y^{(a)} - \hat{\mu}_a(\mathbf{X})] \tag{4.19}$$

Adding this IPW-corrected residual back to our initial guess gives us the Augmented Inverse Propensity Weighting (AIPW) estimator:

$$\mu_a^{AIPW} = \hat{\mu}_a(\mathbf{X}) + \frac{\mathbb{1}[A = a]}{\Pr(A = a \mid \mathbf{X})}(Y - \hat{\mu}_a(\mathbf{X})) \tag{4.20}$$

This perspective reveals AIPW as a corrective exercise: we start with a biased machine learning model, identify the ideal counterfactual residual needed to fix it, and then use IPW to un-confound the observed residuals so they can successfully correct our baseline estimate.

Perspective 2: The Bias-Corrected IPW Estimator

Alternatively, rather than starting with the outcome model, we can start with the standard Inverse Propensity Weighting (IPW) estimator. To estimate the expected potential outcome $\mathbb{E}[Y^{(a)}]$, we reweight the observed outcomes:

$$\mathbb{E}[Y^{(a)}] = \mathbb{E}\left[\frac{\mathbb{1}[A = a]}{\Pr(A = a \mid \mathbf{X})}Y\right] \tag{4.21}$$

In expectation, this estimator is unbiased if the propensity score is correctly specified. However, in practice, the IPW estimator often suffers from high variance and finite-sample bias. If the probability of treatment $\Pr(A = a \mid \mathbf{X})$ is very small for certain individuals, the resulting extreme weights can cause the estimator to become highly unstable. Furthermore, in any finite sample, the IPW weights might randomly over-represent or under-represent certain covariate groups.

To stabilize this estimator, we need a way to measure how much the IPW weights are "drifting" or erring in our specific sample, and then subtract that error. We can measure this drift by testing the IPW estimator on a variable where we *already know* the true marginal expectation: our machine learning predictions, $\hat{\mu}_a(\mathbf{X})$.

Because we can compute the predicted outcome for every individual in the population, the true marginal expectation of the predictions is simply $\mathbb{E}[\hat{\mu}_a(\mathbf{X})]$. We can then see what the IPW estimator *guesses* this average should be:

$$\text{IPW Estimate of Predictions} = \mathbb{E}\left[\frac{\mathbb{1}[A=a]}{\Pr(A=a \mid \mathbf{X})}\hat{\mu}_a(\mathbf{X})\right] \tag{4.22}$$

By the Law of Iterated Expectations, if the IPW weights are perfectly balanced in our sample, this IPW estimate will equal the true expected prediction $\mathbb{E}[\hat{\mu}_a(\mathbf{X})]$. If it does not, the difference represents the exact finite-sample bias, or noise, introduced by the IPW weighting:

$$\text{IPW Error} = \mathbb{E}\left[\frac{\mathbb{1}[A=a]}{\Pr(A=a \mid \mathbf{X})}\hat{\mu}_a(\mathbf{X})\right] - \mathbb{E}[\hat{\mu}_a(\mathbf{X})] = \mathbb{E}\left[\left(\frac{\mathbb{1}[A=a] - \Pr(A=a \mid \mathbf{X})}{\Pr(A=a \mid \mathbf{X})}\right)\hat{\mu}_a(\mathbf{X})\right] \tag{4.23}$$

This error term acts as a **control variate**. If we assume the IPW estimator makes a structurally similar error on the *actual* outcomes Y as it does on the *predicted* outcomes $\hat{\mu}_a(\mathbf{X})$, we can correct the baseline IPW estimator by simply subtracting this known error:

$$\mathbb{E}[Y^{(a)}] = \mathbb{E}\left[\frac{\mathbb{1}[A=a]}{\Pr(A=a \mid \mathbf{X})}Y - \left(\frac{\mathbb{1}[A=a] - \Pr(A=a \mid \mathbf{X})}{\Pr(A=a \mid \mathbf{X})}\right)\hat{\mu}_a(\mathbf{X})\right] \tag{4.24}$$

Through basic algebra, this equation is perfectly equivalent to Perspective 1. However, the intuition is reversed: we start with a high-variance IPW estimator, use our outcome model to measure exactly how much the IPW weights are distorting the sample, and subtract that distortion to stabilize our final estimate.

Perspective 3: The Algebraic Cancellation

In previous sections, re-weighting the distribution allowed us to regress on a pseudo-population where the backdoor set was independent from our treatment (IPW). Conversely, the backdoor adjustment allowed us to use regression to capture the marginal dependence of the treatment relative to the adjustment set. It turns out that combining the two of them into the AIPW formula mechanically forces the errors to cancel out.

By distributing the product in the AIPW functional, we can isolate the components for IPW and the Backdoor Adjustment:

$$\mathbb{E}[Y^{(a)}] = \underbrace{\mathbb{E}\left[\frac{\mathbb{1}[A=a]}{\Pr(A=a \mid \mathbf{X})} \times Y\right]}_{\text{IPW}} + \underbrace{\mathbb{E}[\mathbb{E}[Y \mid A=a, \mathbf{X}]]}_{\text{Backdoor Adjustment}} - \mathbb{E}\left[\frac{\mathbb{1}[A=a]}{\Pr(A=a \mid \mathbf{X})}\mathbb{E}[Y \mid A, \mathbf{X}]\right] \tag{4.25}$$

If $\mathbb{E}[Y \mid A=a, \mathbf{X}]$ is correctly specified, then the backdoor adjustment term is correct. If $\Pr(A=a \mid \mathbf{X})$ is correctly specified, then the IPW term is correct. Conveniently, so long as one of the two is correctly specified, the final term will elegantly cancel out the incorrect term.

First, let's suppose that $\Pr(A=a \mid \mathbf{X})$ (blue means correct) is correctly specified, but that $\mathbb{E}[Y \mid A=a, \mathbf{X}]$ is incorrectly specified (red means incorrect). To avoid ambiguity, let a' denote the dummy variable we are summing over for the distribution of A, while a remains our fixed intervention. Expanding the third term

gives:

$$\begin{aligned}\mathbb{E}\left[\frac{\mathbb{1}[A=a]}{\Pr(A=a\mid\mathbf{X})}\mathbb{E}[Y\mid A,\mathbf{X}]\right] &= \sum_{a',\mathbf{x}}\Pr(a',\mathbf{x})\frac{\mathbb{1}[a'=a]}{\Pr(a\mid\mathbf{x})}\mathbb{E}[Y\mid a',\mathbf{x}]\\ &= \sum_{\mathbf{x}}\Pr(a,\mathbf{x})\frac{1}{\Pr(a\mid\mathbf{x})}\mathbb{E}[Y\mid a,\mathbf{x}]\\ &= \sum_{\mathbf{x}}\Pr(\mathbf{x})\mathbb{E}[Y\mid a,\mathbf{x}]\\ &= \mathbb{E}[\mathbb{E}[Y\mid A=a,\mathbf{X}]]\end{aligned}$$

Even though the red term is incorrectly specified, the correct specification of the IPW weights causes the third term to exactly cancel out with our incorrect backdoor adjustment term, leaving only the correct IPW term.

Now, suppose that $\Pr(A=a\mid\mathbf{X})$ is incorrectly specified, but that the outcome model $\mathbb{E}[Y\mid A,\mathbf{X}]$ is correctly specified. By the law of iterated expectations (and applying the same a' summation logic), we can easily collapse the expression:

$$\mathbb{E}\left[\frac{\mathbb{1}[A=a]}{\Pr(A=a\mid\mathbf{X})}\mathbb{E}[Y\mid A,\mathbf{X}]\right] = \sum_{a',\mathbf{x}}\Pr(a',\mathbf{x})\frac{\mathbb{1}[a'=a]}{\Pr(a\mid\mathbf{x})}\mathbb{E}[Y\mid a',\mathbf{x}] = \mathbb{E}\left[\frac{\mathbb{1}[A=a]}{\Pr(A=a\mid\mathbf{X})}Y\right]$$

Here, the correct specification of the outcome model causes the third term to exactly cancel out with our incorrect IPW term, leaving only the correct backdoor adjustment term.

4.4 Causal Machine Learning

While the statistics perspective is primarily based on an assumed model class (like linear models), machine learning takes an alternative perspective of refining and regularizing extremely powerful models like neural networks and random forests. Such models can also be used as structural equations to infer counterfactuals, but with a few caveats. Every approach in this section assumes we have a treatment A, an outcome Y, and covariates $\mathbf{X}$. The first three—the S-, T-, and X-learners—are *meta-learners* [Künzel et al., 2019]: recipes that turn any off-the-shelf regression algorithm into an estimator of the conditional average treatment effect (CATE). The fourth, Double Machine Learning [Chernozhukov et al., 2018], is instead a meta-*algorithm* for debiasing those nuisance estimates.

4.4.1 S-Learner

The most naive idea is to train an ML model that predicts Y using A and $\mathbf{X}$, and to compare $\hat{Y}(A=1,\mathbf{X}=\mathbf{x})$ to $\hat{Y}(A=0,\mathbf{X}=\mathbf{x})$ to compute a CATE for $\mathbf{X}=\mathbf{x}$. This is called an S-learner because it involves a **S**ingle model. While this *can* work, it's important to understand that ML models are trained with the sole goal of Level 1 (predictive) accuracy. Under this objective, an ML model has no reason to emphasize accurate estimation of the functional dependence on A: as far as it is concerned, A is just another coordinate of $\mathbf{X}$.

ML models are often extremely powerful and therefore subject to overfitting. High-dimensional data naturally creates this issue, meaning that regularization often has the effect of limiting the number of covariates that can influence the output (if all of them could, then we would overfit even with a linear model). As such, the ML model must make "cuts" to variables with weaker signals. As it turns out, causal signals can often be weaker (from a predictive perspective) than non-causal ones. In the homework, we will see that S-learners often give zero dependence on the causal variables.

4.4.2 T-Learner

To emphasize the model dependence on A, a T-learner trains **T**wo models, μ_1 on the $A=1$ data and μ_0 on the $A=0$ data. It then estimates the difference between those models, $\text{CATE}(\mathbf{x})=\mu_1(\mathbf{x})-\mu_0(\mathbf{x})$. As such, there is no way for the optimization to suppress the dependence on A: that dependence is now the gap between two separately fit models, not a single coefficient that regularization can shrink toward zero.

The problem with such an approach is "data starvation," since training these two models involves splitting the dataset into two subsets, D_0 with $A=0$ and D_1 with $A=1$. It turns out that confounding is especially devastating to this learner, because it means that specific regions of covariate space will have primarily $A=0$ or $A=1$ datapoints (such as when the severe cases of a disease are primarily treated and mild cases are primarily untreated). This means that at any covariate value $\mathbf{x}$ where there is some bias and confounding, *at least one* of the two models will be bad. Since our estimate is their difference, we can expect a poor estimate throughout those regions. Figure 4.5 shows this failure on simulated data: each model tracks the truth only where its own arm has data, so their difference inherits the error of whichever model is starved at that $\mathbf{x}$.

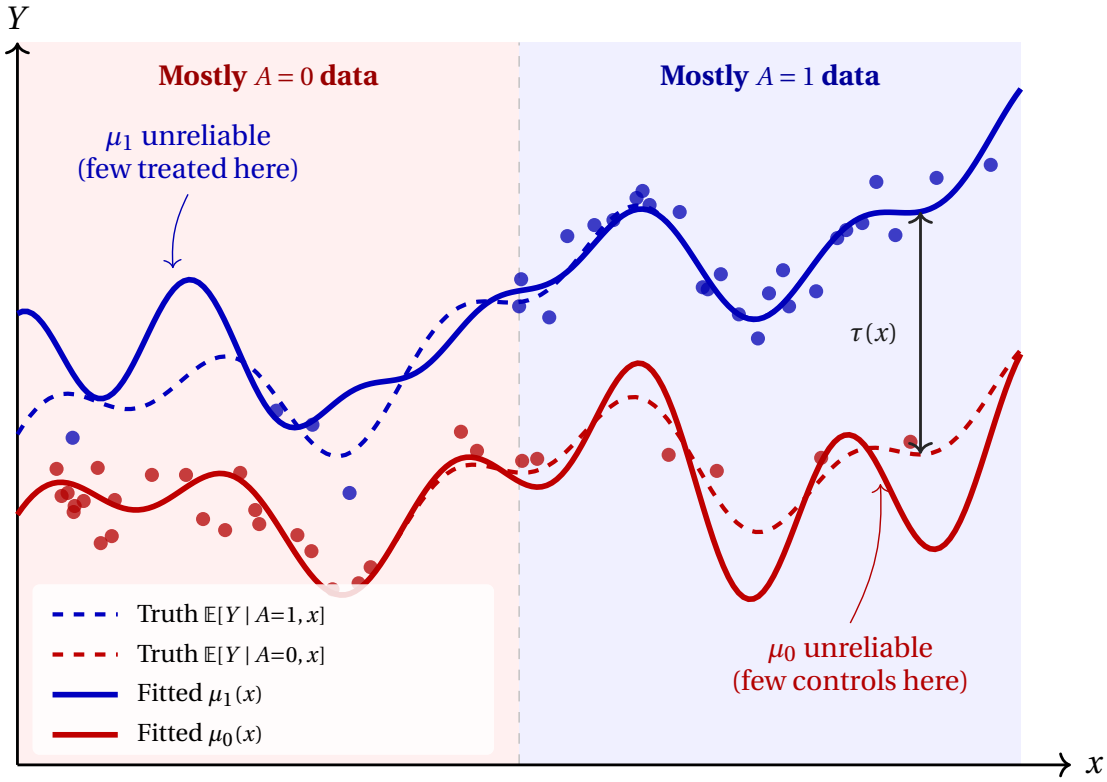


Figure 4.5: Complementary failure of the T-learner under confounding. The dashed curves are the ground truth surfaces $\mathbb{E}[Y \mid A{=}1, x]$ (blue) and $\mathbb{E}[Y \mid A{=}0, x]$ (red). Each surface is complicated, but their gap (the CATE $\tau(x)$) grows only linearly. Confounding concentrates the control data on the left and the treated data on the right, so each fitted model (solid) tracks the truth closely on its data-rich side and is biased and overfit on its data-poor side. At every x, at least one of μ_0 and μ_1 is unreliable, which is why the T-learner struggles. But the other model is reliable at that same x, which is exactly what the X-learner's propensity-weighted blend exploits.

4.4.3 X-Learner

When one of the two T-learner models is unreliable in some region of covariate space, the other is typically well-estimated there. In Figure 4.5, μ_0 is accurate exactly where μ_1 is not, and vice versa. This is a powerful property that the X-learner exploits by interpolating between two estimates of the CATE—one built from μ_1 and one from μ_0.

The idea is to impute a treatment effect for every unit using the model fit on the *opposite* arm:

$$D^1 = Y - \mu_0(\mathbf{X}) \quad \text{(for treated)}$$

$$D^0 = \mu_1(\mathbf{X}) - Y \quad \text{(for control)}$$

We then fit two new ML models to these imputed effects: $\tau_1(\mathbf{X})$ on D^1 (the treated units) and $\tau_0(\mathbf{X})$ on D^0 (the controls). Finally, we blend the two using the propensity score $\pi(\mathbf{X})$ (estimated, e.g., by logistic regression) as the weight:

$$\hat{\tau}(\mathbf{X}) = \pi(\mathbf{X})\tau_0(\mathbf{X}) + (1 - \pi(\mathbf{X}))\tau_1(\mathbf{X})$$

Notice that a large $\pi(\mathbf{x})$ means patients with $\mathbf{X} = \mathbf{x}$ are very likely to be treated, so $\mu_1(\mathbf{x})$ is estimated from plenty of data and is accurate; the imputed effect $D^0 = \mu_1 - Y$ is therefore reliable, and so is τ_0. The weighting gives τ_0 correspondingly more weight.

4.4.4 Double Machine Learning (DML)

While doubly robust estimators give us a safety net against model misspecification, they still typically rely on traditional parametric models. When dealing with high-dimensional confounders or complex, non-linear relationships, we might want to estimate the underlying models—$\mathbb{E}[Y \mid \mathbf{X}]$ and $\Pr(A \mid \mathbf{X})$—using modern machine learning algorithms.

In causal inference, these intermediate models are often referred to as **nuisance parameters** (which specify nuisance functions). The intuition behind the name is simple: we do not actually care about their specific values or predictions. Our only goal is to uncover the causal effect of the treatment. However, because we must adjust for confounding, estimating the relationships between the covariates, the treatment, and the outcome is a necessary "nuisance" we have to deal with along the way.

However, naively plugging machine learning predictions into our causal estimators leads to severe bias. Machine learning models rely heavily on regularization (e.g., tree depth limits, Lasso penalties) to prevent overfitting, which inherently biases their predictions. Double Machine Learning (DML), introduced by Chernozhukov et al. [2018], resolves this by framing causal inference not as a single model, but as a *meta-algorithm* built on two key statistical components: **Neyman Orthogonality** and **Cross-Fitting**.

1. Neyman Orthogonality

An estimator is Neyman orthogonal if it is highly insensitive to small, local estimation errors in the underlying ML nuisance parameters. Because machine learning models use regularization (which introduces bias) and converge slowly, their predictions are inherently "flawed" from a traditional inference standpoint. Orthogonality acts as a mathematical shock absorber: it ensures that the first-order bias from these flawed ML models vanishes. This allows our causal estimate $\hat{\theta}$ to converge at the standard, fast $\sqrt{n}$ rate, essentially shielding our target parameter from the slower convergence of the machine learning algorithms.

The specific orthogonal equation we use depends on our treatment type:

For Binary Treatments (AIPW): It turns out that the Augmented IPW functional derived in Section 4.3 is exactly the Neyman orthogonal score for the Average Treatment Effect (ATE). By predicting $\hat{\mathbb{E}}[Y \mid A, \mathbf{X}]$ and $\hat{\Pr}(A \mid \mathbf{X})$ using arbitrary ML models and plugging them into the AIPW formula, the "error correction" properties of the doubly robust estimator naturally satisfy the orthogonality requirement.

For Continuous Treatments (Partialling Out): When A is continuous, dividing by a propensity density $p(A \mid \mathbf{X})$ that can approach zero causes the AIPW estimator to explode. DML adapts the Frisch-Waugh-Lovell (FWL) theorem to handle this.

Suppose we have a *partially linear* model:

$$Y = \theta A + g(\mathbf{X}) + \epsilon, \quad \mathbb{E}[\epsilon \mid A, \mathbf{X}] = 0 \tag{4.26}$$

$$A = m(\mathbf{X}) + \eta, \quad \mathbb{E}[\eta \mid \mathbf{X}] = 0 \tag{4.27}$$

The "partial linearity" lives entirely in the treatment: θ is the same scalar across all units, and A's effect on Y is linear. The functions g and m, however, are left completely unspecified — they may be arbitrarily complex, non-linear functions of high-dimensional covariates $\mathbf{X}$.

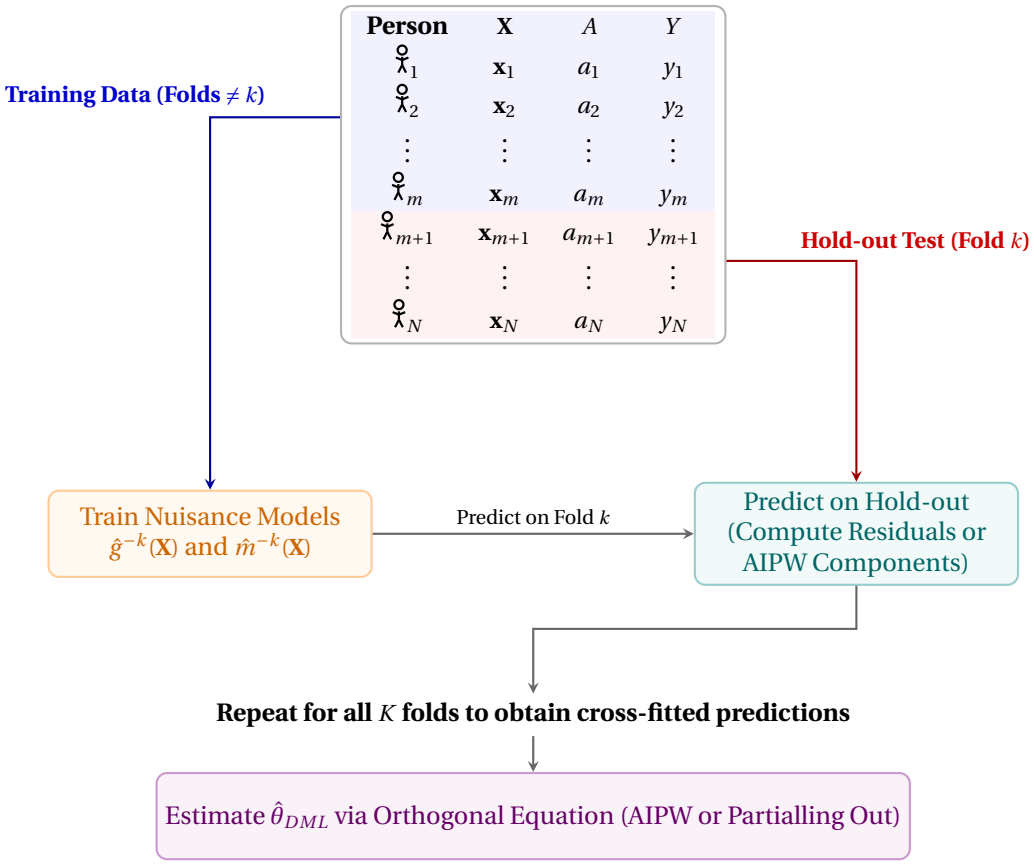


Figure 4.6: The cross-fitting procedure in Double Machine Learning. The dataset is partitioned into training and hold-out folds. For each fold k, machine learning nuisance models are trained exclusively on the blue out-of-fold data. These models are then used to predict and compute the orthogonal components for the red hold-out observations, thereby eliminating own-observation bias.

This is exactly where machine learning earns its place. If we used OLS to estimate g and m, we would implicitly assume they were linear in $\mathbf{X}$, and any non-linearity in the true g or m would corrupt our residuals and bias $\hat{\theta}$. ML methods (gradient boosting, random forests, neural nets, Lasso) can capture non-linear, high-dimensional dependence on $\mathbf{X}$ without committing to a functional form.

The procedure has three steps, only the first of which involves ML:

1. **Estimate the nuisances with ML.** Fit $\hat{g}(\mathbf{X})$ and $\hat{m}(\mathbf{X})$ using any flexible ML method (with cross-fitting — see below).
2. **Residualize.** Compute $\tilde{Y} = Y - \hat{g}(\mathbf{X})$ and $\tilde{A} = A - \hat{m}(\mathbf{X})$. These residuals are what Y and A look like after the (possibly non-linear) effects of $\mathbf{X}$ have been "partialled out."
3. **Estimate θ by OLS on the residuals.** Just regress $\tilde{Y}$ on $\tilde{A}$:

$$\hat{\theta} = \frac{\mathbb{E}[\tilde{A}\tilde{Y}]}{\mathbb{E}[\tilde{A}^2]}.$$

The final step is plain OLS, not ML, which is initially counter-intuitive, since this is supposed to be a "machine learning" method. The partial-linearity assumption tells us θ is a single scalar with no $\mathbf{X}$-dependence, so once $\mathbf{X}$'s nuisance effects are removed, all that's left to estimate is a single slope. The score function $\psi(W;\theta,g,m) = (Y - \theta A - g(\mathbf{X}))(A - m(\mathbf{X}))$ satisfies Neyman orthogonality, which is what guarantees that small errors in $\hat{g}$ and $\hat{m}$ from regularized ML estimation don't bias the final $\hat{\theta}$.

If we relax the partial-linearity assumption and allow $\theta(\mathbf{X})$ to vary with covariates (i.e., heterogeneous treatment effects), the final step is no longer OLS — we'd use a second ML model to estimate $\theta(\mathbf{X})$ from the residuals (the R-Learner, Nie and Wager [2021]). The partial-linearity case is the simplest DML application; everything else scales up from here.

2. Cross-Fitting

Even with an orthogonal score, simply training and evaluating our ML models on the same dataset destroys the estimator's validity. This is due to **own-observation bias**. Machine learning models possess immense capacity and often overfit to the random noise (ϵ_i) of their specific training data. If observation i is used to train $\hat{g}(\mathbf{X})$, and we then use that model to predict $\hat{g}(\mathbf{X}_i)$ to compute the residual for observation i, the prediction will be spuriously correlated with the true underlying error term ϵ_i. This correlation prevents the expected value of our score function from centering at zero, dragging the final estimate away from the true causal effect.

Cross-fitting removes this bias by ensuring that no observation ever helps fit the model that is later used to score it. We split the data into K folds. For each fold k, we fit the nuisance models $\hat{g}^{-k}$ and $\hat{m}^{-k}$ on the other $K-1$ folds and use them *only* to compute the orthogonal score (the residuals, or the AIPW components) for the held-out observations in fold k. Rotating through all K folds yields a full set of out-of-fold scores, which we average to obtain $\hat{\theta}_{DML}$; in practice one often repeats the random fold assignment a few times and averages the results to reduce dependence on the particular split. Because the prediction for observation i never saw ϵ_i during training, the spurious correlation vanishes and the score recenters at zero.

Together, orthogonality and cross-fitting decouple the estimation of the nuisance functions from the estimation of the causal parameter. This is what lets DML deliver valid inference — unbiased estimates and correctly calibrated confidence intervals — while still fully leveraging the predictive power of modern machine learning.

4.5 Synthetic Controls

4.5.1 The Netflix Problem and Matrix Completion

In 2006, Netflix offered a one-million-dollar prize to anyone who could improve their recommendation algorithm by 10%. The challenge was fundamentally a matrix completion problem. Imagine a massive grid where each row represents a user, and each column represents a movie. The entries in the matrix are the ratings (from 1 to 5 stars) that users gave to movies.

Because most users have only watched a tiny fraction of the available movies, this matrix is overwhelmingly empty. The goal of the algorithm is to guess the missing entries.

To do this, we can assume the matrix has a low-rank structure. This means that human preferences are not completely random; they are driven by a few underlying latent factors (like a preference for action, comedy, or a specific director). Because of these shared underlying factors, the rows of the matrix are not entirely independent. We can often approximate an incomplete row as a linear combination of other, complete rows.

Consider a simple 3×4 matrix representing three users and four movies, where the last entry is missing:

$$M = \begin{bmatrix} 1 & 0 & 2 & 3 \\ 0 & 1 & 1 & 2 \\ 2 & 1 & 5 & ? \end{bmatrix}$$

If we assume this matrix is low-rank, we can look for relationships between the users. Notice that the ratings of the third user (Row 3) can be perfectly reconstructed by taking two times the first user's ratings plus one times the second user's ratings: $\text{Row}_3 = 2 \cdot \text{Row}_1 + 1 \cdot \text{Row}_2$.

By applying this exact same linear combination to the final column, we can easily impute the missing value: $2(3) + 1(2) = 8$. The missing entry is 8, an intuition visualized in Figure 4.7(a).

It is worth noting that this exact matching process is a simplified, idealized picture of matrix completion. In practice, real-world data is noisy, and exact linear dependence is rare. Advanced matrix completion algorithms (like alternating least squares or nuclear norm minimization) approximate these missing values rather than relying on perfect, deterministic row matches, but the fundamental logic of borrowing information from similar rows remains the same.

4.5.2 Matrix Rank and Imputability

The success of this matching process depends heavily on the rank of the matrix, denoted as r. The rank represents the number of underlying latent factors driving the observed data. If a matrix has rank r, any row can be expressed as a linear combination of a basis containing *at most* r linearly independent rows.

If our target unit lies in a lower-dimensional subspace spanned by just k donor units (where $k \leqslant r$), we only need those k donor rows to form the perfect linear combination. Furthermore, to solve for the unique weights applied to those k donors, we only strictly need them to match on k linearly independent columns. While having them match on $> r$ entries provides excellent overdetermination to verify the relationship, it is not strictly required.

Theorem 4.5.1 (Low-Rank Matrix Completion via Row Combinations)**.** *Let M be an $N \times T$ matrix of rank r. Suppose row i has a missing entry in column t, but is observed in a list of columns $\mathbf{c}$. To perfectly impute M_{it} via a linear combination of other rows:*

1. *We need a donor pool $\mathbf{d}$ of $k \leqslant r$ rows that are fully observed in column t such that row i lies in the span of $\mathbf{d}$.*
2. *The donor rows must be observed in at least k overlapping columns within $\mathbf{c}$.*
3. *The $k \times k$ sub-matrix formed by the donor rows restricted to these k overlapping columns must be invertible (i.e., the equations must be linearly independent).*

If these conditions hold, we can uniquely solve for the k weights, $\mathbf{w}$, and perfectly impute the missing entry as $M_{it} = \sum_{j \in \mathbf{d}} w_j M_{jt}$.

Proof: Let the row space of M have dimension r. By definition, any row vector M_i can be written as a linear combination of a basis containing at most r rows. Thus, there exists a list of donor rows $\mathbf{d}$ with $|\mathbf{d}| = k \leqslant r$ such that $M_i = \sum_{j \in \mathbf{d}} w_j M_j$.

To compute the unknown weights w_j, we restrict our view to the list of observed columns $\mathbf{c}$. This yields a linear system of $|\mathbf{c}|$ equations: $M_{ic} = \sum_{j \in \mathbf{d}} w_j M_{jc}$ for all $c \in \mathbf{c}$.

As long as we have $|\mathbf{c}| \geqslant k$ observed columns and the $k \times |\mathbf{c}|$ sub-matrix restricted to those columns has full row rank k (condition 3), this system of equations is perfectly determined (or overdetermined) and has a unique solution for the weight vector w. Once w is solved, we apply these exact weights to the unobserved column t to synthesize the missing entry: $M_{it} = \sum_{j \in \mathbf{d}} w_j M_{jt}$. ■

In general, the real world is too noisy for our data to ever form a perfectly low-rank, deterministic matrix. Because of random error, any real-world data matrix will technically be full rank. Still, we can perform decompositions like the Singular Value Decomposition (SVD) to denoise the data, isolate the most important latent factors, and closely approximate the missing counterfactuals using this underlying lower-rank structure.

