BAYESIAN STRUCTURAL CAUSAL INFERENCE WITH PROBABILISTIC PROGRAMMING

A Dissertation Presented

by

SAMUEL A. WITTY

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

September 2023

Robert and Donna Manning College of
Information and Computer Sciences

BAYESIAN STRUCTURAL CAUSAL INFERENCE WITH PROBABILISTIC PROGRAMMING

A Dissertation Presented

by

SAMUEL A. WITTY

Approved as to style and content by:

David Jensen, Chair

Daniel Sheldon, Member

Justin Domke, Member

Vikash Mansinghka, Outside Member

Ramesh K. Sitaraman, Associate Dean for
Educational Programs and Teaching
Robert and Donna Manning College of
Information and Computer Sciences

DEDICATION

For my wife Katharine, and our faithful pup Mira.

ACKNOWLEDGMENTS

During the course of my PhD I have had the good fortune to be surrounded by brilliant and inspiring peers, collaborators, mentors, and friends who made this foray into academic endeavors possible.

My PhD Advisor, David Jensen, has been a constant source of insight, inspiration, and guidance over this past six years of academic pursuits, always encouraging me to be expansive in my thinking and precise in my execution. Even before I was a member of his research group at UMass, the Knowledge Discovery Lab, David inspired me to think about ideas that would later make their way into this thesis, bridging the gap between causal inference and probabilistic programming. Perhaps more important than any specific technical guidance, which was certainly plentiful, David taught me what it meant to study computation itself as a scientific discipline, rather than just as a means to some other end. Anyone who's taken his course on research methods in Computer Science will tell you that David is a fabulous mentor.

In the final three years of my PhD I worked closely with Vikash Mansinghka as a visiting student in his group at MIT, the Probabilistic Computing Project. I learned and accomplished more in this three year period than any other prior, thanks in no small part to Vikash's mentorship and guidance. Vikash helped me take abstract ideas and transform them into concrete research contributions, all while maintaining an overarching perspective of where the work was going. Vikash's vision for the field of probabilistic programming is nothing short of inspiring, and I'm truly grateful that I had the opportunity to play a small role in it. Looking back, it's clear that I would not have completed this thesis had it not been for either David or Vikash's support.

Thanks also to the other members of my committee, Justin Domke and Dan Sheldon, for helpful feedback on earlier drafts of this thesis. While I haven't worked with either Justin or Dan directly, I am inspired by their work on probabilistic inference and grateful to have their support. Additionally, thanks to Laura Balzer for providing additional mentorship and guidance as the second reader on my synthesis project for candidacy to the PhD program.

In the Knowledge Discovery Lab at UMass, I had the good fortune of working with many fellow students who have positively impacted my research trajectory. In our work together on Deep Reinforcement Learning [136], Emma Tosch helped me find my voice as a researcher and taught me how to take programming seriously as an intellectual discipline. In our work together on GP-SLC (see Chapter 4), Kenta Takatsu played an instrumental role and was an excellent thought partner throughout. Very early conversations with Javier Burroni and Dan Garant planted the seeds for what would later become Chapter 5, long before I knew much about probabilistic programming. I was also fortunate to collaborate with other KDL students, staff, and affiliated faculty on a variety of funded projects for IARPA and DARPA. I learned a great deal from working with Katerina Marazopoulou, Daniel Corkill, Kaleigh Clary, Akanksha Atrey, Andy Zane, Justin Clarke, David Westbrook, and Przemyslaw Grabowicz. Thanks also to David Arbour, Amanda Gentzel, Reilly Grant, Purva Pruthi, Sankaran Vaidyanathan, Jack Kenney, Kate Avery, Erica Cai, and Pracheta Amaranath for all of the feedback and camaraderie over the years.

At the Probabilistic Computing Project at MIT, I was also fortunate to work with, and be inspired by, many incredible students and affiliated researchers. Working closely with Alex Lew on causal Bayesian program synthesis (Chapter 5) taught me a tremendous amount about metaprogramming, and probabilistic programming generally. Frequent whiteboarding sessions with Marco Cusamano-Towner taught me about the inner workings of Gen [28], and probabilistic inference. Cameron Freer's feedback on early drafts of simulation-based identifiability (Chapter 6) and Feras Saad's feedback on GP-SLC (Chapter 4) taught me to be precise about mathematical claims. Many discussions with McCoy Becker taught me how to think practically about programming languages, and how to think deeply about performance. Thanks also to Tan Zhi-Xuan, Nishad Gothoskar, Ulrich Shaechtle, Jameson Quinn, Veronica Weiner, Ben Zinberg, Andrew Bolton, Sharan Yalburgi, Austin Garrett, Matin Ghavamizadeh, George Matheos, Amanda Brower, and Rachel Paiste for all of the feedback and support during my time at the Probabilistic Computing Project.

In addition to my time at the Knowledge Discovery Lab and at the Probabilistic Computing Project, I have had the great fortune of collaborating with researchers at other universities and organizations. Most importantly, I'd like to thank the participants in the "Causal Probabilistic Programming Reading Group", organized by Eli Bingham. Starting in 2020 and approximately once per week Eli Bingham, Zenna Tavares, Robert Ness, Alex Lew, Jeremy Zucker, Jimmy Koppel, and I would discuss connections between causal inference and probabilistic programming. These discussions culminated in an early version of the work that would later become Chapter 3, which I presented at the 2021 International Conference on Probabilistic Programming. I'm grateful to be able to continue working with Eli and Zenna at the Basis Research Institute, along with new colleagues Emily Mackevicius, Martin Jankowiak, Raj Agrawal, Rafal Urbaniak, Ria Das, Karen Schroeder, Archana Warrier, and Marjorie Xie. I'm especially grateful to Rafal Urbaniak for his detailed feedback on this thesis. It's a remarkable experience to have such capable thought partners in our shared exploration of this new and exciting research subfield, and I'm excited to see where we take it in the coming years.

I am grateful to the many friends I made at UMass and MIT over these past six years. Among many other activities, I'll fondly remember sharing meals with Ryan and Rachel Harb, rock climbs with Katie Keith, homework cocktails with Conrad Holtsclaw, runs and chats with Justin Svegliato, Dungeons and Dragons with Connor Basich and Sam Baxter, and ping pong sessions with Marco Cusamano-Towner, Nishad Gothoskar, and McCoy Becker. Of course, this effort was also supported by my many friends who knew me before this journey into academia. In no particular order, thanks to Ian Dawud, Nick Shrewsbury, Ross Rivers, Gio Musto, Trevor White, Max Bridges, Devon Dawson, Brian Cantrell, Eric Spencer, Ryan Pollin, Gene Rush, Nate Jones, Liyang Wang, Mike Olson, Ben McDaniel, Andrew Marchev, John Zarcone, Jake Grant, Alexa Clark, Cody Ball, Christin Sluter, Dovrah Plotkin, Austen Higgins-Cassidy, Jacob Neilson-Philips, and many others along the way.

Thanks to my parents, Richard and Ruth Witty, for giving me the freedom to explore my own interests, even if they didn't always appear to be on the path to "success". I'm grateful to have learned to think critically and to seek truth from a young age.

Finally, thanks to my extremely supportive wife Katharine for the several years of patience as I took on this seemingly endless venture. Katharine always listened compassionately as I shared my frustrations about roadblocks, and listened enthusiastically as I shared my excitement while I completed work that would eventually turned into this thesis. Thanks for being my partner for these past several years, and for all the years that follow.

ABSTRACT

BAYESIAN STRUCTURAL CAUSAL INFERENCE WITH PROBABILISTIC PROGRAMMING

SEPTEMBER 2023

SAMUEL A. WITTY

B.S., UNIVERSITY OF MASSACHUSETTS AMHERST

M.S., UNIVERSITY OF MASSACHUSETTS AMHERST

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor David Jensen

Reasoning about causal relationships is central to the human experience. This evokes a natural question in our pursuit of human-like artificial intelligence: how might we imbue intelligent systems with similar causal reasoning capabilities? Better yet, how might we imbue intelligent systems with the ability to *learn* cause and effect relationships from observation and experimentation? Unfortunately, reasoning about cause and effect requires more than just data: it also requires partial knowledge about data generating mechanisms. Given this need, our task then as computational scientists is to design data structures for representing partial causal knowledge, and algorithms for updating that knowledge in light of observations and experiments. In this dissertation, I explore the *Bayesian structural* approach to causal inference in which probability distributions over structural causal models are one such data structure, and probabilistic inference in *multi-world* transformations of those models

as the corresponding algorithmic task. Specifically, I demonstrate that this approach has two distinct advantages over the dominant computational paradigm of causal graphical models: (i) it expands the breadth of compatible assumptions; and (ii) it seamlessly integrates with modern Bayesian modeling and inference technologies to facilitate quantification of uncertainty about causal structure and the effects of interventions. Specifically, doing so allows the emerging and powerful technology of probabilistic programming to be brought to bear on a large and diverse set of causal inference problems.

In Chapter 3, I present an example-driven pedagogical introduction to the Bayesian structural approach to causal inference, demonstrating how priors over structural causal models induce joint distributions over observed and latent counterfactual random variables, and how the resulting posterior distributions capture common motifs in causal inference. In particular, I show how various assumptions about latent confounding influence our ability to estimate causal effects from data and I provide examples of common observational and quasi-experimental designs expressed as probabilistic programs. In Chapter 4, I present an advanced application of the Bayesian structural approach for modeling hierarchical relational dependencies with latent confounders, and how to combine such assumptions with flexible Gaussian process models. In Chapter 5, I present a prototype software implementation for causal inference using probabilistic programming, accommodating a broad class of multi-source observational and experimental data. Finally, in Chapter 6, I present Simulation-Based Identifiability, a gradient-based optimization method for determining if any differentiable and bounded prior over structural causal models converges to a unique causal conclusion asymptotically.

TABLE OF CONTENTS

Page

CHAPTER

LIST OF TABLES

LIST OF FIGURES

Figure **Page**

NOTATION

Here, I provide the mathematical notation used throughout the remainder of this thesis, borrowed heavily from Bengio et al [13].

Numbers and Arrays

a	A scalar (integer or real)
$\boldsymbol{x}$	A vector
$\boldsymbol{X}$	A matrix
$\textsf{\textbf{X}}$	A tensor
$\boldsymbol{I}_n$	Identity matrix with n rows and n columns
$\boldsymbol{I}$	Identity matrix with dimensionality implied by context
x	A scalar random variable
$\mathbf{x}$	A vector-valued random variable
$\mathbf{X}$	A matrix-valued random variable

Sets and Graphs

$\mathbb{A}$	A set
$\mathbb{R}$	The set of real numbers
$\{0, 1\}$	The set containing 0 and 1
$\llbracket n \rrbracket$	The set of all integers between 1 and n
$[a, b]$	The real interval including a and b
$(a, b]$	The real interval excluding a but including b
$\mathbb{A} \backslash \mathbb{B}$	Set subtraction, i.e., the set containing the elements of $\mathbb{A}$ that are not in $\mathbb{B}$
$\mathcal{G}$	A graph
$Pa_{\mathcal{G}}(\mathrm{x}_i)$	The parents of x_i in $\mathcal{G}$

Indexing

x_i	Element i of vector $\boldsymbol{a}$, with indexing starting at 1
$X_{i,j}$	Element i, j of matrix $\boldsymbol{X}$
$\boldsymbol{X}_{i,:}$	Row i of matrix $\boldsymbol{X}$
$\boldsymbol{X}_{:,i}$	Column i of matrix $\boldsymbol{X}$
a_i	Element i of the random vector $\mathbf{a}$
$\mathrm{A}_{i,j}$	Element i, j of random matrix $\mathbf{A}$
$\mathbf{A}_{i,:}$	Row i of random matrix $\mathbf{A}$
$\mathbf{A}_{:,i}$	Column i of random matrix $\mathbf{A}$

Linear Algebra Operations

$\boldsymbol{X}^\top$	Transpose of matrix $\boldsymbol{X}$
$\boldsymbol{X} \odot \boldsymbol{Y}$	Element-wise (Hadamard) product of $\boldsymbol{X}$ and $\boldsymbol{Y}$
$\det(\boldsymbol{X})$	Determinant of $\boldsymbol{X}$

Calculus

$\frac{dy}{dx}$	Derivative of y with respect to x
$\frac{\partial y}{\partial x}$	Partial derivative of y with respect to x
$\nabla_{\boldsymbol{x}} y$	Gradient of y with respect to $\boldsymbol{x}$
$\frac{\partial f}{\partial \boldsymbol{x}}$	Jacobian matrix $\boldsymbol{J} \in \mathbb{R}^{m \times n}$ of $f : \mathbb{R}^n \to \mathbb{R}^m$
$\int f(\boldsymbol{x}) d\boldsymbol{x}$	Definite integral over the entire domain of $\boldsymbol{x}$
$\int_{\mathbb{S}} f(\boldsymbol{x}) d\boldsymbol{x}$	Definite integral with respect to $\boldsymbol{x}$ over the set $\mathbb{S}$

Probability and Information Theory

$P(\mathrm{a})$	A probability distribution over a discrete variable
$p(\mathrm{a})$	A probability distribution over a continuous variable, or over a variable whose type has not been specified
$\mathrm{a} \sim P$	Random variable a has distribution P
$\mathbb{E}_{\mathrm{x} \sim P}[f(x)]$ or $\mathbb{E} f(x)$	Expectation of $f(x)$ with respect to $P(\mathrm{x})$
$\mathrm{Var}(f(x))$	Variance of $f(x)$ under $P(\mathrm{x})$
$\mathrm{Cov}(f(x), g(x))$	Covariance of $f(x)$ and $g(x)$ under $P(\mathrm{x})$
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$

Functions

$f : \mathbb{A} \rightarrow \mathbb{B}$	The function f with domain $\mathbb{A}$ and range $\mathbb{B}$
$f(\boldsymbol{x}; \boldsymbol{\theta})$	A function of $\boldsymbol{x}$ parametrized by $\boldsymbol{\theta}$. (Sometimes I write $f(\boldsymbol{x})$ and omit the argument $\boldsymbol{\theta}$ to lighten notation)
$\log x$	Natural logarithm of x
$\mathbb{1}_{\text{condition}}$	a function which returns 1 if the condition is true and 0 otherwise

Sometimes I use a function f whose argument is a scalar but apply it to a vector, matrix, or tensor: $f(\boldsymbol{x})$, $f(\boldsymbol{X})$, or $f(\textsf{\textbf{X}})$. This denotes the application of f to the array element-wise. For example, if $\textsf{\textbf{C}} = \sigma(\textsf{\textbf{X}})$, then $\mathsf{C}_{i,j,k} = \sigma(\mathsf{X}_{i,j,k})$ for all valid values of i, j and k.

The half minute which we daily devote to the winding-up of our watches is an exertion of labour almost insensible; yet, by the aid of a few wheels, its effect is spread over the whole twenty-four hours.

Charles Babbage

CONTRIBUTIONS

The chapters in this dissertation are based on the following publications[1]:

Chapter 3

Causal Probabilistic Programming without Tears.
Bingham*, Koppel*, Lew*, Ness*, Tavares*, **Witty***, Zucker*. (*Equal Contribution, Alphabetical)
International Conference on Probabilistic Programming. (2021).

Chapter 4

Causal Inference using Gaussian Processes with Structured Latent Confounders.
Witty, Takatsu, Jensen, and Mansinghka.
International Conference on Machine Learning. (2020).

Chapter 5

Bayesian Causal Inference via Probabilistic Program Synthesis.
Witty*, Lew*, Jensen, and Mansinghka. (*Equal Contribution)
International Conference on Probabilistic Programming. (2020).

Chapter 6

SBI: A Simulation-Based Test of Identifiability for Bayesian Causal Inference.
Witty, Jensen, and Mansinghka.
arXiv PrePrint. (2021).

[1] I have significantly expanded on the content in Chapter 3 for this Dissertation. Additionally, in the conclusion, I briefly discuss some of my other contributions connecting causal inference to applications in: (i) evaluating the generalization capabilities of deep reinforcement learning agents [136]; and (ii) searching for fair machine learning classifiers [65], as well as preliminary work on the expressiveness of higher-order probabilistic programming languages for causal models [134].

CHAPTER 1
INTRODUCTION

In this thesis, I hope to contribute both conceptually and pragmatically towards the long-term goal of understanding and engineering intelligent systems. Specifically, I focus on intelligent systems that can reason with uncertainty about probabilistic cause-effect relationships and that can learn from observation and experimentation. Imbuing intelligent systems with causal representations allows them to reason about global consequences of local interventions, giving them the ability to reason about how their decisions influence their environment. The suite of techniques I develop aims to be both conceptually clarifying—reducing previously distinct ideas from the social sciences, economics, statistics, and computer science to probabilistic modeling and inference—and practically useful—opening the door to new capabilities in AI-assisted scientific discovery, data-driven public policy, robotics, and a host of other application areas.

1.1 Motivation

The goal of building intelligent systems that learn from experience is certainly not novel to this thesis. Machine learning, the subfield of artificial intelligence focused on engineering intelligent systems that learn, has been a goal for computer scientists almost as long as the field of computer science has existed. Typically, machine learning methods are partitioned in terms of the characteristics of the data they learn from; supervised learning for feature-label pairs, unsupervised learning for features alone, and reinforcement learning for sequential interaction with an environment that provides feedback. However, perhaps a more meaningful distinction can be made by looking at what can be accomplished by the learned intelligent system. In other words, we might instead wish to categorize methods based on the kinds of questions these intelligent agents are learning to answer, not simply how they learn to answer them.

One class of such questions are of the form "what is?" or what are typically called *associational* questions. For example, a face detection system built using a convolutional neural network architecture may answer the question "given an image as input, how many faces are there and where are they located?" Answering these kinds of questions poses a number of challenging methodological problems, as any such algorithm will have to implicitly or explicitly reason about physically salient concepts such as spatial continuity, occlusion, and symmetry. However, intelligent systems may need to answer questions beyond "what is?", such as questions of the form "what will be if?" Answers to these kinds of questions allow intelligent agents to make decisions that manipulate the world, or to generalize to new environments [136], not just describe its current configuration. These types of questions, and their answers, are typically called *causal*.

Unfortunately, asking intelligent systems to answer causal questions poses a number of technical challenges that don't appear when answering associational questions. These challenges are ones we're intimately familiar with in everyday life. From a very young age we're taught that "correlation does not equal causation"; or in other words, associational relationships may mislead us if interpreted causally. Without this wisdom, we may erroneously infer that banning ice cream will almost entirely eliminate swimsuit sales, as ice cream consumption and swimsuit sales are highly correlated. Instead, we know that the correlation between ice cream consumption and swimsuit sales manifests in our observations because of the confounding effect of seasonality. Simply put, summer weather encourages people to both swim more and eat more ice cream.

While true, the mantra that "correlation does not equal causation" is unnecessarily pessimistic; it ignores the fact that humans are able to make causal judgments every day, many of which accurately predict the result of changes to an environment. A more honest and actionable account of causal inference is given by Judea Pearl that, "behind every causal conclusion there must lie some causal assumption" [96]. Without such assumptions there are an unbounded collection of causal explanations that are compatible with our observations of the world, but that imply different conclusions. With causal assumptions however, we (partially) restrict the space of causal explanations and we may thus come to unique (or a small collection of) causal conclusions that are consistent with data.

1.1.1 Thesis Overview

What then is an appropriate representation for encoding causal assumptions, and what are the underlying recipes for manipulating these representations to yield causal conclusions? In other words, what are the *data structures* and *algorithms* that enable causal reasoning? In this thesis, I explore a representation of causal knowledge as probability distributions (or priors) over structural causal models (SCMs) as one such data structure and probabilistic inference as the underlying algorithmic specification for updating that knowledge in light of data.

On the surface, a fully specified SCM is just a particular template for defining a distribution over observed (endogenous) variables in terms of a collection of deterministic functions of latent (exogenous) random noise variables.[1] However what makes SCMs "causal" is that they permit a kind of model transformation called an *intervention.* Interventions take as input a SCM and returns a SCM in which some of

[1] I elaborate on the definitions of a SCM in Chapter 2 and on the definitions for and intuition behind the Bayesian structural approach in Chapter 3.

the deterministic functions have been modified.[2] As a canonical example, applying an *atomic intervention* to a structural causal model, denoted by Pearl and colleagues as $do(\mathrm{x} = x)$, is accomplished by replacing the structural function $\mathrm{x} = f(Pa(\mathrm{x}))$ with the expression $\mathrm{x} = x$, and leaving all other functions unchanged. This atomic intervention models the scenario where some external force has set a particular attribute, irrespective of its original data generating mechanism. For example, we might use this mathematical description of an intervention to model how the world would change if we imposed new masking requirements in an infectious disease transmission model, whereas our model posits that absent an intervention people choose whether to wear a mask based on their own preferences.

With an intervention semantics in hand, structural causal models implicitly denote more than just a joint distribution over a collection of random variables. They also denote a joint distribution over random variables after some dependency-altering intervention has been applied, which I call *counterfactual* random variables throughout this thesis[3]. In this way, structural causal models explicitly encode causal assumptions, and queries applied to these counterfactual random variables represent answers to the kinds of "what if" questions that can be used to inform decision making.

In practice, choosing a single fully specified SCM is often far too strong of a modeling choice; we don't often know the exact mechanisms by which data (and counterfactuals) are generated in any particular domain. Instead, we would like to

[2] Other related formalisms can be used to define prior distributions over causal models that are not isomorphic to a SCM, such as models where interventions influence the number of entities in a system, not just their attributes [124]. Note also that SCMs can define cyclic dependencies, and can thus reflect causal models with context-dependent causal structure, i.e. conditional branching [134]. However, Markovian causal graphs, the more refined data structure on which most graph-based algorithms operate, do not permit context-dependent structure, as they enforce that the candidate collection of structural causal models are acyclic.

[3] Here, the term "counterfactual" refers to to an instantiation of a random variable in a transformed version of the world in which an intervention has been applied.

posit our uncertainty over a *class* of plausible SCMs, and then use observational and experimental data to further disambiguate between members of that class[4]

In this thesis, I explore the *Bayesian structural* approach to causal inference and show how it can be implemented and supported by probabilistic programming[5]. As the name suggests, the Bayesian structural approach to causal inference involves being Bayesian about structural causal models, that is representing uncertainty explicitly in terms of probability distributions (or priors) over structural functions and exogenous noise. Placing a prior over structural causal models in this way implicitly induces a joint distribution over observed and counterfactual random variables, i.e. a *multi world* construction, and therefore a distribution over counterfactual random variables conditional on observations. Using this approach, causal inference simply reduces to (a carefully constructed) probabilistic inference problem. With this conditional query as a target, a user can then apply their choice of exact or approximate inference algorithm, including Monte Carlo [37, 35], variational [102, 131], or some combination of approaches thereof [33, 115]. In Chapter 3, I expand on this description.

1.1.2 Key Claims

In the remainder of this chapter I summarize five key claims about the Bayesian structural approach as a computational foundation for causal inference, and a summary of how the remainder of the thesis provides evidence in support of each claim.

[4] Pearl's causal graphical approach uses directed acyclic graphs to represent classes of plausible SCMs. In Chapter 2, I discuss the key differences between the graphical and the Bayesian structural representations of causal assumptions.

[5] It is somewhat misleading to describe the contributions of this thesis as an alternative to existing formalisms, as it borrows heavily from existing well-studied data structures. Specifically, the Bayesian structural approach I explore in this thesis borrows the underlying structural formalisms from Pearl's structural approach [97], and expresses causal estimands in terms of counterfactuals reminiscent of potential outcomes [61]. As I'll elaborate on in Chapter 3, what is novel is how a user expresses uncertainty over these borrowed structural objects, and the algorithms that operate over them.

Claim 1. *The Bayesian structural approach provides an expressive substrate for representing practical assumptions for causal inference that cannot be expressed using graph structure alone.*

I provide evidence for Claim 1 in Chapters 4, 5, and 6. In Chapter 4, I show how to use the Bayesian structural approach to estimate causal effects in hierarchical relational settings in which latent confounders are shared among multiple observed instances. In Chapter 5, I show how to use the Bayesian structural approach to model multi-source observational and experimental data, including experiments reflecting atomic interventions, encouragement designs, and other custom program transformations. Finally, in Chapter 6, I show how the Bayesian structural approach can be used to model instrumental variable designs [26] and regression discontinuity designs [73]. None of these applications are covered by the causal graphical approach.

Claim 2. *A large and diverse collection of qualitative findings scattered throughout the causal inference literature emerge as a consequence of the Bayesian structural approach to causal inference.*

I provide evidence in support of Claim 2 in Chapters 3 and 6. In Chapter 3, in a series of worked examples, I show that the Bayesian structural approach agrees with Pearl's graphical approach in a simple linear Gaussian example. In Chapter 6, I show that the Bayesian structural approach produces conclusions that are consistent with known identifiability results for graph-based [96] and econometric quasi-experimental designs, including instrumental variable designs [26], within-subjects designs [43], and regression discontinuity designs [73].

Claim 3. *The Bayesian structural approach can be used to represent broad uncertainty over structural functions and to learn complex nonlinear dependencies from data.*

I provide evidence in support of Claim 3 in Chapters 4 and 6. Specifically, in Chapter 4, I show how to combine rich causal assumptions about relational dependencies in combination with flexible Gaussian process models to estimate causal effects. In Chapter 6, I show how similar Gaussian process models can be used to extend instrumental variable and regression discontinuity designs.

Claim 4. *The Bayesian structural approach can provide valuable insight into causal inference problems even without exact probabilistic inference, which is NP-hard in general.*

I provide evidence in support of Claim 4 in Chapters 4, 5, and 6. In Chapters 4 and 5, I show that using the Bayesian structural approach with approximate inference techniques produces accurate estimates for structure learning and effect estimation tasks. In Chapter 6, I prove that determining when a causal query is identifiable does not require exact probabilistic inference, and instead only requires determining whether there exist two likelihood-equivalent structural causal models that induce different effect estimates, a task that can more easily be approximated via gradient-based optimization of a custom loss function on simulated data.

Claim 5. *The Bayesian structural approach provides a computational foundation on which a software engineering discipline of causal inference can be constructed, enabling modular, composable, and extensible software artifacts that facilitate causal inference.*

I provide evidence in support of Claim 5 in Chapters 3 and 5, demonstrating how the Bayesian structural approach can be implemented in software on top of existing probabilistic programming systems supplemented with program transformation interventions. In Chapter 3, I show how function composition in a probabilistic programming language can be used to implement layers of successively more uncertain model specifications, i.e. a Bayesian hierarchical model of structural causal models. In Chapter 5, I present a prototype software system for causal inference with observational and experimental data, implementing interventions as syntax-rewriting program transformations on a restricted domain-specific language for causal problems, embedded in the Gen probabilistic programming language [28] using a custom interpreter.

CHAPTER 2
BACKGROUND

In this chapter I discuss the necessary background on causal inference, Bayesian statistics, Gaussian process models, and probabilistic programming.

2.1 Causal Inference

Perhaps due to to the diversity of its applications and the relatively minimal contact between academic fields of computer science, economics, statistics, and public policy, many distinct formalisms for causal inference have been proposed over the past several decades. The two most prominent of these formalisms are known as the Neyman-Rubin potential outcomes framework [61], which frames the problem of causal inference principally as a statistical missing data problem, and Pearl's structural approach [97], which instead frames the problem in terms of mathematical logic.

2.1.1 Potential Outcomes

As the name implies, the potential outcomes framework sets up causal inference problems as having two "potential outcomes", denoted with parenthetical notation as $\mathrm{y}_i(1)$ and $\mathrm{y}_i(0)$, corresponding to what the outcome would be for the i'th individual if they receive a binary treatment ($\mathrm{t}_i = 1$) or do not receive that treatment ($\mathrm{t}_i = 0$)[1]. However, in any given dataset, we only observe one of these two potential outcomes, and the other remains latent. Instead, the outcome we observe, y_i, is determined by the actual observed treatment assignment, or as an equation, $\mathrm{y}_i = \mathrm{y}_i(1) \cdot \mathrm{t}_i + \mathrm{y}_i(0) \cdot (1 - \mathrm{t}_i)$.

[1] The potential outcomes formalism easily generalizes to categorical or continuous treatments.

Not being able to observe all potential outcomes is the source of the "fundamental problem of causal inference", namely that without strong assumptions we can only unambiguously estimate (sub)population-level effects, and not effects on a specific individual [61]. However, estimating even these (sub)population-level effects requires assumptions about the data generating mechanism.

In the potential outcomes framework, the most common and well-studied assumptions for causal inference with observational data are known as *strong ignorability*, *stable unit treatment value*, and *overlap*, and are defined as follows:

Assumption 2.1.1. ***Strong Ignorability.*** *Treatment assignment (and thus the mechanism by which we observe potential outcomes) is independent of the value of those potential outcomes conditional on pre-treatment covariates* x_i[2]*. Somewhat more formally:*

$$(\mathrm{y}_i(1), \mathrm{y}_i(0)) \perp\!\!\!\perp \mathrm{t}_i | \mathrm{x}_i$$

Assumption 2.1.2. ***Stable Unit Treatment Value.*** *The potential outcomes for each instance i are independent of the treatment assignment of any other instance j. Again, somewhat more formally*[3]*:*

$$(\mathrm{y}_i(1), \mathrm{y}_i(0)) \perp\!\!\!\perp \mathrm{t}_j, \forall j \neq i$$

Assumption 2.1.3. ***Overlap.*** *Each treatment assignment has nonzero probability under each joint assignment of the covariates.*

$$0 < p(\mathrm{t}_i | \mathrm{x} = x) < 1, \forall x \in support(\mathrm{x})$$

[2] Strictly speaking, more stringent assumptions are necessary to yield identifiability. For some counterexamples and caveats see the work of Cinelli et al. [25].

[3] Typically, the stable unit treatment value assumption (SUTVA) also include an additional assumption about there being no hidden variations in treatment assignment [61]. Unfortunately, this component of the SUTVA assumption is much more difficult to express succinctly in mathematical notation, so I omit it for brevity.

While these three assumptions are not sufficient to answer unit-level queries, such as the individual treatment effect — $(\mathrm{y}_i(1) - \mathrm{y}_i(0))$ for some $i \in [\![n]\!]$ — they are sufficient to estimate the *sample average treatment effect*, $SATE = \sum_{i \in [\![n]\!]} [\mathrm{y}_i(1) - \mathrm{y}_i(0)]$. Using the law of iterated expectations we can come to the following standard result [111]:

$$SATE := \sum_{i \in [\![n]\!]} [\mathrm{y}_i(1) - \mathrm{y}_i(0)] = \mathbb{E}_{\mathrm{x}}[\mathbb{E}[\mathrm{y}_i|\mathrm{t}_i = 1, \mathrm{x}] - \mathbb{E}[\mathrm{y}_i|\mathrm{t}_i = 0, \mathrm{x}]] \tag{2.1}$$

In essence, Equation 2.1 states that if strong ignorability, SUTVA, and overlap hold, then the sample average treatment effect can be estimated from observational data simply by estimating expectations of observable quantities. In other words, we've translated from an expectation of (latent) potential outcomes, $\mathrm{y}_i(1)$ and $\mathrm{y}_i(0)$, to an expectation over observed actual outcomes, y_i. It is worth noting that this equivalence does not directly lead to conclusions about how we should estimate these expectations to achieve the best statistical properties, only that it is possible from the variables we can observe. Answering the question of how best to estimate these quantities is a very active field of research. Interested readers should read Imbens and Rubin's recent textbook [61] for a survey of causal effect estimation methods and their statistical properties.

So far, I have discussed how one specific collection of assumptions leads to a statistical quantity that can then be estimated from data. What if instead we don't want to make these exact assumptions? Unfortunately, the potential outcomes approach does not provide a general algorithmic recipe for translating from assumptions to target statistical quantities[4]. As a result, the statistics literature on causal inference

[4] There has been some progress on algorithmically representing the potential outcomes literature. For example, single-world intervention graphs [106] represent assumptions and concepts in potential outcomes using graphical representations, and the potential outcomes calculus [82] provides an algorithmic solution to identifying nested counterfactuals that are common in the potential outcomes literature, such as mediation analysis. However, none of these approaches represent the parameteric or structural assumptions necessary for the quasi-experimental designs discussed throughout this thesis.

has instead focused on collectively building up what is a essentially a catalog of causal inference assumption templates, and specialized estimation techniques for these common causal inference problems. Unfortunately, this leaves an awkward gap for practitioners who want to make use of custom assumptions.

Here, I describe three quasi-experimental designs [22], i.e. non-experimental settings that share some characteristics with controlled or randomized experiments, that violate at least one of the three assumptions needed to derive Equation 2.1. These quasi-experimental designs are a useful substrate for understanding the trade-offs between existing formalisms, and as a target for our Bayesian structural approach in subsequent chapters.

2.1.1.1 Instrumental Variable Designs

Instrumental variable designs represent settings in which strong ignorability is violated, meaning that treatment assignment and potential outcomes are no longer conditionally independent given covariates. In the nomenclature of causal inference, in these settings treatment and outcome may be confounded. Instead, we may be able to take advantage of a special (collection of) covariate(s) called an *instrument(s)* that can be used to estimate effects despite the fact that treatment and outcome may be confounded.

In order to yield a valid instrument, and thus enable unbiased effect estimation in the presence of confounding, we must make some additional assumptions. While I omit the formal specification of these assumptions [8] for brevity, the most important assumptions are given below:

Assumption 2.1.4. ***Exclusion:*** *The instrument,* z_i*, has no effect on outcome except in its influence on the treatment,* t_i*.*

$$(\mathrm{y}_i(1), \mathrm{y}_i(0)) \perp\!\!\!\perp \mathrm{z}_i$$

Assumption 2.1.5. ***As-if Random:*** *The instrument,* z_i*, is not influenced by any latent variables that also influence treatment,* t_i*, or outcome,* y_i*.*

$$\mathrm{t}_i \perp\!\!\!\perp \mathrm{z}_i | \mathrm{x}_i$$

$$\mathrm{y}_i \perp\!\!\!\perp \mathrm{z}_i | \mathrm{x}_i, \mathrm{t}_i$$

In addition to these assumptions, instrumental variables also require parametric assumptions about how z_i influences t_i, although many different variants of these parametric assumptions may be sufficient [79]. For example, in the discrete case, assuming that the effect of z_i on t_i is monotonic is sufficient. Stating these assumptions in their most general form is out of scope for this thesis [122]. In Chapter 6, I discuss instrumental variable designs with stronger-than-necessary parametric assumptions for continuous treatment and outcome variables.

Instrumental Variable Example: Military Service and Lifetime Earnings. As an example, consider the question of whether military service increases lifetime earnings [7]. To answer this question we could simply compare the lifetime earnings for individuals who did and did not enlist in the military over some time period, but this estimation procedure would likely lead to biased estimates of the effect we're interested in. In actuality, individuals who are inclined to enlist in the military may have certain latent attributes that also influence their subsequent lifetime earnings, such as their level of education or pre-enlistment financial conditions. However, during the period of the Vietnam war military enlistment was not entirely self selecting; many individuals' military service was driven by the mandatory draft. In this setting, within the subset of the population of individuals who were healthy and eligible for service, an individual's observed draft ticket serves as one such instrumental variable. One could reasonably argue that the draft ticket satisfies the exclusion condition;

an individual's draft ticket does not influence their lifetime earnings except through their military service, and also the as-if random condition; the draft tickets are not influenced by any of the individual's latent attributes.

2.1.1.2 Regression Discontinuity Designs

Regression discontinuity designs represent settings in which overlap/positivity is violated, meaning that some assignments of covariates lead to a zero probability of a particular treatment assignment. This poses a problem, as we no longer have data in all regions of treatment/covariate space with which to estimate the conditional expectations in Equation 2.1.