4.5.3 Potential Outcomes as Missing Data

Earlier, we discussed the Fundamental Problem of Causal Inference: we can never observe both potential outcomes, $Y^{(1)}$ and $Y^{(0)}$, for the same individual at the same time.

We can actually view this fundamental problem as a matrix completion problem, as demonstrated in Figure 4.7(b). If we create a matrix where the rows are individual units and the columns are covariates X followed by potential outcomes $Y^{(0)}$ and $Y^{(1)}$, exactly half of our outcome matrix will be missing. When we use methods like matching or Inverse Propensity Weighting (IPW), we are effectively trying to fill in these missing counterfactuals by looking at similar rows (units) in the observed data.

Main Idea 19

Imputing potential outcomes is a matrix completion problem that can leverage the data's low-rank structure to infer counterfactuals.

Exercise 4

What happens if conditional exchangeability does not hold when conditioning on all of the covariates in the matrix? Is the approach still valid?

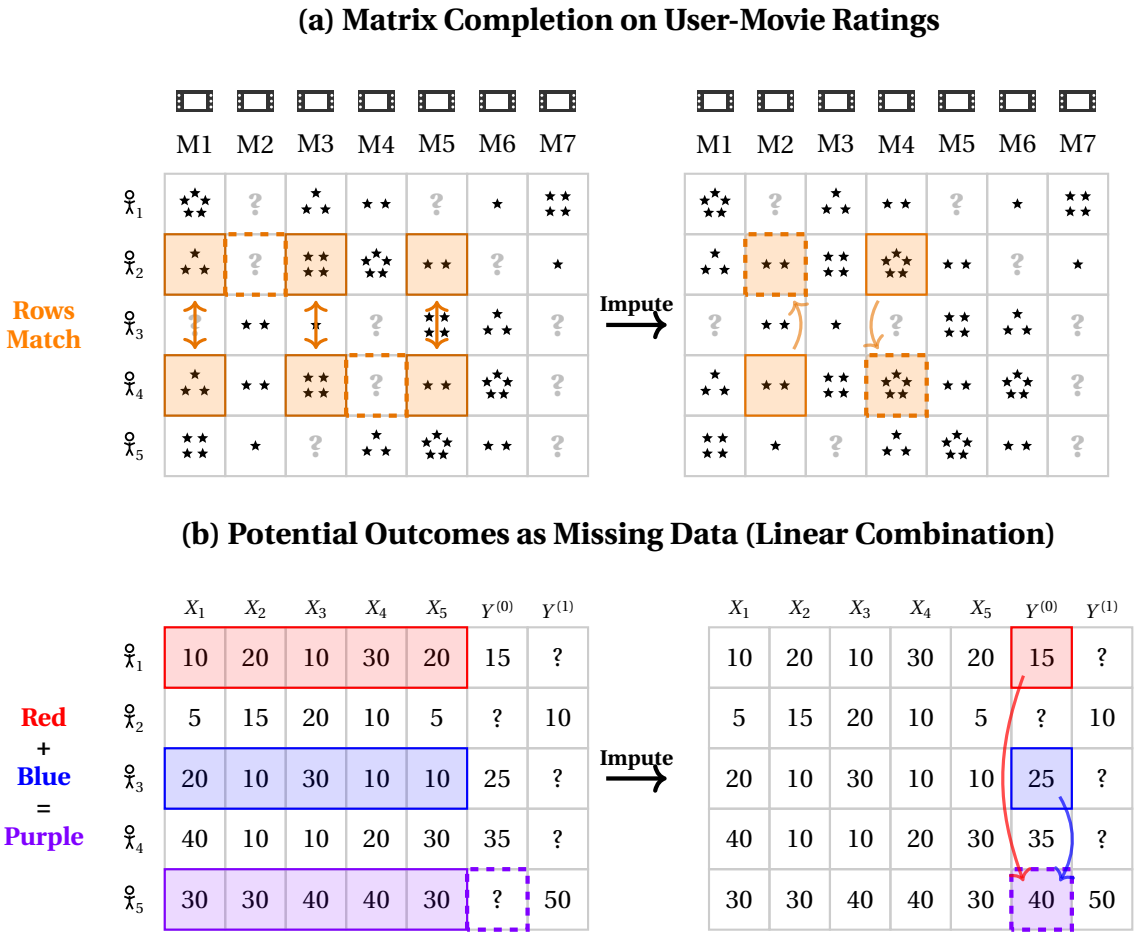


Figure 4.7: Matrix completion and potential outcome imputation using linear combinations. The top example shows two rows that match exactly, making the entry easy to match. The bottom example requires a linear combination (adding the rows).

4.5.4 Synthetic Controls

The Synthetic Control method extends this missing data intuition to time-series data, a dynamic illustrated in Figure 4.8. Instead of just two columns for treatment and control, imagine a matrix where the rows are still different units (such as states, countries, or companies), but the columns are sequential time steps: $t = 1, 2, \ldots, T$.

Suppose unit 1 receives a policy intervention (the treatment) at time T_0. For all time steps before T_0, we observe the untreated outcome for all units. However, for time steps $t > T_0$, unit 1 is treated, meaning its untreated potential outcome, $Y^{(0)}_{1t}$, is entirely missing.

Just like in the Netflix problem, we can reconstruct this missing data by finding a linear combination of the other, untreated rows (called the “donor pool").

If we can find a set of weights $w_2, w_3, \ldots, w_N$ such that the weighted sum of the donor units perfectly matches the treated unit in the *pre-treatment* period, we can use those same weights to project what the treated unit would have looked like in the *post-treatment* period.

Formally, we find weights $\hat{\mathbf{w}}$ that minimize the pre-treatment prediction error:

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \sum_{t=1}^{T_0} \left(Y_{1t} - \sum_{j=2}^{N} w_j Y_{jt} \right)^2$$

To avoid extrapolation and ensure our synthetic unit is grounded in reality, researchers typically constrain these weights so that they are non-negative ($w_j \geqslant 0$) and sum to one ($\sum w_j = 1$).

Once we have learned these optimal weights from the pre-treatment columns, we apply them to the post-treatment columns to impute our missing counterfactual:

$$\hat{Y}_{1t}^{(0)} = \sum_{j=2}^{N} \hat{w}_j Y_{jt} \quad \text{for } t > T_0$$

The resulting sequence, $\hat{Y}_{1t}^{(0)}$, is our “Synthetic Control"—a mathematically constructed clone of our treated unit that did not receive the treatment.

Exercise 5

What rung of causality are synthetic controls addressing? Does it differ depending on whether or not you need to apply the SVD to lower the rank?

The Magic of Panel Data: Bypassing Unobserved Confounders

Why is matching on pre-treatment outcomes so much more powerful than traditional cross-sectional matching?

In standard matching, we try to match units based on observed covariates $\mathbf{X}$. But if there are unobserved variables driving the outcome, our estimates are biased.

Synthetic controls use the time dimension to bypass this. Suppose the world contains k unobserved factors that fluctuate over time — say, the strength of a national health movement, oil prices, or background economic conditions — captured by $U_{1t}, \ldots, U_{kt}$. Each unit i has its own time-invariant *sensitivity* to each of these factors, $\beta_{i1}, \ldots, \beta_{ik}$. The untreated outcome is a simple inner product:

$$Y_{it}^{(0)} = \beta_{i1} U_{1t} + \cdots + \beta_{ik} U_{kt} = \beta_i^\top \mathbf{U}_t,$$

where $\beta_i, \mathbf{U}_t \in \mathbb{R}^k$. We don't observe either vector.

The synthetic control's idea. If we can find a weighted combination of donor outcomes that tracks the treated unit across many pre-treatment periods, we will have implicitly matched the hidden sensitivities. The pre-treatment matching condition is

$$Y_{1t} = \sum_{j=2}^{N} w_j Y_{jt} \quad \text{for } t = 1, \ldots, T_{\text{pre}}.$$

Substituting the factor model on both sides and pulling the weights inside the inner product:

$$\beta_1^\top \mathbf{U}_t = \left(\sum_{j=2}^{N} w_j \beta_j \right)^\top \mathbf{U}_t.$$

Define the *sensitivity deficit*

$$\delta := \beta_1 - \sum_{j=2}^{N} w_j \beta_j,$$

which measures how far the synthetic unit's sensitivities are from the target's. The matching condition collapses to

$$\delta^\top \mathbf{U}_t = 0 \quad \text{for every pre-treatment period } t.$$

Each pre-treatment period forces δ to be orthogonal to one more direction in $\mathbb{R}^k$.

There are two conditions for exact reconstruction:

- **Enough time:** If our T_{pre} periods include k time steps where the $\mathbf{U}_t$ vectors are linearly independent (spanning $\mathbb{R}^k$), then δ is orthogonal to all of $\mathbb{R}^k$, and the only such vector is the zero vector. So $T_{\text{pre}} \geq k$ with linearly independent shocks forces $\delta = \mathbf{0}$.
- **Enough donors:** For $\delta = \mathbf{0}$ to be *achievable* in the first place, the target β_1 must lie in the span of the donor sensitivities $\{\beta_j\}_{j=2}^N$. With k donors whose sensitivities are linearly independent, the span is all of $\mathbb{R}^k$, and the weights w_j that produce the match are uniquely determined.

When both conditions hold, linear algebra gives us

$$\beta_1 = \sum_{j=2}^{N} w_j \beta_j.$$

The weighted combination of donor sensitivities is an exact copy of the treated unit's sensitivities, even though we never observed either side. The long pre-treatment history did the work: k rounds of independently-moving shocks reveal a k-dimensional hidden vector.

Why time variation is essential. If the $\mathbf{U}_t$'s were constant in t, every pre-treatment year would force δ to be orthogonal to the *same* direction, and we'd have one constraint no matter how long we waited. The latent factors must actually move, in linearly independent directions, for each new period to contribute new information.

Main Idea 20

By matching tightly on a long history of pre-treatment outcomes where temporal shocks vary independently, Synthetic Control implicitly balances unobserved, time-invariant confounders. We do not need to observe the confounders to control for them!

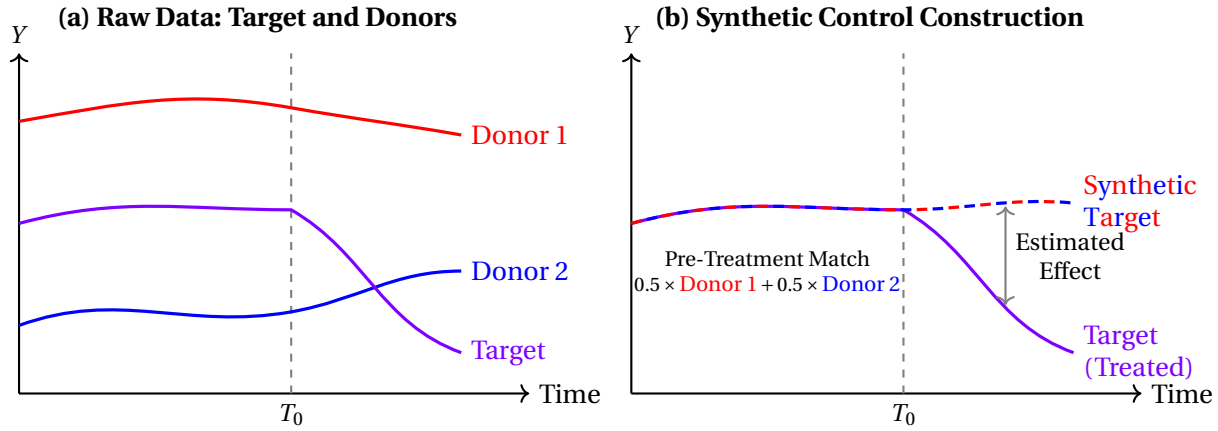


Figure 4.8: Constructing a synthetic control. In (a), the target unit (purple) lies between two donor units (red and blue). In (b), a weighted combination of the donors creates a "Synthetic Target" that matches the pre-treatment trend. The divergence post-T_0 reveals the causal effect.

4.5.5 A Real-World Example: Synthetic California

A famous application of this approach is Abadie, Diamond, and Hainmueller's 2010 study on the effects of Proposition 99, a major 1988 tobacco control program in California [Abadie et al., 2010]. The researchers wanted to know how many fewer packs of cigarettes were sold per capita as a direct result of the policy.

California is a unique state, so simply comparing its smoking rates to the national average or a single other state wouldn't provide a reliable counterfactual. Instead, the researchers used a donor pool of 38 states that did not implement similar tobacco policies.

By applying the synthetic control algorithm to pre-1988 data, they found that a weighted combination of five states—Colorado, Connecticut, Montana, Nevada, and Utah—perfectly mirrored California's historical cigarette sales. "Synthetic California" tracked real California almost exactly for decades leading up to 1988.

Why did this specific combination of states make such a good clone? It was not just because they had similar observed demographics. By matching closely on decades of actual cigarette sales, the algorithm inherently selected a blend of states that perfectly captured California's *unobserved* latent factors—such as unmeasured cultural attitudes toward health or state-specific economic shocks—during that era.

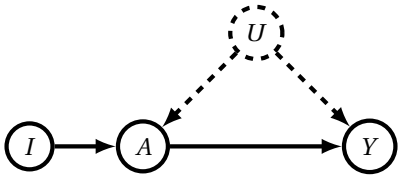
After 1988, however, the two trends drastically diverged. Real California's cigarette sales plummeted, while Synthetic California's sales declined much more slowly. The gap between these two lines provided a highly credible estimate of Proposition 99's causal effect: a reduction of about 26 packs of cigarettes per capita by the year 2000.

4.6 Path Analysis and Instrumental Variables

So far, our methods for causal identification have focused primarily on adjusting for confounding (such as using the backdoor criterion to block spurious paths and condition on observable variables). However, there are many real-world settings where we cannot observe or adjust for all confounders. We've already learned about the frontdoor adjustment, but it turns out we can sometimes also rely on natural sources of randomization that act like Randomized Controlled Trials (RCTs).

A classic example of this is estimating the effect of military service on lifetime income. We might suspect that individuals who serve in the military differ systematically from those who do not (e.g., in socioeconomic background or civilian opportunities), which hopelessly confounds the relationship. However, during the Vietnam War, the U.S. government used a draft lottery based on birthdays to determine who was called to serve. Inherently, there should be absolutely no association between a person's random birthday and their future income. Yet, there was! Because birthdays randomly determined draft eligibility—which in turn heavily influenced whether someone served in the military—researchers could use these birthdays as a natural source of randomization to uncover the true causal effect of military service on earnings [Angrist, 1990].

When a variable like a birthday acts as a random nudger for a treatment, we call it an *instrumental variable*. A canonical causal diagram for this setting explicitly shows the instrument (I) alongside the unobserved confounder (U):



Let's explore how we can formalize this idea using path analysis. If we assume that all equations in our structural causal model are linear with independent noise, then we greatly simplify causal identifiability.

4.6.1 Association

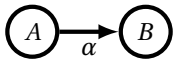
Let's define "association" to be the linear coefficient from regression. For example, if we have

$$A = N_A,$$
$$B = \alpha A + N_B,$$

where N_A, N_B are additive noise and $\text{Var}(A) = \text{Var}(B) = 1$, we will then say

$$\text{assoc}(A, B) = \frac{\text{Cov}(A, B)}{\text{Var}(A)} = \alpha.$$

You can think of an association as something like a derivative. When all variances of the random variables are 1 (which can be achieved when we normalize our data), covariances and associations are the same. We will draw these graphs by putting the association on the arrow.



There are two equivalent ways to read this α, and both are worth holding in mind:

1. It is the *forward structural coefficient* — the α in the equation $B = \alpha A + N_B$. "If I intervene to push A up by one unit, B goes up by α."
2. It is the *regression coefficient of B on A* — the slope you would estimate by running OLS. "If I observe A to be one unit higher, my best linear prediction of B goes up by α."

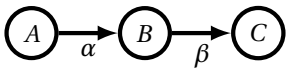
Under the variance normalization, these coincide numerically. They also coincide with the correlation $\text{Corr}(A, B)$, which is symmetric. As a consequence, traversing the arrow in either direction gives the *same* coefficient:

$$\text{assoc}(A, B) = \text{assoc}(B, A) = \alpha.$$

This symmetry is the engine of path analysis. It is worth pausing on, because it can feel counterintuitive: structurally inverting $B = \alpha A + N_B$ to solve for A would give the coefficient $1/\alpha$ on B, not α. But path analysis does not do structural inversion. It composes regression coefficients, and regression coefficients account for noise (you can never recover the value of A exactly from B when N_B is involved). The symmetry $\text{assoc}(A, B) = \text{assoc}(B, A) = \alpha$ holds precisely because regression projects rather than inverts.

4.6.2 Composing Paths

Consider the following graphical model and its corresponding structural equations:

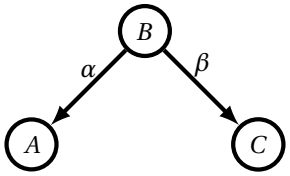


$$A = N_A,$$
$$B = \alpha A + N_B,$$
$$C = \beta B + N_C.$$

We want to write C in terms of A in order to find $\text{assoc}(A, C)$:

$$C = \beta[\alpha A + N_B] + N_C = \alpha\beta A + \beta N_B + N_C.$$

We conclude that $\text{assoc}(A, C) = \alpha\beta$. Notice that we have just multiplied the Greek letters along the path from A to C. The same thing happens when the path forks:



$$B = N_B,$$
$$A = \alpha B + N_A,$$
$$C = \beta B + N_C.$$

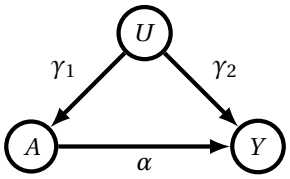
Direct computation gives $\text{assoc}(A, C) = \text{Cov}(A, C) = \alpha\beta$ (under unit variances) — the same multiplication rule, even though one leg of the path is now traversed against its arrow. This is the symmetry from the unit variance: the label on $B \to A$ is the same whether we read it forward or backward.

Main Idea 21

In a Linear SEM with normalized variances, we can compose associations along non-colliding paths. That is, if $A \leftarrow B \rightarrow C$ or $A \rightarrow B \rightarrow C$, then $\text{assoc}(A,C) = \text{assoc}(A,B) \times \text{assoc}(B,C)$.

4.6.3 More Than One Path

Now consider the following model:



$$U = N_U,$$
$$A = \gamma_1 U + N_A,$$
$$Y = \alpha A + \gamma_2 U + N_Y.$$

There are now two non-colliding paths from A to Y: the direct causal path $A \rightarrow Y$ (carrying α), and the backdoor path $A \leftarrow U \rightarrow Y$ (carrying $\gamma_1\gamma_2$). Both contribute to the association:

$$\text{assoc}(A,Y) = \underbrace{\alpha}_{\text{causal path}} + \underbrace{\gamma_1\gamma_2}_{\text{backdoor path}}.$$

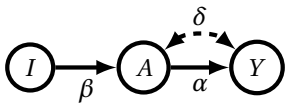
So association is a biased estimator for α, with the bias determined by the backdoor path.

Main Idea 22

$\text{assoc}(A,C)$ is the sum across all non-colliding paths from A to C of the products of associations along each of those paths.

4.6.4 Instrumental Variables

Let's collapse the backdoor $\gamma_1\gamma_2$ into a single combined coefficient δ (it doesn't matter for the analysis whether the confounding comes from one variable U or several — only their net path matters). Now suppose we also have a separate variable I that nudges the treatment A but otherwise has no relationship with Y. This is sometimes called a soft intervention or shift intervention because it just nudges A rather than forcing it to a specific value. We will treat I as an observed random variable.



$$I = N_I,$$
$$A = \beta I + \gamma_1 U + N_A,$$
$$Y = \alpha A + \gamma_2 U + N_Y.$$

The causal effect we want to find is α. Applying the path-analysis rules, the pairwise associations are

$$\text{assoc}(I,A) = \beta,$$
$$\text{assoc}(I,Y) = \beta\alpha,$$
$$\text{assoc}(A,Y) = \alpha + \delta.$$

Notice that $\text{assoc}(A,Y)$ is contaminated by the backdoor coefficient δ — this is the original confounding problem. But $\text{assoc}(I,A)$ and $\text{assoc}(I,Y)$ are both clean (the instrument has no backdoor to A or to Y). Dividing one by the other gives the IV ratio (sometimes called the Wald Ratio):

$$\alpha = \frac{\text{assoc}(I,Y)}{\text{assoc}(I,A)} = \frac{\beta\alpha}{\beta}. \tag{4.28}$$

The β cancels, and the causal effect α is identified.

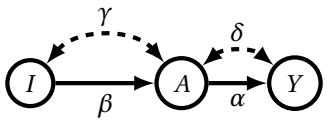
It turns out that variables like I occur somewhat frequently. When I occurs naturally, it is called an instrumental variable (or IV, for short). The path analysis we just went through was invented by Sewall Wright in 1921, and the first known instance of an instrumental variable is by his father, Philip Wright, in 1928. More recently, Angrist, Imbens, and Card have popularized the use of instrumental variables in economics, which earned them a Nobel Prize in 2021. There are three conditions to use I as an instrumental variable to determine the causal effect of A on Y (in addition to linearity):

1. **Exclusion Restriction:** I must cause Y only through A.
2. **Marginal Ignorability:** $I \perp\!\!\!\perp Y^{(a)}$, meaning the instrument is independent of the potential outcomes. Graphically, this means I and Y share no unobserved confounders (no backdoor paths between I and Y).
3. **IV Relevance:** $I \not\perp\!\!\!\perp A$, i.e., I must be associated with A in some way.

The exclusion restriction and marginal ignorability can be understood through parameter counting. For example, if we had an edge $I \to Y$ with a corresponding γ (violating the exclusion restriction), we would have 4 unknowns. Furthermore, a bidirected edge $I \leftrightarrow Y$ (violating marginal ignorability) would also add another parameter for that edge. It is worth noting that IV settings are not necessarily identifiable; they require linearity and some general dimensionality constraints.

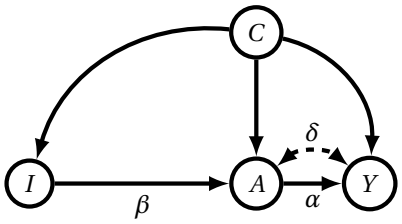
4.6.5 Notes on IV Identification

There are a few extensions to the IV setting that are worth considering. The first thing to note is that the exclusion restriction does not necessarily require $I \to A$ to be a direct causal link — this link could also be confounded:



This case adds an extra parameter (γ) but, perhaps surprisingly, α is still identifiable. We explore this in a homework question.

It is also worth noting that marginal ignorability can be relaxed to *conditional* ignorability. For example:



Now, we can condition on C to get conditional ignorability and apply an IV to get the causal effect for a specific $C = c$. These causal effects can then be summed or integrated over different values of C to recover the overall ATE.

When Effects Are Heterogeneous: LATE and the Compliers

Everything in this section leans on linearity with a single structural coefficient α, which implicitly assumes every unit responds to treatment identically. We have discussed CATEs in the past, but there is a noteworthy re-partitioning of sub-groups due to Imbens and Angrist [1994]. If we drop linearity and let both the instrument and the treatment be binary, then each unit now has two one-step ahead potential *treatments*, $A^{(I=1)}$ and $A^{(I=0)}$ (written $A^{(1)}$ and $A^{(0)}$ for short) — what they would do if drafted versus not. This partitions the population into four types:

- **Always-takers** ($A^{(1)} = A^{(0)} = 1$): volunteer whether drafted or not.
- **Never-takers** ($A^{(1)} = A^{(0)} = 0$): avoid service whether drafted or not.
- **Compliers** ($A^{(1)} = 1$, $A^{(0)} = 0$): serve if and only if drafted.
- **Defiers** ($A^{(1)} = 0$, $A^{(0)} = 1$): serve if and only if *not* drafted.

The instrument, by definition, only moves the compliers and defiers — the always- and never-takers ignore it. A key new assumption is **Monotonicity**: there are no defiers ($A^{(1)} \geqslant A^{(0)}$ for every unit). The draft may push people into service, but it never pushes anyone *out*. Now decompose the two clean associations in the IV ratio. The denominator is the fraction of the population the instrument actually moves:

$$\text{assoc}(I, A) = \mathbb{E}[A \mid I = 1] - \mathbb{E}[A \mid I = 0] = \Pr(\text{complier}).$$

For the numerator, the always- and never-takers receive the same treatment under either value of I, so by the exclusion restriction their outcomes do not change; only the compliers contribute:

$$\text{assoc}(I, Y) = \mathbb{E}[Y \mid I = 1] - \mathbb{E}[Y \mid I = 0] = \mathbb{E}\left[Y^{(1)} - Y^{(0)} \mid \text{complier}\right] \cdot \Pr(\text{complier}).$$

Dividing, the $\Pr(\text{complier})$ cancels — exactly as β canceled in the linear analysis — and we are left with the **Local Average Treatment Effect**:

$$\frac{\mathbb{E}[Y \mid I = 1] - \mathbb{E}[Y \mid I = 0]}{\mathbb{E}[A \mid I = 1] - \mathbb{E}[A \mid I = 0]} = \mathbb{E}\left[Y^{(1)} - Y^{(0)} \mid \text{complier}\right] =: \text{LATE}.$$

As a sanity check, if treatment effects are homogeneous, then the compliers are representative of the whole population, so LATE = ATE and we recover this section's linear result — with monotonicity standing in for linearity as the price of identification. While the CATE recovers effects for subgroups defined by similar *observable* confounders, the LATE partitions the population by its *latent* response to the instrument — no individual's complier status is ever observable (a drafted man who served could be a complier or an always-taker). For example, the draft lottery identifies the effect of service among men who served *because* they were drafted, not among volunteers. The IV ratio averages treatment effects over exactly the subpopulation whose behavior the instrument changes and nobody else. Two valid instruments for the same treatment can disagree, because they move different groups of compliers.

4.6.6 Two-Stage Least Squares (2SLS) and Machine Learning

The ratio of associations $\alpha = \text{assoc}(I, Y)/\text{assoc}(I, A)$ is an elegant theoretical result, but in practice, calculating this ratio directly from sample covariances can be quite limiting. It only works easily for a single instrument, a single treatment, and no additional covariates.

To estimate instrumental variables in more complex, real-world datasets, researchers rely on a technique called **Two-Stage Least Squares (2SLS)**. While 2SLS is a classical linear regression technique, understanding its two-step architecture is essential because it forms the exact foundation for modern machine learning extensions in causal inference.

As the name suggests, 2SLS breaks the IV process into two separate prediction tasks:

1. **Stage 1: Isolate the Exogenous Signal.** First, we regress the treatment A on the instrument I (along with any observed covariates C we need to condition on):

   $$\hat{A} = \hat{\beta}_0 + \hat{\beta}_1 I + \hat{\beta}_2 C.$$

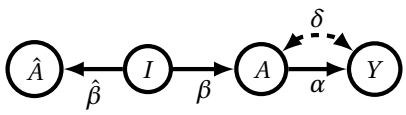
   The predicted values $\hat{A}$ represent the portion of the treatment that is driven *only* by the randomized instrument and the observed covariates. Because I is independent of the unobserved confounders U, our predictions $\hat{A}$ are also independent of U. We have successfully stripped away the unobserved confounding bias.

2. **Stage 2: Estimate the Causal Effect.** Next, we regress the outcome Y not on the actual treatment A, but on the predicted treatment $\hat{A}$ from the first stage:

$$Y = \alpha_0 + \alpha_{IV}\hat{A} + \alpha_2 C + \text{error}.$$

The coefficient α_{IV} is our estimate of the causal effect.

Why does this work? The cleanest way to see 2SLS in action is to add $\hat{A}$ to the causal diagram as a derived node. By construction $\hat{A} = \hat{\beta} I$ in the no-covariate case, so $\hat{A}$ is just a deterministic function of I, and the augmented DAG looks like:



Two things to observe in this augmented graph:

1. The path from $\hat{A}$ to Y that would carry the confounding — namely $\hat{A} \leftarrow I \rightarrow A \leftrightarrow Y$ — is **blocked**. The bidirected edge $A \leftrightarrow Y$ represents an unobserved common cause of A and Y, so this path is really $\hat{A} \leftarrow I \rightarrow A \leftarrow U \rightarrow Y$, with A acting as a collider. Without conditioning, the path through the collider is closed: confounding cannot propagate to $\hat{A}$.
2. The only open path from $\hat{A}$ to Y is the clean causal one, $\hat{A} \leftarrow I \rightarrow A \rightarrow Y$, carrying coefficients $\hat{\beta}, \beta, \alpha$.

Applying path multiplication (with $\hat{\beta} = \beta$ under the variance normalization) and $\text{Var}(I) = 1$:

$$\text{Cov}(\hat{A}, Y) = \beta \cdot \beta \cdot \alpha = \beta^2 \alpha.$$

And the variance of $\hat{A}$ inherits the same factor:

$$\text{Var}(\hat{A}) = \text{Var}(\beta I) = \beta^2.$$

The Stage 2 OLS coefficient is the ratio of these two:

$$\hat{\alpha}_{IV} = \frac{\text{Cov}(\hat{A}, Y)}{\text{Var}(\hat{A})} = \frac{\beta^2 \alpha}{\beta^2} = \alpha.$$

The two factors of β in the numerator (from traversing the augmented path through I) cancel exactly with the two factors in the denominator (from the variance of $\hat{A}$), leaving the causal effect alone. **2SLS is just the IV ratio dressed up as a regression**, with $\hat{A}$ chosen so that path analysis on the regression makes the answer come out right.

The Leap to Machine Learning

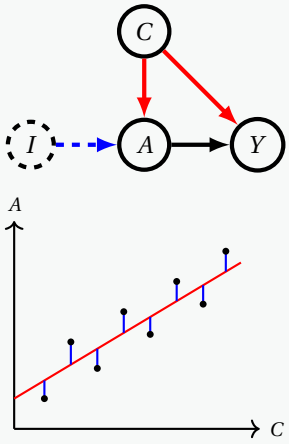
Classical 2SLS assumes that the relationships in both Stage 1 and Stage 2 are strictly linear. However, if the data-generating process is highly complex, non-linear, or involves high-dimensional data (like text or images), standard linear regression will fail to capture the true dynamics.

Modern causal inference extends this exact instrumental variable framework into the realm of Machine Learning (using methods like DeepIV). In these advanced methods, researchers replace the linear regressions in Stage 1 and Stage 2 with flexible, non-parametric ML algorithms like Random Forests or Deep Neural Networks.