More precisely, (sharp) regression discontinuity designs violate the overlap assumptions by having treatment be assigned deterministically according to whether a particular (set of) continuous covariate(s) is above or below a known threshold. In other words, we have one of the following two equations, where $\mathbb{1}$ is the indicator function and $a \in \mathbb{R}$ is known:

$$\mathrm{t}_i = \mathbb{1}_{\mathrm{x}_i > a} \text{ or } \mathrm{t}_i = \mathbb{1}_{\mathrm{x}_i < a}$$

In these settings the question of whether we can estimate causal effects from observational data is somewhat more nuanced [73]. For example, if we assume that the relationship between covariates and outcome is linear, then the sample average treatment effect can be estimated by extrapolating the linear relationships to regions of covariate space in which we have no data, thus estimating the expectations in Equation 2.1. If we are unwilling to make such an assumption, but are willing to assume that the relationship between covariates and outcome is smooth then we can estimate the sample average treatment effect only locally near the discontinuity at a. This query is known as the *conditional average treatment effect* for what are hopefully obvious reasons. In Chapter 6 I elaborate on some of the nuances of the regression discontinuity design, and how the Bayesian structural approach reflects those nuances.

Regression Discontinuity Design Example: Test Taking and Course Enrollment. Perhaps the most standard example of a regression discontinuity design is also its first known formal application; assessing the effect of honorary awards on lifelong academic achievement [125]. The causal hypothesis under scrutiny was that receiving an award leads to higher long term academic achievement, as recognition leads to "favorable attitudes towards intellectualism". However, receiving an award and eventually obtaining an advanced degree may be confounded by many other factors, such as the quality of the student's early education, and by proxy their test scores during their early education. In this setting, even if we believe that such test scores satisfy the strong ignorability assumption, the way observational data is generated poses a problem. Specifically, in our observational data students are only provided an academic award if their test performance exceeds some threshold. This is exactly the setting described mathematically above.

2.1.1.3 Structured Latent Confounding

Settings with structured latent confounding (often referred to as multi-level, clustered, or panel data) represent a particular violation of the stable unit treatment value assumption, meaning that potential outcomes for one individual may be statistically dependent on the treatment assignment for another, perhaps due to latent confounders that are shared amongst individuals. Instead, in these settings we can assume that data is observed according to coherent groups, and that strong ignorability holds locally within each group. Letting w_i be the group assignment of unit i, then we have the following:

$$(\mathrm{y}_i(1), \mathrm{y}_i(0)) \perp\!\!\!\perp \mathrm{t}_i | \mathrm{x}_i, \mathrm{w}_i$$

In settings where the cardinality of w_i is fixed, and does not scale with the number of units n, then this is exactly equivalent to the strong ignorability condition. However, in some settings the number of groups may continue to increase as n increases. Again, like the instrumental variable and regression discontinuity designs, whether we can estimate the effect unambiguously depends on whether we are willing to make (strong) parametric assumptions. For example, if we assume linear and additive relationships between treatment, covariates, and outcome, then we can estimate effects even if the size of each group remains finite as $n \to \infty$ [56, 134].

Structured Latent Confounding Example: Evaluating Kindergarten Retention Policy. One natural area in which individual units are partitioned into grouped structure is when trying to make causal inferences in educational settings. For example, if we are interested in understanding the effect of retaining students in kindergarten on subsequent academic performance [58], we may wish to better inform policy decisions by first analyzing observational data. One challenge, however, is that statistical dependencies within a given school may differ from statistical dependencies across schools, as the schools' policies may simultaneously influence whether students are retained and their subsequent academic achievements. In this setting, we can model the observational data using a multi-level model, accounting for heterogeneity between schools.

2.1.2 Pearl's Structural Approach

Here, I describe structural causal models, the key mathematical object underlying Pearl's graphical approach, as well as the Bayesian structural approach that I explore throughout this thesis. This section aims to both: (i) introduce the notation used throughout the remainder of this thesis; and (ii) provide background on causal graphical models as a formalism for causal reasoning and inference.

Definition 2.1.1. ***Structural Causal Models.*** *A structural causal model (SCM) is a four-tuple* $\mathbb{M} = (\mathbb{V}, \mathbb{F}, \mathbb{X}, \mathbb{U})$*, where:* $\mathbb{V} = \{\mathbf{t}, \mathbf{y}, \mathbf{x}_1, \ldots, \mathbf{x}_d\}$ *is a set of observed variables,* $\mathbb{F} = \{f_t, f_y, f_{x_1}, \ldots, f_{x_d}\}$ *is a set of deterministic functions,* $\mathbb{X} = \{\boldsymbol{\epsilon}_t, \boldsymbol{\epsilon}_y, \boldsymbol{\epsilon}_{x_1}, \ldots, \boldsymbol{\epsilon}_{x_d}\}$ *is a set of exogenous latent noise variables, and* $\mathbb{U} = \{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{d'}\}$ *is a set of latent confounder variables. Each observed random variable in* $\mathbb{V}$ *is assigned deterministically according to its corresponding structural function, e.g.* $\mathbf{y} = f_y(\mathbf{t}, \boldsymbol{u}_1, \boldsymbol{\epsilon}_y)$*, where* $\mathbf{t} = f_t(\cdot)$[5]*. By construction, each* $\boldsymbol{\epsilon} \in \mathbb{X}$ *is an argument of exactly one structural function, and each* $\boldsymbol{u} \in \mathbb{U}$ *is an argument of at least two structural functions.*[6]

Definition 2.1.2. ***Probabilistic Structural Causal Model.*** *A probabilistic structural causal model is a tuple* $\mathbb{M}_p = (\mathbb{M}, p(\mathbb{X}), p(\mathbb{U}))$*, where:* $\mathbb{M}$ *is a structural causal model,* $p(\mathbb{X})$ *is a distribution over values of latent exogenous noise variables,* $\boldsymbol{\epsilon} \in \mathbb{X}$*, and* $p(\mathbb{U})$ *is a distribution over values of latent confounders,* $\boldsymbol{u} \in \mathbb{U}$*.*

Definition 2.1.3. ***Atomic Intervention.*** *An atomic intervention is a mapping* $\mathbb{I} : \mathcal{T} \times \mathcal{F} \to \mathcal{F}$*, where* $\mathcal{T}$ *is the domain of* $\mathbf{t}$ *and* $\mathcal{F}$ *is the space of structural functions* $\mathbb{F}$*. Specifically, given an intervention assignment* $\boldsymbol{t} \in \mathcal{T}$ *and a collection of structural functions* $\mathbb{F} \in \mathcal{F}$*, an atomic intervention* $\mathbb{I}(\boldsymbol{t}, \mathbb{F})$ *produces a collection of structural functions* $\mathbb{F}'$ *in which* f_t *has been replaced with the function* $\mathbf{t} = \boldsymbol{t}$*, and all other structural functions are left unchanged*[7]*.*

[5] Note that each structural function defines a process for generating a vector of instances, $\mathbf{y}$, in which each element y_i represents a single data instance. Additionally, the cardinality of each $\mathrm{v} \in \mathbb{V}$ may differ, such as when $\mathbf{t}$ and $\mathbf{y}$ are length n vectors of students' time spent studying and their grades respectively, and $\mathbf{x}_1$ is a length m vector of course difficulties, where n is the number of students and m is the number of courses. I elaborate on this example in Chapter 4.

[6] Here, I slightly modify the definition in Pearl's classic textbook "Causality" [97] to distinguish between confounders and exogenous noise, and to clarify that I only consider interventions on a single variable, $\mathbf{t}$, and I only consider queries on a single variables, $\boldsymbol{y}$. In full generality SCMs (and the structural Bayesian approach) applies when interventions are applied to any subset of endogenous variables and with multiple outcomes of interest.

[7] The assumption that such an atomic intervention exists is referred to as "modularity" or "autonomy" [4, 97]. Interventions are often denoted using the notation $do(\mathbf{t} = \boldsymbol{t})$. I choose the

Given a probabilistic structural causal model $\mathbb{M}_p$, we can define a collection of *counterfactual* random variables $\mathbb{V}(\boldsymbol{t}) = \{\mathbf{y}(\boldsymbol{t}), \mathbf{x}_1(\boldsymbol{t}) \ldots, \mathbf{x}_d(\boldsymbol{t})\}$, which are induced by the pushforward measure of $\mathbb{F}'$ applied to samples drawn from $p(\mathbb{X})$ and $p(\mathbb{U})$, where $\mathbb{F}' = \mathbb{I}(\boldsymbol{t}, \mathbb{F})$[8]. In this way, we can think of a probabilistic structural causal model as implicitly defining a (potentially infinite) exchangeable sequence of random variables indexed by intervention assignment, $\mathbb{V}, \mathbb{V}(\boldsymbol{t}_1), ..., \mathbb{V}(\boldsymbol{t}_k)$, each of which conceptually maps to a parallel world in which a different intervention has been applied. Here, exchangeability follows directly from Di Finneti's Theorem, as $\mathbb{V}, \mathbb{V}(\boldsymbol{t}_1), .., \mathbb{V}(\boldsymbol{t}_k)$ are all conditionally independent given $\mathbb{X}$ and $\mathbb{U}$ by construction. As a result, we can trivially marginalize out any collection of counterfactual random variables as desired without consequence, only considering the relevant counterfactual worlds that help us answer a specific question. It is worth noting, however, that even given a single known collection of structural functions, $\mathbb{F}$, any pair of factual and counterfactual variable sets $\mathbb{V}, \mathbb{V}(\boldsymbol{t}_1), ..., \mathbb{V}(\boldsymbol{t}_k)$ are not independent, as they share the same sampled values of $\mathbb{X}$ and $\mathbb{U}$, albeit with (slightly) different structural functions $\mathbb{F}$ and $\mathbb{F}'$. I denote the joint distribution over factual and counterfactual variables as $p(\mathbb{V}, \mathbb{V}(\boldsymbol{t}_1), ..., \mathbb{V}(\boldsymbol{t}_k)|\mathbb{M}_p)$ for reasons that will become obvious when we induce a distribution over $\mathbb{M}_p$ in Chapter 3.

Thusfar, I have described the structural approach given a single known probabilistic structural causal model $\mathbb{M}_p$. However, in practice we will most often not wish to choose a single model a priori, and instead would prefer to specify broader uncertainty over a collection of candidate structural causal models.

functional notation shown here to emphasize that counterfactuals are themselves random variables via a pushforward measure through $\mathbb{I}$. This corresponds to the expository figures shown in Chapter 3.

[8] While these random variables are invoked using transformations of structural causal models ala Pearl, they are reminiscent of the potential outcomes framework's framing of causal inference as a missing data problem, hence the similar parenthetical notation. This observation that counterfactuals in the structural approach are equivalent to potential outcomes is not novel to this thesis, e.g. see Chapter 7 in Pearl's book "Causality" [97].

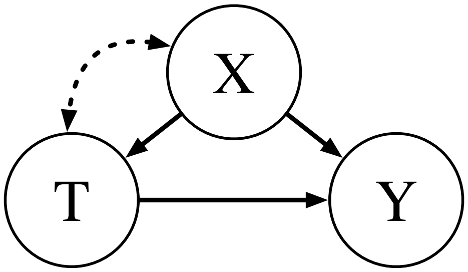


Figure 2.1: **Example backdoor causal graph.** Here, $p(\mathbf{y}(\boldsymbol{t}))$ can be estimated from data using $\mathbf{x}$ as a backdoor adjustment set. This is true despite the possible latent confounding between $\mathbf{t}$ and $\mathbf{x}$.

Typically, in Pearl's formalism uncertainty over structural causal models is encoded in the structure of a directed acyclic graph known as a causal graph. A causal graph is best thought of as a partial specification of a structural causal model, constraining the set of arguments to each structural function. Specifically, each node in a graph $\mathcal{G}$ represents a random variable in $\mathbb{V}$, and the set of incoming edges to each node represents the set of arguments to that variable's structural function. In addition to directed edges, causal graphs also contain bidirected edges between pairs of nodes in $\mathcal{G}$, representing the possibility of latent confounders. By omitting bidirected edges between pairs of variables $\mathbf{v}_1$ and $\mathbf{v}_2$ we are making the assumption that there does not exist a latent variable $\mathbf{u} \in \mathbb{U}$ that is an argument to both $\mathbf{v}_1$ and $\mathbf{v}_2$'s corresponding structural functions. These causal graphs are also sometimes called nonparametric structural equation models, as they place no restriction on the structural functions themselves except for their collection of arguments.

Perhaps surprisingly, this minimal specification is sometimes enough to draw causal conclusions from data. For example, the backdoor adjustment formula [97] states that if there exists a collection of nodes $\mathbb{Z} \subset \mathbb{V}$ in $\mathcal{G}$ that *block all backdoor paths* from $\mathbf{t}$ to $\mathbf{y}$, the causal query $p(\mathbf{y}(\boldsymbol{t}))$ can be expressed equivalently as $\mathbb{E}_{\boldsymbol{z} \sim p(\mathbb{Z})}[p(\mathbf{y}|\mathbf{t} = \boldsymbol{t}, \mathbb{Z} = \boldsymbol{z})]$, which only involves probabilistic expressions that can be estimated from observational data. See Figure 2.1 for a visual representation of one such graph. This backdoor

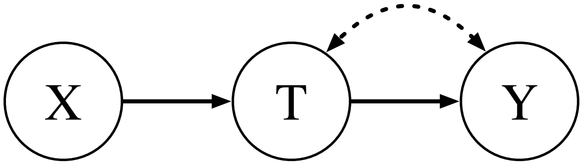


Figure 2.2: **Example instrumental variable graph.** Both the exclusion and as-if random assumptions are satisfied with this causal graph. However, additional parametric assumptions are necessary to estimate the effect of $\mathbf{t}$ on $\mathbf{y}$. These parametric assumptions cannot be expressed using graph structure alone.

adjustment formula is a special case of the more general do-calculus [97], which is a sound and complete algorithm for recovering similar adjustment formula from causal graph structure alone[9]. Perhaps even more surprisingly, in some cases $p(\mathrm{y}(\mathrm{t}))$ can be estimated from data even when a bidirected edge between $\mathbf{t}$ and $\mathbf{y}$ exists without additional parametric assumptions.

While these results are impressive, many practical assumptions cannot be expressed in terms of graph structure alone; they require additional restrictions on structural functions. To see this, let us again revisit our quasi-experimental design examples from Section 2.1.1.

2.1.2.1 Instrumental Variable Designs: A Graphical Perspective

Of the assumptions used in the instrumental variable design, two of them can be expressed in terms of the structure of a causal graph. Specifically, the exclusion and as-if random assumptions can be expressed graphically as follows:

Assumption 2.1.6. ***As-if Random:*** *There does not exist a bidirected edge between any instrument,* $\mathbf{x}$*, and either the treatment,* $\mathbf{t}$*, or the outcome,* $\mathbf{y}$*, in* $\mathbb{G}$*.*

[9] A particularly observant reader will notice the similarities between Pearl's backdoor adjustment formula and Equation 2.1 derived from the law of iterated expectation. In fact, modulo notation, the conclusions are equivalent. What is different however is that in Pearl's formalism the assumption of strong ignorability is derived from the assumed causal graph, and is not given axiomatically as in the potential outcomes framework.

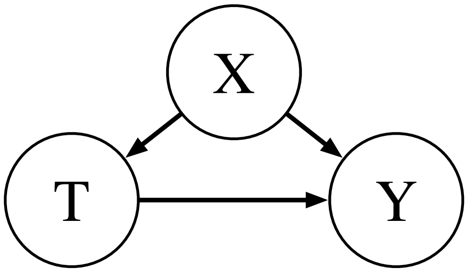


Figure 2.3: **Example regression discontinuity graph.** The assumption that $\mathbf{t}$ is assigned deterministically according to the value of $\mathbf{x}$ is not reflected in this causal graphical representation. As a result, methods that only consider graph structure would incorrectly conclude that estimands such as the sample average treatment effect could be unambiguously estimated from data. Instead, the story is much more nuanced.

Assumption 2.1.7. ***Exclusion:*** *There does not exist a directed path between any instrument,* $\mathbf{x}$*, and the outcome,* $\mathbf{y}$*, that does not pass through treatment,* $\mathbf{t}$*.*

We can see an example of these two assumptions being satisfied in the graph shown in Figure 2.2[10]. However, as I discussed in Section 2.1.1, instrumental variable designs also require additional parametric assumptions. Unfortunately, these parametric assumptions cannot be expressed in terms of graph structure alone.

2.1.2.2 Regression Discontinuity Designs: A Graphical Perspective

Unlike the instrumental variable design, in which some of the design-specific assumptions could be encoded in terms of graph structure, the design-specific assumptions behind the regression discontinuity design cannot. For example, the closest graphical representation to the regression discontinuity design shown in Figure 2.3 looks nearly identical to the backdoor adjustment graph in Figure 2.1. In fact, taken naïvely the graph in Figure 2.3 should be even more straightforward to reason about, as it does not contain any possible latent confounders. Unfortunately, the parametric restrictions relating $\mathbf{t}$ and $\mathbf{x}$ tell a different story, as I discussed in Section 2.1.1. Just

[10] Here, we use $\mathbf{x}$ to denote the instrument instead of $\mathbf{z}$ to remain consistent with our SCM definition.

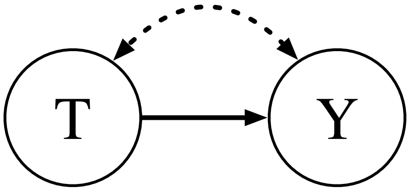


Figure 2.4: **Example structured latent confounding graph.** The assumption that any latent confounders, $\mathbf{u}$, are shared between instances of $\mathbf{t}$ and $\mathbf{y}$ that belong to the same group is not reflected in this causal graphical representation. As a result, methods that only consider graph structure would incorrectly conclude that estimands such as the sample average treatment effect could not be unambiguously estimated from data. Like the instrumental variable and the regression discontinuity examples, the truth is more nuanced.

like in the instrumental variable design, automated algorithms that operate on graphs do not apply in these common settings.