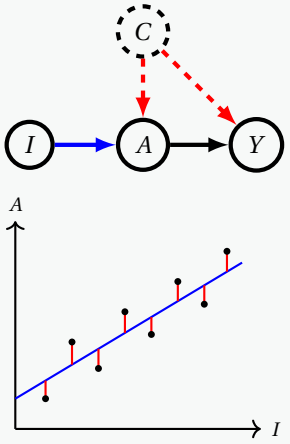
The core logic remains exactly the same — using a first-stage model to isolate the unconfounded signal from the instrument, and a second-stage model to map that clean signal to the outcome — but the ML models allow us to estimate causal effects in highly complex environments where classical equations fall short.

The Duality Between 2SLS and Partialling Out

When we partialled out a confounder, we predicted the treatment from that *confounder* and ran the final regression on the *residuals*. In 2SLS we instead predict the treatment from the *instrument* and run the final regression on the *predictions*. The contrast is the whole point, and it is easiest to see in one picture: the same graph $I \to A \to Y$ with the confounding path $A \leftarrow C \to Y$, drawn twice, with whichever variable we cannot observe drawn dashed. Beneath each graph we plot the regression of A on the observed variable, coloring the clean instrument signal blue and the confounding red — so in both panels, blue is what we keep and red is what we throw away.



Partialling out. C is observed, I is not. The *residual* of A on C strips out the red confounding and keeps the blue instrument signal, even though I itself is never measured.



Instrumental variables. C is unobserved, I is observed. The *prediction* of A from I keeps the blue instrument signal; the red confounding is left behind in the discarded residual.

Regression splits the variation of its target into a part explained by the regressors, which lives in the predictions, and an orthogonal, independent part, which lives in the residuals. Because every variable carries its own exogenous noise, that orthogonal part behaves like a miniature randomized controlled trial. Instrumental variables capture this clean variation directly, in the predictions, when the confounder is unobserved. Partialling out captures the very same variation indirectly, in the residuals, once an observed confounder has been removed. We will explore this more on the problem set!

4.7 Chapter Summary: A Taxonomy of Estimation Strategies

Over the course of this chapter, we have explored a variety of strategies for identifying and estimating causal effects from observational data. A recurring theme has been the connection between traditional econometric techniques and modern machine learning paradigms. Whether we are treating confounding as a covariate shift problem or framing counterfactual imputation as a recommendation system, the underlying goal remains the same: isolating the causal signal from spurious noise.

Table 4.1 provides a high-level summary of the five primary approaches we have discussed. When deciding which method to apply in practice, the choice almost entirely depends on what assumptions you are willing to make about your data generating process, and whether you are trying to estimate population-level averages or individual-level counterfactuals.

As you move forward into applying these tools, remember that every single method in this table relies on one foundational, untestable assumption: **knowing the correct causal structure in advance.**

In the previous chapter, we acted as causal "chefs," assuming a domain expert handed us the true DAG so we could identify the required adjustment sets. In this chapter, we acted as "cooks," assuming we knew exactly which variables constituted $\mathbf{X}$ for our regressions, which variable was a valid instrument for 2SLS, or which pre-treatment variables guaranteed latent unconfoundedness for our synthetic controls.

Method	Estimand & Pearl's Level	ML Analogue	Key Identification Assumptions	Modeling / Structural Assumptions
Outcome Regression	**ITE / CATE** (Level 3)	Supervised Learning	**Conditional Exchangeability:** $Y^{(a)} \perp\!\!\!\perp A \mid \mathbf{X}$. All confounders are observed.	**Parametric Form:** Traditional estimators (e.g., OLS) assume a fixed functional form to predict individual counterfactuals. Semi-parametric settings can be addressed using partialling out, so long as the treatment effect is linear.
Inverse Probability Weighting	**ATE** (Level 2)	Domain Adaptation / Importance Weighting	**Conditional Exchangeability** and **Positivity:** $0 < \Pr(A = a \mid \mathbf{X}) < 1$.	**Parametric Form:** Propensity models assume a functional form to reweight the population, not to predict individual outcomes.
Doubly Robust / Double ML	**ATE / CATE** (Level 2 / 3)	Ensemble / Orthogonal Learning	**Conditional Exchangeability** and **Positivity**.	**Partially Linear (Semi-parametric) Model:** DML often assumes a linear treatment effect, while using non-parametric ML to flexibly model the confounding nuisance parameters.
Synthetic Controls	**Unit-Level ITE** (Level 3)	Matrix Completion / Collaborative Filtering	**Latent Unconfoundedness:** Unobserved confounders are balanced by matching on a long history of pre-treatment outcomes ($T_{pre} \geq k$). **Positivity (Geometric):** The treated unit lies within the linear span (or convex hull) of the donor pool.	**Interactive Fixed Effects (Low-Rank):** Assumes outcomes are driven by a low-rank structure of latent factors. Instead of ruling out unobserved confounding, it uses donor histories to mathematically absorb it.
Instrumental Variables	**ATE** (Level 2)	Natural Experiments / Perturbation	**Exclusion Restriction**, **Ignorability**, and **Relevance**.	**Parametric Form:** Traditional 2SLS assumes linear relationships along the causal paths to estimate the average effect for the affected subpopulation.

Table 4.1: A summary of causal inference methods, their target estimands on Pearl's Ladder of Causation, ML analogues, and core assumptions.

But what happens when we step into an entirely unfamiliar domain? What if we are handed a modern dataset with hundreds of variables, and there is no domain expert available to draw the arrows for us? If we guess the graph incorrectly, all the sophisticated machine learning estimators in Table 4.1 will confidently compute the wrong causal effect.

This brings us to the final frontier of our causal pipeline, which we will tackle in the next chapter: **Causal Discovery**. Instead of using a known graph to analyze data, we will learn how to use the data to build the graph. By systematically testing for conditional independencies and exploiting the mathematical properties of statistical noise, we will explore algorithmic methods (like the PC Algorithm and LiNGAM) that attempt to reverse-engineer the underlying Structural Causal Model directly from purely observational data.

Chapter 5

Causal Discovery

The previous chapters operated under a common assumption: a domain expert hands us the causal DAG, and our job is to extract estimands from that structure. The do-calculus, instrumental variables, and all of the regression-based approaches take the graph as input. In this chapter, we ask where the graph comes from in the first place.

Causal discovery is the task of learning the existence and direction of causal links from data alone, before any quantification of their magnitude. It sits at the foundation of every causal claim but is methodologically distinct from estimation: estimation answers "how big is the effect?", whereas discovery answers "is there an effect at all, and which way does it run?"

The intellectual roots of causal discovery trace to philosopher Hans Reichenbach's *Common Cause Principle* (1956): if two variables are statistically dependent, then either one causes the other, or they share a common cause. This deceptively simple claim turned a statistical observation (dependence) into a constrained search problem (which causal structure could have produced this dependence?). The principle remained philosophical until the early 1990s, when Spirtes, Glymour, and Scheines at Carnegie Mellon turned it into an algorithm. Their PC algorithm [Spirtes et al., 2000], the founding work of constraint-based causal discovery, uses systematic tests of conditional independence to narrow down which causal structures could have generated the data. The score-based alternative [Chickering, 2002] and asymmetry-based methods (LiNGAM and its successors) followed, each tackling the same question with different statistical machinery.

The motivation today extends well beyond philosophy. Recent research on causal discovery for root-cause analysis [Ikram et al., 2022] has shown that pre-specified graphs from architectural call topology are systematically incomplete. They omit dependencies through shared infrastructure (CPU contention, memory, queues, caches), and they fail to update as systems evolve. If we can learn the graph, diagnosing outages reduces to following a DAG to the root cause. The same logic recurs wherever intervention is too costly or unethical to run: gene regulatory networks in drug discovery, effective connectivity in the brain, climate teleconnections like ENSO and the Indian monsoon.

The mathematical insight that drives the chapter is one we already have. In Chapter 3, d-separation rules told us which conditional independencies a known graph implied. In this chapter, we invert that logic: *conditional independence statements can be tested from data,* so the same correspondence between graphs and independencies lets us read causal structure off observed independencies in reverse. This inversion turns out not to pin down a unique DAG — only an *equivalence class* of DAGs that share certain observable features. The first section below develops a constructive algorithm [Spirtes et al., 2000] for recovering the easier of those features (the undirected skeleton); the second characterizes what else is identifiable from data and produces the CP-DAG that summarizes the full equivalence class. We then touch on score-based [Chickering, 2002] and asymmetry-based (LiNGAM and its successors) alternatives, and discuss how interventional data can refine these structures further. This part of the class will feel like computer science.

5.1 Skeleton Discovery

The intellectual move that makes causal discovery possible is the one we have been building toward implicitly. Throughout this class, we have used the structural causal model to derive d-separation rules: if a graph $\mathcal{G}$ contains a certain pattern of edges, then certain conditional independencies must hold in any distribution it generates. Causal discovery turns this around. Conditional independencies are statistical claims, and statistical claims can be *tested* from data. If we know which graphs imply which independencies, then independencies we observe in data tell us which graphs are still in the running.

This section assembles the tools that make this inversion work — statistical tests for conditional independence, the assumptions that connect those tests back to graphical structure, and a constructive procedure that uses them to recover the undirected backbone of the true causal DAG.

5.1.1 Testing Conditional Independence

The basic operation of constraint-based discovery is the conditional independence test: given variables X, Y, and a conditioning set $\mathbf{Z}$, decide whether $X \perp\!\!\!\perp Y \mid \mathbf{Z}$ holds in the data. The right test depends on the data type.

Discrete data. The standard tool is the **conditional chi-squared test** (or its close cousin, the G-test). The data is first partitioned into strata defined by levels of $\mathbf{Z}$. Within each stratum, we count the empirical co-occurrences of X and Y and compare them to what we would expect if X and Y were independent — namely, the product of their marginals.

Concretely, for binary X and Y within a fixed stratum $Z = z$, let n_{xy} denote the observed count of $(X = x, Y = y)$, with row and column marginals $n_{x\cdot} = \sum_y n_{xy}$ and $n_{\cdot y} = \sum_x n_{xy}$, and stratum total n. The chi-squared statistic within this stratum is

$$\chi^2_z = \sum_{x,y} \frac{(\text{observed} - \text{expected})^2}{\text{expected}} = \sum_{x,y} \frac{(n_{xy} - n_{x\cdot}\, n_{\cdot y}/n)^2}{n_{x\cdot}\, n_{\cdot y}/n}.$$

The expected count is the product of marginals because $\Pr(X = x, Y = y \mid Z = z) = \Pr(X = x \mid Z = z)\Pr(Y = y \mid Z = z)$ when $X \perp\!\!\!\perp Y \mid Z = z$, and the empirical estimates of those marginal probabilities are exactly $n_{x\cdot}/n$ and $n_{\cdot y}/n$. Summing χ^2_z across all strata gives the conditional chi-squared statistic, which is asymptotically chi-squared distributed under the null hypothesis of conditional independence. We reject the null when the statistic exceeds a chosen threshold.

Continuous (linear Gaussian) data. When the variables are jointly Gaussian, conditional independence is equivalent to *zero partial correlation*:

$$X \perp\!\!\!\perp Y \mid \mathbf{Z} \quad \Longleftrightarrow \quad \rho_{XY\cdot\mathbf{Z}} = 0,$$

where $\rho_{XY\cdot\mathbf{Z}}$ is the partial correlation of X and Y controlling for $\mathbf{Z}$. This is a direct connection to machinery we already have. Partial correlation is exactly the correlation of the FWL residuals from the previous chapter: if we residualize both X and Y against $\mathbf{Z}$ (call the residuals $\tilde{X}$ and $\tilde{Y}$), then

$$\rho_{XY\cdot\mathbf{Z}} = \mathrm{Corr}(\tilde{X}, \tilde{Y}).$$

To convert a sample partial correlation into a hypothesis test, we apply **Fisher's z-transformation**:

$$z = \tfrac{1}{2}\log\frac{1+\hat{\rho}}{1-\hat{\rho}},$$

which (asymptotically) has variance $1/(n - |\mathbf{Z}| - 3)$. Under the null of zero partial correlation, the transformed statistic is approximately standard normal, and we can reject at standard significance levels.

These tests are imperfect: at finite sample sizes they make Type I errors (false positives) and Type II errors (false negatives). Constraint-based discovery algorithms inherit this imperfection, and errors in early tests can propagate destructively through the algorithm. This is one of the key practical concerns when running causal discovery on real data.

5.1.2 The Causal Markov Condition and Faithfulness

Conditional independence (CI) tests give us a statistical signal. But this signal is only useful for discovery if the relationship between data and graph is tight enough that we can move in both directions: graph → CI and CI → graph. Two assumptions handle these directions separately, one of which we have already seen.

Definition 5.1.1 (Causal Markov Condition)**.** A distribution P is **Markov** with respect to a DAG $\mathcal{G}$ if every variable in $\mathcal{G}$ is independent of its non-descendants given its parents. Locally, every d-separation in $\mathcal{G}$ corresponds to a conditional independence in P:

$$X \perp\!\!\!\perp_d Y \,|\, \mathbf{Z} \text{ in } \mathcal{G} \implies X \perp\!\!\!\perp Y \,|\, \mathbf{Z} \text{ in } P.$$

The Causal Markov Condition is what we have implicitly been using all along: it says the graph faithfully reproduces what the system generates. Every conditional independence that the graph implies via d-separation must hold in the data. Without it, we could not even use a known graph to derive estimands — the entire program of earlier chapters would collapse.

Definition 5.1.2 (Faithfulness)**.** A distribution P is **faithful** to a DAG $\mathcal{G}$ if every conditional independence in P corresponds to a d-separation in $\mathcal{G}$:

$$X \perp\!\!\!\perp Y \,|\, \mathbf{Z} \text{ in } P \implies X \perp\!\!\!\perp_d Y \,|\, \mathbf{Z} \text{ in } \mathcal{G}.$$

Equivalently, P contains no conditional independencies beyond those forced by the graph.

Faithfulness is the reverse direction, and it is the assumption that gives us license to interpret a CI test result as a structural claim. Together, the two assumptions yield a biconditional:

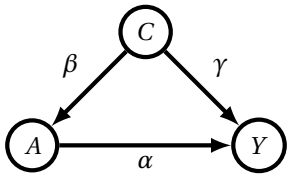
$$X \perp\!\!\!\perp Y \,|\, \mathbf{Z} \text{ in } P \quad \iff \quad X \perp\!\!\!\perp_d Y \,|\, \mathbf{Z} \text{ in } \mathcal{G}.$$

This is the engine of constraint-based causal discovery. Every CI test we run is, modulo statistical noise, a query about the structure of $\mathcal{G}$.

A third assumption is implicit in everything we do in this section: **causal sufficiency**, the requirement that every common cause of two measured variables is itself measured. When this fails, the FCI algorithm and its descendants extend the techniques here to ADMGs, but we will not pursue that direction.

5.1.3 When Faithfulness Fails: Path Cancellation

The Causal Markov Condition is essentially automatic in the structural causal model framework — if the data really were generated by $\mathcal{G}$, the d-separations must hold by construction. Faithfulness is more fragile. A standard counter-example, drawing on the path-analysis machinery from the previous chapter, is the following:



Path analysis tells us that

$$\text{assoc}(A, Y) = \underbrace{\alpha}_{\text{direct path}} + \underbrace{\beta\gamma}_{\text{backdoor through } C}.$$

For a special choice of parameters — specifically $\alpha = -\beta\gamma$ — the two paths cancel exactly, and $\text{assoc}(A, Y) = 0$. A marginal independence test will conclude $A \perp\!\!\!\perp Y$, even though A structurally causes Y. Conditioning on C would unmask the direct effect (the backdoor is now blocked, leaving only α), so $A \not\perp\!\!\!\perp Y \,|\, C$. The graph is *unfaithful* to the distribution it generates: there is a conditional independence in P ($A \perp\!\!\!\perp Y$ marginally) that does not correspond to any d-separation in $\mathcal{G}$.

Faithfulness violations of this kind are called *path cancellations*, and they require knife-edge tuning of the structural coefficients — the set of parameter values that produces exact cancellation has Lebesgue measure zero in the parameter space. A standard pragmatic stance is that natural data-generating processes do not land on this measure-zero set, so faithfulness holds "generically." When it does not, no purely observational discovery method can recover the missing edge.

5.1.4 Recovering the Skeleton

We now turn to the first concrete computational task of causal discovery: recovering the *skeleton* of the true DAG — the undirected graph obtained by erasing all arrow directions. For every pair of variables (X, Y), we need to decide whether they are adjacent in the true DAG. The logic rests on a single lemma.

Lemma 5.1.3 (Parents separate non-adjacent pairs)**.** *Let $\mathcal{G}$ be a DAG and let X, Y be two non-adjacent variables in $\mathcal{G}$. Then either*

$$X \perp\!\!\!\perp_d Y \mid \mathbf{PA}(X) \qquad \textit{or} \qquad X \perp\!\!\!\perp_d Y \mid \mathbf{PA}(Y),$$

and by the Causal Markov Condition, the corresponding conditional independence holds in any Markovian distribution.

Proof: Since $\mathcal{G}$ is acyclic, at least one of $Y \notin \mathbf{DE}(X)$ or $X \notin \mathbf{DE}(Y)$ must hold ($\mathbf{DE}$ is the descendents operator). Without loss of generality, assume $Y \notin \mathbf{DE}(X)$. By the local Markov property,

$$X \perp\!\!\!\perp_d \left(\mathbf{V} \setminus (\{X\} \cup \mathbf{DE}(X) \cup \mathbf{PA}(X))\right) \Big| \mathbf{PA}(X).$$

Since X and Y are non-adjacent, $Y \notin \mathbf{PA}(X)$, so Y lies in this set, giving $X \perp\!\!\!\perp_d Y \mid \mathbf{PA}(X)$. ■

The contrapositive is just as important: if X and Y *are* adjacent, then *no* set separates them — the direct edge is an active path under any conditioning, and Faithfulness rules out accidental cancellations that could mimic separation. So skeleton discovery reduces to a mechanical procedure: for each pair (X, Y), search for a separating set; if one exists, delete the edge; if not, the edge is real.

Naive search: too expensive. A naive implementation would test, for each of the $\binom{n}{2}$ pairs, every one of the 2^{n-2} subsets of the remaining variables until either a separating set is found or every subset has been ruled out. This is $O(n^2 \cdot 2^n)$ independence tests — hopelessly expensive for anything beyond a handful of variables.

Restricting the search to current neighbors. Lemma 5.1.3 tells us we don't need every subset of $\mathbf{V} \setminus \{X, Y\}$. Whenever a separating set exists, the parents of X (or Y) are one — and the parents of a variable are, by definition, among its neighbors in the true skeleton, which we denote $\mathbf{NB}(X)$. We don't know the true skeleton, but we can maintain a *working* skeleton $\mathcal{G}$ that always contains it: start with the complete graph (trivially a superset of the truth) and remove edges only when a separating set has been found. Then at every step,

$$\mathbf{PA}(X) \subseteq \mathbf{NB}_{\text{true}}(X) \subseteq \mathbf{NB}_{\mathcal{G}}(X),$$

and likewise for Y. This gives us the anchor that licenses restricting the search:

Lemma 5.1.4 (Separating sets live in the working neighborhood)**.** *Let $\mathcal{G}$ be a working skeleton that contains the true skeleton, and let X, Y be non-adjacent in the true graph. Then a separating set for X, Y exists as a subset of $\mathbf{NB}_{\mathcal{G}}(X) \cup \mathbf{NB}_{\mathcal{G}}(Y)$.*

Proof: By Lemma 5.1.3, $\mathbf{PA}(X)$ or $\mathbf{PA}(Y)$ separates X and Y. Since $\mathcal{G}$ contains every true edge, $\mathbf{PA}(X) \subseteq \mathbf{NB}_{\mathcal{G}}(X)$ and $\mathbf{PA}(Y) \subseteq \mathbf{NB}_{\mathcal{G}}(Y)$. So at least one separating set sits inside $\mathbf{NB}_{\mathcal{G}}(X) \cup \mathbf{NB}_{\mathcal{G}}(Y)$. ■

We therefore never need to look outside the working skeleton's neighborhoods to find separating sets.

Searching by increasing size. We further sharpen the strategy by checking conditioning sets in order of increasing cardinality. There is exactly one set of size 0 (the empty set), at most $n-2$ of size 1, $\binom{n-2}{2}$ of size 2, and so on — the count grows steeply with size. So we test all size-0 candidates first across all pairs, then all size-1 candidates, then all size-2, and so on. This ordering has two benefits:

- Every deleted edge shrinks the neighborhood of *every* vertex it touched, which shrinks the search space for every remaining pair. Aggressive early deletion compounds.
- It gives us a clean stopping rule.

When to stop. Suppose at some point the maximum degree of the working $\mathcal{G}$ is $\delta_{\mathcal{G}}$, and we have just finished testing all conditioning sets of size $d > \delta_{\mathcal{G}}$. Then no neighborhood in $\mathcal{G}$ is large enough to hold a conditioning set of size d or larger. By Lemma 5.1.4, if a separating set exists, then there must be one that is a subset of a neighborhood — so no separating set we haven't already tried can exist. The remaining edges are genuine, and we terminate.

The total cost is bounded by the maximum degree δ of the true skeleton: for each pair, the search space is the power set of its neighborhood, at most 2^{δ} subsets (which is approximately the number of separating sets of size δ and smaller, since we halt afterwards). Across all $\binom{n}{2}$ pairs, the algorithm performs $O\left(n^2 \cdot 2^{\delta}\right)$ independence tests — exponential in degree, but *polynomial in n* for any graph of bounded degree. For the sparse graphs that arise in essentially every real-world causal system, this is tractable.

Algorithm 1 formalizes the procedure.

Algorithm 1 Skeleton Discovery(DATA) → Skeleton Graph $\mathcal{G}$, SepSets $\mathbb{S}$

```
𝒢 ← A complete undirected graph                    # Start by connecting all pairs X − Y
𝕊 ← empty dictionary                               # Stores the variables used to delete an edge: (X, Y) → Z
C ← 0                                              # Size of the conditioning set
while C ⩽ |V| − 2 do
  for all adjacent pairs X − Y in 𝒢 do
    ℕ ← (NB_𝒢(X) ∪ NB_𝒢(Y)) \ {X, Y}              # Get neighbors of X and Y
    for all subsets Z ⊆ ℕ where |Z| = C do
      if X ⫫ Y | Z in the DATA then
        Delete edge X − Y from 𝒢
        Store the separation set: 𝕊[(X, Y)] ← Z
        break                                      # Edge is gone, move to the next pair
      end if
    end for
  end for
  C ← C + 1
end while
return 𝒢, 𝕊
```

The output is the undirected skeleton of the true DAG, together with a record of which conditioning set separated each missing edge — the separating sets of "SepSets". The skeleton on its own is missing the orientation information we need to read off causal structure; the SepSets, as we will see in the next section, are exactly what lets us begin to recover that information.

5.2 Equivalence Classes and Edge Orientation

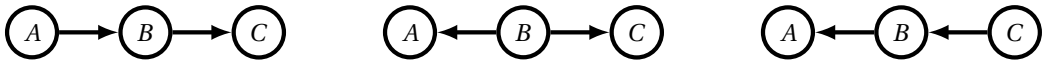
In the previous section, we recovered the undirected skeleton of the causal DAG by testing conditional independencies. Each edge in the skeleton corresponds to a true adjacency in the underlying causal structure, but the directions remain unknown. Can we orient any of them from observational data alone?

The answer is: some, yes; some, no. There are pairs of DAGs that share the same skeleton, generate exactly the same set of conditional independencies, and are therefore indistinguishable from any amount

of observational data. There are also edge directions that, once a few are pinned down, become forced by purely deductive logic. This section characterizes both — the part of edge direction that is fundamentally unidentifiable, and the part we can extract by careful reasoning.

5.2.1 Markov Equivalence

Even with Markov and Faithfulness both assumed, there is a deep limit on what observational data can teach us about a DAG. Consider three random variables A, B, C and the following three DAGs:



Chain: $A \to B \to C$ Fork: $A \leftarrow B \to C$ Reverse Chain: $A \leftarrow B \leftarrow C$

All three of these structures imply the same single conditional independence: $A \perp\!\!\!\perp C \mid B$, and no others. The chain, fork, and reverse chain are *statistically indistinguishable* — no amount of CI testing on observational data can tell them apart. Notably, the collider $A \to B \leftarrow C$ is the odd one out: it implies $A \perp\!\!\!\perp C$ marginally (without conditioning on anything), and conditioning on B *opens* an association rather than closing one. That distinct independence signature makes the collider statistically identifiable.

Definition 5.2.1 (Markov Equivalence)**.** Two DAGs $\mathcal{G}_1$ and $\mathcal{G}_2$ over the same vertex set are **Markov equivalent** if they imply the same set of conditional independence statements:

$$\mathcal{I}(\mathcal{G}_1) = \mathcal{I}(\mathcal{G}_2),$$

where $\mathcal{I}(\mathcal{G})$ denotes the complete set of conditional independencies entailed by $\mathcal{G}$ via d-separation.

Markov equivalence partitions the set of all DAGs over $\mathbf{V}$ into mutually exclusive *Markov Equivalence Classes* (MECs). All DAGs within a class produce the same observational signature, and no purely observational method can distinguish among them. The best we can hope to recover from observational data is the MEC of the true DAG.

> **Main Idea 23**
>
> [Verma-Pearl Characterization] Two DAGs $\mathcal{G}_1$ and $\mathcal{G}_2$ are Markov equivalent if and only if they share:
>
> 1. the same **skeleton** (the undirected graph obtained by erasing all arrow directions), and
> 2. the same **v-structures** (*unshielded colliders*): triples $X \to Z \leftarrow Y$ where X and Y are not adjacent.

This characterization [Verma and Pearl, 1990, Andersson et al., 1997] pins down exactly what observational data can reveal: the skeleton and the v-structures, and nothing else. The chain, fork, and reverse chain in the example above share the same skeleton ($A - B - C$) and have no v-structures, so they sit in the same MEC. The collider has the same skeleton but does contain a v-structure, so it sits in its own MEC of size one — which is why it is the one observational signature standing apart from the others.

The previous section already covered the first ingredient: the skeleton is the output of Algorithm 1. The rest of this section deals with the second ingredient (v-structures), and then with a deductive cleanup phase that propagates as many additional orientations as logic allows.

5.2.2 Identifying V-Structures

V-structures are unshielded colliders: triples $A \to B \leftarrow C$ where A and C are not adjacent. Verma-Pearl tells us they are identifiable, but *how* do we extract them from data? The answer is hidden in the SepSets $\mathbb{S}$ we recorded during skeleton discovery.

Consider an unshielded triple $A - B - C$ in the skeleton: A is adjacent to B, B is adjacent to C, but A is not adjacent to C. Because A and C are missing an edge, the algorithm recorded a SepSet $\mathbf{Z} = \mathbb{S}[(A, C)]$ that separated them. Ask: is B inside $\mathbf{Z}$?

- If $B \in \mathbf{Z}$, then conditioning on B *helped* block the path between A and C. This means B acts as a non-collider on the path $A - B - C$ — a chain ($A \to B \to C$ or $A \leftarrow B \leftarrow C$) or a fork ($A \leftarrow B \to C$), all of which are blocked by conditioning on B.
- If $B \notin \mathbf{Z}$, then conditioning on $\mathbf{Z}$ separated A and C *without* needing B. If B were a non-collider on the path, leaving it out would leave that path open, contradicting $A \perp\!\!\!\perp C \mid \mathbf{Z}$. So B must be a collider: $A \to B \leftarrow C$.

This logic gives us our first orientation rule, often called **Rule 0**, which Algorithm 2 formally applies to every unshielded triple in the skeleton.

Algorithm 2 Apply $\mathscr{R}_0(\mathscr{G}, \mathbb{S}) \to$ Partially Directed Graph $\mathscr{G}'$, Boolean

```
𝒢' ← 𝒢                                          # Create a copy of the skeleton
applied_rule ← False
for all missing edges (A, C) with SepSet Z = 𝕊[(A, C)] do
  for all B ∈ NB_{𝒢'}(A) ∩ NB_{𝒢'}(C) do
    # Check all unshielded triples of the form A − B − C
    if B ∉ Z then
      Replace A − B − C with A → B ← C in 𝒢'
      applied_rule ← True
    end if
  end for
end for
return 𝒢', applied_rule
```

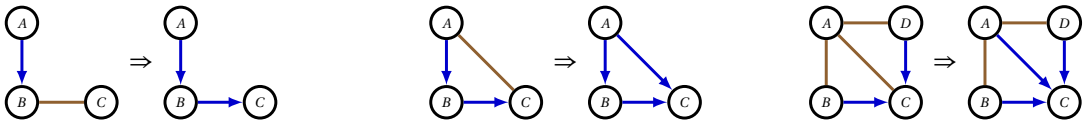
After Rule 0, the graph $\mathscr{G}'$ is a **partially directed acyclic graph (PDAG)**: a graph with both directed and undirected edges and no directed cycles. The specific PDAG produced by Rule 0 — skeleton with all v-structures oriented — is called the **pattern**. It is not yet the final answer, because there can be additional edges whose orientation is forced by avoiding either a new v-structure or a directed cycle.

5.2.3 Propagating Orientations: Meek's Rules

Because Rule 0 has already found *all* the v-structures, any future edge orientation must not introduce a new one. And because the true graph is a DAG, no orientation can introduce a directed cycle. Together, these two constraints force the direction of additional edges beyond those in v-structures.

We can repeatedly apply the following three logical rules, discovered by Christopher Meek (1995), until no more edges can be oriented:

Rule 1 (No New V-Structures) Rule 2 (No Cycles) Rule 3



The logic behind these rules is purely deductive:

- **Rule 1:** If we oriented $B - C$ as $B \leftarrow C$, we would create a new v-structure ($A \to B \leftarrow C$). Since we already found all v-structures, this is illegal. Thus, it must be $B \to C$.
- **Rule 2:** If we oriented $A - C$ as $A \leftarrow C$, we would create a directed cycle ($A \to B \to C \to A$). Since causal graphs are acyclic, this is illegal. Thus, it must be $A \to C$.
- **Rule 3:** This rule prevents a combination of cycles and v-structures. Suppose we tried to orient $A - C$ as $A \leftarrow C$. To avoid creating the cycle $A \to B \to C \to A$, we would be forced to orient $A - B$ as $A \leftarrow B$. To avoid the cycle $A \to D \to C \to A$, we would be forced to orient $A - D$ as $A \leftarrow D$. But if we do both, we

create $B \to A \leftarrow D$, which is a brand new v-structure! To avoid this paradox, $A - C$ must be oriented as $A \to C$.

Meek proved that these rules are *sound and complete*. Sound means they will never make a false orientation (assuming the skeleton and v-structures are correct). Complete means that once these three rules can no longer be applied, no further orientations can be logically deduced from observational data alone.

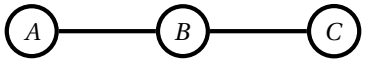
5.2.4 The CP-DAG

The output of Rule 0 followed by exhaustive application of the Meek rules is a partially directed acyclic graph in which every edge whose direction is invariant across the MEC has been oriented, and every undirected edge represents genuine ambiguity within the MEC. This object is the **CP-DAG**.

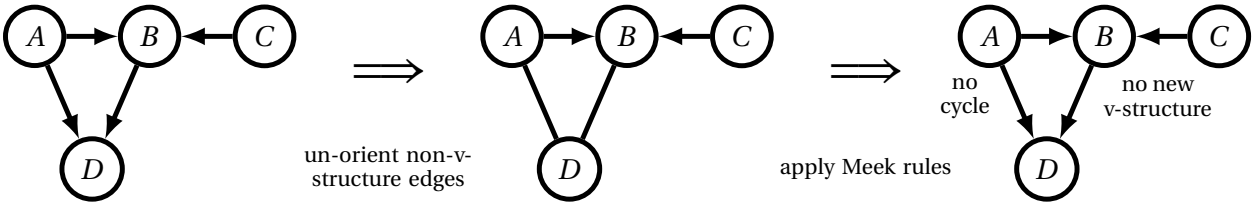
Definition 5.2.2 (CP-DAG / Essential Graph)**.** Given a Markov Equivalence Class $\mathscr{M}$, the **completed partially directed acyclic graph (CP-DAG)** is a partially directed graph in which:

- edge (X, Y) is drawn **directed** $(X \to Y)$ if every DAG in $\mathscr{M}$ orients it that way, and
- edge (X, Y) is drawn **undirected** $(X - Y)$ if some DAGs in $\mathscr{M}$ orient it one way and others orient it the opposite way.

The CP-DAG — equivalently called the *essential graph* — captures every invariant orientation in the MEC and represents the remaining ambiguity with undirected edges. For the three-DAG MEC of chain, fork, and reverse chain, the pattern and the CP-DAG happen to coincide because the MEC contains no v-structures and the Meek rules have nothing to propagate from:



At the other extreme, an unshielded collider $A \to B \leftarrow C$ has its CP-DAG fully oriented: the v-structure pins down both edges and the MEC has size one. The interesting case is in between, where v-structures pin down some edges directly and the Meek rules pin down others by propagation. Consider the DAG on the left below:



The graph contains two colliders, one of each type:

- At B: $A \to B \leftarrow C$, with A and C not adjacent. **Unshielded v-structure.**
- At D: $A \to D \leftarrow B$, with A and B adjacent (via $A \to B$). *Shielded* — not a v-structure.

The v-structure at B pins down $A \to B$ and $C \to B$ in every member of the MEC. The shielded collider at D contributes nothing to the v-structure characterization, because a shielded collider has the same observational signature as a chain or fork through the same triple. Stripping the orientation from every edge that is not part of a v-structure gives the pattern (middle): two directed edges and two undirected ones.

But the pattern is not yet fully oriented. If $D \to B$ were a valid orientation of the B-D edge, it would create a new unshielded collider $C \to B \leftarrow D$ (since C and D are not adjacent), changing the v-structure set and dropping us out of the MEC. So $B \to D$ is forced (Rule 1). And once $A \to B \to D$ is in place, the reverse

orientation $D \to A$ would create a directed cycle, so $A \to D$ is forced too (Rule 2). These two additional orientations — annotated on the rightmost graph — bring us to the CP-DAG, which in this example coincides with the original DAG: this MEC has size one.

This is precisely the gap between a pattern and a CP-DAG. The pattern captures only what the v-structures themselves impose; the CP-DAG additionally captures every orientation forced by avoiding new v-structures and directed cycles.

5.2.5 Putting It Together: The PC Algorithm

What we have built across this section and the previous one is the **PC algorithm** (named after its inventors, **P**eter Spirtes and **C**lark Glymour), the classic constraint-based method for causal discovery from observational data [Spirtes et al., 2000]. Putting the pieces in one place:

1. **Skeleton discovery** (Algorithm 1). Start with the complete undirected graph. Iteratively delete edges $X - Y$ for which a separating set exists in $\mathbf{NB}_{\mathcal{G}}(X) \cup \mathbf{NB}_{\mathcal{G}}(Y)$, growing the conditioning-set size C from 0 upward. Record the separating set $\mathbb{S}[(X, Y)]$ for each deleted edge.
2. **V-structure identification** (Algorithm 2). For every unshielded triple $A - B - C$ in the skeleton, check whether B is in the SepSet $\mathbb{S}[(A, C)]$. If not, orient as $A \to B \leftarrow C$. The result is the pattern.
3. **Meek propagation.** Apply Rules 1–3 to the pattern until no more orientations can be deduced. The result is the CP-DAG of the true DAG's Markov Equivalence Class.

This procedure recovers everything observational data can possibly recover about the causal structure, and abstains on the rest.

5.2.6 Consistency and Theoretical Guarantees

A fundamental concept in statistical learning is **consistency** — the property that, as the amount of data grows to infinity, an algorithm's output converges in probability to the true underlying structure.

The PC algorithm is **asymptotically consistent** for causal discovery. Given an infinite sample size ($n \to \infty$) and perfect conditional independence tests, the algorithm recovers the true CP-DAG of the data-generating process exactly.

This guarantee rests on three assumptions, two of which we have developed in the previous section, and a third that we have not yet leaned on explicitly:

1. **Causal Markov Condition.** The joint distribution is Markovian with respect to the true causal DAG.
2. **Faithfulness.** All conditional independencies in the data are structural — no path cancellations or other measure-zero coincidences.
3. **Causal Sufficiency.** There are no unmeasured hidden confounders. Every common cause of two or more measured variables is included in the dataset.

Causal Sufficiency is the one most likely to fail in practice. If it does, the PC algorithm can misidentify the skeleton (an unmeasured common cause induces marginal dependence that no observed conditioning set can resolve) and can introduce spurious v-structures. The standard remedy is the *FCI algorithm* (Fast Causal Inference), which extends the PC framework to output a partial ancestral graph rather than a CP-DAG — an object that admits unmeasured confounders in its semantics. We will not develop FCI here, but it lives in the same constraint-based family.

5.2.7 Key Properties in Practice

While the algorithm is consistent in the limit, its practical application on finite datasets has distinct characteristics:

- **Computational complexity.** The cost is $O(n^2 \cdot 2^\delta)$ independence tests, exponential in the maximum degree δ of the true skeleton. For sparse graphs — which describes most real-world causal systems — this is tractable. For dense graphs, the cost can blow up.
- **Compounding errors.** Each conditional independence test has some probability of a false positive or false negative on finite data. A single early error in skeleton discovery propagates: an incorrectly deleted edge can corrupt many downstream SepSets, which can cause Rule 0 to mis-orient v-structures, which can cause the Meek rules to mis-propagate further. The algorithm has no built-in mechanism for revisiting earlier mistakes.
- **Inherent limits.** The algorithm does not orient edges that are unidentifiable from observational conditional independencies alone. Any edges left undirected in the final CP-DAG represent genuine, mathematically unresolvable causal uncertainty — not a failure of the algorithm. Resolving them requires additional information, typically from interventional experiments or from parametric assumptions of the kind we will discuss with LiNGAM later in the chapter.

The next section develops the GES algorithm, which takes a different tack on the same problem: rather than testing CIs and deducing structure, GES treats causal discovery as an explicit search over the space of equivalence classes. Both PC and GES output CP-DAGs, and both are limited by the same fundamental observational ceiling. Going beyond that ceiling — to identify edges that no purely observational method can orient — requires either interventional data or parametric assumptions, both of which we will see later in the chapter.

5.3 Greedy Equivalence Search (GES)

While the PC algorithm is elegant, it suffers from "compounding errors" that are hard to analyze: an incorrect output to one test of conditional independence can have downstream consequences later in the algorithm. Chickering [2002] introduced an alternative approach to the PC algorithm that shifts the paradigm from constraint-based testing to *score-based* learning. Instead of using conditional independence tests to systematically eliminate edges, score-based methods introduce a "score function" that quantifies how well a specific DAG fits a dataset, and then perform a greedy search to find the DAG that maximizes this score.

5.3.1 Bayesian Information Criterion

The most popular score function used in GES (Greedy Equivalence Search) is the **Bayesian Information Criterion (BIC)**, which is grounded in Bayesian statistics. In general, a Bayesian approach to statistics involves finding the most likely graphical model $\mathcal{G}$ given the data $\mathbf{D}$. While this posterior probability is difficult to reason about directly, Bayes' rule allows us to flip the conditioning:

$$\Pr(\mathcal{G} \mid \mathbf{D}) = \frac{\Pr(\mathbf{D} \mid \mathcal{G}) \Pr(\mathcal{G})}{\Pr(\mathbf{D})}$$

Because $\Pr(\mathbf{D})$ is the same for every model choice, and assuming a uniform prior $\Pr(\mathcal{G})$ over the possible DAGs, maximizing the left-hand side is equivalent to maximizing $\Pr(\mathbf{D} \mid \mathcal{G})$. This quantity is called the *marginal likelihood* (or model evidence) because it implicitly averages over all parameter values θ consistent with $\mathcal{G}$:

$$\Pr(\mathbf{D} \mid \mathcal{G}) = \int \Pr(\mathbf{D} \mid \mathcal{G}, \theta) \Pr(\theta \mid \mathcal{G}) \, d\theta.$$

This is what we want to maximize, not the more familiar maximum likelihood $\Pr(\mathbf{D} \mid \mathcal{G}, \hat{\theta}_{\mathcal{G}})$. The distinction matters: maximum likelihood always prefers more complex models (more parameters can only fit better), whereas marginal likelihood penalizes complexity automatically. Integrating a prior across many parameters spreads it thin, so high-dimensional models pay a cost in the average that low-dimensional ones do not.

The marginal likelihood has no closed form in general, but a *Laplace approximation* — Taylor-expanding the integrand around the MLE and treating the curvature as Gaussian — gives a clean asymptotic expression. On the log scale:

$$\mathrm{BIC}(\mathcal{G}) \approx \log \Pr(\mathbf{D} \mid \mathcal{G}) \approx \ell(\mathbf{D}; \hat{\theta}_{\mathcal{G}}) - \frac{d}{2} \log(n).$$

Here, $\ell(\mathbf{D}; \hat{\theta}_{\mathcal{G}})$ is the log-likelihood evaluated at the Maximum Likelihood Estimator (MLE) for the parameters $\hat{\theta}_{\mathcal{G}}$, and the $-\frac{d}{2}\log(n)$ term is the volume of the Gaussian posterior around the MLE — the asymptotic form of the natural Bayesian penalty on complexity, with d the number of parameters in the graph and n the sample size.

5.3.2 Score Decomposability

A vital property of the BIC score that makes the greedy search computationally tractable is **decomposability**. The total BIC score of a DAG $\mathcal{G}$ can be expressed as a sum of local scores, one per node:

$$\mathrm{BIC}(\mathcal{G}) = \sum_{i=1}^{|\mathbf{V}|} \mathrm{score}(V_i \mid \mathbf{PA}(V_i)).$$

This falls out of the factorization that defines a DAG model. The joint distribution factors as a product of local conditionals,

$$\Pr(V_1, \ldots, V_{|\mathbf{V}|}) = \prod_{i=1}^{|\mathbf{V}|} \Pr(V_i \mid \mathbf{PA}(V_i)),$$

and under the standard *parameter modularity* assumption — each local conditional $\Pr(V_i \mid \mathbf{PA}(V_i))$ has its own parameter set θ_i with its own prior, independent of the others — the same factorization carries through the marginal likelihood. Writing $\mathbf{D}_i$ for the columns of $\mathbf{D}$ corresponding to V_i and its parents,

$$\Pr(\mathbf{D} \mid \mathcal{G}) = \prod_{i=1}^{|\mathbf{V}|} \int \Pr(\mathbf{D}_i \mid \mathbf{PA}(V_i), \theta_i) \Pr(\theta_i \mid \mathcal{G})\, d\theta_i = \prod_{i=1}^{|\mathbf{V}|} \Pr(\mathbf{D}_i \mid \mathbf{PA}(V_i)).$$

Each local marginal on the right depends only on V_i and its parents. Taking logs turns the product into a sum, and applying the Laplace approximation to each local integral gives the BIC's local-score form:

$$\mathrm{score}(V_i \mid \mathbf{PA}(V_i)) = \ell_i(\mathbf{D}; \hat{\theta}_i) - \frac{d_i}{2} \log n,$$

where ℓ_i is the local log-likelihood at the local MLE $\hat{\theta}_i$, and d_i is the number of parameters in $\Pr(V_i \mid \mathbf{PA}(V_i))$. The total ℓ and d that appear in the global BIC formula are just $\sum_i \ell_i$ and $\sum_i d_i$, so BIC inherits the decomposition.

Asymptotic form of local score differences. We will use one asymptotic fact about these local scores repeatedly in the rest of the section. The GES algorithm will often add edges, e.g., between $Z \to V_i$. The change in V_i's local log-likelihood from adding Z as a parent is a sum of log-density ratios over the n data points:

$$\ell_i\big(\mathbf{D}; \hat{\theta}_i^{\mathrm{new}}\big) - \ell_i\big(\mathbf{D}; \hat{\theta}_i^{\mathrm{old}}\big) = \sum_{j=1}^{n} \log \frac{\Pr\big(V_i^{(j)} \mid \mathbf{PA}(V_i)^{(j)}, Z^{(j)}\big)}{\Pr\big(V_i^{(j)} \mid \mathbf{PA}(V_i)^{(j)}\big)}.$$

By the law of large numbers, $\frac{1}{n}$ times this sum converges to its expectation, the *conditional mutual information*

$$I\big(V_i; Z \mid \mathbf{PA}(V_i)\big) \equiv \mathbb{E}\left[\log \frac{\Pr\big(V_i \mid \mathbf{PA}(V_i), Z\big)}{\Pr\big(V_i \mid \mathbf{PA}(V_i)\big)}\right].$$

This quantity is non-negative, and equals zero exactly when $V_i \perp\!\!\!\perp Z \mid \mathbf{PA}(V_i)$. So the asymptotic improvement in log-likelihood from adding Z as a parent of V_i is $n \cdot I(V_i; Z \mid \mathbf{PA}(V_i))$, scaling linearly in the sample size.

Weighed against the $\frac{\Delta d}{2} \log n$ complexity penalty in BIC, this $n \cdot I$ scaling is what determines whether the algorithm wants to add a given edge.

Critically, each local score depends only on V_i and its parents — not on the rest of the graph. When the algorithm evaluates adding or removing a single edge, only one node's parent set has changed, and we only need to recompute that one local score. Every other local score can be cached. This is what makes the greedy search computationally feasible: GES can evaluate thousands of potential edge modifications in fractions of a second.

5.3.3 Searching Over Equivalence Classes

If we want to greedily search for the DAG with the highest BIC score, we run into a major problem: the space of possible DAGs is astronomically large, and standard greedy search algorithms easily get trapped in local optima.

Furthermore, score functions like BIC are *score equivalent*: any two DAGs in the same Markov Equivalence Class (MEC) yield the exact same score. Searching over individual DAGs would therefore redundantly visit many equivalent states. The natural alternative is to search over MECs instead.

The canonical representation of an MEC is the CP-DAG we developed in the previous section. But running the search directly on CP-DAGs is expensive: nearly every candidate edge modification we contemplate would knock the graph out of valid CP-DAG form, and we would have to re-apply the full set of **Meek rules** to canonicalize back into a CP-DAG before evaluating the next candidate. Meek's rules are cheap individually, but invoking them across thousands of candidate moves at every iteration of the search is computationally wasteful.

The cleaner approach is to keep the working state at the level of DAGs — which carry no canonicalization machinery — and then search across all DAGs within a Markov equivalence class. The DAG we hold at any moment is then just a chosen *representative* of the MEC under consideration. To search across all such representative DAGs, the right tool is the **covered edge reversal**.

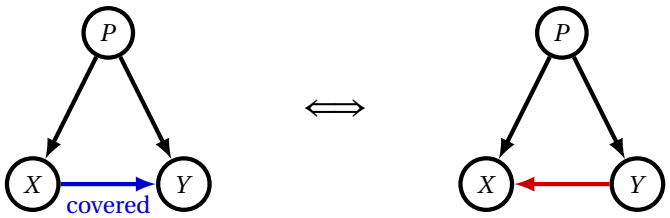
Definition 5.3.1 (Covered Edge)**.** A directed edge $X \to Y$ in a DAG $\mathcal{G}$ is **covered** if X and Y have the exact same set of parents, excluding X itself. Formally:

$$\mathbf{PA}(Y) \setminus \{X\} = \mathbf{PA}(X).$$

If an edge is covered, reversing it (changing $X \to Y$ to $X \leftarrow Y$) preserves the skeleton of the graph and neither creates nor destroys any v-structures. Because Markov equivalence is determined entirely by skeleton and v-structures (the Verma-Pearl characterization from the previous section), reversing a covered edge guarantees that the new DAG sits in the exact same Markov Equivalence Class as the original.

> **Main Idea 24**
>
> Reversing a covered edge guarantees that the new DAG sits in the exact same Markov Equivalence Class as the original.

Let's look at a visual example:



DAG 1: $\mathbf{PA}(X) = \{P\}$, $\mathbf{PA}(Y) = \{P, X\}$ DAG 2: $\mathbf{PA}(X) = \{P, Y\}$, $\mathbf{PA}(Y) = \{P\}$

In DAG 1, $\mathbf{PA}(Y) \setminus \{X\} = \{P\} = \mathbf{PA}(X)$, so the blue edge is covered. Reversing it yields DAG 2, which has the same skeleton and no new v-structures — the two DAGs sit in the same MEC, and we moved between them without consulting Meek's rules at all.

This brings us to the foundational theorem that makes score-based search over equivalence classes feasible.

Theorem 5.3.2 (Chickering's Theorem, 1995)**.** *Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be two DAGs. $\mathcal{G}_1$ and $\mathcal{G}_2$ are Markov equivalent if and only if there exists a finite sequence of covered edge reversals that transforms $\mathcal{G}_1$ into $\mathcal{G}_2$.*

Proof Sketch : **($\Longleftarrow$):** Assume there is a sequence of covered edge reversals transforming $\mathcal{G}_1$ into $\mathcal{G}_2$. A single covered edge reversal does not change the skeleton or the v-structures of a DAG, and by induction neither does any finite sequence of such reversals. By the Verma-Pearl criteria, $\mathcal{G}_1$ and $\mathcal{G}_2$ must therefore be Markov equivalent.

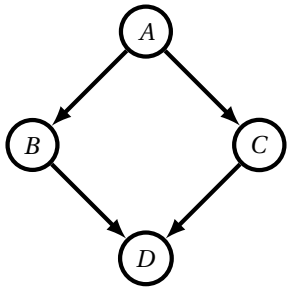
($\Longrightarrow$): Assume $\mathcal{G}_1$ and $\mathcal{G}_2$ are Markov equivalent. They share the same skeleton and v-structures, but may disagree on the orientation of some non-v-structure edges. Chickering showed constructively that whenever the orientations differ, we can always find at least one covered edge in $\mathcal{G}_1$ whose reversal brings $\mathcal{G}_1$ closer to $\mathcal{G}_2$ (reducing the number of disagreeing edges). Iterating this process eventually transforms $\mathcal{G}_1$ into $\mathcal{G}_2$ exactly. ■

Armed with Chickering's theorem, GES has its navigation engine. By chaining covered edge reversals, the algorithm can "rotate" through DAGs inside the current equivalence class — staying within the MEC the entire time, with no need to invoke Meek's rules between candidate evaluations. This lets it safely evaluate whether a candidate edge addition or deletion is legal across the entire MEC (i.e., that the move does not create a directed cycle in any DAG representative of the class), rather than getting blocked by the arbitrary orientation of a single representative.

5.3.4 The GES Search Phases

The natural way to use a decomposable score is to do greedy hill climbing on it: start from the empty graph, evaluate every possible single-edge addition, accept the one that improves the BIC most, and repeat. But pure forward greedy doesn't suffice for causal discovery, and a small example shows why.

Consider a "diamond" structure in which the true graph has two parallel directed paths from A to D:



Because both paths contribute to the dependence of D on A, the marginal mutual information $I(A;D)$ can easily exceed each of $I(A;B)$, $I(A;C)$, $I(B;D)$, $I(C;D)$ individually. (Contrast this with a chain $A \to B \to C$, where the data processing inequality forces $I(A;C) \leqslant \min(I(A;B), I(B;C))$ — the shortcut never wins, and a chain-shaped truth presents no such trap for greedy search.) Since BIC improvements scale as $n \cdot I$ for each candidate edge, the greedy forward search may pick the shortcut $A \to D$ as the highest-scoring first move from the empty graph. It then continues, adding the four true edges as more conditional dependencies emerge. By the time all four are in place, D is conditionally independent of A given $\{B, C\}$, and the local-score contribution from keeping $A \to D$ has collapsed to

$$I(A; D \mid B, C) = 0,$$

with no remaining likelihood gain to offset the complexity penalty. The shortcut is now a net negative in the final graph — but a forward-only search has no mechanism to notice. It only adds edges.

This is what the backward phase of GES exists to fix. Putting both phases together with the covered-edge-reversal machinery from the previous subsection, GES is a two-phase greedy algorithm:

1. **Forward Equivalence Search (FES):** The algorithm typically starts with an empty graph (though it can be initialized with background knowledge). It iteratively considers all valid single-edge additions to the current DAG, using covered edge reversals when needed to navigate to a DAG representative of the current MEC where the addition is cycle-free. It scores every candidate, accepts the addition with the largest BIC improvement, and repeats. FES stops when no further addition improves the score.
2. **Backward Equivalence Search (BES):** Starting from the graph FES converged to, BES evaluates all possible edge removals (again, navigating among DAG representatives via covered edge reversals as needed). It greedily drops the edge whose removal increases the BIC score the most — typically by reducing the complexity penalty without significantly harming the data likelihood. BES terminates when no further removal improves the score. In the diamond example, BES is exactly the phase that removes the shortcut $A \to D$.

5.3.5 Theoretical Guarantees and Trade-offs

By relying on equivalence classes rather than single DAGs, GES perfectly avoids the local optima that plague standard greedy network searches. **As with the PC algorithm, GES is asymptotically consistent.** In the infinite sample limit, it is mathematically proven to perfectly identify the true CP-DAG of the data-generating process, provided the standard assumptions of Causal Sufficiency, the Markov Condition, and Faithfulness hold.

PC and GES are testing the same things. At the level of individual edge decisions, PC and GES are doing more similar work than the constraint-based vs score-based dichotomy suggests. The asymptotic score difference for adding Z to X's parent set is

$$\text{score}\big(X \mid \mathbf{PA}(X) \cup \{Z\}\big) - \text{score}\big(X \mid \mathbf{PA}(X)\big) \longrightarrow n \cdot I(X; Z \mid \mathbf{PA}(X)) - \frac{\Delta d}{2} \log n,$$

and the $n \cdot I(X; Z \mid \mathbf{PA}(X))$ piece is exactly the population quantity that a PC-style CI test for $X \perp\!\!\!\perp Z \mid \mathbf{PA}(X)$ estimates. Every edge decision GES considers is, in this sense, a conditional independence test in disguise. The two algorithms differ only in the *decision rule* applied to the same signal: PC compares a sample estimate of the effect size against a binary significance threshold, while GES compares it against the continuous complexity penalty $\frac{\Delta d}{2} \log n$. Both are consistent in the limit because they correctly classify the same conditional-independence structure — they just take different finite-sample paths to that classification.

The output is a minimal I-map. The CI-test correspondence above also gives an immediate structural guarantee about GES's output. A DAG $\mathcal{G}$ is an **I-map** for a distribution P if every d-separation in $\mathcal{G}$ corresponds to a conditional independence in P, and a **minimal I-map** if additionally removing any edge would introduce a d-separation that does not hold in P. Reading the score-difference formula in the limit, an edge $Z \to X$ is kept exactly when $I(X; Z \mid \mathbf{PA}(X)) > 0$ (so the $O(n)$ likelihood term dominates the $O(\log n)$ penalty) and removed otherwise. After BES converges, every remaining edge corresponds to a genuine conditional dependence in the data — which is exactly the minimal I-map condition. Equivalently, BES is the phase that brings the implicit CI tests below threshold.

When to use which. When choosing between PC and GES, practitioners typically weigh dataset size and graph density. Constraint-based methods like the PC algorithm are extremely fast for sparse graphs, but because they rely on binary true/false independence tests, they suffer heavily from compounding statistical errors on small datasets. Conversely, score-based methods like GES are more robust to finite-sample statistical noise because they evaluate the graph holistically, but they can become computationally expensive on networks with hundreds of highly connected variables.

5.4 Intervention MECs and Verifying Intervention Sets

Once we have learned a Markov Equivalence Class (MEC) from observational data, we are often left with undirected edges. To fully orient the graph, we must look beyond observational data and begin running experiments.

For example, if the edge $X - Y$ is undirected, we can intervene on X. If we see the distribution of Y change, we can conclusively orient the edge as $X \to Y$. In this section, we will mathematically formalize the information gained from interventions and use it to design efficient sequences of experiments that fully reveal the true causal structure.

5.4.1 Interventions as Mechanism Changes

Earlier in this text, we focused primarily on **do-interventions**, where we forced a variable to a specific constant value and "mutilated" the graph by cutting its incoming edges.

However, when designing experiments to orient a Markov Equivalence Class, we can use a much broader class of interventions. We have generally seen three types of experimental setups:

1. **Do-interventions:** Fix a variable at a specific constant value (e.g., forcing a dosage to exactly 20mg).
2. **RCTs:** Randomize a variable's value independent of its natural parents (e.g., assigning a dosage by flipping a coin).
3. **Shift Interventions:** Shift a variable's natural value by some amount δ (e.g., giving a patient 10mg *more* than their natural baseline).

All of these can be elegantly modeled as changes to the underlying Structural Causal Model (SCM). Suppose our natural SCM assigns each variable via a mechanism $V_i = f_i(\mathbf{PA}(V_i))$. An intervention simply replaces this natural mechanism with a new mechanism, g_i.

Definition 5.4.1. An intervention I on a DAG $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ is defined by a target set $\mathbf{V}_I \subseteq \mathbf{V}$ and a set of new mechanisms $\{g_i\}_{i \in \mathbf{V}_I}$. The post-intervention distribution $\Pr(\mathbf{V} \mid I)$ is generated by:

$$V_i = \mathbb{1}[V_i \notin \mathbf{V}_I] f_i(\mathbf{PA}(V_i)) + \mathbb{1}[V_i \in \mathbf{V}_I] g_i(\mathbf{PA}(V_i)) \tag{5.1}$$

($\mathbb{1}$ here is used to denote an indicator function.)

This definition perfectly captures our three experimental types. If the new mechanism $g_i(\mathbf{PA}(V_i)) \perp\!\!\!\perp \mathbf{PA}(V_i)$ (as is the case for both do-interventions and randomized RCTs), we call it a **hard** intervention because it completely severs the variable from its natural parents.

Conversely, if the new mechanism g_i still depends on the natural parents (as in a shift intervention where $g_i(\mathbf{PA}(V_i)) = f_i(\mathbf{PA}(V_i)) + \delta$), we call it a **soft** intervention. Regardless of whether an intervention is hard or soft, any local change to the mechanism provides us with structural information about the graph.

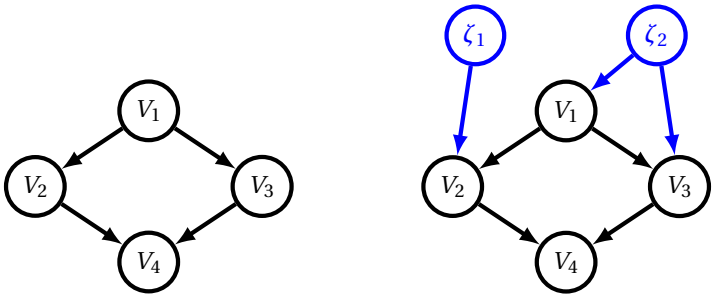
5.4.2 The Intervention Graph

To understand how interventions help us orient edges, we can draw them directly onto our causal DAG. Given a DAG $\mathcal{G}$ and a set of possible interventions $\mathcal{I} = \{I_\emptyset, I_1, \ldots, I_K\}$ (where $I_\emptyset$ is the observational regime), we define the **intervention graph** $\mathcal{G}_\mathcal{I}$ as follows:

- **Nodes:** We keep all original variables $\mathbf{V}$. We add an indicator node ζ_k for every intervention k, where $\zeta_k = 1$ when intervention k is active and 0 otherwise.[1]
- **Edges:** We keep all original edges $\mathbf{E}$. We add a directed edge $\zeta_k \to V_i$ for every variable targeted by intervention k.

[1] The original formulation by Eberhardt [2007] included an additional master switch node ζ^* that selected which intervention regime was active, with each ζ_k a deterministic function of ζ^*. This formalism is useful when formally combining data across regimes because it models the dependence between the intervention choices, e.g., if we combine many singe-node intervention datasets then there is no chance of two interventions happening together, so they cannot be independent exogeneour variables. Still, this detail is not needed for the graphical orientation arguments developed here.

Let's look at an example. Suppose we have a DAG on four vertices and we design two experiments: I_1 targets $\{V_2\}$ and I_2 targets $\{V_1, V_3\}$.