2.1.2.3 Structured Latent Confounding: A Graphical Perspective

Again, and perhaps not surprisingly, representing structured latent confounding using causal graphical models poses some conceptual challenges. Ignoring the assumption that confounders are shared between individuals belonging to the same groups, we may come to the causal graph shown in Figure 2.4. This awkwardness comes from the fact that causal graphical models, unlike probabilistic graphical models, do not contain a semantics for plates[11]. Unfortunately, this graphical model would lead us to believe that no conclusions can be drawn about the effect of $\mathbf{t}$ on $\mathbf{y}$, when in reality we may be able to leverage the structured latent confounding assumption to our advantage.

[11] In Chapter 4, we show a somewhat informal graphical representation of the structured latent confounder setup using plates. However, graph-based algorithms such as the do-calculus do not contain an explicit semantics of plates, and thus cannot use them for reasoning about identifiability or for producing adjustment formula. There does exist a partial semantics for plates in causal models, including work on d-separation in relational settings [81]. However, this work does not progress as far as the do-calculus.

2.2 Bayesian Statistics

In this section, I provide an extremely brief introduction to Bayesian statistics and probabilistic inference. For additional background, I recommend reading one of the many excellent books on the topic, including those by Gellman [44] and Murphy [85].

The key idea behind Bayesian statistics is conceptually simple, though challenging in practice: represent uncertain belief about the world in terms of a joint probability distribution over models and data, and then use axioms of probability to update that belief in light of data. Somewhat informally, let $p(\theta, \mathbf{x})$ denote a joint distribution over parameters, $\theta \in \Theta$, and data, $\mathbf{x} \in \mathbb{X}$, defined over the appropriate product measure space. It is common to decompose this joint distribution into what is known as the prior, $p(\theta)$, and the likelihood, $p(\mathbf{x}|\theta)$. Intuitively, this joint distribution represents our uncertain belief about the underlying world, and our uncertain belief about the process for generating data. Once we observe data, we can invert the conditioning operations as follows by using a straightforward application of Bayes' theorem:

$$p(\theta|\mathbf{x}) = \frac{p(\mathbf{x}|\theta)p(\theta)}{p(\mathbf{x})} = \frac{p(\mathbf{x}|\theta)p(\theta)}{\int p(\mathbf{x}|\theta)p(\theta)d\theta} \tag{2.2}$$

On the left we have what we want, the conditional probability over parameters given data, a quantity known as the posterior distribution. On the right however, we are faced with a challenge; the integral $\int_{\theta\in\Theta} p(\mathbf{x}|\theta)p(\theta)d\theta$, also known as the data marginal, is often intractable to compute exactly, especially if θ is high dimensional or continuous[12]. In settings where the posterior distribution is analytically intractable, we are forced to turn to approximate methods. In Section 2.2.1, I outline one of many templates for approximating this posterior using a technique known as importance

[12] There is a special class of probabilistic models known as conjugate models that lend themselves to a closed-form analytic solution for the posterior distributions. Not only is the posterior tractable for such models, it is also of the same distributional family of the prior. For example, the posterior distribution of a sequence of Bernoulli trials with a beta prior distribution over the weight parameter is itself beta distributed. Gaussian process models (see Section 2.3) are one such conjugate prior distribution over the space of smooth continuous functions.

sampling. The derivations in Section 2.2.1 are heavily informed by Kevin Murphy's excellent textbook *Machine Learning: A Probabilistic Perspective* [85].

2.2.1 Approximate Inference — Importance Sampling

In order to approximate the posterior distribution, importance sampling draws a collection of samples from some proposal distribution q, and then weights them according to the prior density, the likelihood, and the density under the proposal. The result of this procedure is a collection of samples that either: (i) approximate expectations of functions of derived random variables (including posterior densities); or (ii) can be sampled from to produce approximate samples from the posterior. In its most general form, importance sampling is described as a technique for approximating expectations of functions as follows:

$$\begin{aligned}\mathbb{E}_{x\sim p(\mathrm{x})}[f(x)] &\coloneqq \int f(x)p(x)dx && (2.3)\\ &= \int f(x)q(x)\frac{p(x)}{q(x)}dx && (2.4)\\ &\approx \frac{1}{N}\sum_{i=1}^{N} f(x_i)\frac{p(x_i)}{q(x_i)}; \quad x_i \overset{\text{iid}}{\sim} q(\mathrm{x}) && (2.5)\end{aligned}$$

Here $q(\mathrm{x})$ must be a distribution that is absolutely continuous with respect to $p(\mathrm{x})$, i.e. $p(\mathrm{x} = x) > 0 \rightarrow q(\mathrm{x} = x) > 0$ for all $x \in \mathit{support}(\mathrm{x})$. This simple derivation follows from the weak law of large numbers, which states that the empirical mean of a collection of samples drawn from some distribution is an unbiased estimate of its expectation. While this may appear to solve a different problem than the posterior inference we are interested in, the general idea can be applied to an expectation over the posterior, i.e. the case where $\theta \sim p(\theta|\mathbf{x})$, as follows:

$$
\begin{aligned}
\mathbb{E}_{\theta\sim p(\theta|\mathbf{x})}[f(\theta)] &\coloneqq \int f(\theta)p(\theta|\mathbf{x})d\theta && (2.6)\\
&= \frac{1}{p(\mathbf{x})}\int f(\theta)p(\mathbf{x}|\theta)p(\theta)d\theta && (2.7)\\
&= \frac{\int f(\theta)p(\mathbf{x}|\theta)p(\theta)d\theta}{\int p(\mathbf{x}|\theta)p(\theta)d\theta} && (2.8)\\
&\approx \frac{\sum_{i=1}^{N} f(\theta_i)p(\mathbf{x}|\theta_i)p(\theta_i)/q(\theta_i)}{\sum_{i=1}^{N} p(\mathbf{x}|\theta_i)p(\theta_i)/q(\theta_i)}; \quad \theta_i \overset{\text{iid}}{\sim} q(\theta) && (2.9)
\end{aligned}
$$

An interesting special case is when f is given by the dirac-Delta function at a point θ', i.e. $f(\theta) = \delta(\theta - \theta')$, where $\delta(x) = 0$ if $x \neq 0$ and $\int_{\mathbb{X}} \delta(x)dx = 1$ if $0 \in \mathbb{X}$. This special case corresponds to pointwise posterior density evaluation at θ', and results in the following simplification:

$$
\begin{aligned}
\mathbb{E}_{\theta\sim p(\theta|\mathbf{x})}[\delta(\theta-\theta')] = p(\theta'|x) &= \frac{p(x|\theta')p(\theta')}{\int p(x|\theta)p(\theta)d\theta} && (2.10)\\
&\approx \frac{p(x|\theta')p(\theta')}{\sum_{i=1}^{N} p(x|\theta_i)p(\theta_i)/q(\theta_i)}; \quad \theta_i \overset{\text{iid}}{\sim} q(\theta) && (2.11)
\end{aligned}
$$

To sample from the posterior, rather just compute its density, we can use a technique called *sampling importance resampling* (SIR) [113], which generates approximate samples from $p(\theta|\mathbf{x})$ by sampling with replacement from the weighted collection of samples from the proposal $q(\theta)$. Given a collection of such samples $\theta_i \overset{\text{iid}}{\sim} q(\theta)$, the weight for each is given by the following:

$$
w_i = \frac{p(\mathbf{x}|\theta_i)p(\theta_i)/q(\theta_i)}{\sum_{i=1}^{N} p(\mathbf{x}|\theta_i)p(\theta_i)/q(\theta_i)} \qquad (2.12)
$$

.

While this approximation is unbiased for all $q(\theta)$ that are absolutely continuous with respect to $p(\theta)$, and thus $p(\theta|x)$, the choice of $q(\theta)$ can have a significant impact on the variance of the estimator. Intuitively, to minimize variance we would like $q(\theta)$ to be close as possible to $p(\theta|x)$. As an example, which I do not use in this thesis, recent work on *amortized importance sampling* uses a neural network to approximate a data-dependent proposal distribution $q(\theta; \mathbf{x})$ trained on synthetic data from $p(\theta, \mathbf{x})$ [72].

In the remainder of this thesis, I use a variety of approximate inference techniques that on the surface may appear to be quite distinct from the importance sampling method presented here. For example, in Chapter 5 I use Sequential Monte Carlo [35], and in Chapter 4 I use a combination of Elliptical Slice Sampling [86] and Random Walk Metropolis Hastings [53]. Looking at their implementations alone, the only attribute these methods seem to share with importance sampling is that they approximate expectations with samples. In fact, it is possible to show that all of these sampling-based algorithms can be expressed as importance sampling in an augmented state-space with the inclusion of carefully selected auxiliary variables [41]. This unification is not unique to sampling-based methods, and recent work has made substantial progress organizing and unifying the previously disparate collection of variational, sampling-based, and other classes of methods for (approximate) probabilistic inference [5, 27, 33, 75, 89]. As I discuss in Chapter 3, reducing causal inference to probabilistic inference means that we can seamlessly bring to bear all of the impressive advancements in probabilistic modeling and inference technology for causal inference problems.

2.3 Gaussian Processes

Gaussian processes are a flexible technique for probabilistic modeling. Again, somewhat informally, Gaussian processes are distributions over deterministic functions, $\mathrm{y}_i = f(\mathrm{x}_i) = \phi(\mathrm{x}_i)^\top \theta$, $\theta \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\phi : \mathbb{X} \to \mathbb{R}^d$ is some basis function, $\boldsymbol{\mu}$ is a length d mean vector and $\boldsymbol{\Sigma}$ is a $d \times d$ symmetric positive definite covariance matrix. Perhaps surprisingly, the distribution $p(\mathbf{y}|\mathbf{x})$ of a collection of outputs $\mathbf{y} = [\mathrm{y}_1, \ldots, \mathrm{y}_n]$ conditional on a vector of inputs $\mathbf{x} = [x_1, \ldots, x_n]$ can be tractably evaluated even in the limit as $d \to \infty$ by using the *kernel trick* [103]. In these settings, a Gaussian process can instead be summarized by a mean function, $m : \mathbb{X} \to \mathbb{R}$ and covariance function, $k : \mathbb{X} \times \mathbb{X} \to \mathbb{R}$, which I refer to as the kernel function [103]. By definition, any finite collection of draws from a Gaussian process prior are jointly Gaussian distributed, $\mathbf{y} \sim \mathcal{N}(\boldsymbol{m}, \boldsymbol{K})$, where $m_i = m(x_i)$ and $K_{i,j} = k(x_i, x_j)$ for all $i, j \in [\![n]\!]$. It is common to set the prior mean function to $m(x_i) = 0$, which I do throughout this thesis.[13] Intuitively, the kernel function defines a notion of similarity between inputs, and the Gaussian process model simply states that instances with similar inputs (i.e. high kernel function) typically have similar outputs (high covariance).

This identity is useful for two reasons: (i) it provides an explicit likelihood, which can be used to perform inference over latent variables [71, 127]; and (ii) it enables closed-form out-of-sample probabilistic prediction [103]. I take advantage of both of these characteristics in this thesis, performing approximate inference over latent confounders and sampling counterfactual outcomes in Chapter 4.

To sample from the posterior, we first extend the joint distribution to include both observed outcomes $\mathbf{y}$ and unobserved outcomes $\mathbf{y}^*$ as follows, where $\mathbf{x}^*$ is the collection of inputs corresponding to $\mathbf{y}^*$:

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{y}^* \end{bmatrix} | \mathbf{x}, \mathbf{x}^* \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \boldsymbol{K} & \boldsymbol{K}^{*\top} \\ \boldsymbol{K}^* & \boldsymbol{K}^{**} \end{bmatrix}\right) \tag{2.13}$$

Here, for a collection of m unobserved instances, $\boldsymbol{K}^*$ is the $n \times m$ matrix of kernel functions applied to each combination of observed and unobserved instances' inputs. Similarly, $\boldsymbol{K}^{**}$ is the $m \times m$ matrix of kernel functions applied to each combination of

[13] A vector $\mathbf{y}$ sampled from a Gaussian process should not be confused with a vector $\mathbf{y}$ with each element sampled i.i.d from a Gaussian distribution. Gaussian processes sample all elements of $\mathbf{y}$ jointly from a multivariate Gaussian, where the "similarity" between elements of $\mathbf{x}$ determines the covariance between elements of $\mathbf{y}$.

unobserved instances' inputs. As the joint distribution over $\mathbf{y}$ and $\mathbf{y}^*$ is Gaussian, the conditional distribution $p(\mathbf{y}^*|\mathbf{y}, \mathbf{x}, \mathbf{x}^*)$ is also Gaussian, and is given by the following:

$$\mathbf{y}^*|\mathbf{y}, \mathbf{x}, \mathbf{x}^* \sim \mathcal{N}(\boldsymbol{K}^*\boldsymbol{K}^{-1}\mathbf{y}, \boldsymbol{K}^{**} - \boldsymbol{K}^*\boldsymbol{K}^{-1}\boldsymbol{K}^{*\top}) \tag{2.14}$$

This formulation describes a noise-free Gaussian process model. To model additive Gaussian noise on the observed outputs, simply replace $\boldsymbol{K}$ with $\boldsymbol{K} + \boldsymbol{I}_n\sigma^2$, where σ^2 is the noise variance. I assume additive Gaussian noise throughout this thesis, and leave extensions with non-Gaussian noise [121] as future work.

2.3.1 Kernel Specification

Intuitively, the kernel function defines what it means for two inputs x_i and x_j to be similar, and thus induce similar outcomes y_i and y_j. By choosing a particular kernel, or even a particular prior over kernels, we implicitly place an inductive bias over the choice of functions we would expect to see *a priori*[14]. In this thesis, I exclusively use finite-dimensional Gaussian processes, or infinite-dimensional Gaussian processes with (variants of) automatic relevance determination (ARD) kernels for Gaussian process models, which are a generalization of the commonly used squared exponential kernel. For $x_i, x_j \in \mathbb{R}^d$, the ARD kernel is given as follows:

$$k(x_i, x_j) = s \cdot \exp\left[-\sum_{k=1}^{d} \frac{(x_{i,k} - x_{j,k})^2}{\lambda_k}\right] \tag{2.15}$$

Here, $s \in \mathbb{R}^+$ and $\lambda_k \in \mathbb{R}^+$ for all $k \in [\![d]\!]$ are hyperparameters that dictate the typical *shape* of functions drawn for the Gaussian process prior, and are known as the scale and lengthscale respectively. We can represent broader uncertainty over

[14] For this reason, Gaussian process models are best described as semi-parametric estimators (rather than fully non-parametric estimators) when applied to causal inference problems. This terminology aims to avoid confusion when discussing non-parametric structural equation models, i.e. causal graphs, which place no inductive bias on structural functions besides their collection of arguments.

Gaussian process models by additionally placing priors and subsequently performing inference over these respective hyperparameters, which I do throughout this thesis.

2.4 Probabilistic Programming

Probabilistic programming is an emerging sub-field of computer science combining insights from probability theory, generative machine learning, and programming languages. Simply put, the trace-based class of probabilistic programming languages (PPLs) that I discuss in this thesis extend imperative programming languages with declarative programming interfaces for specifying probabilistic models, and backend support for sampling, scoring, and storing random choices made during execution in special *trace* data structures[15]. In addition, PPLs provide (partial) automation for manipulating these traces of random choices, allowing developers and users to implement probabilistic inference algorithms.

There are two primary reasons why one may wish to use a probabilistic programming language: (i) they provide an expressive substrate for writing probabilistic models that extend beyond Bayesian networks, leveraging programming constructs such as control-flow, looping, and recursion; and (ii) they provide a software engineering foundation for probabilistic modeling and inference, enabling the development of software artifacts that are modular, extensible, and reusable[16]. To understand a bit more about how probabilistic programming brings software engineering to probabilistic modeling and

[15] It is challenging to give a formal account of PPLs generally in a way that is language-agnostic, as the mapping to familiar concepts in probability and measure theory can depend heavily on the syntax and semantics of the language. In Gen, the language I use throughout this thesis, programs define joint distributions over dictionary-like traces which may vary in length, but must be finite length with probability 1. For additional theoretical detail, see Marco Cusamano-Towner's PhD thesis [29].

[16] While I do not explore the implications of language expressiveness for causal models with rigorous PL formalisms in this thesis, preliminary results from early work in my graduate studies [134] indicate that control flow can have important implications for causal inference. Recent work by other researchers has also established expressivity of causal models as an important area of study [60, 124]

inference (and, by extension, to causal inference in the Bayesian structural approach), it is helpful to dig deeper into the design of Gen [28], the probabilistic programming language I use throughout this thesis.

2.4.1 Gen and the Generative Function Interface

For a casual user, Gen [28] is a general-purpose probabilistic programming language in which one can write and reason about probabilistic models as code. Leveraging Julia's just-in-time compiler and fast array processing [15], the default implementation of Gen, `Gen.jl`, provides fast probabilistic inference that is practical for some near real-time applications in robotics and computer vision settings [49]. While these accomplishments are impressive and should be celebrated, computational performance is often not the key bottleneck when using programming languages, as evidenced by the remarkable adoption of high-level interpreted programming languages such as Python. Instead, programming languages should be performant and ergonomic, allowing programmers to quickly iterate, extend, and validate their (probabilistic) code.

It is certainly misleading to describe Gen as *just* a fast implementation for probabilistic modeling and inference. Instead, and perhaps more significantly, Gen defines the *generative function interface* (GFI). As the name implies, the GFI is an abstract collection of methods that must satisfy pre-defined contractual properties. For example, the definition of one such interface method, `generate`, is:

> `generate` (**obtaining a trace subject to constraints**) This method takes a choice map u, arguments x, and returns: (i) a trace (t, x) such that $t \cong u$, sampled using the internal proposal distribution (denoted $t \sim q(\cdot; x, u)$; as well as (ii) a weight $w := p(t; x)/q(t; x, u)$.