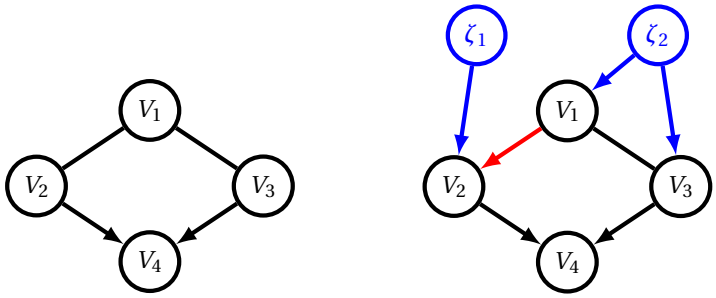
Original DAG $\mathcal{G}$ Intervention Graph $\mathcal{G}_{\mathcal{I}}$

This framework, pioneered by Eberhardt [2007], formally proves our early intuition that "interventions change downstream nodes." Ancestors of the intervened node are d-separated from ζ_k, while descendants are not.

5.4.3 Intervention Markov Equivalence Classes

How does $\mathcal{G}_{\mathcal{I}}$ help us learn the structure? Let's look at the observational CP-DAG for our four-node graph. Without interventions, the only unshielded collider is $V_2 \to V_4 \leftarrow V_3$, leaving $V_1 - V_2$ and $V_1 - V_3$ undirected.

However, once we add our intervention nodes ζ_1 and ζ_2 to the CP-DAG, we can orient one previously-undirected edge — but, importantly, not the other:



Observational CP-DAG Intervention CP-DAG

The key to orienting edges with interventions is realizing that **the orientation of the intervention edge is strictly known**. Because ζ_1 represents our external intervention, it has no parents, and it only points *towards* its target V_2.

Because we know the direction is exactly $\zeta_1 \to V_2$, we can orient adjacent undirected edges like $V_1 - V_2$ in two ways:

1. **Colliders:** If the true structure is $V_1 \to V_2$, then $V_1 \to V_2 \leftarrow \zeta_1$ forms a new unshielded collider in the intervention graph. Independence tests will reveal this, forcing us to orient $V_1 \to V_2$.
2. **Meek's Rules:** If the true structure is $V_1 \leftarrow V_2$, then $\zeta_1 \to V_2 \to V_1$ is not a collider. Because we know $\zeta_1 \to V_2$ is fixed, Meek's rules dictate that $V_2 - V_1$ must be oriented as $V_2 \to V_1$ to avoid creating a false cycle or false collider.

What about the second undirected edge, $V_1 - V_3$? Even though intervention I_2 targets both endpoints, the edge remains undirected. With ζ_2 pointing to both V_1 and V_3, neither orientation of $V_1 - V_3$ creates a new unshielded collider — in either case, ζ_2 is a parent of both endpoints, never a collider. And the Meek-rule logic fails too: the symmetric structure $\zeta_2 \to V_1, \zeta_2 \to V_3$ doesn't force a unique direction along the edge between them. This is exactly the content of the "exactly one endpoint" clause in the Hauser-Bühlmann

lemma below: an intervention orients an edge only when one of its endpoints is intervened on and the other is not.

Definition 5.4.2. Two DAGs $\mathcal{G}$ and $\mathcal{G}'$ are $\mathcal{I}$-**Markov Equivalent** if they share the same conditional independencies in their augmented intervention graphs: $\mathcal{I}_{\perp\!\!\!\perp}(\mathcal{G}_{\mathcal{I}}) = \mathcal{I}_{\perp\!\!\!\perp}(\mathcal{G}'_{\mathcal{I}})$. The set of all such equivalent graphs is denoted $M_{\mathcal{I}}(\mathcal{G})$.

Lemma 5.4.3 (Hauser and Bühlmann [2012])**.** *For a set of **hard** interventions $\mathcal{I}$ (including the observational regime $I_\emptyset$), two DAGs $\mathcal{G}$ and $\mathcal{G}'$ are $\mathcal{I}$-Markov equivalent if and only if:*

- *They have the same skeleton and v-structures (standard observational MEC).*
- *For every intervention $I_k \in \mathcal{I}$ and every edge of the (shared) skeleton with* exactly one *endpoint in I_k, that edge has the same orientation in $\mathcal{G}$ and $\mathcal{G}'$.*

This characterization extends to soft interventions: Yang et al. [2018] showed that, under mild faithfulness-like assumptions, general interventions identify the same equivalence class as hard interventions, so the lemma applies in both settings.

The "exactly one endpoint" clause covers single-node and multi-node interventions in one condition. For a single-node intervention $I_k = \{V_i\}$, every edge incident to V_i has exactly one endpoint in I_k, so all such edges get oriented — the intuitive "intervene on V_i, orient its neighbors" picture. For a multi-node intervention I_k, only edges that *cross the boundary* of I_k are oriented; edges entirely inside I_k are exactly the $V_1 - V_3$ situation above.

> **Main Idea 25**
>
> Interventions orient edges incident to the intervened nodes — but only when *exactly one* endpoint of the edge is targeted. An edge with both endpoints in the same intervention remains undirected, because the symmetric $\zeta_k \to V_i, \zeta_k \to V_j$ structure neither forms a new collider nor propagates orientation via Meek's rules.

5.4.4 Verifying Intervention Sets

Experiments are expensive. The natural question is: what is the minimum number of interventions required to fully orient the graph? Before answering, it is worth distinguishing two related settings.

> **Verification.** We already have a candidate DAG $\mathcal{G}$ in mind — typically the output of an observational discovery algorithm paired with whatever domain knowledge picks one DAG out of its MEC. We want to design a set of interventions $\mathcal{I}$ that confirms $\mathcal{G}$ by reducing the interventional MEC to a single graph: $|M_{\mathcal{I}}(\mathcal{G})| = 1$. Because we already know what we are looking for, the interventions can be tailored to the specific structure of $\mathcal{G}$.
>
> **Search.** We do not have a candidate DAG; we know only the observational MEC. We want to design interventions that will identify the true DAG, whatever it turns out to be in the MEC. This problem is generally harder because the intervention set must succeed across every DAG consistent with the MEC — and it often requires *adaptive* strategies, where each intervention's design depends on the results of previous ones.

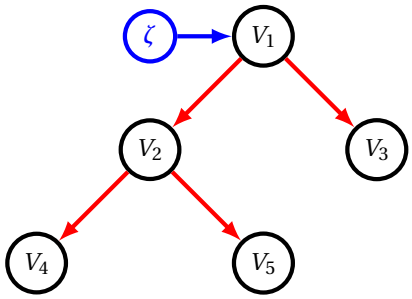
The two problems can differ sharply in cost. As we will see, verification on a tree takes just one intervention — but only because we already know which node is the root. The corresponding search problem (orient an unknown tree from observational and interventional data alone) requires more, since the first intervention has to be chosen without knowing which node is upstream of which. This section focuses on the verification problem.

Definition 5.4.4. For a DAG $\mathcal{G}$, a set of interventions $\mathcal{I}$ is a **verifying intervention set** if it reduces the equivalence class to a single graph: $|M_{\mathcal{I}}(\mathcal{G})| = 1$.

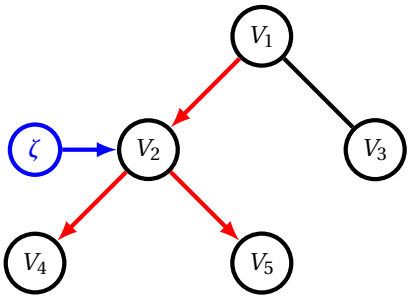
Let's look at verifying sets for two extreme topological cases: Trees and Cliques.

Trees

When given a directed tree with no v-structures, a single intervention on the root node is sufficient to verify the entire structure. Compare this to intervening on a downstream node, which fails to verify the graph.



Intervention on V_1 (Verifies)



Intervention on V_2 (Fails)

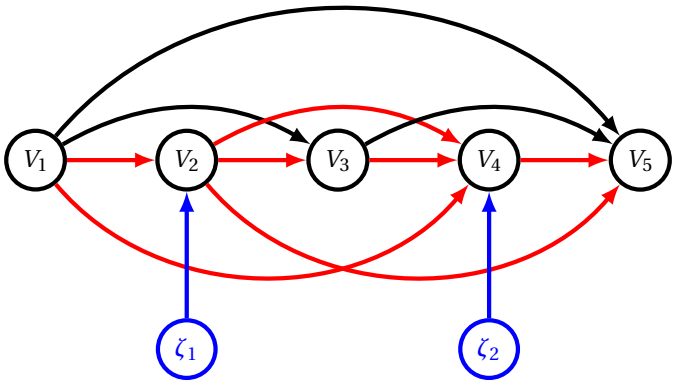
By intervening on the root V_1 (left graph), we form colliders that orient $V_1 \to V_2$ and $V_1 \to V_3$. Once these are oriented, Meek's rules dictate that all subsequent edges must be directed downwards to avoid creating new v-structures.

Conversely, intervening on V_2 (right graph) tells us that $V_1 \to V_2$ (as it forms a collider with $\zeta \to V_2$), and pushes orientations down to V_4 and V_5. However, it leaves us completely blind to $V_1 - V_3$, leaving ambiguity as to where the true root of the tree is.

This single-intervention bound is specifically a *verification* bound: we are allowed to inspect $\mathcal{G}$ before choosing the target, so we can pick the root. In the search version — where we have only the unoriented tree skeleton and no idea where the root sits — a single intervention cannot guarantee full orientation, since the strategy must succeed regardless of which node turns out to be the root.

Cliques

Consider a fully connected DAG (a clique) on 5 nodes, where $V_i \to V_j$ for all $i < j$. Because there are no missing edges, there are zero unshielded colliders. The observational CP-DAG is completely undirected!



To verify this clique, it is sufficient to intervene on just the even nodes: V_2 and V_4. This directly orients all edges connected to V_2 and V_4 (shown in red). Once the red edges are locked in, the remaining black edges ($V_1 \to V_3$, $V_3 \to V_5$, $V_1 \to V_5$) are automatically oriented by Meek's rules to prevent cycles.

In general, a clique on n nodes can always be verified using $\lfloor n/2 \rfloor$ single-node interventions on the even-indexed vertices. In general, a clique on n nodes can always be verified using $\lfloor n/2 \rfloor$ single-node interventions on the even-indexed vertices. If we allow multi-node interventions, Shanmugam et al. [2015] showed that the cost drops sharply to $\Theta(\log n)$, using a construction based on *separating systems* from combinatorial design theory — families of vertex subsets that distinguish every pair of nodes through their

intervention patterns. They also proved this bound is tight, so the linear/logarithmic gap between single-node and multi-node interventions is fundamental.

Both of these figures presume that we already know the topological order of the clique — i.e., which node is V_1, which is V_2, and so on. This is the verification setting: with the order in hand, we can name the targets as "the even-indexed vertices." In the search setting we only have the (fully undirected) clique CP-DAG, no such labeling is available, and identifying the topological order is itself part of what the interventions must accomplish. The cost in the search setting is generically higher.

5.4.5 Algorithmic Verification and Search

The Trees and Cliques examples relied on inspecting $\mathcal{G}$ by hand to pick clever target sets. For general DAGs we need an algorithm. The combinatorial-design view of intervention sets was developed by Shanmugam et al. [2015], who studied the clique case using separating systems. Choo et al. [2022] generalized this lens to arbitrary DAGs and gave a tight bound in terms of a graph-theoretic quantity called the **verification number**, denoted $\nu(\mathcal{G})$.

To see what $\nu(\mathcal{G})$ actually counts, recall **covered edges** from Chickering's Theorem in the GES section, i.e., edges $V_i \to V_j$ where $\mathbf{PA}(V_i) = \mathbf{PA}(V_j) \setminus \{V_i\}$. An intervention on V_i breaks every covered edge incident to V_i at once, because adding $\zeta_i \to V_i$ changes $\mathbf{PA}(V_i)$ but no other parent set. Choo et al. [2022] show that an intervention set verifies $\mathcal{G}$ exactly when its augmented graph $\mathcal{G}_{\mathcal{I}}$ contains no covered edges, so $\nu(\mathcal{G})$ is the *minimum vertex cover of the covered-edge subgraph of* $\mathcal{G}$. In our K_5 clique example, the covered edges turn out to be the path $V_1 - V_2 - V_3 - V_4 - V_5$, and its minimum vertex cover is $\{V_2, V_4\}$ — exactly the "intervene on the even nodes" rule. The computation is polynomial because the covered edges of a DAG always form a forest, and minimum vertex cover on a forest is solvable in linear time.

The verification number is the right way to talk about the cost of running experiments, but it leaves open the harder question. In the search problem we have only the observational MEC and need to discover the true DAG without knowing it in advance. Choo et al. [2022] resolve this in terms of the verification number, and the bound is tight.

Theorem 5.4.5 (Choo et al. 2022)**.** *Let $\mathcal{G}$ be a DAG on n nodes. There exists an adaptive search algorithm that, given only the observational MEC of $\mathcal{G}$, identifies $\mathcal{G}$ using*

$$\mathcal{O}\big(\nu(\mathcal{G}) \cdot \log n\big)$$

atomic interventions in the worst case. Moreover, this bound is tight: there exist essential line graphs on n nodes for which any search algorithm requires $\Omega\big(\nu(\mathcal{G}) \cdot \log n\big)$ interventions in the worst case.

The upper-bound algorithm is built on **graph separators** — sets of vertices whose removal breaks the graph into small components. Intuitively, intervening on a good separator orients the edges incident to it and splits the unresolved portion of the MEC into independent subproblems of roughly half the size, yielding the $\log n$ factor through divide-and-conquer. The full paper is a satisfying read for students with more background in graph theory, and we recommend it to anyone interested in the algorithmic side of causal discovery.

The shape of the result is worth dwelling on. The verification number $\nu(\mathcal{G})$ depends only on the target DAG (not on the size of its MEC) so the cost of search is governed by the intrinsic complexity of $\mathcal{G}$, with $\log n$ overhead for the ambiguity left on the table by observational data alone.

5.5 Independent Component Analysis

The interventions we've considered so far have been explicit and known: we set a variable, observe the consequences, and design the experiment around what we did. But interventions happen in the wild without our knowledge too — a regulator quietly tightens a standard, a supplier swaps a component, a sensor is recalibrated. The downstream data look like ordinary observations whose noise distribution has been silently nudged.

This unannounced shifting turns out to be useful. Gaussian noise, while mathematically convenient, is the *worst case* for identifying causal direction — it leaves us trapped in the kind of symmetries that produce

Markov equivalence classes and, as we are about to see, rotational ambiguities. Non-Gaussianity breaks those symmetries: it helps us disentangle mixed signals (the subject of this section) and, as we will see in the LiNGAM section that follows, disentangle causal direction itself. Random shifts and unobserved interventions tend to push noise away from Gaussianity, which is exactly the resource these methods exploit.

We begin with the disentanglement-of-signals problem in its classical form: the cocktail party.

5.5.1 The Cocktail Party

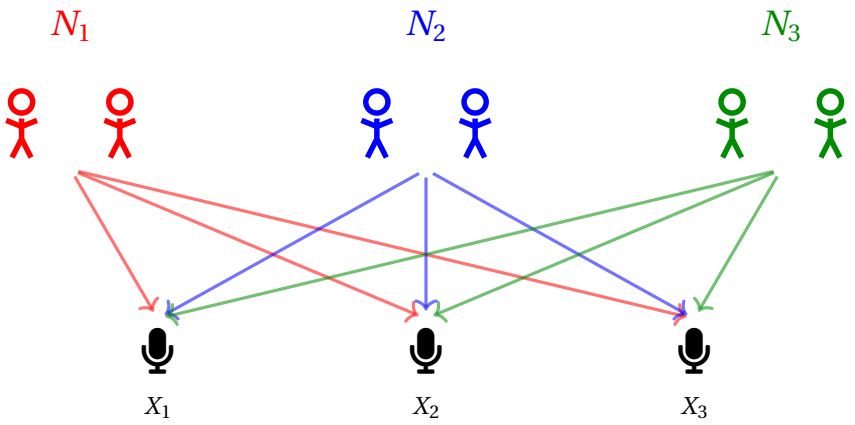


Figure 5.1: The cocktail party. Three conversations (N_1, N_2, N_3, color-coded) take place in a room; each of three microphones records a linear mixture of all three conversations. ICA recovers the original sources from the microphone signals by searching for a demixing matrix $\mathbf{W} = \mathbf{M}^{-1}$.

Figure 5.1 shows the setup: n conversations $N_1, \ldots, N_n$ take place in a room, and n microphones each pick up a different linear mixture of all of them. Writing $\mathbf{X} = (X_1, \ldots, X_n)^T$ for the microphone signals and $\mathbf{N} = (N_1, \ldots, N_n)^T$ for the conversations, the observations are related to the sources by an unknown mixing matrix $\mathbf{M}$:

$$\mathbf{X} = \mathbf{M}\mathbf{N}.$$

The goal of independent component analysis (ICA) is to recover the original independent sources $\mathbf{N}$ from the mixed observations $\mathbf{X}$ by searching for a demixing matrix $\mathbf{W} = \mathbf{M}^{-1}$ such that $\mathbf{N} = \mathbf{W}\mathbf{X}$.

To expose what makes the problem hard — and what makes it solvable — it is enough to work through the two-source case in detail:

$$\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} = \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix} \begin{pmatrix} N_1 \\ N_2 \end{pmatrix}. \tag{5.2}$$

When our sources have *Gaussian* noise, we are not able to uniquely identify $\mathbf{N}$ because many transformations can yield two independent variables. For example, define two new variables:

$$Y_1 = b_{11} N_1 + b_{12} N_2$$
$$Y_2 = b_{21} N_1 + b_{22} N_2$$

If everything is Gaussian, then $Y_1 \perp\!\!\!\perp Y_2 \Leftrightarrow \mathrm{Cov}(Y_1, Y_2) = 0$. Because we have assumed the original sources are independent ($N_1 \perp\!\!\!\perp N_2$), we have:

$$\mathrm{Cov}(Y_1, Y_2) = b_{11} b_{21} + b_{12} b_{22} = \begin{pmatrix} b_{11} & b_{12} \end{pmatrix} \begin{pmatrix} b_{21} \\ b_{22} \end{pmatrix}. \tag{5.3}$$

Hence, *any orthogonal basis change* on N_1, N_2 also gives perfectly independent components. If $\mathbf{M}^{-1}\mathbf{X}$ gives independent components, so does $\mathbf{B}\mathbf{M}^{-1}\mathbf{X}$ for any orthogonal matrix $\mathbf{B}$. The distributions are rotationally invariant, making the true sources unidentifiable. Fortunately, the same is not true for *non-Gaussian* noise.

Theorem 5.5.1 (Darmois-Skitovich)**.** *Let $N_1, \dots, N_n$ be independent, non-degenerate random variables. If there exist coefficients $\alpha_1, \dots, \alpha_n$ and $\beta_1, \dots, \beta_n$ that are all non-zero such that the two linear combinations*

$$Y_1 = \alpha_1 N_1 + \dots + \alpha_n N_n \quad \textit{and} \quad Y_2 = \beta_1 N_1 + \dots + \beta_n N_n$$

are independent, then each N_i is normally distributed.

Read in its contrapositive, this is exactly what licenses ICA: when the sources are non-Gaussian, mixing them always destroys independence, so the only independent components to be found are the true sources themselves. Before stating this precisely, it helps to see *why* the Gaussian is the lone exception.

Proof Intuition: The theorem has a clean geometric picture in the two-variable case. Independent random variables N_1, N_2 have a joint density that factorizes: $p(N_1, N_2) = f(N_1)\, g(N_2)$. For independent Gaussians, this factorization combines with the quadratic exponent,

$$e^{-N_1^2}\, e^{-N_2^2} \;=\; e^{-(N_1^2+N_2^2)} \;=\; e^{-r^2},$$

so the joint density depends only on the radius $r = \sqrt{N_1^2 + N_2^2}$. The bivariate Gaussian density is *rotationally symmetric*: its level sets are concentric circles (Figure 5.2).

This rotational symmetry is what makes Gaussians robust to mixing. Any rotation of the coordinate system $\mathbf{Y} = \mathbf{R}_\theta \mathbf{N}$ leaves the circular level sets in place, so Y_1, Y_2 are also independent, with the same marginals as N_1, N_2. Any pair of orthogonal linear combinations of independent Gaussians is itself a pair of independent Gaussians.

The Gaussian is the *only* distribution for which this works. Demanding that a factorized density also depend only on r^2 gives the functional equation

$$f(N_1)\, g(N_2) = h(N_1^2 + N_2^2).$$

Setting $N_2 = 0$ shows $f(N_1) = h(N_1^2)/g(0)$, so f depends only on N_1^2; similarly g depends only on N_2^2. Taking logs in the original equation,

$$\log f(N_1) + \log g(N_2) = \log h(N_1^2 + N_2^2),$$

and writing $F(N_1^2) = \log f(N_1)$, $G(N_2^2) = \log g(N_2)$, the equation becomes

$$F(a) + G(b) = \log h(a + b),$$

where $a = N_1^2$ and $b = N_2^2$. The right side depends only on the sum $a + b$, so holding $a + b$ fixed while shifting a up by δ (and b down by δ) cannot change anything — meaning $F'(a) = G'(b)$ for all a, b. Both derivatives must then be the same constant, and F and G are linear with the same slope, which we will call α. Translating back,

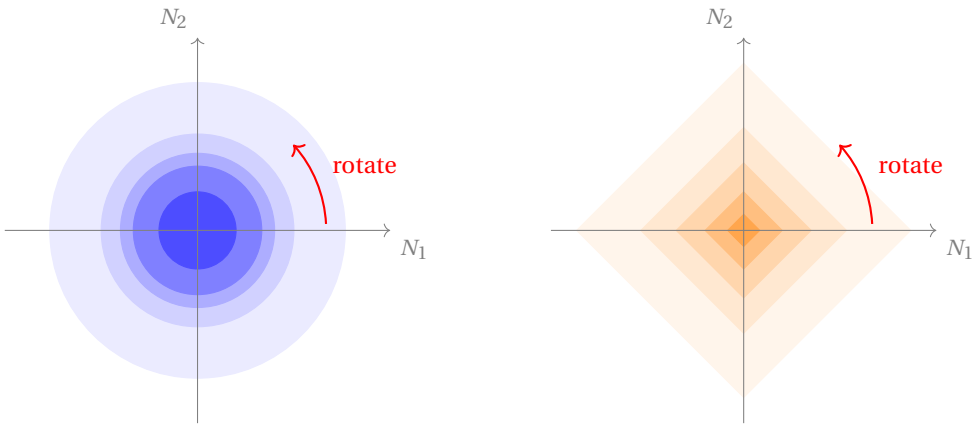
$$f(N_1) = c_1\, e^{-\alpha N_1^2}, \qquad\qquad g(N_2) = c_2\, e^{-\alpha N_2^2},$$

with the same α for both marginals. The Gaussian is forced. So if rotating independent N_1, N_2 preserves their independence, N_1 and N_2 must already have been Gaussian.

The full Darmois-Skitovich theorem generalizes this from rotations to arbitrary non-trivial linear combinations. The standard proof uses **characteristic functions** — complex-valued cousins of moment-generating functions, $\phi_X(t) = \mathbb{E}[e^{itX}]$ — to convert the joint-independence condition into a functional equation whose only solution is a quadratic exponent (the log of a Gaussian characteristic function). The geometric argument for rotations and the analytic argument for general linear combinations are two faces of the same theorem. ■

That picture pays off immediately. The obstruction was a rotation *within* the subspace of Gaussian components, so a lone Gaussian source has nothing to rotate against and causes no trouble — we can afford exactly one.

Corollary 5.5.2 (Identifiability)**.** *Suppose the independent sources $N_1, \dots, N_n$ are non-degenerate and at most one of them is Gaussian. Then any invertible demixing $\mathbf{W}$ that makes the components of $\mathbf{WX}$ mutually independent recovers $N_1, \dots, N_n$ uniquely, up to scaling and permutation.*



Gaussian: circular level sets
(rotation preserves independence)

Laplacian: diamond level sets
(rotation breaks independence)

Figure 5.2: The joint density of two independent random variables. **Left:** for two independent Gaussians, $f(N_1)\,g(N_2) = e^{-(N_1^2+N_2^2)/2}/(2\pi)$ depends only on r^2, so its level sets are concentric circles. A rotation of the coordinates leaves the picture unchanged, and the rotated components Y_1, Y_2 are still independent. **Right:** for two independent Laplacians, $f(N_1)\,g(N_2) \propto e^{-|N_1|-|N_2|}$ has level sets aligned with the axes (diamonds). A rotation deforms this shape, and the rotated components are dependent. The Gaussian is the unique distribution combining independent factorization with rotational symmetry.

This is exactly what lets us perform Independent Component Analysis and assign true meaning to the recovered components. With identifiability established, the question becomes how to actually find a good demixing matrix in practice. Two families of algorithms have emerged, one for each side of the same coin: *minimizing dependence* between the recovered components, and *maximizing non-Gaussianity* of each of them.

5.5.2 Minimizing Dependence

The first family takes the most direct route. The defining assumption of ICA is that the true sources are mutually independent, so we can measure dependence directly and search for the demixing matrix that drives it to zero.

The natural measure is **mutual information** (which we saw earlier when we discussed GES):

$$I(\hat{N}_1;\ldots;\hat{N}_n) = \underbrace{\sum_{i=1}^{n} H(\hat{N}_i)}_{\text{Sum of Marginal Entropies}} \underbrace{- H(\hat{N}_1,\ldots,\hat{N}_n)}_{\text{Joint Entropy}},$$

which equals zero exactly when the recovered components are mutually independent and is positive otherwise. Equivalently, it is the KL divergence between the joint distribution of the recovered components and the product of their marginals — literally measuring how far the joint is from "looking like" independent sources.

The catch is that mutual information is difficult to estimate from samples in high dimensions. Practical algorithms therefore optimize tractable surrogates. The classical example is the **Infomax** algorithm of Bell and Sejnowski [1995], which maximizes the entropy of a nonlinearly transformed output and, under standard preprocessing, is equivalent to minimizing the mutual information of the recovered components. Kernel-based methods (e.g., kernel ICA) provide another route, using reproducing kernel Hilbert space norms as a dependence measure that is easier to estimate.

5.5.3 Maximizing Non-Gaussianity

The second family searches for components that are as far from Gaussian as possible. The intuition comes from the Central Limit Theorem: the sum of independent random variables is strictly *more* Gaussian than either summand. So any incorrect demixing yields outputs that are still mixtures — $\alpha_1 N_1 + \alpha_2 N_2 + \ldots$ — and these mixtures will look more Gaussian than the true sources N_i. Pushing the recovered components as far from Gaussian as possible therefore reverses the mixing.

To do this mathematically, ICA defines a **contrast function** (often denoted J). A contrast function operationalizes the measurement of "Gaussian-ness" by calculating the difference between the expected value of some function evaluated on our actual data, and the expected value of that same function if the data were perfectly Gaussian.

Let Y be our candidate component (standardized to have mean 0 and variance 1), and let Z be a standard normal Gaussian variable ($Z \sim \mathcal{N}(0,1)$). A general contrast function takes the form:

$$J(Y) \propto \left(\mathbb{E}[G(Y)] - \mathbb{E}[G(Z)]\right)^2 \tag{5.4}$$

where $G(\cdot)$ is some non-quadratic function. If Y is perfectly Gaussian, $\mathbb{E}[G(Y)]$ equals $\mathbb{E}[G(Z)]$ and the contrast function drops to zero. The larger $J(Y)$ is, the less Gaussian our component is.

Two common measures of non-Gaussianity fit into this framework:

- **Kurtosis:** Kurtosis measures how "fat" the tails of a distribution are. It is the fourth-order moment: $\text{kurt}(Y) = \mathbb{E}[Y^4] - 3(\mathbb{E}[Y^2])^2$. For a standard Gaussian Z, $\mathbb{E}[Z^4] = 3$, so its kurtosis is exactly zero. Using $G(y) = y^4$ in our contrast function recovers absolute kurtosis as the objective.
- **Negentropy:** A Gaussian distribution has the largest possible entropy (randomness) among all distributions with a specified mean and variance. Negentropy is defined as $J(Y) = H(Z) - H(Y)$, where H is the information-theoretic entropy. Maximizing negentropy pushes the distribution as far from Gaussian as possible. Because true negentropy is computationally difficult to estimate from samples, FastICA approximates it with the contrast-function form above using smooth functions for G such as $\log\cosh(y)$ or $\exp(-y^2/2)$.

By using gradient ascent on a contrast function of this form, FastICA and related algorithms efficiently cut through the rotational ambiguity of Gaussian variables and isolate the true, non-Gaussian sources.

Two perspectives on the same thing. The two approaches are more closely related than they look. After whitening — centering $\mathbf{X}$ and decorrelating it so that $\text{Cov}(\mathbf{X}) = \mathbf{I}$ — the demixing matrix $\mathbf{W}$ is constrained to be a rotation. On rotations, the mutual information of the recovered components decomposes (up to a constant) as the negative of the sum of their marginal negentropies. Minimizing dependence then reduces to maximizing total non-Gaussianity, and the two families coincide. The non-Gaussianity view gives a more tractable per-component objective; the dependence-minimization view is more transparent about what ICA is ultimately trying to achieve.

5.6 LiNGAMs

Building directly on the concepts of Independent Component Analysis, Shimizu et al. [2006] introduced the concept of a **Li**near **N**on-**G**aussian **A**cyclic **M**odel (LiNGAM). The central idea is that if we restrict our Structural Causal Models to be linear and have non-Gaussian noise, the true causal direction becomes completely identifiable from observational data alone.

5.6.1 Algorithm 1: ICA-LiNGAM

Because we just established how ICA recovers independent signals from linear mixtures, let's view the causal discovery problem as a pure ICA problem.

Consider a concrete linear SCM on three variables, V_1, V_2, and V_3, arranged in a chain: V_1 influences V_2, and V_2 influences V_3, but V_1 has no *direct* effect on V_3.

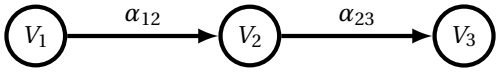


Figure 5.3: A small linear chain. V_1 drives V_2 (weight α_{12}), which drives V_3 (weight α_{23}). There is no direct $V_1 \to V_3$ edge, so V_1 affects V_3 only indirectly, through V_2.