As Harold Abelson and Gerald Jay Sussman describe repeatedly in their highly influential software engineering book "Structure and Interpretation of Computer Programs" [1], clean abstractions are necessary for engineering complex software systems. Applying this principle to generative models, abstractions like the GFI are central to engineering complex probabilistic modeling and reasoning systems. To elaborate on this point a bit further, three tangible benefits of such an interface are the following:

Model Composition. Gen and related systems enable composition of generative models in two ways. First, GFI methods such as `generate` shown above are defined recursively for generative functions that themselves call other generative functions. As we will see in Chapter 3, this allows programmers to define model components as distinct blocks of code, and call them in the body of other models the same way one would write and compose ordinary functions in an imperative programming language. Second, Gen provides a set of closed program transformations called combinators. Here, closure means that a combinator applied to a generative function that satisfies the GFI returns a generative function that also satisfies the GFI automatically. For example, applying the `map` combinator to a generative function `f` defining a distribution over a single data instance returns a generative function `g` defining a distribution over an array of independent and identically distributed draws from `f`[17]. As we'll see in Chapter 3, interventions are closely related to Gen's combinators, although they require more introspection into the structural causal model than just the GFI can provide.

Bayesian Workflow. For most applications, generative modeling is still very much a human-in-the-loop endeavor. In practice, it is difficult to encode assumptions that directly reflect our beliefs or to choose inference algorithms that traverse the posterior

[17] Other probabilistic programming languages include similar automated transformations of generative models for concise composition, albeit implemented somewhat differently. For example, `plate` constructs in Pyro [16] broadcast traced random choices along new tensor dimensions. It is worth noting however that in addition to being implemented using effect handlers instead of function combinators, Pyro's plate semantics differ somewhat from Gen's Map combinator in that they only assert conditional independence given all ancestors in the program.

landscape effectively on first attempt. As a result, many researchers have explored a practical set of techniques known as the "Bayesian Workflow" [46]. In essence, the Bayesian workflow is a guide for introspecting, testing, and selecting models in practice, using Bayesian methodologies and tooling. Many of the techniques used as a part of the Bayesian workflow, such as simulation-based calibration [123] and its variants [140], are model-agnostic by design, only requiring that models and approximate inference algorithms support (posterior) sampling and/or density evaluation. Therefore, rather than implementing components of the Bayesian workflow from scratch for each model, they can instead be implemented agnostically using GFI methods. Then, any model written in Gen or related systems automatically enables the Bayesian workflow for free. In Chapter 6, I discuss SBI, a supplement to the Bayesian workflow for identifiability questions that are common to causal inference problems.

Programmable Inference. Of the many probabilistic programming languages that have emerged in the scientific literature, the vast majority provide automated *universal inference* algorithms for conditioning and marginalizing joint distributions. When using these systems, a programmer is only responsible for encoding their knowledge in the form of a probabilistic program, i.e. a joint distribution, and is not responsible for designing an inference algorithm. For example, Church [48] automates single-site Metropolis Hastings [53], which produces asymptotically consistent samples from the posterior for any program written in its Turing complete language. While asymptotically correct, this general-purpose algorithm often fails to converge in realistic settings with finite compute resources. Stan [24], on the other end of the spectrum, automates variations of Hamiltonian Monte Carlo (HMC) [55, 88] for probabilistic programs that are isomorphic to graphical models, and contain only continuous latent variables. HMC is often fast, but does not permit inference over discrete latent variables, such as those necessary for representing random adjacency matrices of directed acyclic graphs.

Rather than fully automating probabilistic inference, some recent languages provide programmable high-level interfaces for programmers to implement custom inference algorithms [83]. Inference algorithms written in Gen [28] are implemented only in terms of GFI methods, abstracting away details of the individual model or modeling language. In Gen, users are free to use off-the-shelf inference algorithms in the standard library, or to write custom inference programs themselves that directly interact with GFI methods. For example, importance sampling (see Section 2.2.1) requires sampling and density evaluation, but does not require introspection into how those methods are implemented. Therefore, a programmer can implement a custom inference method that will still be valid for any model that implements the GFI, including but not limited to any model written in Gen.jl. However, that same user may want to write a custom algorithm that alternates between importance sampling with a custom neural network proposal, and random walk Metropolis Hastings over disjoint blocks of latent variables. Gen's inference programming capabilities are particularly well suited for pseudo-marginal Monte Carlo methods [6], in which approximate inference methods are composed to perform inference jointly on all latent variables. Pyro [16] similarly allows users to design custom variational families for stochastic variational inference and custom proposal families for importance sampling.

CHAPTER 3

BAYESIAN STRUCTURAL CAUSAL INFERENCE

In this chapter, I discuss the foundations of the Bayesian structural approach to causal inference, in which uncertainty over causal mechanisms is explicitly represented as prior distributions over the probabilistic structural causal models discussed in Section 2.1.1. Importantly, the framework presented in this chapter does not claim that probability theory subsumes causal inference [95], only that with care it can act as an alternative to other means of partial specification. In fact, this prior serves a conceptually similar role to a causal graph, representing uncertain belief about the space of SCMs *a priori* [12].

Unlike subsequent Chapters 4, 5, and 6, which present new methods with empirical comparisons and/or theoretical proofs, this chapter lays out a general-purpose conceptual framework for representing and reasoning about causal inference problems. As a result, its novelty is a bit more subtle. In particular, this chapter focuses on the novel claim that Bayesian structural causal inference implemented with probabilistic programs is an expressive, modular, and extensible framework for representing and reasoning about the critical assumptions that enable effective and accurate causal inference, and that this framework easily generalizes to unanticipated inference scenario. I provide evidence for this novel claim by representing common designs as probabilistic programs, and analyzing (some of) their implications. In Section 3.4, I elaborate on comparisons to existing literature.

3.1 Overview

Put succinctly, a Bayesian structural causal model is defined as follows:

Definition 3.1.1. ***Bayesian structural causal model.*** *A Bayesian structural causal model, $p(\mathbb{M}_p)$, is a distribution over probabilistic structural causal models, $\mathbb{M}_p \in \mathcal{M}_p$, i.e. its density function $p : \mathcal{M}_p \to \mathbb{R}^+$ satisfies the usual axioms of probability. Namely, that $1 = \int_{\mathcal{M}_p} dp(\mathbb{M}_p)$.*

While this definition is straightforward, some care must be taken for the Bayesian structural causal model to be coherent. A Bayesian SCM is a distribution over probabilistic SCMs, requiring a measure over structural functions $\mathbb{F}$ and a random measure over probability distributions $p(\mathbb{X})$ and $p(\mathbb{U})$. First, as the space of all functions $f : \mathbb{R} \to \mathbb{R}$ is not measurable [10], in this thesis I restrict my attention to distributions over structural functions that are fully specified by a collection of parameters defined on measurable domains, $f(\cdot; \theta_f)$ where $\theta_f \in \mathbb{R}^{d_f}$, with a corresponding prior, $p(\theta_f)$. While Gaussian processes implicitly may contain an infinite collection of parameters ($d_f \to \infty$), they can be thought of as operating on a Cartesian product of Lebesgue measures for finite n[103]. Similarly, I restrict my attention to distributions over exogenous noise and confounders that are fully specified by a collection of measurable parameters, $p(\mathbb{X}; \theta_x)$ and $p(\mathbb{U}; \theta_u)$ where $\theta_x \in \mathbb{R}^{d_x}$ and $\theta_u \in \mathbb{R}^{d_u}$, and with corresponding priors, $p(\theta_x)$ and $p(\theta_u)$. While these parameterizations might at first appear to be restrictive for practical use, this template covers a broad range of models, from linear models to Bayesian neural networks [87]. As an example, a user can express assumptions using Gaussian process priors [103], which I show in Chapter 4. Even so, relaxing these restrictions, and opening the door to a broader class of Bayesian nonparametric priors over structural causal models is an exciting area of future work.

In Section 2.2, I discussed how a hierarchical Bayesian model composes a prior distribution over parameters, $p(\theta)$, and a likelihood for data given parameters, $p(\mathbb{X}|\theta)$, to fully specify a unique joint distribution, $p(\mathbb{X}, \theta)$, and by extension a marginal distribution over data, $p(\mathbb{X})$, or a posterior distribution of parameters conditional on data, $p(\theta|\mathbb{X})$. Similarly, a Bayesian structural causal model, $p(\mathbb{M}_p)$, provides all of the necessary ingredients for the marginal and conditional quantities we are interested in regarding factual and counterfactual quantities. Using the conditional distribution $p(\mathbb{V}, \mathbb{V}(\mathbf{t}_1), ..., \mathbb{V}(\mathbf{t}_n)|\mathbb{M}_p)$ induced by pushing forward randomness from $p(\mathbb{V})$ and $p(\mathbb{X})$ through $\mathbb{F}$ as described in Section 2.1, we have the following joint distribution:

$$p(\mathbb{V}, \mathbb{V}(\boldsymbol{t}_1), ..., \mathbb{V}(\boldsymbol{t}_k)) = \int_{\mathcal{M}_p} p(\mathbb{V}, \mathbb{V}(\boldsymbol{t}_1), ..., \mathbb{V}(\boldsymbol{t}_k)|\mathbb{M}_p) dp(\mathbb{M}_p) \tag{3.1}$$

and the conditional distribution:

$$p(\mathbb{V}(\boldsymbol{t}_1), ..., \mathbb{V}(\boldsymbol{t}_k)|\mathbb{V}) = \frac{p(\mathbb{V}, \mathbb{V}(\boldsymbol{t}_1), ..., \mathbb{V}(\boldsymbol{t}_k))}{p(\mathbb{V})} \tag{3.2}$$

Practitioners are often not interested in counterfactual outcomes directly, and instead are interested in some causal query, $Q(\mathbb{V}, \mathbb{V}(\boldsymbol{t}_1), ..., \mathbb{V}(\boldsymbol{t}_k))$, such as the sample average treatment effect, $Q = \sum_{i=1}^{n}(y_i(t) - y_i)/n$. Finally, we have that the conditional distribution over counterfactual outcomes $p(\mathbb{V}(\boldsymbol{t}_1), ..., \mathbb{V}(\boldsymbol{t}_k)|\mathbb{V})$ and the causal query Q induce a pushforward distribution over causal effect, $p(Q|\mathbb{V})$, as follows, where $\mathcal{M}_Q$ is the subset of all probabilistic SCMs in $\mathcal{M}_p$ that induce a causal effect Q:

$$p(Q|\mathbb{V}) = \frac{1}{p(\mathbb{V})} \int_{\mathcal{M}_Q} p(\mathbb{V}, \mathbb{V}(\boldsymbol{t}_1), ..., \mathbb{V}(\boldsymbol{t}_k)|\mathbb{M}_p) dp(\mathbb{M}_p) \tag{3.3}$$

Remarkably, in just a few equations we have represented the essence of the causal inference problem in purely probabilistic terms, the conditional distribution of answers to our query, Q, given observational data, $\mathbb{V}$. In other words, we have managed to reduce the problem of causal inference to one of purely probabilistic inference, albeit over functions of counterfactual variables[1]. While this reduction does not fully

[1] Importantly, this reduction is only possible if given a prior over structural causal models, representing our untestable assumptions about causal relationships.

solve our original causal inference problem, as we've reduced a hard causal inference problem to a hard probabilistic inference problem, it does allow us to bring to bear all of the impressive probabilistic modeling and inference technology to yield (often approximate) inferences, including probabilistic programming languages like Gen [28].

3.2 Linear Example

To see how the Bayesian approach works a bit more intuitively, let us consider an example where the space $\mathcal{M}_p$ contains all linear structural causal models between three observed (or endogenous) random variables $\mathbb{V} = \{\mathbf{t}, \mathbf{x}, \mathbf{y}\}$, representing *treatment*, *covariates*, and *outcome* respectively. For now we'll make the following simplifying assumptions:

- There exists a single latent confounder $\mathbb{U} = \{\mathbf{u}\}$ that may influence all three observed variables in $\mathbb{V}$.
- $\mathbf{x}$ can influence $\mathbf{t}$, and both $\mathbf{x}$ and $\mathbf{t}$ can influence $\mathbf{y}$.
- Exogenous noise is additive. (e.g. $\mathbf{y} = f_y(\mathbf{t}, \mathbf{x}, \mathbf{u}, \epsilon_y) = f'_y(\mathbf{t}, \mathbf{x}, \mathbf{u}) + \epsilon_y$).
- Exogenous noise and latent confounders are mean-zero Gaussian distributed.
- Each instance depends only on the corresponding instance of its potential causes. (e.g. $\mathrm{t}_i = f_t(\mathrm{x}_i, \mathrm{u}_i, \epsilon_{t_i})$)

This simple scenario can be succinctly described by the appropriately simple family of structural causal models as follows, corresponding to Definition 2.1.1, and parameterized by linear weights β:

```julia
@gen function linear_scm(noise::Dict{Symbol, Array{Float64, 1}},
                         confounders::Dict{Symbol, Array{Float64, 1}},
                         parameters::Dict{Symbol, Float64})

    u = confounders[:u]
    x = parameters[:beta_ux] * u + noise[:x]
    t = parameters[:beta_xt] * x + parameters[:beta_ut] * u + noise[:t]
    y = parameters[:beta_ty] * t + parameters[:beta_xy] * x + parameters[:beta_uy] * u + noise[:y]

    return data = Dict(:x => x, :t => t, :y => y)
end
```

Figure 3.1: **Implementation of the linear structural causal model in Gen.** This implementation is equivalent to the mathematical description of the structural causal model (see Definition 2.1.1) given in Equations 3.4.

$$
\begin{aligned}
\mathrm{x}_i &= \beta_{ux}\mathrm{u}_i + \epsilon_{\mathrm{x}_i} \\
\mathrm{t}_i &= \beta_{xt}\mathrm{x}_i + \beta_{ut}\mathrm{u}_i + \epsilon_{\mathrm{t}_i} \\
\mathrm{y}_i &= \beta_{ty}\mathrm{t}_i + \beta_{xy}\mathrm{x}_i + \beta_{uy}\mathrm{u}_i + \epsilon_{\mathrm{y}_i}
\end{aligned}
\tag{3.4}
$$

Using a probabilistic programming language such as Gen, we can translate this simple model to code, as shown in Figure 3.1. As discussed in Section 2.4, this Gen program supports all of the operations we would expect on probability distributions, including the ability to draw samples and to evaluate densities (and often gradients) pointwise. In that sense, the mathematical description in Equations 3.4 and the code description in Figure 3.1 are equivalent, and not *merely* an implementation detail for representing mathematical concepts.[2] Using this linear structural causal model, we can place distributions on exogenous noise variables and the latent confounder to construct a linear probabilistic structural causal model corresponding to Definition 2.1.2. In standard mathematical notation these distributions are defined as follows:

[2] In Chapter 5 I show why this framing is conceptually useful, allowing us to write Gen programs that describe distributions over causal programs themselves, a procedure known as Bayesian program synthesis [114].

$$\epsilon_{\mathrm{x}_i} \sim \mathcal{N}(0, \sigma_{\mathrm{x}}^2) \quad \epsilon_{\mathrm{t}_i} \sim \mathcal{N}(0, \sigma_{\mathrm{t}}^2) \quad \epsilon_{\mathrm{y}_i} \sim \mathcal{N}(0, \sigma_{\mathrm{y}}^2) \quad \mathrm{u}_i \sim \mathcal{N}(0, \sigma_{\mathrm{u}}^2) \tag{3.5}$$

Again, we can write the probabilistic structural causal model as Gen code, as can be seen in `linear_probabilistic_scm` in Figure 3.2, this time calling our earlier implementation of `linear_scm` in Figure 3.1 without needing to re-implement the SCM from scratch. As we'll see throughout this chapter, the ability to compose probabilistic programs as ordinary function calls provides an ergonomic and straightforward means of implementing complex and extensible models.

Figure 3.3 shows a visual representation of the conditional distribution induced by `linear_probabilistic_scm` if we knew the probabilistic structural causal model exactly, i.e. if we called `linear_probabilistic_scm` with a single collection of arguments[3]. Up until this point, all uncertainty comes from the exogenous noise, or individual-level variation in how data comes to be generated. In practice however, it is almost never reasonable to assume a single structural causal model exactly. Instead, we would like to represent uncertainty over the structural functions themselves as well.

3.2.1 Hierarchical Bayesian Extension

To represent broader uncertainty over probabilistic structural causal models, we can borrow insight from the broader practice of Bayesian statistics and represent our uncertain belief in terms of a prior probability distribution. To do this we create a new Gen program, `linear_bayesian_scm` in Figure 3.4, by: (i) generating parameters

[3] The figures in this chapter are inspired by Figures 1 and 3 in the abstract formalism presented in Oliver Maclaren and Ruanui Nicholson's evocative paper describing how inverse problems in engineering relate to causal inference [80].

```julia
@gen function noise_model(parameters::Dict{Symbol, Float64}, n::Int64)

    eps_x, eps_t, eps_y = [], [], []

    for i in 1:n
        push!(eps_x, @trace(Normal(0, parameters[:var_x]), :eps_x => i))
        push!(eps_t, @trace(Normal(0, parameters[:var_t]), :eps_t => i))
        push!(eps_y, @trace(Normal(0, parameters[:var_y]), :eps_y => i))
    end

    return noise = Dict(:x => eps_x, :t => eps_t, :y => eps_y)
end
```

```julia
@gen function confounder_model(parameters::Dict{Symbol, Float64}, n::Int64)
    u = []

    for i in 1:n
        push!(u, @trace(Normal(0, parameters[:var_u]), :u => i))
    end

    return confounders = Dict(:u => u)
end
```

```julia
@gen function linear_probabilistic_scm(parameters::Dict{Symbol, Float64}, n::Int64)

    noise = @trace(noise_model(parameters, n))

    confounders = @trace(confounder_model(parameters, n))

    return data = @trace(linear_scm(noise, confounders, parameters))
end
```

Figure 3.2: **Implementation of the linear probabilistic structural causal model in Gen.** This implementation is equivalent to the mathematical description of a probabilistic SCM (see Definition 2.1.1) with distributions over exogenous noise and confounders given by Equations 3.5 and the SCM given by Equations 3.4. Importantly, Gen's generative function interface is closed under composition, allowing us to implement models like those shown here by calling the inner Gen function as if it were ordinary code.

from some prior distribution; and then (ii) calling `linear_probabilistic_scm` using those parameters.[4]

Unlike `linear_probabilistic_scm`, which takes as input a collection of dictionaries containing numeric values for `means`, `variances`, and `weights`, our implementation for `linear_bayesian_scm` in Figure 3.4 instead takes as input a collection of dictio-

[4] Note that while the following code is valid Julia and Gen code, it is written to optimize clarity, not computational performance. For example, in practice we would replace the explicit loop with a call to Gen's Map combinator to denote conditional independence. Gen's internal representation (IR) automatically leverages this information to reduce likelihood computations, etc. See the Gen paper [28] for insight into achieving performance in Gen.