Each node contributes one structural equation — a variable equals the weighted sum of its parents plus its own independent noise:

$$\begin{aligned} V_1 &= N_1, \\ V_2 &= \alpha_{12} V_1 + N_2, \\ V_3 &= \alpha_{23} V_2 + N_3. \end{aligned}$$

Listing the variables in causal order as $\mathbf{V} = (V_1, V_2, V_3)^T$ and their noises as $\mathbf{N} = (N_1, N_2, N_3)^T$, these three scalar equations collapse into the single matrix equation $\mathbf{V} = \mathbf{AV} + \mathbf{N}$:

$$\begin{pmatrix} V_1 \\ V_2 \\ V_3 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ \alpha_{12} & 0 & 0 \\ 0 & \alpha_{23} & 0 \end{pmatrix} \begin{pmatrix} V_1 \\ V_2 \\ V_3 \end{pmatrix} + \begin{pmatrix} N_1 \\ N_2 \\ N_3 \end{pmatrix}.$$

Two things about $\mathbf{A}$ are worth highlighting. The entry $\mathbf{A}_{ij}$ is nonzero *precisely* when the graph has an edge from variable j to variable i: the only nonzeros are the coefficient of V_1 in V_2's row (α_{12}) and the coefficient of V_2 in V_3's row (α_{23}) — one per arrow. In particular, the (V_3, V_1) entry is zero, even though V_1 clearly influences V_3; but that influence runs *through* V_2, and $\mathbf{A}$ stores direct edges, not ancestry. Because each variable is listed after all of its parents, every nonzero falls strictly below the diagonal: $\mathbf{A}$ is strictly lower triangular for all acyclic models.

Suppose each variable $V_i \in \mathbf{V}$ is a linear function of the other variables in $\mathbf{V}$, specified by coefficients in a strictly lower-triangular adjacency matrix $\mathbf{A}$, plus its own independent source of noise N_i. In matrix notation, the entire Structural Causal Model is:

$$\mathbf{V} = \mathbf{AV} + \mathbf{N} \tag{5.5}$$

We can rearrange this using the identity matrix $\mathbf{I}$:

$$(\mathbf{I} - \mathbf{A})\mathbf{V} = \mathbf{N} \tag{5.6}$$

Notice that $\mathbf{N}$ is a set of mutually independent noise components! This matches the exact formulation of Independent Component Analysis, where we search for a demixing matrix $\mathbf{W}$ such that $\mathbf{WV}$ yields independent components. Therefore, the ICA demixing matrix $\mathbf{W}$ must be a permuted and scaled version of $(\mathbf{I} - \mathbf{A})$.

There is just one problem: ICA returns independent components and a demixing matrix *up to permutations of the rows*, since it doesn't really care which independent component is the first one, second one, etc. Let's return to the DAG in Figure 5.3. Recall that its true demixing matrix is

$$\mathbf{I} - \mathbf{A} = \begin{pmatrix} 1 & 0 & 0 \\ -\alpha_{12} & 1 & 0 \\ 0 & -\alpha_{23} & 1 \end{pmatrix},$$

with rows ordered (V_1, V_2, V_3) to match the columns. Instead of this matrix, ICA may hand back a matrix with its first two rows swapped (the rows could also be scaled arbitrarily):

$$\mathbf{W} = \begin{pmatrix} -\alpha_{12} & 1 & 0 \\ 1 & 0 & 0 \\ 0 & -\alpha_{23} & 1 \end{pmatrix}.$$

If you were to try to interpret this matrix as a graph, reading row 1 as V_1's equation, row 2 as V_2's, row 3 as V_3's, you would get nonsense.

How do we tell which alignment is the correct one? The diagonal gives it away. When the rows and columns are matched correctly, every diagonal entry is a variable's coefficient on itself, which is always 1, so the diagonal is automatically free of zeros. Any *incorrect* alignment is a non-trivial permutation of the rows, and this is where acyclicity does the work. Take the swap above: it puts V_2's row in slot 1 and V_1's row in slot 2. For both diagonal entries to be nonzero we would need an edge $V_1 \to V_2$ *and* an edge $V_2 \to V_1$, which is a 2-cycle. The first exists, but the second cannot, so slot 2 is stuck with a zero ($\mathbf{W}_{2,2} = 0$). The same trap springs for any wrong permutation: it must contain a cycle of slots $i_1, i_2, \ldots, i_m$ (with $m \geqslant 2$) in which V_{i_2}'s row sits in slot i_1, V_{i_3}'s row in slot i_2, and so on around to V_{i_1}'s row in slot i_m. Keeping the diagonal zero-free along that cycle would require edges making i_1 a parent of i_2, i_2 a parent of i_3, …, and i_m a parent of i_1 — a directed cycle $i_1 \to i_2 \to \cdots \to i_m \to i_1$ in the graph, which acyclicity forbids. Notice this argument never used sparsity: it holds even for a fully connected DAG, the densest an acyclic graph can be, because the one edge needed to close the loop is exactly the edge acyclicity rules out. The correct alignment is therefore the *unique* permutation with a zero-free diagonal. The solution is to permute the rows until the diagonal has no zeros; the only ordering that achieves it swaps rows 1 and 2 back, restoring

$$\begin{pmatrix} 1 & 0 & 0 \\ -\alpha_{12} & 1 & 0 \\ 0 & -\alpha_{23} & 1 \end{pmatrix} = \mathbf{I} - \mathbf{A}.$$

Dividing each row by its (now nonzero) diagonal entry sets the diagonal to 1, and $\hat{\mathbf{A}} = \mathbf{I} - \mathbf{W}_{\text{normalized}}$ returns

$$\hat{\mathbf{A}} = \begin{pmatrix} 0 & 0 & 0 \\ \alpha_{12} & 0 & 0 \\ 0 & \alpha_{23} & 0 \end{pmatrix}.$$

Now, the edges $V_1 \to V_2$ and $V_2 \to V_3$ are recovered, and nothing else.

Putting this all together, we get the ICA-LiNGAM procedure:

1. Perform ICA on the data to find a demixing matrix $\mathbf{W}$ such that $\mathbf{WV}$ gives independent components.
2. Permute the rows of $\mathbf{W}$ to find the unique arrangement where there are no zeros on the diagonal.
3. Divide each row of this permuted matrix by its diagonal element. This normalizes the matrix so that the diagonal entries are exactly 1. This normalized matrix is now equal to $(\mathbf{I} - \mathbf{A})$.
4. Solve for the adjacency matrix by calculating $\hat{\mathbf{A}} = \mathbf{I} - \mathbf{W}_{\text{normalized}}$.

At this point $\hat{\mathbf{A}}$ already *is* the recovered graph — its nonzero pattern gives the edges and its entries give their weights. If we also want the causal ordering written out explicitly, we can read it off by topologically sorting $\hat{\mathbf{A}}$ into a lower-triangular matrix.

5.6.2 The Bivariate Intuition and DirectLiNGAM

While the ICA matrix formulation is elegant, it can sometimes be computationally unstable due to the optimization landscape of contrast functions. However, the fundamental intuition for why ICA works can be exploited more directly. To better understand this, let's look at the bivariate case.

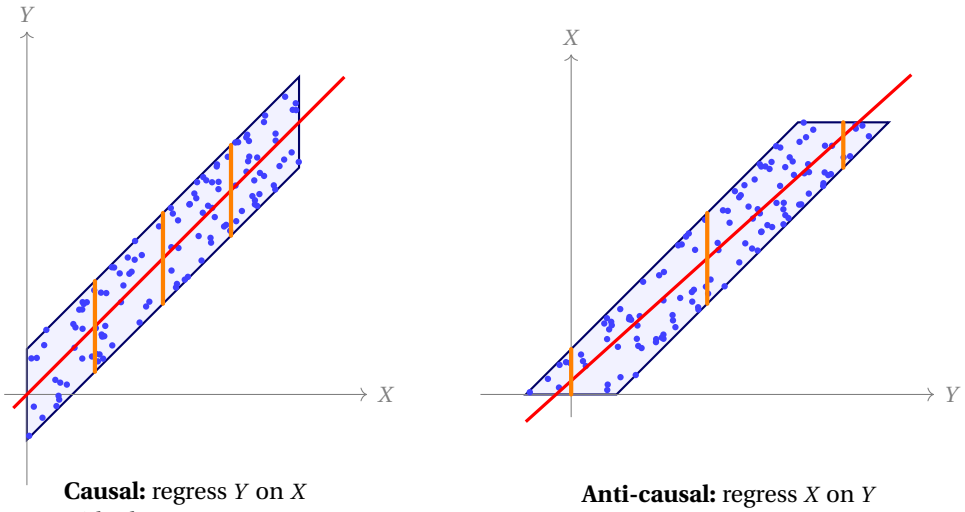
Consider the causal relationship $X \to Y$:

$$\begin{aligned} X &= N_X \\ Y &= \alpha X + N_Y \end{aligned}$$

By assumption, the noise terms are independent ($N_X \perp\!\!\!\perp N_Y$), centered ($\mathbb{E}[N_X] = \mathbb{E}[N_Y] = 0$), and crucially, **non-Gaussian**.

In the true causal direction, we predict Y using X as input via standard linear regression, which yields the coefficient $\alpha = \frac{\text{Cov}(X,Y)}{\text{Var}(X)}$. The residuals are:

$$R_{Y \mid X} = Y - \alpha X = N_Y \tag{5.7}$$



Causal: regress Y on X
residuals same at every X

Anti-causal: regress X on Y
residuals shrink at extreme Y

Figure 5.4: The bivariate LiNGAM asymmetry, with $X \sim \text{Uniform}(0,6)$ and $Y = X + N_Y$, $N_Y \sim \text{Uniform}(-1,1)$. The joint support is a parallelogram. **Left:** regressing Y on X in the true causal direction, the parallelogram has vertical left and right edges. The OLS line $\hat{Y} = X$ splits each vertical edge in half, and the residual $Y - \hat{Y} = N_Y$ has the same uniform $[-1,1]$ distribution at every X. So $R_{Y|X} \perp\!\!\!\perp X$. **Right:** regressing X on Y in the anti-causal direction (axes swapped), the parallelogram now has horizontal short edges. The vertical extent of the cloud — the conditional range of X given Y — is full in the middle of the parallelogram but shrinks to nothing at the extreme corners. The residual $R_{X|Y}$ depends on Y, and the anti-causal SCM fails to produce independent noise. Darmois-Skitovich is what guarantees this asymmetry whenever the noise is non-Gaussian.

Because we defined $N_X \perp\!\!\!\perp N_Y$, and $X = N_X$, it is trivially true that the residual is independent of the explanatory variable: $R_{Y\,|\,X} \perp\!\!\!\perp X$.

Now, let's look at the anti-causal direction. We proceed by rewriting the structural equation backwards, regressing X on Y. This gives the reverse coefficient $\alpha' = \frac{\text{Cov}(X,Y)}{\text{Var}(Y)}$.

$$X = \alpha' Y + R_{X\,|\,Y} \tag{5.8}$$

The residual is therefore $R_{X\,|\,Y} = X - \alpha' Y$. Let's replace X and Y with our original structural equations to view this entirely in terms of the underlying noise variables:

$$R_{X\,|\,Y} = N_X - \alpha'(\alpha N_X + N_Y) = (1 - \alpha'\alpha)N_X - \alpha' N_Y \tag{5.9}$$

We now see that both Y and $R_{X\,|\,Y}$ are two different linear combinations of N_X and N_Y. Because LiNGAM assumes the noise variables are *non-Gaussian*, Corollary 5.5.2 (from the Darmois-Skitovich theorem in the previous section) tells us that these two specific linear combinations *must* be dependent! For a visual example, see Figure 5.4. Notice that when the axes are swapped, the conditional range of X given Y collapses at the extreme corners of the parallelogram, so the residual depends on Y.

Theorem 5.6.1 ([Shimizu et al., 2006])**.** *For a LiNGAM, if the true SCM is $X = N_X$ and $Y = \alpha X + N_Y$ with non-Gaussian noise, then there does not exist an SCM in the reverse direction $Y = N'_Y$ and $X = \alpha' Y + N'_X$ where the reverse noise terms are independent.*

Because the residual is dependent on the explanatory variable in the reverse direction, we can conclusively rule it out.

Main Idea 26

Causal direction in a LiNGAM is uniquely encoded in the independence between a regression residual and its explanatory variable.

Algorithm 2: DirectLiNGAM. Introduced a few years after the original paper, DirectLiNGAM [Shimizu et al., 2011] takes this exact residual intuition and applies it iteratively. In any DAG there must be at least one exogenous root node with no direct causes. DirectLiNGAM finds a root with a regression test: regress every other variable on a candidate and form the residuals; the candidate is a root exactly when it is independent of all of those residuals, since any variable that has a parent would fail the test for at least one of its regressions. When several variables qualify, it commits to the single most independent one.

Once we have a root, we strip its effect out using the residualization (partialling out) idea from Chapter 4.1: using the linear model we just fit, we replace every remaining variable with its residual after regressing that root out of it. We are now looking at the "conditioned" distribution with that root's influence removed, so some new variable becomes exogenous and takes over as the next root. Repeating this "peels off" the DAG one root at a time until we have the full topological ordering.

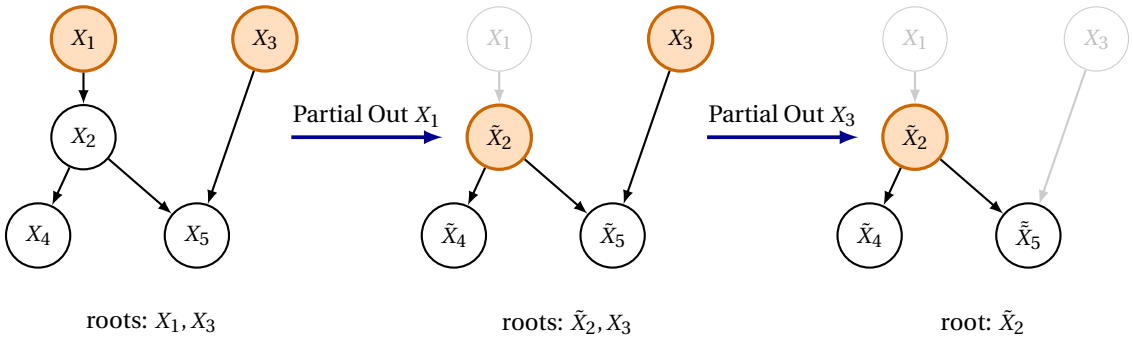


Figure 5.5: Three rounds of DirectLiNGAM on a five-node graph, peeling one root at a time. **Left:** X_1 and X_3 are both roots, but the algorithm commits to the single most independent one (here X_1) and regresses it out of the rest. A tilde marks a *current residual*, the part of a variable left after the chosen roots are partialled out, so X_2, X_4, X_5 become $\tilde{X}_2, \tilde{X}_4, \tilde{X}_5$, while X_3 is untouched because it is already independent of X_1. **Middle:** with X_1 gone, $\tilde{X}_2$ joins X_3 as a root; the test now selects X_3 and partials it out. **Right:** $\tilde{X}_2$ is the lone remaining root and is taken next; residualizing it exposes $\tilde{X}_4$ and $\tilde{X}_5$, which follow. The recovered order is $X_1 \prec X_3 \prec X_2 \prec X_4 \prec X_5$ — notably *not* the index order, since the algorithm re-checks which variable is most independent every round.

Once the ordering is fixed, recovering the actual edges is straightforward: regress each variable on the variables that precede it in the order. The regression coefficients are exactly the entries of $\hat{\mathbf{A}}$ — a coefficient near zero means no edge, while a nonzero one is a weighted arrow. This final pass both prunes the graph and estimates the edge weights. Putting the pieces together, DirectLiNGAM is:

1. **Identify the next root.** For each remaining candidate, regress every other remaining variable on it and form the residuals. A candidate that is independent of all of its residuals has no remaining parent, so it is a root. When more than one variable qualifies, pick the single most independent candidate as the next variable in the order.
2. **Append and residualize (partial out).** Add that one root to the end of the causal ordering, then replace every remaining variable with its residual after regressing the root out of it. That root's influence is now removed.
3. **Repeat.** With one variable's influence partialled out, some new variable has become exogenous; loop back to step 1 until every variable has been placed in the ordering.
4. **Estimate the edges.** Given the final topological order, regress each variable on its predecessors. The resulting coefficients fill in $\hat{\mathbf{A}}$ (near-zero entries are absent edges), recovering the full weighted DAG.

Figure 5.5 traces the first few rounds on a five-node graph.

5.7 Modern Causal Discovery

This chapter covered the basics of DAG-learning:

1. Constraint-based search using conditional independence.
2. Extensions to score-based learning.
3. Integrating multiple intervented environments to reduce MECs to a single DAG.
4. Using asymmetries from non-Gaussian noise to uncover causal direction without interventions.

The area is still pretty active, with three notable ideas that we did not cover: (1) extensions to the asymmetry idea, (2) differentiable causal discovery, and (3) causal discovery on ADMGs with unobserved confounding.

5.7.1 Beyond LiNGAM: Nonlinearity as an Alternative to Non-Gaussianity

LiNGAM identifies causal direction by requiring *non-Gaussian* noise; the linear-Gaussian case is famously the one place where observational data cannot distinguish $X \to Y$ from $Y \to X$. A parallel result, due to Hoyer et al. [2008], shows that there is a clean dual: if the structural mechanism is *nonlinear* with additive noise — $Y = f(X) + N_Y$ with f a smooth nonlinear function and $N_Y \perp\!\!\!\perp X$ — then the causal direction is identifiable even when N_Y is Gaussian. The reverse model $X = g(Y) + N_X$ with $N_X \perp\!\!\!\perp Y$ can fit the same joint distribution only when f and the noise satisfy a particular system of differential equations, and this fails for generic nonlinear f. These models are called **Additive Noise Models (ANMs)**.

Peters et al. [2014] extended this from pairs to full DAGs and introduced the **RESIT** algorithm (Regression with Subsequent Independence Test). RESIT peels off root variables one at a time by regressing each candidate against the remaining variables and testing whether the residuals are independent of the candidate — structurally analogous to DirectLiNGAM, but with nonlinear regressors (typically Gaussian processes or boosted trees) in place of linear ones. **Post-nonlinear models** [Zhang and Hyvärinen, 2009] push the family further to $Y = g(f(X) + N_Y)$, recovering both LiNGAM and ANMs as special cases.

The unifying picture across these identifiability results is that linear-plus-Gaussian is the *unique* symmetric corner in the space of bivariate models. Escape either axis with non-Gaussian noise, or nonlinear mechanisms, and the joint distribution of observational data carries enough asymmetry to read off the causal direction.

> **Main Idea 27**
>
> Causal direction is identifiable from observational data in every regime except the linear-Gaussian one. LiNGAM exploits non-Gaussianity; ANMs exploit nonlinearity; post-nonlinear models exploit both. The lone exception is linear systems with Gaussian noise, which is exactly the setting in which classical statistics has always lived.

5.7.2 Differentiable Causal Discovery

Zheng et al. [2018] introduced a third paradigm beyond constraint-based and score-based DAG learning: **differentiable causal discovery**, recasting the discrete combinatorial problem of "find the best DAG" as a smooth, gradient-based optimization. The trick in the " NOTEARS" algorithm was an algebraic acyclicity constraint, $h(W) = \text{tr}(e^{W \circ W}) - d = 0$, that vanishes if and only if the weighted adjacency matrix W defines a DAG. This made it possible, for the first time, to learn graph structure using deep learning machinery.

The area has grown rapidly. **DAGMA** [Bello et al., 2022] replaced the matrix-exponential trick with a log-determinant constraint that runs orders of magnitude faster. **GraN-DAG** [Lachapelle et al., 2020] extends the framework to nonlinear causal mechanisms via neural networks — bringing the ANM identifiability

theory above into direct contact with continuous optimization. **DiBS** [Lorch et al., 2021] provides a fully differentiable Bayesian version. The methodology interfaces naturally with interventional discovery too: **ENCO** [Lippe et al., 2022] uses the same continuous-optimization framework with interventional data instead of (or in addition to) observational data.

The most important caveat is a 2021 result by Reisach et al. [2021] showing that early differentiable methods exploited an artifact of synthetic benchmarks called *varsortability*: in many random DAG simulations, causally upstream variables happen to have lower marginal variance, and NOTEARS-style methods were partially learning the variance ordering rather than the true causal structure. This sparked a serious correction in the field, and current work focuses on making differentiable methods robust to variance scale and to evaluation on more realistic benchmarks.

> **Main Idea 28**
>
> Differentiable causal discovery recasts structure learning as smooth optimization with an algebraic acyclicity constraint, making causal discovery compatible with deep learning infrastructure. The field is still calibrating how much of its measured performance reflects causal recovery versus benchmark artifacts.

5.7.3 Causal Discovery with Unobserved Confounding

A serious limitation of every algorithm in this chapter so far is the assumption of **causal sufficiency**: that the observed variables include all common causes, so the underlying graph really is a DAG over what we see. In practice this is almost never true — there are nearly always lurking variables we did not measure. To handle this, we need to move from DAGs to **Acyclic Directed Mixed Graphs (ADMGs)**, which allow both directed edges $X \to Y$ (direct causation) and bidirected edges $X \leftrightarrow Y$ (shared unobserved cause).

The algorithmic question becomes: which patterns in the observational data force us to conclude that an unobserved confounder must exist? The key observation is that certain conditional dependencies between two variables persist no matter what subset of the remaining observed variables we condition on. Under causal sufficiency, every such dependence could in principle be blocked by some adjustment set; persistent unblockable dependence is therefore evidence of a latent common cause, drawn as a bidirected edge.

The most famous algorithm exploiting this is **Fast Causal Inference (FCI)** [Spirtes et al., 1995, 2000], which extends the PC algorithm to the ADMG setting. As with PC, FCI cannot identify the graph uniquely; it recovers an equivalence class, but now the class is over ADMGs rather than DAGs, and the equivalence class representative is called a **Partial Ancestral Graph (PAG)** [Zhang, 2008]. A PAG looks like a graph whose edges carry endpoint marks — arrowheads, tails, and circles for "unknown" — recording exactly what the data does and does not determine about the orientation at each end.

> **Main Idea 29**
>
> Dropping the causal sufficiency assumption forces us out of DAGs and into ADMGs, where bidirected edges encode unobserved confounding. FCI extends PC-style constraint-based discovery to this setting, recovering the underlying graph up to a partial ancestral graph rather than a CPDAG.

Chapter 6

Reflections and Future Directions

In this chapter, we first summarize the key takeaways, then close with a few modern applications and directions for future research.

6.1 Key Takeaways

The first key insight is that *causality is a missing data problem*, formalized through the potential outcomes framework. The missingness isn't random — it is biased by confounding features — and that bias can be corrected with the tools of modern machine learning: reweighting, regression, and matrix completion. This positions causality as a set of guiding principles for the modern ML toolkit.

The second lesson, sitting alongside the first, is that *context is critical*. To better understand data context and the flow of dependence, we studied *data-generating processes*, which we modeled graphically using DAGs. Once we have a DAG, we can use it to understand how regression and probability carve up signals and to predict what would happen under intervention.

Finally, we turned to causal discovery, which brought out the *fundamental differences between observational and interventional data*. Observational data can typically recover only a Markov equivalence class, and resolving the remaining causal directions requires interventions or specific asymmetries. Our graphical framework was especially useful for understanding the value of interventions, since they augment the data-generating process as new variables. The problem sets even showed us how to model other data contexts, such as sampling bias.

6.2 Other Topics in Causality

We now survey nine directions where these challenges are being met. Some are already mature applications; others are active research frontiers. None are fully solved, and all rely on the machinery this book has built.

6.2.1 Algorithmic Root Cause Analysis

One of the most powerful and immediate applications of causal discovery in modern industry is **Algorithmic Root Cause Analysis (RCA)**. In large-scale tech companies (like Amazon, Google, and Uber), software is broken down into hundreds of interconnected "microservices." When a user experiences a failure—such as a website crashing or a checkout page taking ten seconds to load—engineers must figure out exactly which microservice caused the anomaly so they can automatically page the correct engineering team to fix it.

Traditional statistical monitoring struggles with this because anomalies *propagate*. If a backend database slows down, every service that relies on it will also slow down. A naive monitoring system might see fifty

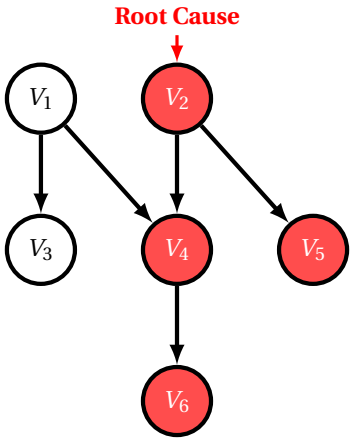


Figure 6.1: Anomaly propagation in an SCM. The root cause V_2 has anomalous exogenous noise; its descendants (V_4, V_5, V_6) become anomalous through propagation along the directed edges, even though their own exogenous noise terms are normal. Non-descendants of V_2 (V_1, V_3) are unaffected. RCA recovers the root cause as the topologically highest anomalous node — the only red node whose own N_i deviates from baseline.

different services simultaneously reporting high latency and page fifty different engineering teams. Figure 6.1 shows a small example: a single failure at V_2 cascades to make four nodes appear anomalous, even though only one of them is the true source. By combining causal discovery (to learn the microservice dependency graph) with Structural Causal Models, we can mathematically trace the anomaly back to its true source. **Counterfactual attribution** methods [Budhathoki et al., 2022] formalize this: given an anomalous outcome, they compute each node's Shapley-style contribution to the deviation by asking "what would the outcome have been if this node's mechanism had behaved normally?", yielding a ranked list of likely root causes rather than a single guess.

> **Main Idea 30**
>
> In an interconnected system, anomalies propagate along the directed edges of a causal graph. Algorithmic Root Cause Analysis isolates the true source of a failure by mapping observed data back to the unobserved exogenous noise variables (N_i) of the Structural Causal Model.

6.2.2 Algorithmic Fairness

Modern AI systems make consequential decisions in lending, hiring, criminal justice, and healthcare. The classical statistical fairness literature defines fairness through correlational criteria — *demographic parity*, *equalized odds*, *calibration* — that compare outcomes across protected groups. These criteria are tractable but pairwise incompatible: no classifier can simultaneously satisfy more than one of them in any realistic setting. More fundamentally, they conflate dependencies that arise through legitimate causal pathways with those that arise through discrimination.

Causal fairness reframes the question in terms of *counterfactuals*: would the model's decision change if the applicant's race or sex had been counterfactually different, holding everything else fixed? The **counterfactual fairness** criterion of Kusner et al. [2017] requires that the model's output be invariant under counterfactual interventions on protected attributes. Refinements based on **path-specific effects** [Chiappa, 2019, Nabi and Shpitser, 2018] go further, distinguishing direct discrimination (Race → Decision) from discrimination that flows through legitimate mediators (Race → Test Score → Decision), and they let policymakers decide which pathways are tolerable.

The technical core is the do-calculus and SWIG machinery from Chapter 3, applied not to estimate a treatment effect but to certify its absence along forbidden pathways. This is the formal core of **causal**

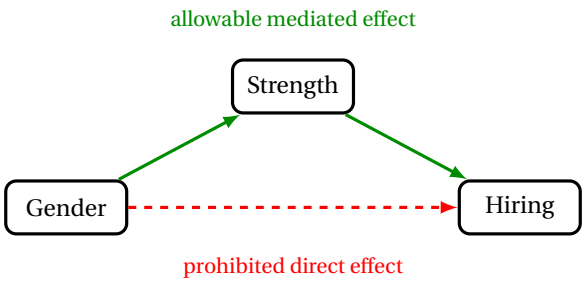


Figure 6.2: Path-specific fairness for a hypothetical hiring decision. The mediated pathway Gender → Strength → Hiring is treated as legitimate when the job genuinely requires physical capability: gender legitimately predicts strength, and strength legitimately predicts fit. The direct pathway Gender → Hiring is prohibited; it represents discrimination unmediated by any job-relevant variable. Path-specific counterfactual fairness asks whether the model's decision would change under a counterfactual intervention on Gender that operated only along the prohibited pathway.

mediation analysis [VanderWeele, 2015, Pearl, 2012], which has its own substantial literature on identification and estimation of direct and indirect effects.

> **Main Idea 31**
>
> Causal fairness asks not whether the model's outcomes are correlated with protected attributes, but whether the model would have made a different decision had the protected attribute been counterfactually different. The hard question becomes which pathways from the attribute to the decision are considered legitimate.

6.2.3 Counterfactual Analysis in AI Architectures

A generative model trained on faces can produce a new face. But can it answer a counterfactual question — "what would this face look like if the person had been older?" — while keeping every other attribute (identity, expression, lighting) fixed? This is the question of *counterfactual generation*, where modern generative AI meets the third rung of Pearl's ladder.

The technical move is to interpret a deep generative architecture as a Structural Causal Model: each latent variable plays the role of an exogenous noise ε_i, and the network's forward pass realizes the structural equations $V_i = f_i(\mathbf{PA}(V_i), \varepsilon_i)$. **Deep Structural Causal Models** [Pawlowski et al., 2020] make this explicit, layering causal graphs over the latent space of variational autoencoders. Counterfactual GANs and causal diffusion models make analogous moves for adversarial and score-based architectures. To compute a counterfactual, the model runs Pearl's three-step procedure: *abduct* the noise from the observed image, *act* by intervening on the variable of interest, and *predict* the new image by running the structural equations forward.

The active challenges here are about identifiability. A standard generative model has no notion of "holding identity fixed while changing age" — it has only a joint distribution over images. Forcing it to respect counterfactual operations requires either causally-structured latent spaces or post-hoc interventions on disentangled representations. Neither is fully solved, and the field is exploring how much causal structure can be imposed before generative quality degrades.

Counterfactual reasoning also drives much of modern *explainability*. To explain why a model produced a given output, we ask a counterfactual question about the model itself: which inputs, had they been different, would have changed the prediction? Gradient-based attribution makes this concrete — a feature with a large gradient in the output is one the prediction is locally most sensitive to, so large gradients are read as evidence that a feature matters. This is the abduct-act-predict loop in miniature: perturb an input and watch how the output responds.