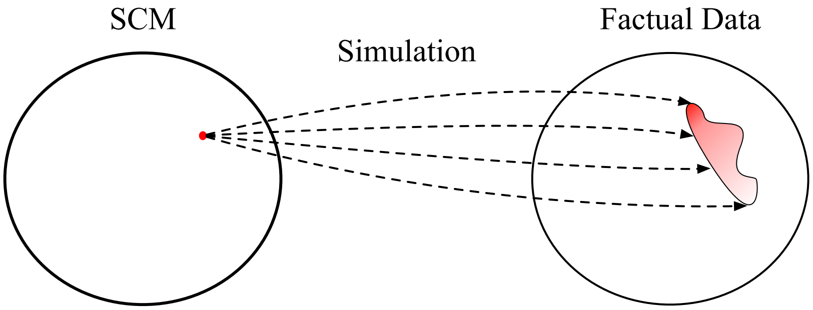


Figure 3.3: **Visual representation of sampling from a single probabilistic structural causal model.** Here, uncertainty over simulated data comes exclusively from exogenous noise and latent confounders, and not uncertainty over structural functions. In this and subsequent illustrative figures I show induced distributions as having crisp boundaries for aesthetic purposes. In practice, the support of induced distributions over factual (and later counterfactual) data typically does not depend on the particular probabilistic SCM.

```
 1 @gen function linear_bayesian_scm(parameters_prior::Dict{Symbol, Distribution}, n::Int64)
 2 
 3     parameters = Dict()
 4 
 5     for (variable, prior) in parameters_prior
 6         parameters[variable] = @trace(prior, variable)
 7     end
 8 
 9     return data = @trace(linear_probabilistic_scm(variances, weights, n))
10 end
```

Figure 3.4: **Implementation of the linear Bayesian structural causal model in Gen.** This implementation is equivalent to the mathematical description of a Bayesian SCM (see Definition 3.1.1) with user-specified priors over parameters in the probabilistic SCM defined by Equations 3.4 and 3.5. Again, implementing this extension is straightforward, as calling generative functions is equivalent to composing probabilistic models.

naries containing probability distributions, representing our priors. Connecting the two is remarkably straightforward: simply sample numeric values from each prior (lines 5-7), and then use those sampled values to generate a collection of instances from our already defined `linear_probabilistic_scm` (line 9). Even in this simple example, we are seeing some of the benefits of using a probabilistic programming languages in how models freely compose.

Figure 3.5 shows a visual representation of this new Bayesian extension of our original linear structural causal model. Intuitively, adding uncertainty to the space

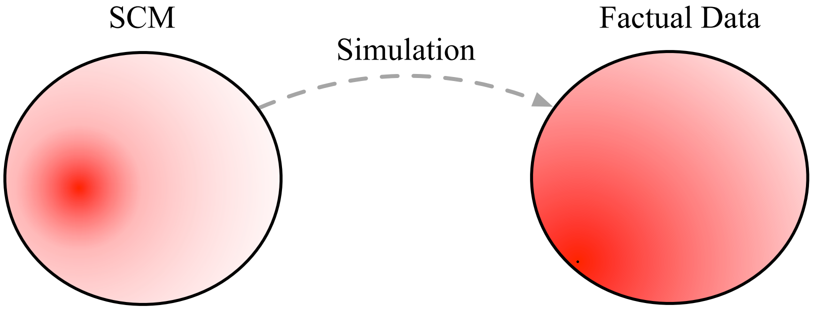


Figure 3.5: **Visual representation of the joint distribution of probabilistic structural causal models and data.** Adding a prior distribution to the space of structural causal models in our Bayesian variant broadens the resulting marginal distribution over data. Here, and in the remainder of this chapter, darker colors represent higher density.

of probabilistic SCMs increases our resulting uncertainty in the factual data we may observe.

3.2.2 Posterior Inference

Not only does this Bayesian extension allow us to represent broader uncertainty over probabilistic SCMs, which is then propagated to data, we can also use the `linear_bayesian_scm` in combination with data to yield a posterior distribution over probabilistic SCMs. As a simple example, we could run Gen's implementation of sampling importance resampling [113] (see Section 2.2.1) to generate a single sample from the posterior over probabilistic SCMs[5]. Figure 3.6 shows a visual representation of the posterior distribution over probabilistic SCMs after conditioning on data, $\mathbb{V}$. As we'll see later in more depth, Figure 3.6 is already alluding to a problem we often encounter in causal inference problems; multiple probabilistic structural causal models may induce the same distribution over factual data. Therefore, after running inference

[5] As written, the code in `linear_bayesian_scm` could not yet be used to condition on $\mathbb{V}$, as the assignment for `x`, `t`, and `y` in `linear_scm` are not traced random variables, i.e. they do not use Gen's `@trace` syntax. See Section 3.2.8 for discussion on pushing randomness from exogenous noise to endogenous random variables.

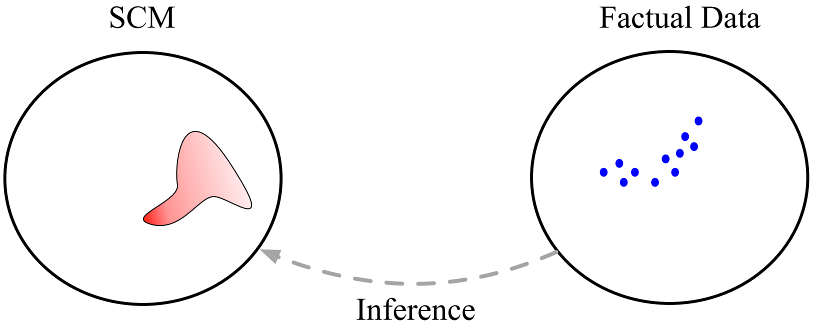


Figure 3.6: **Visual representation of the posterior distribution of probabilistic structural causal models given data.** Given a Bayesian SCM and observations, $\mathbb{V}$, we can use an (approximate) inference algorithm to infer likely probabilistic SCMs that could have generated the data.

we will not be able to distinguish between them, and the conclusions they imply about our causal query of interest.

To see this concretely in our linear Gaussian example, we begin by analyzing the joint density of data, $\mathbb{V}$, conditional on model parameters, θ, marginalizing out the latent confounders, $\mathbb{U}$. As we'll see later, this joint density is all we'll need to conclude whether data can yield unique causal conclusions[6]. As all of our variables are Gaussian and all functions between them are linear, the joint density conditional on data (i.e. the data likelihood) is also Gaussian, with the following covariance matrix[7]:

$$[\mathbf{u}, \mathbf{x}, \mathbf{t}, \mathbf{y}]^\top | \theta \sim \mathcal{N}(0, \Sigma) \tag{3.6}$$

with covariance terms given by the matrix form of our linear structural causal model,

[6] Here we show a closed form expression for the data likelihood, and not the full posterior. If we assume conjugate priors we could also yield a closed form expression for the posterior, however that will not be necessary to make the expository argument. In Chapter 6 I use a similar analysis to develop a general approach to determining if causal queries are identifiable given data.

[7] The analysis here is a light elaboration on similar frequentist-style analysis by Bollen [17] and D'Amour [30].

$$
\begin{aligned}
\boldsymbol{W} &= \boldsymbol{A} \cdot \boldsymbol{B} \cdot \boldsymbol{C} \cdot \boldsymbol{D} \\
\Sigma &= \boldsymbol{W} \cdot \boldsymbol{W}^{\top}
\end{aligned}
\tag{3.7}
$$

$$
\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \beta_{uy} & \beta_{xy} & \beta_{ty} & \sigma_y^2 \end{bmatrix} \quad \boldsymbol{B} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \beta_{ut} & \beta_{xt} & \sigma_t^2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
\boldsymbol{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \beta_{ux} & \sigma_x^2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \boldsymbol{D} = \begin{bmatrix} \sigma_u^2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

As the joint distribution on $[\mathbf{u}, \mathbf{x}, \mathbf{t}, \mathbf{y}]$ is Gaussian, the marginal distribution of $[\mathbf{x}, \mathbf{t}, \mathbf{y}]$ is also Gaussian with covariance given by the corresponding lower right 3×3 block component of Σ. As a result, the six combinations of covariances (i.e. unique elements of Σ) between pairs of endogenous random variables, $\mathbb{V} = \{\mathbf{x}, \mathbf{t}, \mathbf{y}\}$ is given by the following collection of equations:

$$
\begin{aligned}
cov(\mathbf{x},\mathbf{x}) &= \beta_{ux}^2\sigma_u^2 + \sigma_x^2 \\
cov(\mathbf{x},\mathbf{t}) &= \beta_{ux}\sigma_u^2(\beta_{ut} + \beta_{ux}\beta_{xt}) + \beta_{xt}\sigma_x^2 \\
cov(\mathbf{x},\mathbf{y}) &= \beta_{ux}\sigma_u^2(\beta_{ty}\beta_{ut} + \beta_{ux}(\beta_{ty}\beta_{xt} + \beta_{xy}) + \beta_{uy}) + \sigma_x^2(\beta_{ty}\beta_{xt} + \beta_{xy}) \\
cov(\mathbf{t},\mathbf{t}) &= \beta_{xt}^2\sigma_x^2 + \sigma_t^2 + \sigma_u^2(\beta_{ut} + \beta_{ux}\beta_{xt})^2 \\
cov(\mathbf{t},\mathbf{y}) &= \beta_{ty}\sigma_t^2 + \beta_{xt}\sigma_x^2(\beta_{ty}\beta_{xt} + \beta_{xy}) + \sigma_u^2(\beta_{ut} + \beta_{ux}\beta_{xt})(\beta_{ty}\beta_{ut} + \beta_{ux}(\beta_{ty}\beta_{xt} + \beta_{xy}) + \beta_{uy}) \\
cov(\mathbf{y},\mathbf{y}) &= \beta_{ty}^2\sigma_t^2 + \sigma_u^2(\beta_{ty}\beta_{ut} + \beta_{ux}(\beta_{ty}\beta_{xt} + \beta_{xy}) + \beta_{uy})^2 + \sigma_x^2(\beta_{ty}\beta_{xt} + \beta_{xy})^2 + \sigma_y^2
\end{aligned}
\tag{3.8}
$$

Unfortunately, this system of equations does not have a unique solution for any parameters θ in terms of the six pairwise covariance terms. In fact, there are an infinite collection of parameters in the pre-image of any given covariance matrix, Σ, forming what Alex D'Amour calls an "ignorance region" [30]. Perhaps this is not surprising, as we have six "knowns" on the left-hand side of the system of equations that we are attempting to relate to ten unknown parameters on the right-hand side. Therefore, we should never hope to recover all of the parameters from data alone. However, later I'll discuss the perhaps more surprising conclusion that we can sometimes construct a unique solution for a subset of the parameters of interest that are particularly informative for our causal questions, even in the presence of latent confounders.

But how does this anecdote of non-injectivity relate to our Bayesian story thusfar? To see the connections, note that the six covariance terms form a sufficient statistic for the joint distribution over $[\mathbf{x}, \mathbf{t}, \mathbf{y}]$. This implies that any two collections of parameters, θ_1 and θ_2, that induce the same covariance matrix over $[\mathbf{x}, \mathbf{t}, \mathbf{y}]$ will, by definition, have equal likelihood, i.e. $p(\mathbf{x}, \mathbf{t}, \mathbf{y}|\theta_1) = p(\mathbf{x}, \mathbf{t}, \mathbf{y}|\theta_2)$. By extension, the posterior odds ratio for any pairs reduces to the prior odds ratio, $p(\theta_1|\mathbf{x}, \mathbf{t}, \mathbf{y})/p(\theta_2|\mathbf{x}, \mathbf{t}, \mathbf{y}) = p(\theta_1)/p(\theta_2)$, even as $n \to \infty$. We'll call this inability to disambiguate parameters (and causal conclusions) non-identifiability. I substantially elaborate on this argument and its implications in Chapter 6.

```
function intervention(model, intervention_assignment::Array{Float64, 1})
    ...
    return intervened_model
end
```

```
@gen function linear_intervened_scm(noise::Dict{Symbol, Array{Float64, 1}},
                                    confounders::Dict{Symbol, Array{Float64, 1}},
                                    parameters::Dict{Symbol, Float64})

    u = confounders[:u]
    x = parameters[:beta_ux] * u + noise[:x]
    t = $(intervention_assignment) # [1., ..., 1.]
    y = parameters[:beta_ty] * t + parameters[:beta_xy] * x + parameters[:beta_uy] * u + noise[:y]

    return data_cf = Dict(:x => x, :t => t, :y => y)
end
```

Figure 3.7: **Mock implementation of an intervention program transformation in Gen.** This (mock) implementation for `intervention` corresponds to Definition 2.1.3, taking as input a `model` and an `intervention_assignment` and returning an `intervened_model`. `linear_intervened_scm` shows the equivalent Gen program to the return value of `intervention(linear_scm, ones(n))`.

3.2.3 Intervention Program Transformations

Thus far, I have showed how to compose generative models representing SCMs in Gen to represent increasing layers of uncertainty, first over data in Figure 3.2 with `linear_probabilistic_scm` and then over probabilistic SCMs themselves in Figure 3.4 with `linear_bayesian_scm`. However, these programs do not yet address our interest in drawing *causal* conclusions from data. To do so, we must take advantage of the intervention model transformation as described in Definition 2.1.3. Applying this intervention to the SCM $\mathbb{M}$ in Equation 3.4, we obtain the intervened SCM $\mathbb{M}'$ as follows, where t is the intervention assignment:

$$\begin{aligned} \mathrm{x}_i &= \beta_{ux}\mathrm{u}_i + \epsilon_{\mathrm{x}_i} \\ \mathrm{t}_i &= t \\ \mathrm{y}_i(t) &= \beta_{ty}t + \beta_{xy}\mathrm{x}_i + \beta_{uy}\mathrm{u}_i + \epsilon_{\mathrm{y}_i} \end{aligned} \tag{3.9}$$

```
@gen function linear_twin_probabilistic_scm(parameters::Dict{Symbol, Float64},
                              intervention_assignment::Array{Float64, 1},
                              n::Int64)

    noise = @trace(noise_model(parameters, n))

    confounders = @trace(confounder_model(parameters, n))

    linear_intervened_scm = intervention(linear_scm, intervention_assignment)

    data = @trace(linear_scm(noise, confounders, weights), :factual)
    data_cf = @trace(linear_intervened_scm(noise, confounders, weights), :counterfactual)

    return data[:y], data_cf[:y]
end
```

Figure 3.8: **Implementation of the linear twin world probabilistic structural causal model in Gen.** Similar to `linear_probabilistic_scm` in Figure 3.5, `linear_twin_probabilistic_scm` extends the `linear_scm` with a distribution over exogenous noise and latent confounders. However, using our intervention program transformation, `linear_twin_probabilistic_scm` induces a distribution over factual, e.g. `data`, and counterfactual, e.g. `data_cf`, data.

To implement this transformation, we need an implementation of `intervention` shown in Figure 3.7 that takes as input a `model` and an `intervention_assignment`, and returns an `intervened_model` with an intervention applied according to Definition 2.1.3. I defer until Chapter 7 an in-depth discussion of how to implement such an intervention transformation in a way that is: (i) model-agnostic; (ii) accommodates Bayesian nonparametric approaches like GPs; (iii) covers the full space of Gen programs (with arbitrary control flow); and (iv) requires minimal introspection into the intervened model. That said, in Chapter 5, I show a partial solution in Figure 5.3 implemented using syntax transformations of symbolic code objects in a restricted domain-specific modeling language I call "MiniStan".

Suspending disbelief about a general implementation for now, imagine that we applied an implementation of `intervention` in to `linear_scm` with `intervention_assignment` given by a length n vector of ones. This program transformation would return a Gen program equivalent to `linear_intervened_scm` shown in Figure 3.7, which, not surprisingly, closely resembles the structural equations shown in Equation 3.9.

```julia
@gen function SATE(y::Array{Float64, 1}, y_cf::Array{Float64, 1})
    return sum(y - y_cf)/length(y)
end
```

```julia
@gen function linear_queried_probabilistic_scm(parameters::Dict{Symbol, Float64},
                                               intervention_assignment::Array{Float64, 1},
                                               n::Int64)

    y, y_cf = @trace(linear_twin_probabilistic_scm(parameters, intervention_assignment, n))
    return answer = SATE(y, y_cf)
end
```

Figure 3.9: **Implementation of the linear queried probabilistic structural causal model in Gen.** Using our probabilistic program over factual and counterfactual data, `linear_twin_probabilistic_scm`, `linear_queried_probabilistic_scm` induces a distribution over answers to our causal query by straightforwardly applying `linear_twin_probabilistic_scm`'s outputs to the deterministic query function, here, `SATE`. See Table 3.1 for a description of some other common queries.

3.2.4 Causal Queries

We need one final ingredient to tell a complete story of causal inference reducing to probabilistic probabilistic inference: our causal query, Q. As discussed in Section 3.1, Q is a function of factual and counterfactual endogenous variables, codifying causal questions of interest. In this linear example, we consider the *sample average treatment effect* (SATE), $Q = \sum_{i=1}^{n}(y_i(t) - y_i)/n$. As the name implies, answering a SATE query is akin to answering the English-language question "On average over the individuals I've measured, how much greater would their numerical outcomes have been if their treatment was forced to be t, rather than the treatment they actually received naturally?" This query is fairly straightforward. However, the space of causal queries is vast. While I don't explore it in detail in this thesis, Table 3.1 shows a small survey of standard causal queries in the literature, and how they translate to functions, Q[8].

[8] I have omitted some of the more complex queries, such as the nested counterfactual queries required for mediation analysis [98] or for actual causal queries [51], i.e. queries of the form "why did X event happen?". This omission is a consequence of the decision to simplify the definition of SCMs for ease of exposition and clarity in Section 2.1 to only consider fixed interventions on treatment variables, and not because of any inherent limitations of the Bayesian structural approach.

Query	**Treatment**	$Q(\mathbb{V}, \mathbb{V}(\boldsymbol{t}_1), ..., \mathbb{V}(\boldsymbol{t}_k))$
Average Treatment Effect (ATE)	Continuous	$\lim_{n \to \infty} \sum_{i=1}^{n} (y_i(t) - y_i)/n$
Average Treatment Effect (ATE)	Binary	$\lim_{n \to \infty} \sum_{i=1}^{n} (y_i(1) - y_i(0))/n$
Average Treatment Effect on the Treated (ATT)	Binary	$\lim_{n \to \infty} \sum_{i \in \mathcal{I}_t} (y_i(1) - y_i(0))/\|\|\mathcal{I}_t\|\|$, where $\mathcal{I}_t = \{i \in \llbracket n \rrbracket \| \mathrm{t}_i = 1\}$
Average Treatment Effect on the Untreated (ATU)	Binary	$\lim_{n \to \infty} \sum_{i \in \mathcal{I}_t} (y_i(1) - y_i(0))/\|\|\mathcal{I}_t\|\|$, where $\mathcal{I}_t = \{i \in \llbracket n \rrbracket \| \mathrm{t}_i = 0\}$
Conditional Average Treatment Effect (CATE)	Continuous	$\lim_{n \to \infty} \sum_{i \in \mathcal{I}_x} (y_i(t) - y_i)/\|\|\mathcal{I}_x\|\|$, where $\mathcal{I}_x = \{i \in \llbracket n \rrbracket \| \mathrm{x}_i = x\}$
Conditional Average Treatment Effect (CATE)	Binary	$\lim_{n \to \infty} \sum_{i \in \mathcal{I}_x} (y_i(1) - y_i(0))/\|\|\mathcal{I}_x\|\|$, where $\mathcal{I}_x = \{i \in \llbracket n \rrbracket \| \mathrm{x}_i = x\}$
Individual Treatment Effect (ITE)	Continuous	$y_i(t) - y_i$, for some $i \in \llbracket n \rrbracket$
Individual Treatment Effect (ITE)	Binary	$y_i(1) - y_i(0)$, for some $i \in \llbracket n \rrbracket$

Table 3.1: **Common causal queries.** Here I show a collection of (some) common causal queries used throughout the causal inference literature. All of these "average" queries can be converted to "sample" variants by omitting the $\lim_{n \to \infty}$. For example, the sample average treatment effect is defined as $\sum_{i=1}^{n} (y_i(t) - y_i)/n$ for continuous treatment, where n is the number of observed instances. Additionally, these queries can trivially be composed to form new queries, e.g. *conditional average treatment effect on the treated.* I have omitted these combinations from the table for brevity, as they are intuitive and straightforward to construct. Finally, I only define the conditional average treatment effect for discrete covariates, as conditioning on a continuous variable requires significant care.

Again, we can straightforwardly implement the sample average treatment effect as an ordinary Julia function, and then compose it with our `linear_twin_probabilistic_scm` to construct a new expanded probabilistic program. At this point this story of composition should not be particularly surprising; we continue to implement model composition as ordinary function calls, with special `@trace` constructs when the callee itself includes traced random variables. This simple wrapping can be seen in Figure 3.9.

With this twin world construction using our intervention program transformation and our new causal query, we can now visualize the full end-to-end mapping from structural causal models to causal queries. As shown in Figure 3.10, a single choice of probabilistic SCM (top left) induces a counterfactual SCM (bottom left) via the intervention transformation. The combination of these two structural causal models induce a distribution over factual data, $\mathbb{V}$, and counterfactual data, $\mathbb{V}(\boldsymbol{t})$, which then propagate to a distribution over causal queries, Q. As Figure 3.10 shows, if we knew the structural causal model exactly, we would therefore know the answer to our query[9].

3.2.5 Composing Intervention Program Transformations with Hierarchical Priors over Probabilistic SCMs

Now that we have a probabilistic program that maps the entire path from probabilistic structural causal model parameters to answers to our causal query, we can again extend the model to include priors in a similar way to how we implemented `linear_bayesian_scm` in Figure 3.4. Figure 3.11 shows exactly that extension, using `linear_queried_probabilistic_scm` as a sub-component. Just as

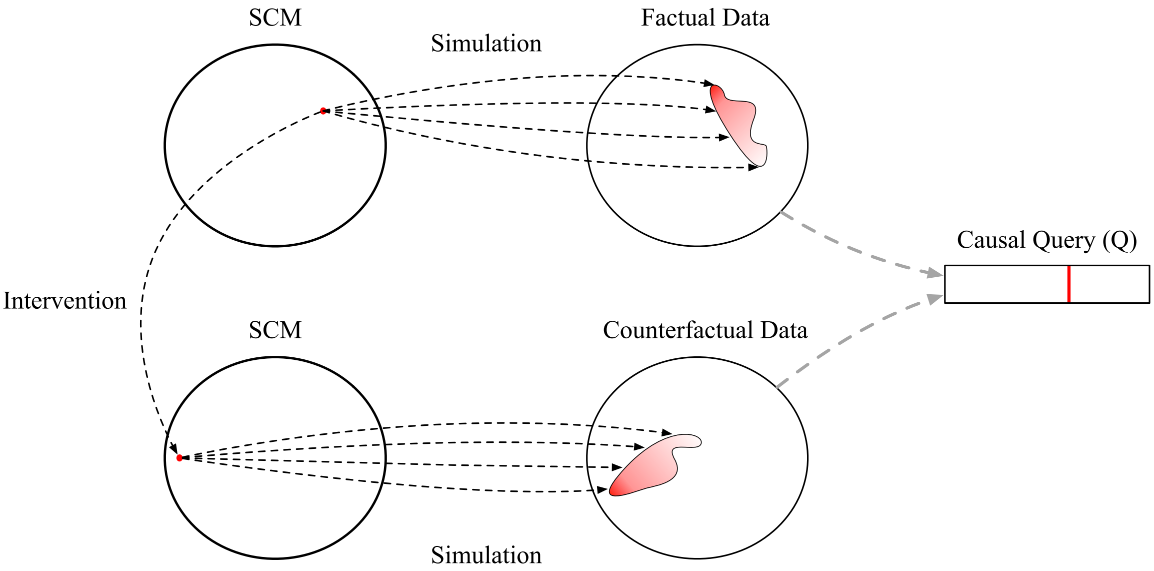


Figure 3.10: **Visual representation propagating a single probabilistic structural causal model to a causal query.** Just as in Figure 3.3, uncertainty over simulated data comes exclusively from exogenous noise and latent confounders, and not uncertainty over structural functions. In the limit of infinite data, fully determined SCMs often induce fully determined answers to causal queries, as shown here. However, this is not the case for finite data, or for some queries even asymptotically.

[9] This is true for this model because we have assumed additive noise. With less stringent assumptions, unique probabilistic structural causal models may still result in uncertainty over Q.

```julia
@gen function linear_queried_bayesian_scm(parameters_prior::Dict{Symbol, Distribution},
                                          intervention_assignment::Float64,
                                          n::Int64)

    parameters = Dict()

    for (variable, prior) in parameters_prior
        parameters[variable] = @trace(prior, variable)
    end

    return answer = @trace(linear_queried_probabilistic_scm(parameters, intervention_assignment, n))
end
```

Figure 3.11: **Implementation of the linear queried Bayesian structural causal model in Gen.** Identical to how `linear_bayesian_scm` extended `linear_probabilsitic_scm` with a prior distribution over parameters representing broader uncertainty, so too does `linear_queried_bayesian_scm` shown here extend `linear_queried_probabilistic_scm`. This program represents the joint distribution of parameters, factual and counterfactual data, and answers to our causal query.

in `linear_bayesian_scm`, extending a model that takes a single set of parameters to a model that instead places uncertainty over those parameters is exceptionally straightforward, we simple sample parameters from some collection of priors and then call `linear_queried_probabilistic_scm` with the sampled parameters. The only difference is that in `linear_queried_bayesian_scm` the nested model we call includes a distribution over answers to the causal query we are interested in, not just factual data as in `probabilistic_linear_scm`.

Figure 3.12 shows a visual representation of how placing a prior distribution on probabilistic structural causal models propagates through to uncertainty over intervened probabilistic structural causal models, factual and counterfactual data, and finally our causal query. While somewhat cartoonish, Figure 3.12 tells a realistic story for a reasonable prior; before seeing any data we should have broad uncertainty over how an intervention will influence the outcomes we are interested in.

Now that we have a probabilistic program representing a joint distribution over all of the quantities we are interested in, we can condition that distribution on observed factual data to induce a posterior distribution over causal effects, $p(Q|\mathbb{V})$. Figure 3.13 shows a visual representation of exactly that posterior distribution. Here, our updated

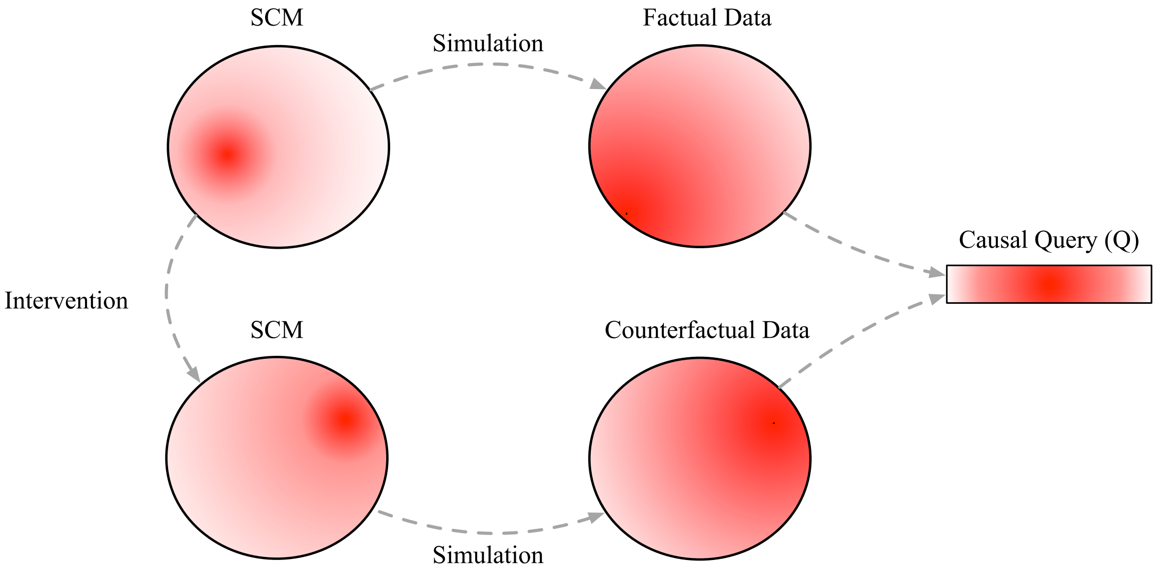


Figure 3.12: **Visual representation propagating a distribution over probabilistic structural causal model to a distribution over causal queries.** This represents the joint distribution over probabilistic structural causal models, factual and counterfactual data, and the resulting queries in the Bayesian structural approach. Before seeing any data our uncertain belief about the space of probabilistic structural causal models propagates to broad uncertainty over answers to causal queries.

belief about the space of probabilistic structural causal models propagates to the the space of intervened probabilistic structural causal models, counterfactual data, and finally the causal query itself.

In this particularly simple linear example, it turns out that the sample average treatment effect can be entirely summarized by the linear weight between treatment and outcome, β_{ty}, as can be seen by the following simple derivation[10]:

[10] Often econometrics textbooks will refer to these linear weights as the "causal effect" parameter. I avoid this terminology because it confuses the derivation of a mathematical fact (that for this particular collection of simple assumptions the effect is fully determined by β_{ty}) with a definition of an effect. In general, causal effects will not be reducible to a single parameter. As we'll see in subsequent chapters, this is not a problem, as we can probabilistically impute latent counterfactual outcomes and estimate resulting causal effects directly.

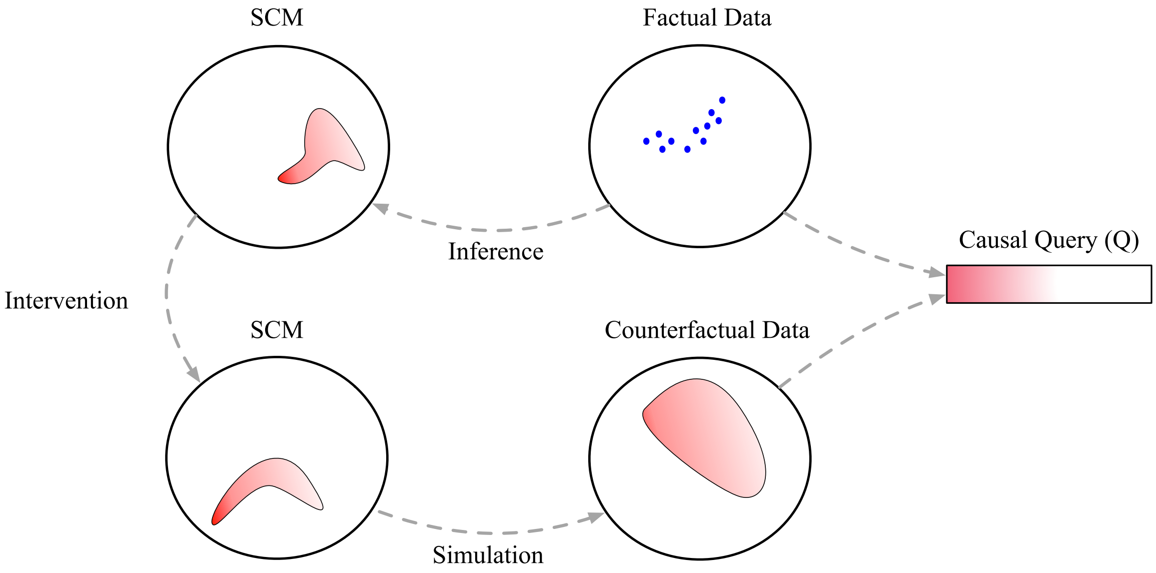


Figure 3.13: **Visual representation of the posterior distribution of probabilistic structural causal models given data, and the resulting distribution over causal queries.** Conditioning a Bayesian SCM on observations, $\mathbb{V}$, induces a posterior distribution over probabilistic SCMs, which is then propagated to a distribution over interventional SCMs, counterfactual data, and finally causal queries. Here, multiple causal explanations are consistent with observed data. Therefore, the posterior distribution $p(Q|\mathbb{V})$ shown on the right does not collapse even as $n \to \infty$.

$$\begin{aligned} Q(\mathbb{V}, \mathbb{V}(t)) &:= \sum_{i=1}^{n} (\mathrm{y}(t)_i - \mathrm{y}_i)/n \\ &= \sum_{i=1}^{n} ((\beta_{ty} t + \cancel{\beta_{xy} \mathrm{x}_i} + \cancel{\beta_{uy} \mathrm{u}_i} + \cancel{\epsilon_{\mathrm{y}_i}}) - (\beta_{ty} \mathrm{t}_i + \cancel{\beta_{xy} \mathrm{x}_i} + \cancel{\beta_{uy} \mathrm{u}_i} + \cancel{\epsilon_{\mathrm{y}_i}}))/n \\ &= \beta_{ty} \sum_{i=1}^{n} (t - \mathrm{t}_i)/n \end{aligned} \tag{3.10}$$

On the one hand, this is a convenient result; even though there are many parameters that are needed to uniquely specify a probabilistic structural causal model, we only need a single one to answer the causal question we are interested in. However, as discussed in Section 3.2.2, the presence of the latent confounder $\mathbf{u}$ makes it challenging to estimate even just β_{ty}, even as $n \to \infty$. This is not particularly surprising: any observed dependence could be explained by either a strong effect of treatment on

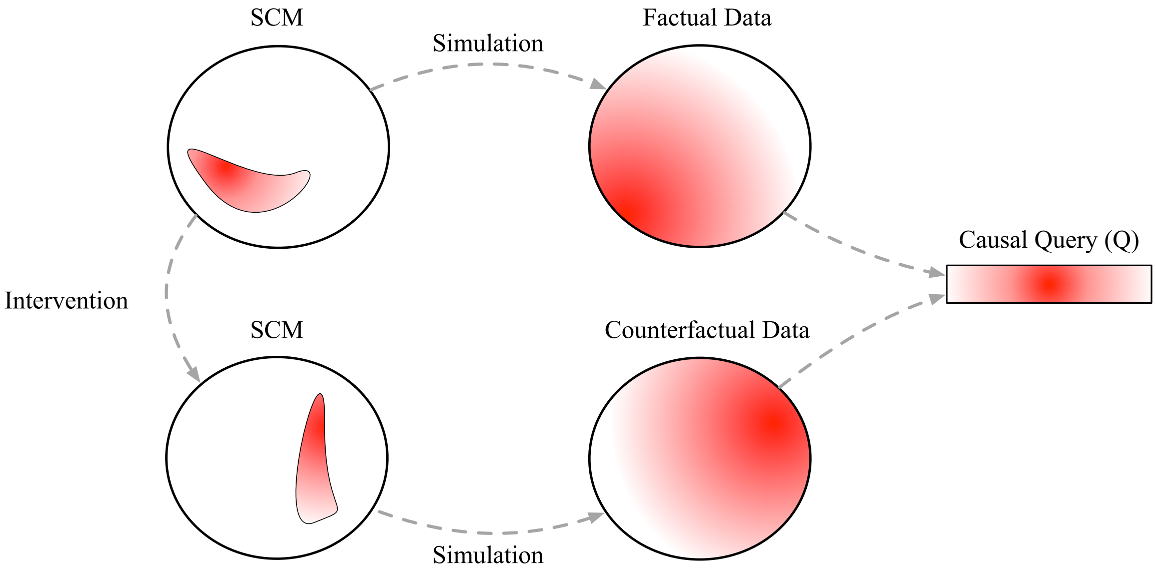


Figure 3.14: **Visual representation propagating a distribution over probabilistic structural causal model to a distribution over causal queries with stronger causal assumptions.** Adding an additional assumption to our Bayesian structural causal model materializes in a change to our prior distribution over probabilistic structural causal models. In this example, we place a dirac-delta prior on $\beta_{uy} = 0$.

outcome, or a strong latent confounder. However, not all latent confounders cause these kinds of problems. To see this, let's explore what happens when we make slightly stronger assumptions.