There is a catch, however, and it is one our problem sets already exposed: the S-learner regularization bias, scaled up. When the S-learner fits a single model $f(X, A)$ for the response, a coefficient penalty can shrink A's contribution toward zero — the optimizer has no way to know that A is the variable whose effect we will eventually want to read off. The same shrinkage that hides a treatment's effect in estimation flattens a causal feature's gradient in attribution: a feature can be genuinely causal yet, once a regularizer has suppressed its coefficient, register only a small gradient and be ranked unimportant. Regularization can quietly kill the causal features, so even when the causal signal is present in the data, the model does not reliably emphasize it. The generative case is the same pathology at far greater scale: one network learns the entire joint distribution, and the training objective (reconstruction loss plus a KL or weight-decay penalty) gives no preferential status to the latent direction we eventually want to intervene on. Image content gets encoded redundantly across latents during training, so changing one of them later is no longer a clean intervention. The architectural fixes above are the deep-learning analog of moving from the S-learner to the T-learner: rather than asking one model to implicitly respect a causal structure the loss doesn't see, build the structure into the model.

Main Idea 32

A generative model can be reinterpreted as a Structural Causal Model whose latents are exogenous noise, unlocking counterfactual generation in principle; the same counterfactual lens underlies gradient-based explainability. The catch in both is the S-learner problem at scale — the training objective gives no preferential status to causally-meaningful directions, so a regularizer can suppress a causal feature's gradient and intervention on a single latent is rarely a clean operation.

6.2.4 Out of Distribution Generalization

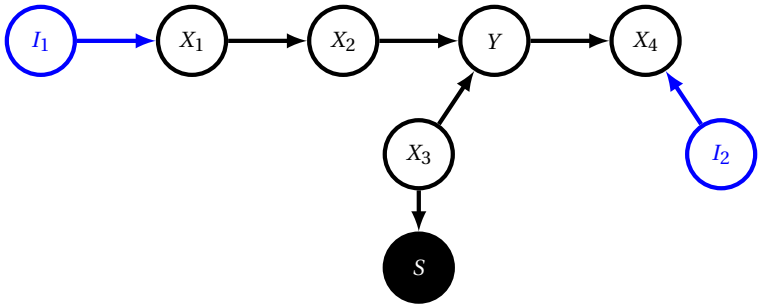


Figure 6.3: A selection diagram for OOD generalization across environments. The environment indicators I_1 and I_2 (blue) mark mechanisms that change between populations: I_1 acts on the upstream cause X_1, while I_2 acts on the downstream variable X_4. The chain $X_1 \to X_2 \to Y$ together with the extra cause X_3 generates the outcome Y, and X_3 also drives the selection node S (black, conditioned on by the sampling process). Crucially, X_4 is a *collider*: it is caused both by Y and by the environment through I_2 ($Y \to X_4 \leftarrow I_2$). A predictor built from the causes of Y (here X_2, X_3) satisfies $Y \perp \{S, I_1, I_2\} \mid \mathbf{X}$, so its predictive relationship is invariant across environments and transports to new settings. Adding the mere correlate X_4 as a predictor conditions on the collider, opening the path $Y \to X_4 \leftarrow I_2$ and injecting a spurious, environment-dependent association — which is exactly why causes are the more robust predictors.

Standard machine learning assumes that training and test data are drawn from the same distribution. In practice they rarely are: medical models trained on one hospital fail at another, language models trained on web text struggle with courtroom transcripts, vision models trained on daylight images fail at night. *Out-of-distribution (OOD) generalization* asks how to learn predictors that retain accuracy under distribution shift. The central thesis of the causal approach is that **features that are causal generalize; features that are merely correlational do not**.

The cleanest formalization is **Invariant Causal Prediction** (ICP) [Peters et al., 2016]. Given data from multiple environments (e.g., multiple hospitals), find the set of features whose conditional distribution of

the outcome is the same across environments. Under faithfulness, this set is exactly the causal parents of the outcome, and a predictor using only these features generalizes optimally to any new environment generated by interventions on non-target variables. **Invariant Risk Minimization** [Arjovsky et al., 2019] relaxes the assumptions and recasts the problem as a regularized loss, making it tractable for deep models. **Anchor regression** [Rothenhäusler et al., 2021] interpolates between OLS and ICP, trading worst-case generalization against in-distribution accuracy. **Causal Information Splitting** [Mazaheri et al., 2023] introduced some preliminary ideas on how to extend this to *proxies* of the true causal parents and children, particularly when those proxies may include both direct causal parents as well as effects that may break down.

A complementary thread is **transportability** [Pearl and Bareinboim, 2014], which formalizes when a causal effect estimated in one population can be transferred to another whose data-generating process differs in known ways. The transportability calculus extends do-calculus: it asks which combinations of observational and experimental data from a source population identify a target query in a different population. Figure 6.3 illustrates the basic criterion — the intervention nodes I_1 and I_2 mark where the source population's mechanisms differ from the target, a selection node S encodes the sampling bias in the source data, and an adjustment set $\mathbf{X}$ that d-separates Y from $\{I_1, I_2, S\}$ certifies that the conditional $\Pr(Y \mid \mathbf{X})$ transports between the two populations. A closely related problem is selection bias, which we examined in Problem Set 2 — explicitly modeling the selection mechanism let us identify effects that would otherwise be lost. Transportability is the formal underpinning of much modern OOD work: it answers *when* a learned model can be moved to a new environment with provable guarantees, rather than just empirically hoping it will.

The connection back to the book is the invariance theme that ran through Chapters 3 and 5. Just as a causal SCM produces an invariant data-generating mechanism across interventions, a causal predictor produces invariant performance across environments. The active research questions are about scaling these guarantees to high-dimensional, nonlinear settings without requiring an explicit graph — exactly where causality, representation learning, and OOD generalization converge.

> **Main Idea 33**
>
> Causal features generalize across environments because the causal mechanism is invariant under interventions on non-causal variables. OOD methods turn this principle into algorithms by searching for features whose predictive relationship with the outcome is stable across multiple training distributions.

6.2.5 Causal Feature Learning

Up to this point, every causal graph in this book has assumed that we already have the right variables. In practice, we don't... especially when working with raw images, audio, sensor streams, or text. We have pixel intensities, frequency bins, accelerometer readings, etc. *Causal Feature Learning* [Chalupka et al., 2017] asks how to construct the right macro-variables from microscopic data, in the sense that the resulting variables admit a clean causal description.

A motivating example is medical imaging. A radiologist's diagnosis depends on tumor presence, not on the raw pixel values that encode it. The relevant causal variable is at a coarser level of description than the data itself, and the same is true for almost every applied causal problem with high-dimensional inputs. Chalupka et al. [2017] formalize this as a coarsening problem: given a microscopic causal graph, find the coarsest macro-variable that preserves causal sufficiency for a query of interest. The resulting macro-variables are exactly those equivalence classes of microstates that the intervention cannot distinguish.

This intersects with all of modern representation learning, but with a different criterion. Standard deep representation learning rewards features that are *predictive*; causal feature learning rewards features that are *intervenable* — ones that admit a well-defined response to do-operations. The active research thread is how to discover such features without supervision, and how to integrate them with the discovery algorithms of Chapter 5 — leading naturally to the broader program of causal representation learning.

Main Idea 34

Real causal problems start with data, not with variables. Causal feature learning asks how to construct the right variables — ones that are coarse enough to be intervenable and fine enough to preserve causal sufficiency — from raw, high-dimensional observations.

6.2.6 Causal Representation Learning

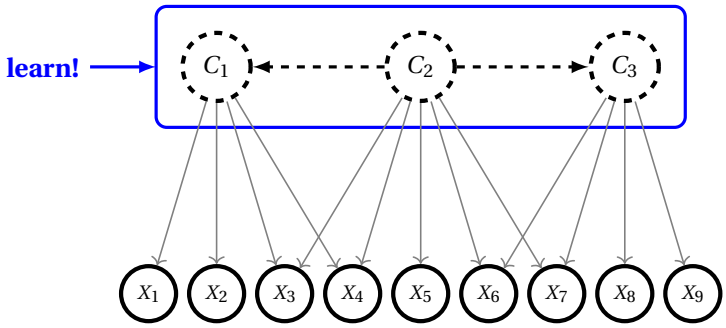


Figure 6.4: The setup of causal representation learning. Latent causal concepts C_1, C_2, C_3 (dashed, unobserved) form a causal structure among themselves and jointly generate the observed variables $X_1, \ldots, X_9$ (solid). Each observation is a mixture of multiple concepts. The goal of causal representation learning is to recover both the concepts and the causal relationships among them (the highlighted box) from observations alone, typically with the help of auxiliary information such as interventions, multi-environment data, or sparsity constraints.

Causal Representation Learning (CRL) is the broader research program of which causal feature learning is one branch. The goal is to use deep learning to recover not just the right variables but their causal structure, directly from observational and interventional data, without requiring labels for the latent factors. The unifying reference is Schölkopf et al. [2021], which charted the agenda and connected it to identifiable independent component analysis, disentanglement, and self-supervised learning.

The core technical question is *identifiability*: when is a learned representation guaranteed to recover the true causal factors? Classical disentanglement is impossible without inductive bias: Locatello et al. [2019] showed that infinitely many disentangled representations are consistent with any given observational dataset. Identifiability is recoverable when we have access to auxiliary information: **iVAE** [Khemakhem et al., 2020] achieves identifiability using auxiliary variables that modulate the prior over the latents; subsequent work extends this to settings with interventional data [Squires et al., 2023], multi-environment data [Zhang et al., 2024, Jin and Syrgkanis, 2024], and sparsity constraints on the causal structure.

The non-Gaussian identification results we proved for LiNGAM (Chapter 5.6) are exactly the kind of result CRL extends to deep nonlinear architectures. Where LiNGAM proves identifiability for linear structural equations with non-Gaussian noise, CRL extends the same logic — often using the same Darmois-Skitovich machinery — to neural-network-parameterized models with rich auxiliary information.

Main Idea 35

Causal Representation Learning extends the identifiability arguments of classical ICA and LiNGAM into the deep learning regime, using auxiliary information (interventions, environments, sparsity) to recover the causal factors underlying high-dimensional observations.

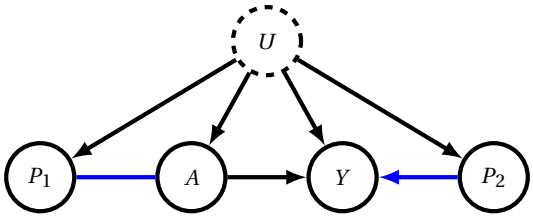


Figure 6.5: Proximal causal inference with negative controls. The treatment A and outcome Y are confounded by an *unobserved* common cause U (dashed). Rather than measuring U, we exploit two proxies, or negative controls: P_1 (a negative-control exposure) and P_2 (a negative-control outcome), each generated by U. Two such proxies of the hidden confounder, properly placed, carry enough information to identify the effect of A on Y without ever observing U. The edge linking P_1 with A is drawn undirected (CPDAG style), since either orientation is compatible with the framework. The blue edges are permitted but not required for identification.

6.2.7 Proximal Causal Inference and Unobserved Confounding

Throughout the estimation chapter, we addressed unobserved confounding with specific structural tools: instrumental variables when we had a source of natural randomization, and front-door adjustment when we had a fully mediated pathway. Neither tool applies universally, and a significant strand of modern research asks how to identify causal effects in the much more common setting where the confounder is unobserved and we have neither a clean instrument nor a clean mediator.

Proximal causal inference [Miao et al., 2018, Tchetgen et al., 2024] offers one answer through *negative controls*. A negative control outcome is a variable affected by the unobserved confounder but not by the treatment; a negative control exposure is a variable affected by the treatment and confounding, but not by the outcome of interest. Two such proxies of the unobserved confounding, properly placed, give us enough information to triangulate the unobserved confounder and recover the causal effect even though the confounder itself remains hidden (Figure 6.5). The proximal g-formula generalizes the back-door adjustment to this setting, replacing direct conditioning on the confounder with an integral over the proxies. The methodological challenge shifts from "find the right adjustment set" to "find good negative controls," which turns out to be more tractable in many applications. The proxies need only satisfy milder "bridging conditions", not full sufficiency.

A complementary line of research attacks unobserved confounding by exploiting *mixture structure* [Wang and Blei, 2019, Gordon et al., 2023].[1] When the unobserved confounder is the population from which a sample was drawn (one of several patient subgroups, sites, or batches in a meta-analysis) the joint distribution of the observables becomes a mixture of population-specific distributions. Mazaheri et al. [2025] introduced **Synthetic Potential Outcomes**, a framework that combined both the mixture and proximal causal inference perspectives to "synthetically sample" from counterfactual distributions using higher-order multi-linear moments of the observable data. Their key conceptual move was to group populations by their *causal response* to interventions, rather than by similarity of covariates (as in Gaussian mixtures) or correlations between covariates (as in latent factor models).

> **Main Idea 36**
>
> When unobserved confounding precludes the classical identification tools, two modern strategies recover causal effects without ever measuring the confounder. Proximal causal inference triangulates the confounder using negative-control variables; mixture-based identification exploits the mathematical structure of mixture models to separate populations by their causal response.

[1] Wang and Blei [2019] has received notable criticism by Ogburn et al. [2019] for some of its theoretical identifiability results, though they have since updated their assumptions to correct this.

6.2.8 Intervention Models and Discovery

Chapter 5.4 covered the verification problem: given a target DAG, how few interventions do we need to confirm it? The dual problem — *discovery from interventions* — is one of the most active areas of modern causal research, driven by the fact that experimental data is increasingly cheap in domains like single-cell biology, online experimentation, and robotics.

Classical results extend the foundations from Chapter 5.4: Hauser and Bühlmann [2012] characterized interventional Markov equivalence, and Eberhardt [2007] and Shanmugam et al. [2015] bounded the number of interventions required for full identification. Recent work has pushed in three complementary directions. First, **adaptive intervention design**, where each intervention is chosen based on the results of previous ones, can dramatically reduce the experimental budget — Choo et al. [2022] showed the gap is exactly $\Theta(\log n)$ between adaptive search and verification. Second, **soft and imperfect interventions** [Eberhardt, 2007, Yang et al., 2018] relax the assumption that interventions perfectly fix a variable, which is unrealistic in biology where the experimenter cannot fully knock out gene expression. Third is **intervention discovery from observational data** [Squires et al., 2020, Jaber et al., 2020, Brouillard et al., 2020], which is common in genomic perturbation screens. In this setting, the data already contains interventions, but the experimenter does not know which variable was perturbed in each sample, and the task is to recover both the graph and the intervention targets simultaneously.

The connection to the rest of this book is the intervention-as-information thread we developed in Chapter 5.4, pushed to settings where the assumptions get more realistic and the experimental budgets get tighter.

> **Main Idea 37**
>
> Interventional data is the resource that makes causal discovery tractable beyond the limits of observational identifiability. Active research focuses on minimizing experimental cost, handling imperfect interventions, and recovering structure when intervention targets themselves are unknown.

6.2.9 Causal Reinforcement Learning

Reinforcement learning has been spectacularly successful at problems where the agent learns from on-policy interaction with the environment. But many high-stakes decision problems (e.g., medical treatment, drug development, policy intervention) only allow off-policy data: we observe what doctors did and what happened, but we cannot randomly assign treatments. Another typical example is the vast corpus of driving data used to train Tesla's self-driving feature. Naïve application of standard RL to off-policy data systematically miscredits actions when there is unobserved confounding, leading to behaviors such as "lurching" when red lights turn green (before the cars ahead have started moving). *Causal Reinforcement Learning* brings the identification machinery of this book to sequential decision problems.

The simplest case is **causal bandits**: a contextual bandit problem where the arms are not just labels but interventions in a causal graph. If the graph is known, an agent can use the structure to share information across arms — pulling one arm tells you something about the expected reward of others, in a way standard bandit algorithms cannot exploit [Lattimore et al., 2016]. More general work [Bareinboim et al., 2015] addresses off-policy evaluation in MDPs with unobserved confounders, using do-calculus identification to determine when a behavioral policy's value can be recovered from observational logs alone.

The frontier is the broader program of **causal reinforcement learning** [Bareinboim et al., 2024], which gives the agent access to all three rungs of the hierarchy at once (pure observation, controlled experiment, and counterfactual reasoning) and lets it exploit whichever data are available. World models with explicit causal structure (e.g., robotics or language model agents) are an increasingly active research thread, and the question of when a learned policy generalizes to new environments is rapidly becoming a causal question rather than a statistical one.

Main Idea 38

Reinforcement learning generalizes naturally to the causal setting: actions are interventions, environments are SCMs, and off-policy evaluation under unobserved confounding becomes a do-calculus identification problem. This is one of the most active interfaces between causality and modern machine learning.

Closing Thoughts on AI

It is worth emphasizing that several of our key takeaways point to shortcomings of modern AI architectures.

First, we notice that Rung 1 (prediction) scales very well as data grows, while the other rungs tend to *worsen*. In particular, large and diverse datasets span many (potentially latent) populations, introducing confounding that obscures causal signals. Disentangling these signals requires conditioning on that confounding, which implicitly splits those large datasets back into small ones. If the confounding is latent, we end up facing very difficult latent-variable problems.

Related to this scaling is the inability of modern AI to preserve invariance. If you ask an AI image generator to modify your image, it produces a completely new image with no notion of what should stay fixed in the counterfactual (causally upstream concepts should remain constant, while downstream ones are free to change). To get this right, you need causal models.

Rung 3 is impossible to validate directly, since counterfactuals involve cross-world comparisons. Still, AI's failures at Rung 2 directly imply a deficiency at Rung 3 as well — and our S-learner experiment made this concrete, since the regularizer there suppressed exactly the causal feature that a counterfactual query would need to read off. This calls into question the ability of AI to assign blame in legal settings.

Finally, modern AI is trained on averages and largely ignores the rare cases from which human causal intuition is built. Outliers are natural experiments — distributional surprises that reveal the mechanisms that averages hide. This is exactly what LiNGAM, ICA, and intervention-based discovery exploit, and it is why human reasoning leans so heavily on asymmetries and tails. An AI that cannot reason from the tails cannot assess risk, recommend policy, or think like a scientist.

When photography matured in the 19th century, realistic painting lost much of its market value. What flourished instead was what photography could not do: impressionism, abstraction, the interpretive eye. As AI turns prediction and generation into cheap commodities, the three capabilities of synthesizing contexts, understanding invariance, and learning from anomalies should only grow in value. If photography taught painters to see differently, the rise of AI may teach us to value differently, placing a new premium on understanding *why*.

Bibliography

Ronald Aylmer Fisher. Statistical methods for research workers. 1934.

Donald B Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, 66(5):688, 1974.

Peter J Bickel, Eugene A Hammel, and J William O'Connell. Sex Bias in Graduate Admissions: Data from Berkeley: Measuring bias is harder than is usually assumed, and the evidence is sometimes contrary to expectation. *Science*, 187(4175):398–404, 1975.

Gerwyn Morris, Michael Berk, Ken Walder, Adrienne O'Neil, Michael Maes, and Basant K. Puri. The lipid paradox in neuroprogressive disorders: Causes and consequences. *Neuroscience & Biobehavioral Reviews*, 128:35–57, September 2021. ISSN 0149-7634. doi: 10.1016/j.neubiorev.2021.06.017. URL https://www.sciencedirect.com/science/article/pii/S0149763421002566.

A. G. Shaper, Goya Wannamethee, and Mary Walker. ALCOHOL AND MORTALITY IN BRITISH MEN: EXPLAINING THE U-SHAPED CURVE. *The Lancet*, 332(8623):1267–1273, December 1988. ISSN 0140-6736, 1474-547X. doi: 10.1016/S0140-6736(88)92890-5. URL https://www.thelancet.com/journals/lancet/article/PIIS0140-6736(88)92890-5/fulltext.

Charles D. Keeling, Robert B. Bacastow, Arnold E. Bainbridge, Carl A. Ekdahl Jr., Peter R. Guenther, Lee S. Waterman, and John F. S. Chin. Atmospheric carbon dioxide variations at Mauna Loa Observatory, Hawaii. *Tellus*, 28(6):538–551, 1976. ISSN 2153-3490. doi: 10.1111/j.2153-3490.1976.tb00701.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.2153-3490.1976.tb00701.x. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.2153-3490.1976.tb00701.x.

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

G. W. Imbens and D. B. Rubin. *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press, 2015.

Tyler J. VanderWeele. Concerning the consistency assumption in causal inference. *Epidemiology*, 20(6): 880–883, November 2009. ISSN 1531-5487. doi: 10.1097/EDE.0b013e3181bd5638.

Thomas S Richardson and James M Robins. Single world intervention graphs (SWIGs): A unification of the counterfactual and graphical approaches to causality. 2013.

I. Shpitser and J. Pearl. Identification of joint interventional distributions in recursive semi-Markovian causal models. In *Proc. 20th AAAI Conference on Artifical Intelligence*, pages 1219–1226, 2006. doi: 10.5555/1597348.1597382.

Yimin Huang and Marco Valtorta. Pearl's calculus of intervention is complete. In *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*, pages 217–224, 2006.

James M. Robins, Andrea Rotnitzky, and Lue Ping Zhao. Estimation of Regression Coefficients When Some Regressors Are Not Always Observed. *Journal of the American Statistical Association*, 89(427):846–866, 1994. ISSN 0162-1459. doi: 10.2307/2290910. URL https://www.jstor.org/stable/2290910.

Sören R. Künzel, Jasjeet S. Sekhon, Peter J. Bickel, and Bin Yu. Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the National Academy of Sciences of the United States of America*, 116(10):4156–4165, March 2019. ISSN 0027-8424. doi: 10.1073/pnas.1804597116. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC6410831/.

Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal*, 21(1):C1–C68, February 2018. ISSN 1368-4221. doi: 10.1111/ectj.12097. URL https://doi.org/10.1111/ectj.12097.

X Nie and S Wager. Quasi-oracle estimation of heterogeneous treatment effects. *Biometrika*, 108(2): 299–319, June 2021. ISSN 0006-3444. doi: 10.1093/biomet/asaa076. URL https://doi.org/10.1093/biomet/asaa076.

Alberto Abadie, Alexis Diamond, and Jens Hainmueller. Synthetic Control Methods for Comparative Case Studies: Estimating the Effect of California's Tobacco Control Program. *Journal of the American Statistical Association*, 105(490):493–505, June 2010. ISSN 0162-1459. doi: 10.1198/jasa.2009.ap08746. URL https://doi.org/10.1198/jasa.2009.ap08746. _eprint: https://doi.org/10.1198/jasa.2009.ap08746.

Joshua D. Angrist. Lifetime Earnings and the Vietnam Era Draft Lottery: Evidence from Social Security Administrative Records. *The American Economic Review*, 80(3):313–336, 1990. ISSN 0002-8282. URL https://www.jstor.org/stable/2006669.

Guido W. Imbens and Joshua D. Angrist. Identification and Estimation of Local Average Treatment Effects. *Econometrica*, 62(2):467–475, 1994. ISSN 0012-9682. doi: 10.2307/2951620. URL https://www.jstor.org/stable/2951620.

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

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

Azam Ikram, Sarthak Chakraborty, Subrata Mitra, Shiv Saini, Saurabh Bagchi, and Murat Kocaoglu. Root Cause Analysis of Failures in Microservices through Causal Discovery. *Advances in Neural Information Processing Systems*, 35:31158–31170, December 2022. URL https://papers.nips.cc/paper_files/paper/2022/hash/c9fcd02e6445c7dfbad6986abee53d0d-Abstract-Conference.html.

Thomas Verma and Judea Pearl. Causal networks: Semantics and expressiveness. In *Machine intelligence and pattern recognition*, volume 9, pages 69–76. Elsevier, 1990.

Steen A. Andersson, David Madigan, and Michael D. Perlman. A Characterization of Markov Equivalence Classes for Acyclic Digraphs. *The Annals of Statistics*, 25(2):505–541, 1997. ISSN 0090-5364. URL https://www.jstor.org/stable/2242556.

Frederick Eberhardt. Causation and intervention. *Unpublished doctoral dissertation, Carnegie Mellon University*, 93, 2007.

Alain Hauser and Peter Bühlmann. Characterization and greedy learning of interventional markov equivalence classes of directed acyclic graphs. *The Journal of Machine Learning Research*, 13(1):2409–2464, 2012.

Karren Yang, Abigail Katcoff, and Caroline Uhler. Characterizing and Learning Equivalence Classes of Causal DAGs under Interventions. In *Proceedings of the 35th International Conference on Machine Learning*, pages 5541–5550. PMLR, July 2018. URL https://proceedings.mlr.press/v80/yang18a.html.

Karthikeyan Shanmugam, Murat Kocaoglu, Alexandros G Dimakis, and Sriram Vishwanath. Learning causal graphs with small interventions. *Advances in Neural Information Processing Systems*, 28, 2015.

Davin Choo, Kirankumar Shiragur, and Arnab Bhattacharyya. Verification and search algorithms for causal dags. *Advances in Neural Information Processing Systems*, 35:12787–12799, 2022.

A. J. Bell and T. J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159, November 1995. ISSN 0899-7667. doi: 10.1162/neco.1995.7.6.1129.

Shohei Shimizu, Patrik O Hoyer, Aapo Hyvärinen, Antti Kerminen, and Michael Jordan. A linear non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7(10), 2006.

Shohei Shimizu, Takanori Inazumi, Yasuhiro Sogawa, Aapo Hyvarinen, Yoshinobu Kawahara, Takashi Washio, Patrik O Hoyer, Kenneth Bollen, and Patrik Hoyer. DirectLiNGAM: A direct method for learning a linear non-Gaussian structural equation model. *Journal of Machine Learning Research-JMLR*, 12(Apr): 1225–1248, 2011.

Patrik Hoyer, Dominik Janzing, Joris M Mooij, Jonas Peters, and Bernhard Schölkopf. Nonlinear causal discovery with additive noise models. *Advances in neural information processing systems*, 21, 2008.

Jonas Peters, Joris M Mooij, Dominik Janzing, and Bernhard Schölkopf. Causal discovery with continuous additive noise models. 2014.

Kun Zhang and Aapo Hyvärinen. On the identifiability of the post-nonlinear causal model. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 647–655, Arlington, Virginia, USA, June 2009. AUAI Press. ISBN 978-0-9749039-5-8. URL https://dl.acm.org/doi/10.5555/1795114.1795190.

Xun Zheng, Bryon Aragam, Pradeep K Ravikumar, and Eric Xing. DAGs with NO TEARS: Continuous Optimization for Structure Learning. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/e347c51419ffb23ca3fd5050202f9c3d-Paper.pdf.

Kevin Bello, Bryon Aragam, and Pradeep Ravikumar. DAGMA: learning DAGs via M-matrices and a log-determinant acyclicity characterization. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, pages 8226–8239, Red Hook, NY, USA, November 2022. Curran Associates Inc. ISBN 978-1-7138-7108-8.

Sébastien Lachapelle, Philippe Brouillard, Tristan Deleu, and Simon Lacoste-Julien. Gradient-based neural dag learning. *Proceedings of the Eighth International Conference on Learning Representations*, 2020.

Lars Lorch, Jonas Rothfuss, Bernhard Schölkopf, and Andreas Krause. DiBS: Differentiable Bayesian Structure Learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 24111–24123. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/ca6ab34959489659f8c3776aaf1f8efd-Paper.pdf.

Phillip Lippe, Taco Cohen, and Efstratios Gavves. Efficient Neural Causal Discovery without Acyclicity Constraints. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=eYciPrLuUhG.

Alexander Reisach, Christof Seiler, and Sebastian Weichwald. Beware of the Simulated DAG! Causal Discovery Benchmarks May Be Easy to Game. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 27772–27784. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/e987eff4a7c7b7e580d659feb6f60c1a-Paper.pdf.

Peter Spirtes, Christopher Meek, and Thomas Richardson. Causal inference in the presence of latent variables and selection bias. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, UAI'95, pages 499–506, San Francisco, CA, USA, August 1995. Morgan Kaufmann Publishers Inc. ISBN 978-1-55860-385-1. URL https://dl.acm.org/doi/10.5555/2074158.2074215.

Jiji Zhang. On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias. *Artificial Intelligence*, 172(16-17):1873–1896, 2008.

Kailash Budhathoki, Lenon Minorics, Patrick Bloebaum, and Dominik Janzing. Causal structure-based root cause analysis of outliers. In *Proceedings of the 39th International Conference on Machine Learning*, pages 2357–2369. PMLR, June 2022. URL https://proceedings.mlr.press/v162/budhathoki22a.html.

Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. Counterfactual Fairness. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/a486cd07e4ac3d270571622f4f316ec5-Paper.pdf.

Silvia Chiappa. Path-specific counterfactual fairness. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7801–7808, 2019.

Razieh Nabi and Ilya Shpitser. Fair Inference on Outcomes. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), April 2018. ISSN 2374-3468. doi: 10.1609/aaai.v32i1.11553. URL https://ojs.aaai.org/index.php/AAAI/article/view/11553.

Tyler VanderWeele. *Explanation in causal inference: methods for mediation and interaction*. Oxford University Press, 2015.

Judea Pearl. The causal mediation formula–a guide to the assessment of pathways and mechanisms. *Prevention Science: The Official Journal of the Society for Prevention Research*, 13(4):426–436, August 2012. ISSN 1573-6695. doi: 10.1007/s11121-011-0270-1.

Nick Pawlowski, Daniel Coelho de Castro, and Ben Glocker. Deep Structural Causal Models for Tractable Counterfactual Inference. In *Advances in Neural Information Processing Systems*, volume 33, pages 857–869. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/0987b8b338d6c90bbedd8631bc499221-Abstract.html.

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

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Dominik Rothenhäusler, Nicolai Meinshausen, Peter Bühlmann, and Jonas Peters. Anchor regression: Heterogeneous data meet causality. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 83(2):215–246, 2021.

Bijan Mazaheri, Atalanti Mastakouri, Dominik Janzing, and Michaela Hardt. Causal information splitting: Engineering proxy features for robustness to distribution shifts. In Robin J. Evans and Ilya Shpitser, editors, *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, volume 216 of *Proceedings of Machine Learning Research*, pages 1401–1411. PMLR, July 2023. URL https://proceedings.mlr.press/v216/mazaheri23a.html.

Judea Pearl and Elias Bareinboim. External Validity: From Do-Calculus to Transportability Across Populations. *Statistical Science*, 29(4):579–595, November 2014. ISSN 0883-4237, 2168-8745. doi: 10.1214/14-STS486. URL https://projecteuclid.org/journals/statistical-science/volume-29/issue-4/External-Validity-From-Do-Calculus-to-Transportability-Across-Populations/10.1214/14-STS486.full.

Krzysztof Chalupka, Frederick Eberhardt, and Pietro Perona. Causal feature learning: an overview. *Behaviormetrika*, 44(1):137–164, January 2017. ISSN 0385-7417, 1349-6964. doi: 10.1007/s41237-016-0008-2. URL http://link.springer.com/10.1007/s41237-016-0008-2.

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward Causal Representation Learning. *Proceedings of the IEEE*, 109(5): 612–634, May 2021. ISSN 1558-2256. doi: 10.1109/JPROC.2021.3058954. URL https://ieeexplore.ieee.org/document/9363924/.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations. In *Proceedings of the 36th International Conference on Machine Learning*, pages 4114–4124. PMLR, May 2019. URL https://proceedings.mlr.press/v97/locatello19a.html.

Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvarinen. Variational Autoencoders and Nonlinear ICA: A Unifying Framework. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, pages 2207–2217. PMLR, June 2020. URL https://proceedings.mlr.press/v108/khemakhem20a.html.

Chandler Squires, Anna Seigal, Salil Bhate, and Caroline Uhler. Linear causal disentanglement via interventions. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *ICML'23*, pages 32540–32560, Honolulu, Hawaii, USA, July 2023. JMLR.org.

Kun Zhang, Shaoan Xie, Ignavier Ng, and Yujia Zheng. Causal Representation Learning from Multiple Distributions: A General Setting. In *Proceedings of the 41st International Conference on Machine Learning*, pages 60057–60075. PMLR, July 2024. URL https://proceedings.mlr.press/v235/zhang24br.html.

Jikai Jin and Vasilis Syrgkanis. Learning Linear Causal Representations from General Environments: Identifiability and Intrinsic Ambiguity. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=dB99jjwx3h.

W. Miao, Z. Geng, and E. Tchetgen Tchetgen. Identifying causal effects with proxy variables of an unmeasured confounder. *Biometrika*, 105(4):987–993, 2018. doi: 10.1093/biomet/asy038.

Eric Tchetgen, Andrew Ying, Yifan Cui, Xu Shi, and Wang Miao. An Introduction to Proximal Causal Inference. *Statistical Science*, 39, August 2024. doi: 10.1214/23-STS911.

Y. Wang and D. M. Blei. The blessings of multiple causes. *Journal of the American Statistical Association*, 114(528):1574–1596, 2019. doi: 10.1080/01621459.2019.1686987.

Spencer L. Gordon, Bijan Mazaheri, Yuval Rabani, and Leonard Schulman. Causal Inference Despite Limited Global Confounding via Mixture Models. In *Proceedings of the Second Conference on Causal Learning and Reasoning*, pages 574–601. PMLR, August 2023. URL https://proceedings.mlr.press/v213/gordon23a.html.

Elizabeth L Ogburn, Ilya Shpitser, and Eric J Tchetgen Tchetgen. Comment on "blessings of multiple causes". *Journal of the American Statistical Association*, 114(528):1611–1615, 2019.

Bijan Mazaheri, Chandler Squires, and Caroline Uhler. Synthetic Potential Outcomes and Causal Mixture Identifiability. In *Proceedings of The 28th International Conference on Artificial Intelligence and Statistics*, pages 4276–4284. PMLR, April 2025. URL https://proceedings.mlr.press/v258/mazaheri25a.html.

Chandler Squires, Yuhao Wang, and Caroline Uhler. Permutation-Based Causal Structure Learning with Unknown Intervention Targets. In *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 1039–1048. PMLR, August 2020. URL https://proceedings.mlr.press/v124/squires20a.html.

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*, volume 33, pages 9551–9561. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/6cd9313ed34ef58bad3fdd504355e72c-Abstract.html.

Philippe Brouillard, Sébastien Lachapelle, Alexandre Lacoste, Simon Lacoste-Julien, and Alexandre Drouin. Differentiable Causal Discovery from Interventional Data. In *Advances in Neural Information Processing Systems*, volume 33, pages 21865–21877. Curran Associates, Inc., 2020. URL https://papers.nips.cc/paper/2020/hash/f8b7aa3a0d349d9562b424160ad18612-Abstract.html.

Finnian Lattimore, Tor Lattimore, and Mark D Reid. Causal Bandits: Learning Good Interventions via Causal Inference. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/b4288d9c0ec0a1841b3b3728321e7088-Paper.pdf.

Elias Bareinboim, Andrew Forney, and Judea Pearl. Bandits with Unobserved Confounders: A Causal Approach. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/795c7a7a5ec6b460ec00c5841019b9e9-Paper.pdf.

Elias Bareinboim, Junzhe Zhang, Sanghack Lee, and Snu Ac Kr. An Introduction to Causal Reinforcement Learning. 2024.

Appendix A

Mathematical Preliminaries

A.1 Probability Review

One problem with Hume's regularity theory is that it cannot capture probabilistic causal effects. This would prohibit us from ever saying "gambling made me broke" or "a lottery ticket is a bribe" because gambling does not always cause someone to lose money, and lottery tickets are not always worth money. To get around this, we will use probability.

A.1.1 What is Probability?

A probability measure $\Pr(\cdot)$ is a function that takes events like "you will get an A in this class" and returns a number between 0 and 1. You can interpret this as model uncertainty or as some inherent randomness.

To formalize this mathematically, we need a way to denote events and their combinations. We will use lowercase letters like a and b to represent specific propositions or events. We use the logical operator $\vee$ to denote "OR" (the union of events), meaning $a \vee b$ represents the event that a happens, b happens, or both.

There are three axioms of probability:

1. Positivity: $\Pr(y) \geqslant 0$.
2. Unit Measure: Sure propositions have $\Pr(y) = 1$.
3. Additivity: $\Pr(a \vee b) = \Pr(a) + \Pr(b)$ if a and b are disjoint events.

A.1.2 Random Variables

A random variable is kind of a misnomer. It's actually a *function* which maps events (which have a probability measure) to a number (discrete or continuous). This is because numbers are easy to work with. We will use the uppercase Roman alphabet for random variables and the lowercase Roman alphabet to abbreviate events. Here is an example:

- X is a "Bernoulli" random variable which maps the events of a coin flip to 0 or 1.
- $X = 1$ is an event, which has a probability.
- $X = x$ is an event which has not been specified. We often abbreviate this event as just x, allowing us to write shorthand like $\Pr(x) = \frac{1}{2}$ instead of $\Pr(X = x) = \frac{1}{2}$.

We will often use binary numbers for true/false events. For example $X = 1$ means the event happened and $X = 0$ means it didn't.

A.1.3 Conditional and Joint Probability

Notationally, we use a comma (,) to represent the logical "AND" for joint probabilities. Therefore, a *joint* probability such as $\Pr(y, x)$ gives the probability of both events happening together.

We use the pipe symbol $\mid$ to represent "given that" for conditional probabilities. Conditional probabilities allow us to take into account partial information. A joint probability can be decomposed into a conditional probability using the chain rule:

$$\Pr(y, x) = \Pr(x) \Pr(y \mid x). \tag{A.1}$$

For example, if $Y = 1$ represents getting an A in this class and X represents doing the homeworks, then the baseline $\Pr(Y = 1)$ may be .7, while $\Pr(Y = 1 \mid X = 1) = .9$ and $\Pr(Y = 1 \mid X = 0) = 0$.

The *marginal* probability $\Pr(Y = 1)$ encodes some of the randomness in X while the *conditional* decodes the randomness due to uncertainty in whether or not you do the homework.

We can derive Bayes' Rule using the chain rule:

$$\begin{aligned} \Pr(y, x) = \Pr(x) \Pr(y \mid x) &= \Pr(y) \Pr(x \mid y) \\ \Pr(y \mid x) &= \frac{\Pr(x \mid y) \Pr(y)}{\Pr(x)} \end{aligned} \tag{A.2}$$

A.1.4 Law of Total Probability

We will use calligraphic fonts to denote the full set (or sample space) of all possible values a random variable can take, and the capital Sigma symbol $\sum$ to denote a sum over those values.

Suppose $\mathscr{Y} = \{y_1, y_2, \ldots, y_d\}$ is a set of mutually exclusive and exhaustive possible values for Y. Axioms 1 and 2 tell us that:

$$\sum_{y \in \mathscr{Y}} \Pr(y) = \Pr(y_1) + \Pr(y_2) + \ldots + \Pr(y_d) = 1. \tag{A.3}$$

This also holds for conditional probabilities:

$$\sum_{y \in \mathscr{Y}} \Pr(y \mid x) = \Pr(y_1 \mid x) + \Pr(y_2 \mid x) + \ldots + \Pr(y_d \mid x) = 1. \tag{A.4}$$

This means we can compute marginal probabilities from joint probabilities by summing over (or "marginalizing out") the other variable:

$$\sum_{y \in \mathscr{Y}} \Pr(x, y) = \Pr(x) \sum_{y \in \mathscr{Y}} \Pr(y \mid x) = \Pr(x). \tag{A.5}$$

Again, the same thing holds for conditional probabilities:

$$\Pr(x \mid z) = \sum_{y \in \mathscr{Y}} \Pr(x, y \mid z). \tag{A.6}$$

A.1.5 Independence

We say that X, Y are marginally independent if knowing something about one variable does not reduce or change the uncertainty about the other variable:

$$\Pr(y \mid x) = \Pr(y) \tag{A.7}$$

or equivalently

$$\Pr(x, y) = \Pr(x) \Pr(y). \tag{A.8}$$

(Note: In causal diagrams and structural equations, you will often see this structural independence denoted with the perpendicular symbol as $X \perp\!\!\!\perp Y$.)

We will also use the notion of conditional independence. We say that two variables X and Y are conditionally independent given a third variable Z if, once we already know the value of Z, learning the value of X provides no additional information about Y:

$$\Pr(y \mid x, z) = \Pr(y \mid z) \tag{A.9}$$

or equivalently

$$\Pr(x, y \mid z) = \Pr(x \mid z) \Pr(y \mid z). \tag{A.10}$$

(We denote conditional independence as $X \perp\!\!\!\perp Y \mid Z$.)

A.1.6 Expectation

We denote the expected value (or mean) of a random variable with a stylized capital $\mathbb{E}[\cdot]$, and we use the colon-equals symbol := to mean "is defined as".

We can find the expected value of a random variable X taking values in $\mathscr{X}$ using:

$$\mathbb{E}[X] := \sum_{x \in \mathscr{X}} \Pr(x) \cdot x. \tag{A.11}$$

A fundamentally important property of expectation is that it is a **linear operator**. This means the expectation of a sum is exactly equal to the sum of the expectations, and constants can be pulled outside of the operator:

$$\mathbb{E}[aX + bY] = a\mathbb{E}[X] + b\mathbb{E}[Y] \tag{A.12}$$

This property (the Linearity of Expectation) holds *regardless* of whether X and Y are independent or dependent!

Another incredibly powerful tool is the **Law of Total Expectation** (sometimes called the Tower Rule). It states that the expected value of X can be found by taking the weighted average of the *conditional* expected values of X given some other variable Y:

$$\mathbb{E}[X] = \mathbb{E}[\mathbb{E}[X \mid Y]] = \sum_{y \in \mathscr{Y}} \Pr(y) \cdot \mathbb{E}[X \mid Y = y] \tag{A.13}$$

This allows us to break complex, population-level expectations into smaller, subgroup-level conditional expectations. As we will see, marginalizing over subgroups like this is a foundational operation in observational causal inference.

A.1.7 Variance and Covariance

While expectation tells us the center of a distribution, **variance** tells us how spread out the data is around that center. It is defined as the expected squared deviation from the mean:

$$\mathrm{Var}(X) := \mathbb{E}[(X - \mathbb{E}[X])^2] = \mathbb{E}[X^2] - (\mathbb{E}[X])^2 \tag{A.14}$$

Unlike expectation, variance is *not* linear. If you multiply a variable by a constant a, the variance scales quadratically: $\mathrm{Var}(aX) = a^2\mathrm{Var}(X)$.

Covariance measures the *linear* relationship between two variables—how much they vary together.

$$\mathrm{Cov}(X, Y) := \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])] = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y] \tag{A.15}$$

If two variables are independent ($X \perp\!\!\!\perp Y$), their covariance is exactly zero (we say they are uncorrelated). However, the reverse is generally *not* true: a covariance of zero only implies there is no linear relationship, but the variables might still be highly dependent in a non-linear way.

A crucial algebraic property of covariance is that it is a **bilinear** operator. This means it behaves linearly in *both* of its arguments. You can expand it exactly the same way you distribute multiplication over addition in standard algebra (like using the FOIL method). For any constants a, b, c, d and random variables W, X, Y, Z:

$$\mathrm{Cov}(aW + bX, cY + dZ) = ac\mathrm{Cov}(W, Y) + ad\mathrm{Cov}(W, Z) + bc\mathrm{Cov}(X, Y) + bd\mathrm{Cov}(X, Z) \tag{A.16}$$

Because the covariance of a variable with itself is simply its variance ($\mathrm{Cov}(X,X) = \mathrm{Var}(X)$), this bilinearity gives us a straightforward way to calculate the variance of a sum of variables:

$$\begin{aligned}\mathrm{Var}(X+Y) &= \mathrm{Cov}(X+Y, X+Y)\\ &= \mathrm{Cov}(X,X) + \mathrm{Cov}(X,Y) + \mathrm{Cov}(Y,X) + \mathrm{Cov}(Y,Y)\\ &= \mathrm{Var}(X) + \mathrm{Var}(Y) + 2\mathrm{Cov}(X,Y)\end{aligned}$$

A.1.8 Entropy and Mutual Information

Variance and covariance summarize spread and linear association. Information theory provides complementary quantities that measure uncertainty and any kind of statistical dependence, including non-linear ones. These show up throughout the book — in ICA, in the GES score, and in causal representation learning — so we collect the definitions here.

The **entropy** of a discrete random variable X taking values in $\mathscr{X}$ is

$$H(X) := -\sum_{x\in\mathscr{X}} \Pr(x)\log\Pr(x) = \mathbb{E}[-\log\Pr(X)]. \tag{A.17}$$

The base of the logarithm sets the units (bits for base 2, nats for the natural log). Entropy measures uncertainty: it is zero when X is deterministic and largest when X is uniformly distributed over $\mathscr{X}$. For a continuous variable with density $p(x)$, the analogous quantity is the **differential entropy** $h(X) = -\int p(x)\log p(x)\,dx$.

The **joint entropy** $H(X,Y)$ is defined the same way using the joint probability $\Pr(x,y)$, and the **conditional entropy** $H(Y \mid X)$ measures the remaining uncertainty in Y once X is known:

$$H(Y \mid X) := H(X,Y) - H(X) = \sum_{x\in\mathscr{X}} \Pr(x)\cdot H(Y \mid X = x). \tag{A.18}$$

The **mutual information** between X and Y is the reduction in uncertainty about one variable from learning the other:

$$I(X;Y) := H(X) - H(X \mid Y) = H(X) + H(Y) - H(X,Y). \tag{A.19}$$

Equivalently, it is the KL divergence between the joint distribution and the product of marginals:

$$I(X;Y) = \sum_{x,y} \Pr(x,y)\log\frac{\Pr(x,y)}{\Pr(x)\Pr(y)}. \tag{A.20}$$

Mutual information satisfies several properties that make it the natural measure of dependence:

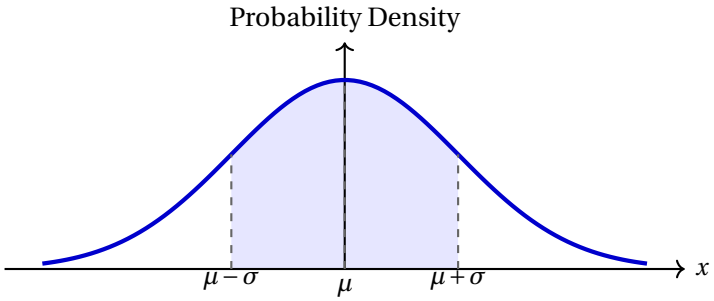
- $I(X;Y) \geqslant 0$, with equality if and only if $X \perp\!\!\!\perp Y$.
- It is symmetric: $I(X;Y) = I(Y;X)$.
- It detects *any* kind of statistical dependence, not just linear association.

The conditional mutual information $I(X;Y \mid Z)$ is defined analogously and is zero exactly when $X \perp\!\!\!\perp Y \mid Z$.

> **Main Idea 39**
>
> Covariance is zero whenever there is no *linear* dependence between two variables; mutual information is zero only when there is no dependence *at all*. This is why mutual information appears wherever we need to test for full independence rather than just uncorrelatedness — in ICA, in the GES score, and in non-linear causal discovery.

A final fact worth noting, since it bridges directly to the next subsection: of all distributions on $\mathbb{R}$ with a given variance, the Gaussian has the largest differential entropy. The amount by which a distribution falls short of this maximum is called its **negentropy**, and is used as a measure of non-Gaussianity in algorithms like FastICA.

A.1.9 Gaussian Distributions and the Central Limit Theorem



The **Gaussian** (or Normal) distribution is the most ubiquitous probability distribution in statistics, famously shaped like a bell curve, with a density function that is proportional to an exponential of a negative quadratic, $p(x) \propto e^{-x^2}$. We denote that a variable follows a Gaussian distribution with mean μ and variance σ^2 as $X \sim \mathcal{N}(\mu, \sigma^2)$.

Gaussians have two incredibly unique mathematical properties that we exploit heavily in causal discovery algorithms (like ICA and LiNGAMs):

1. **Linear combinations remain Gaussian:** If X and Y are Gaussian, then any linear combination $aX + bY$ is also perfectly Gaussian.
2. **Uncorrelated implies Independent:** Earlier, we stated that zero covariance does not guarantee independence. The Gaussian distribution is the major exception! If two variables are jointly Gaussian and their covariance is zero, they are strictly independent. This makes them mathematically easy to work with, but also causes "rotational ambiguity" in causal discovery, which is why algorithms like LiNGAM explicitly require *non-Gaussian* data.

Why are Gaussians so common in nature? This is explained by the **Central Limit Theorem (CLT)**. The CLT states that if you take the sum of many independent random variables, their normalized sum will converge toward a Gaussian distribution, *regardless of what the original underlying distributions were.*

This is the exact intuition driving Independent Component Analysis (ICA): if you have two independent sound signals (like voices), any mixture of those signals will mathematically look "more Gaussian" than the original distinct voices. By actively forcing the signals to be as non-Gaussian as possible, we can successfully un-mix them!

A.2 Linear Algebra Preliminaries

Throughout this course, especially when discussing causal discovery algorithms (like ICA and LiNGAMs) and missing data methods (like synthetic controls), we rely on foundational concepts from linear algebra. This section provides a brief review of these concepts, focusing heavily on the intuition behind matrix rank and low-rank approximations.

A.2.1 Vectors and Matrices

A vector $\mathbf{v} \in \mathbb{R}^n$ is an ordered list of n numbers, which we typically represent as a column:

$$\mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix} \tag{A.21}$$

A matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$ is a rectangular array of numbers with n rows and m columns. We can think of a matrix as a collection of m column vectors, or n row vectors.

The transpose of a matrix, denoted $\mathbf{A}^\top$, flips the matrix over its diagonal, turning rows into columns and vice versa.

A.2.2 Linear Independence and Span

A set of vectors $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k\}$ is **linearly independent** if no vector in the set can be written as a linear combination of the others. Mathematically, this means the only solution to the equation:

$$c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + \ldots + c_k\mathbf{v}_k = \mathbf{0} \tag{A.22}$$

is when all the scalar coefficients $c_i = 0$. If there is a non-zero solution, the vectors are **linearly dependent**, meaning at least one vector is redundant and simply lies in the space already spanned by the others.

The **span** of a set of vectors is the set of all possible linear combinations we can create from them.

A.2.3 Matrix Rank

The **rank** of a matrix $\mathbf{A}$ is a measure of the "information content" or "dimensionality" of the matrix.

- The **column rank** is the maximum number of linearly independent columns in $\mathbf{A}$.
- The **row rank** is the maximum number of linearly independent rows in $\mathbf{A}$.

A fundamental theorem of linear algebra states that the column rank and row rank are always exactly equal. We simply call this the rank of the matrix, denoted $\text{rank}(\mathbf{A})$.

For any $n \times m$ matrix, the rank cannot exceed the smaller of its two dimensions: $\text{rank}(\mathbf{A}) \leqslant \min(n, m)$. If a matrix achieves this maximum possible rank, we say it has **full rank**. This means every column (or row) provides completely new, non-redundant information.

Low-Rank Matrices: If $\text{rank}(\mathbf{A}) = r$ where $r \ll \min(n, m)$, we call $\mathbf{A}$ a **low-rank matrix**. This is a remarkably powerful concept in data science and causal inference. If a large 1000×1000 dataset has a rank of only $r = 3$, it implies that all 1,000,000 data points are actually generated by just 3 underlying hidden factors. The vast majority of the data is mathematically redundant.

A.2.4 Singular Value Decomposition (SVD)

Every matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$, regardless of its shape or rank, can be factored into three matrices:

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top \tag{A.23}$$

Where:

- $\mathbf{U}$ is an $n \times n$ orthogonal matrix (its columns are the left singular vectors).
- $\mathbf{V}$ is an $m \times m$ orthogonal matrix (its columns are the right singular vectors).
- Σ is an $n \times m$ diagonal matrix containing the **singular values** ($\sigma_1 \geqslant \sigma_2 \geqslant \ldots \geqslant \sigma_k \geqslant 0$) in descending order.

The rank of a matrix is exactly equal to the number of non-zero singular values in Σ.

Truncated SVD: If a matrix is full rank but is highly driven by just a few underlying factors, we can approximate it by taking only the top k largest singular values and setting the rest to zero. This creates a low-rank approximation of our original dataset, effectively filtering out noise and leaving only the strongest structural signals.

A.2.5 Matrix Completion

In causal inference, particularly when estimating treatment effects over time (like in Synthetic Controls), we frequently frame the problem of unobserved potential outcomes as a **missing data** problem.

Suppose we have a matrix $\mathbf{M}$ where the rows represent different units (e.g., states, patients) and the columns represent time steps. The entries are the observed outcomes. However, some entries are missing (e.g., the counterfactual potential outcomes of the treated units had they not received treatment).

If we assume the complete matrix $\mathbf{M}$ is **full rank**, it is mathematically impossible to guess the missing entries. Every row and column is completely independent, so observing the first 9 entries of a row tells us absolutely nothing about the 10th.

However, if we assume the complete matrix $\mathbf{M}$ has a **low rank** (meaning the outcomes of all states are driven by a small number of shared latent factors, like the national economy or seasonal trends), the matrix is highly constrained.

Matrix completion algorithms search for a fully observed, low-rank matrix $\hat{\mathbf{M}}$ that perfectly matches our observed data points. Because the low-rank assumption forces the rows and columns to be linear combinations of one another, the algorithm can elegantly "fill in" the missing counterfactuals by leveraging the structure of the observed data. This is typically done by solving an optimization problem that minimizes the rank (often proxied by the nuclear norm) of the matrix while heavily penalizing errors on the observed entries.

A.3 Graph Theory Basics

Causal inference relies heavily on the language of graph theory to represent data-generating processes. A graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ consists of a set of vertices (or nodes) $\mathbf{V}$ representing random variables, and a set of edges $\mathbf{E}$ representing the relationships between them.

A.3.1 Basic Terminology

- **Directed vs. Undirected Edges:** An edge can be undirected ($X - Y$) to represent a symmetric relationship (like an unoriented correlation), or directed ($X \to Y$) to represent an asymmetric, causal relationship.
- **Paths:** A path is a sequence of distinct adjacent nodes. A *directed path* is a path where all arrows point in the same direction (e.g., $X \to Y \to Z$).
- **Cycles:** A cycle is a directed path that starts and ends at the exact same node (e.g., $X \to Y \to Z \to X$). This would imply that a variable causes itself, which violates the arrow of time.
- **DAG:** A **Directed Acyclic Graph** is a graph where all edges are directed and there are absolutely no cycles. This is the foundational structure used to model causal relationships.

A.3.2 Kinship Terminology

Because causal graphs are directed, we borrow kinship terms from family trees to describe the relationships between nodes:

- **Parents:** $\mathbf{PA}(X)$ is the set of all nodes with a directed edge pointing directly into X.
- **Children:** $\mathbf{CH}(X)$ is the set of all nodes that X points directly to.
- **Ancestors:** $\mathbf{AN}(X)$ is the set of all nodes that have a directed path leading to X (parents, grandparents, etc.).
- **Descendants:** $\mathbf{DE}(X)$ is the set of all nodes that can be reached by a directed path originating from X (children, grandchildren, etc.).

A.3.3 Topological Sorting

A **topological sort** (or topological ordering) of a DAG is a linear ordering of its vertices such that for every directed edge $X \to Y$, node X comes before node Y in the ordering.

Because a DAG has no cycles, it is mathematically guaranteed to have at least one valid topological ordering. This concept is crucial for algorithms like LiNGAM and GES, which attempt to discover the causal

structure by finding a sequence where ancestors are strictly evaluated before their descendants. In a topological sort, the matrix representation of a DAG (its adjacency matrix) can be written as a strictly lower-triangular matrix.

A.3.4 Topological Sorting Algorithm

A **topological sort** of a Directed Acyclic Graph (DAG) is a linear ordering of its vertices such that for every directed edge $U \to V$, node U comes before node V in the ordering. This is fundamentally a way to flatten a graph into a sequence that respects the hierarchy of parents and children.

To compute this ordering programmatically, we typically use **Kahn's Algorithm**, which iteratively strips away the root nodes of the graph. The algorithm relies on the concept of an **in-degree**, which is simply the number of incoming edges a node has. (A node with an in-degree of 0 has no parents and is therefore an exogenous root).

The algorithm proceeds as follows:

1. **Initialize:** Calculate the in-degree for every node in the graph.
2. **Queue Roots:** Find all nodes with an in-degree of 0 and place them into a queue.
3. **Process and Remove:** While the queue is not empty:
    - Remove a node from the front of the queue and append it to your final sorted list.
    - For every child of this removed node, conceptually "delete" the edge connecting them. Mathematically, this means you subtract 1 from the in-degree of each child.
4. **Queue New Roots:** If any child's in-degree drops to 0 as a result of step 3, it means all of its parents have already been processed. Add this child to the queue.
5. **Repeat** steps 3 and 4 until the queue is completely empty.

If the algorithm finishes and the sorted list contains every node in the graph, you have successfully found a topological ordering.

> **Main Idea 40**
>
> Kahn's Algorithm also acts as a built-in cycle detector. If the algorithm finishes but there are still nodes left over that never made it into the queue, it mathematically guarantees that the graph contains a cycle! A cycle creates a situation where no remaining node ever reaches an in-degree of 0, causing the algorithm to stall.

A.4 Ordinary Least Squares (OLS) Regression

Throughout this course, we frequently use linear regression to model relationships, estimate causal effects, and perform causal discovery (e.g., DirectLiNGAM). Ordinary Least Squares (OLS) is a method for estimating the parameters of a linear regression model.

A.4.1 The Univariate Case

Suppose we want to model the relationship between a single predictor X and an outcome Y as a linear function:

$$Y = \beta X + \epsilon \tag{A.24}$$

Here, β is the coefficient representing the slope, and ϵ is the error term (or residual). OLS finds the estimate $\hat{\beta}$ that minimizes the sum of squared errors between the predicted values ($\hat{Y} = \hat{\beta} X$) and the actual values Y.

By taking the derivative of the squared error with respect to β and setting it to zero, we arrive at the closed-form solution for the OLS estimator:

$$\hat{\beta} = \frac{\text{Cov}(X, Y)}{\text{Var}(X)} \tag{A.25}$$

This is the exact formula used to isolate linear causal effects in the LiNGAM chapter.

A.4.2 The Multivariate Matrix Formulation

If we have multiple predictors, we can write the model in matrix notation: $\mathbf{Y} = \mathbf{X}\beta + \epsilon$. Here, $\mathbf{X}$ is a design matrix where each column is a variable, and β is a vector of coefficients. The OLS estimator generalizes to:

$$\hat{\beta} = (\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top \mathbf{Y} \tag{A.26}$$

Again, you can get this by taking the derivative of the squared errors (which is quadratic), setting it equal to zero, and solving.

A.4.3 Residuals and Geometric Orthogonality

The residual is the difference between the observed value and the value predicted by our model: $\hat{\epsilon} = Y - \hat{Y}$.

A crucial mathematical property of OLS is its **geometric interpretation**. To understand this, we must distinguish between the data samples and the model features. If we have n data points, the target variable can be represented as a single n-dimensional vector $\mathbf{Y}$, where each entry is an observed data value. Similarly, each *feature* (or predictor) in our model is represented by a column in the design matrix $\mathbf{X}$—an n-dimensional vector containing the data values for that specific feature.

These feature columns span a linear subspace in $\mathbb{R}^n$ known as the *column space* of $\mathbf{X}$. OLS effectively uses the projection matrix $\mathbf{X}(\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top$ to project the target vector $\mathbf{Y}$ directly onto this feature subspace, yielding our predictions $\hat{\mathbf{Y}}$. Because the shortest distance between a point (the vector $\mathbf{Y}$) and a plane (the column space of $\mathbf{X}$) is a perpendicular line, the error vector (the residuals, $\hat{\epsilon}$) is mathematically forced to be *orthogonal* to every feature vector in the predictor space.

In statistical terms, this geometric orthogonality in $\mathbb{R}^n$ means **OLS residuals are always completely uncorrelated with the input predictors**: $\text{Cov}(X, \hat{\epsilon}) = 0$.

> **Main Idea 41**
>
> Uncorrelatedness is not the same as independence. OLS guarantees that $\text{Cov}(X, \hat{\epsilon}) = 0$ by definition. However, in causal discovery (like DirectLiNGAM), we look for a stricter condition: whether the residuals are statistically *independent* of the predictors ($X \perp\!\!\!\perp \hat{\epsilon}$). If the noise is non-Gaussian, independence only holds in the true causal direction.