3.2.6 Stronger Causal Assumptions as Priors

In the proceeding sections we have seen how the assumptions expressed as priors imply that our causal query of interest, the sample average treatment effect, can not be unambiguously estimated from data. However, what if instead we were willing to make slightly stronger assumptions? Specifically, what happens if we assume that the confounder does not have an effect on the outcome, and that it only affects the treatment and covariates. In other words, we place a dirac-delta prior on $\beta_{uy} = 0$, rather than any arbitrary prior over the reals.

These stronger assumptions expressed in terms of priors can be seen visually in Figure 3.14. All that has changed from the previous collection of assumptions reflected in Figure 3.12 is that the space of probabilistic structural causal models is smaller; all of the transformations are left unchanged. Just as in Figure 3.12, before seeing any data we are uncertain about the causal effect. However, this simple restriction combined with observational data is enough to yield unique causal conclusions as $n \to \infty$.

Here, setting β_{uy} to 0 in the system of equations in Equation 3.8 does not result in a unique solution to all of the parameters in θ, as there are still more parameters than there are equations. However, recall from Equation 3.10 that we don't need to estimate all of the parameters in θ to answer our causal query; we only need to estimate β_{ty}. In fact, using a computer algebra system, we can see that the modified collection of Equations in 3.8 results in the following unique solution:

$$\beta_{ty} = \frac{cov(\mathbf{t},\mathbf{y})cov(\mathbf{x},\mathbf{x}) - cov(\mathbf{x},\mathbf{t})cov(\mathbf{x},\mathbf{y})}{cov(\mathbf{t},\mathbf{t})cov(\mathbf{x},\mathbf{x}) - cov(\mathbf{x},\mathbf{t})^2} \tag{3.11}$$

As each of these covariance terms over observable random variables have a unique maximum likelihood asymptotically, so too does β_{ty}. By the Bernstein von-Mises theorem [34] $p(\beta_{ty}|\mathbb{V})$, and by extension $p(Q|\mathbb{V})$, thus converges to that unique maximum likelihood as $n \to \infty$. Putting this result in context, we can see that strengthening our assumptions about how latent confounders relate to observable variables results in qualitatively different conclusions about our ability to answer causal questions. Before, no amount of data could rescue us from ambiguity, and now given enough data we can answer the question of interest.

Figures 3.14 and 3.15 shows a visual representation of how these stronger assumptions again propagate through to the distribution over causal queries. In Figure 3.14, even though we have concentrated our prior distribution over a smaller region of probabilistic structural causal models before seeing any data our uncertainty over answers to the causal query remain high. However, Figure 3.15 shows how these

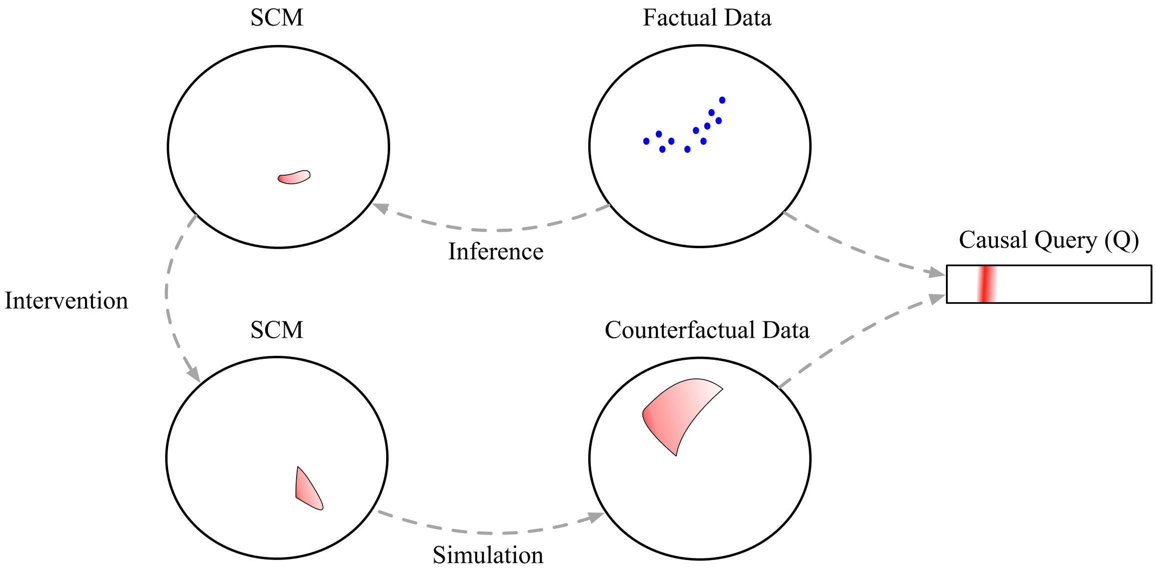


Figure 3.15: **Visual representation of the posterior distribution of probabilistic structural causal models given data, and the resulting distribution over causal queries with stronger causal assumptions.** By restricting the space of probabilistic structural causal models a-priori, conditioning on data not leads to asymptotically unambiguous causal conclusions.

stronger assumptions combined with data yield significantly less ambiguous causal conclusions. In other words, stronger causal assumptions supplement data to yield strong causal conclusions, but both are necessary.

Reflecting these additional assumptions in our Gen implementations is straightforward, we simply change the prior distribution over β_{uy} in `weights_prior` passed as an argument to `linear_queried_bayesian_scm` in Figure 3.11[11].

While the analysis shown in this section is unique to the specific linear Gaussian set of assumptions, it turns out that some of the conclusions are much more broad. In fact, when translated into a causal graphical model the additional assumption that $\beta_{uy} = 0$ corresponds exactly to the causal graph shown in Figure 2.1. As we discussed in Section 2.1, Pearl's graphical theory already tells us that $p(\mathbf{y}(\boldsymbol{t}))$ can

[11] Not all representations of stronger causal assumptions will have such a simple no-code change, as we'll see in the Gen code representations of the quasi-experimental designs from Section 2.1.

be unambiguously estimated from data without any parametric assumptions, so it shouldn't be surprising that that conclusions holds for our particular linear case. However, the restriction on whether **u** influences **y** is just one kind of assumption we may want to add to enable causal inferences from data.

3.2.7 Quasi-Experimental Designs

As we discussed in Section 2.1, there are other practical assumptions that can't be expressed by graph structure alone, but that can be represented as priors over probabilistic structural causal models. To see that, let's revisit our three quasi-experimental designs from Section 2.1.

Regression Discontinuity Design. Figure 3.16 shows the Gen code for implementing a simple version of the regression discontinuity design from Section 2.1 with a similar parameterization to our linear model throughout this chapter. To implement this model we simple remove the latent confounder **u** and change the assignment mechanism in line `7` of `linear_rdd_scm` from its original linear function to the expression `t = x .> 0.`, which is an array representation of the expression `if (x > 0) 1 else 0 end`. How we add priors, apply an intervention, and propagate counterfactual outcomes through to our causal query is conceptually identical to the linear example, with some minor syntax changes to account for the difference in arguments between the two models. The important lesson here is that adding new assumptions, even those outside of the realm of causal graphical models, can be remarkably straightforward using the Bayesian structural approach. In Chapter 6, I unpack the implications of the regression discontinuity assumptions with both linear parameterizations like the ones shown here and when using a finite dimensional Gaussian process prior.

```
@gen function linear_rdd_scm(noise::Array{Float64, 1},
                             confounders::Array{Float64, 1},
                             parameters::Dict{Symbol, Float64})

    x = noise[:x]
    # Array representation of t_i = 1 if x_i > 0., else t_i = 0
    t = x .> 0.
    y = parameters[:beta_ty] * t + parameters[:beta_xy] * x + eps_y

    return data = Dict(:x => x, :t => t, :y => y)
end
```

Figure 3.16: **Implementation of the regression discontinuity design in Gen.** To implement the regression discontinuity design we remove confounders, **u**, and modify the assignment mechanism for treatment, **t**. Despite its simplicity, regression discontinuity designs yield remarkably different conclusions to what the corresponding graphical model would lead one to believe.

```
 1 @gen function linear_probabilistic_slc_scm(parameters::Dict{Symbol, Float64}, n_i::Int64, n_o::Int64)
 2 
 3     noise = @trace(noise_model(parameters, n_o * n_i))
 4 
 5     confounders = @trace(confounder_model(parameters, n_o))
 6 
 7     # Tile each length n_o vector of confounders to a vector of length n_o * n_i
 8     tiled_confounders = Dict()
 9 
10     for (confounder, values) in confounders
11         tiled_confounders[confounder] = repeat(values, inner=n_i)
12     end
13 
14     return linear_scm(noise, tiled_confounders, weights)
15 end
```

Figure 3.17: **Implementation of the structured latent confounder design in Gen.** To implement the structured latent confounder model, we modify `linear_probabilistic_scm` to sample `n_o` instances of latent confounders, and then tile those confounders (lines 8-12) so that they are shared amongst multiple instances of observed covariates, treatment, and outcome. Note that doing so only requires modifying how the confounder model is called, and can be applied agnostically to any user-specified choice of confounder model.

Structured Latent Confounding. Figure 3.17 shows the Gen code for implementing a simple version of the structured latent confounding example from Section 2.1, again with a similar parameterization to our linear model throughout this chapter. Recall that the structured latent confounding assumption is that the same object-level latent confounder is shared between multiple instances. Implementing this change is again remarkably straightforward; we sample a length `n_o` vector of latent confounders `u_o`, and then tile those samples to match the number of data instances `n_i * n_o`. For example, if the `confounder_model` samples a vector `[0., 1.]` and `n_i=3`, then line `11` will produce a vector `[0., 0., 0., 1., 1., 1.]`. Again, an extremely minor change in code has remarkable implications with respect to what causal conclusions we can draw from data. As I show in Section 4.4 of Chapter 4, it turns out that this assumption leads to identifiability for the linear case. In Chapter 4 I expand on this example with rich Gaussian process priors over structural functions.

Instrumental Variable Design. Unlike the regression discontinuity design and the structured latent confounding example, but similar to our modified model in Section 3.2.6, implementing the instrumental variable designs model does not require any changes to our original linear structural causal model programs. Instead, the exclusion and as-if random assumptions described in Chapter 2 can be implemented by modifying the choice of priors over weight parameters in the linear model. Specifically, the exclusion restriction corresponds to assuming a dirac-delta prior on $\beta_{xy} = 0$, i.e. there is no effect from the instrument, $\mathbf{x}$, to the outcome, $\mathbf{y}$, except the effect mediated through treatment, $\mathbf{t}$. Similarly, the as-if random restriction corresponds to assuming a dirac-delta prior on $\beta_{ux} = 0$, i.e. there is no effect of the latent confounder, $\mathbf{u}$, on the instrument, $\mathbf{x}$.

While these instrumental variable design assumptions look identical to the stronger assumptions in Section 3.2.6, they only enable identifiability because of the specific parameterization we chose for our original linear structural causal model. If we had instead chosen a model with non-linear functions and non-additive exogenous noise, the exclusion and as-if random conditions would be insufficient. This is not true of the assumptions in Section 3.2.6; any choice of parameterization would lead to identical identification results.

3.2.8 A Note on Traced Randomness and Reparameterization

The code expressions in `linear_scm` closely resemble the mathematical description of structural causal models generally, as well as the particular choice of parameterization in Equation 3.4. However, this choice of representation is not particularly convenient for downstream probabilistic inference. In Gen, and most probabilistic programming languages, intermediate computations necessary for implementing downstream inference algorithms can only occur at special *traced* addresses, denoted in Gen using the custom `@trace` Julia macro. For example, Gen automates incremental resampling, likelihoods, and likelihood gradients at traced addresses. In other words, if we want to condition on data, such as `x`, we will need `x` in the code to be assigned according to `@trace(some_dist, :x => i)`, rather than the current version which assigns `x` using ordinary (untraced) Julia code. To address this practical concern we can rewrite `linear_twin_probabilistic_scm` as shown in Figure 3.18.

This reparameterization combines the distribution over noise variables with the choice of structural function, placing all `@trace` assignments exactly where we will eventually condition on data. In other words, here we had to manually marginalize out exogenous noise, $\mathbb{X}$, having our code instead directly reflect the pushforward distribution, $p(\mathbb{V}|\mathbb{U}, \mathbb{F})$. In this thesis I assume that this pushforward reparameterization, or change of variables, is always tractable. In this linear Gaussian example, we rely on the simple fact that for the composition of equations $\mathrm{y}_i = \beta \mathrm{t}_i + \alpha u_i + \gamma \epsilon_{y_i}$ and $\epsilon_{y_i} \sim \mathcal{N}(0, 1)$, the conditional density $p(y_i | f_y, u_i)$ is given by $\mathcal{N}(y_i; \beta \mathrm{t}_i + \alpha u_i, \gamma)$.

While the implementations for the remaining chapters rely on a manual implementation of this pushforward, similar to Figure 3.18, in Chapter 7 I discuss some plausible approaches to automating these transformations.

```julia
@gen function linear_twin_probabilistic_scm(parameters::Dict{Symbol, Float64},
                              intervention_assignment::Array{Float64, 1},
                              n::Int64)

    u = @trace(confounder_model)[:u]

    x, t, y = [], [], []

    for i in 1:n
        mean_x = parameters[:beta_ux] * u[i]
        x[i] = @trace(Normal(mean_x, parameters[:var_x]), :x => i)

        mean_t = parameters[:beta_xt] * x[i] + parameters[:beta_ux] * u[i]
        t[i] = @trace(Normal(mean_t, parameters[:var_t]), :t => i)

        mean_y = parameters[:beta_ty] * t[i] + parameters[:beta_xy] * x[i] + parameters[:beta_uy] * u[i]
        y[i] = @trace(Normal(mean_y, parameters[:var_y]), :y => i)
    end

    y_cf = y + parameters[:beta_ty] * (intervention_assignment - t)

    data = Dict(:x => x, :t => t, :y => y)
    data_cf = Dict(:x => x, :t => intervention_assignment, :y => y_cf)

    return data, data_cf
```

Figure 3.18: **Re-implimentation of `linear_twin_probabilistic_scm` from Figure 3.8 to permit conditioning on x, t, y.** Unlike the implementation in Figure 3.8, which faithfully resembled the mathematical description of structural causal models, this implementation assigns all observed variables, `x`, `t`, and `y` using Gen's `@trace` construct for tracking random choices. In doing so, we enable conditioning on `x`, `t`, and `y`, rather than on their corresponding noise variables.

3.3 Choosing a Formalism for Causal Inference: Strengths and Limitations of the Bayesian Approach

For forward-looking practitioners, existing alternatives to the Bayesian structural approach presented in this thesis offer a somewhat unsatisfying choice: use Pearl's graphical formalisms and ignore any non-graphical assumptions or hope that your particular problem fits within a small collection of well-studied templates, such as the regression discontinuity or regression discontinuity design quasi-experimental designs. This thesis aims to be more aspirational, providing an alternative formalism in which causal assumptions can be expressed in terms of computational structures without being restricted to graph structure alone. However, doing so is not without consequence. In this section, I enumerate some of these consequences, and describe circumstances in which one should prefer existing approaches. For example, some limitations of the Bayesian structural approach are the following:

First, the Bayesian structural approach requires explicit priors on structural functions, confounder distributions, and exogenous noise distribution, whereas existing approaches may yield causal conclusions with weaker and less committal descriptions of uncertainty. For example, in this chapter we came to interesting conclusions about linear structural causal models and their variants, but these analyses were not conclusive about any other parametric family of models. If we were to instead swap out these assumptions for a more expressive prior over polynomial as opposed to linear functions we might yield yet different conclusions. The do-calculus however, would tell us that the graphical structure in Figure 2.1 will always lead to identifiability, regardless of structural function, noise distribution, etc. Instead, we would like a formalism that allows us to restrict functional form when necessary, such as in this linear instrumental variable design, but leave structural functions flexible when not. In the following chapters, I will show how using the Bayesian structural approach with Bayesian nonparametric priors over structural functions allows us to do just that. However, these nonparameteric priors still implicitly posit some assumptions about smoothness and the family of noise distributions, e.g. we assume Gaussian noise in subsequent chapters. In Chapter 7, I discuss some opportunities for future work to address these remaining limitations of the Bayesian structural approach.

Second, reducing causal inference to probabilistic inference does not fully solve the problem of causal inference, as exact probabilistic inference in even discrete models is a NP-hard computational problem. Therefore, probabilistic inference often turns to approximation methods, such as the variants of importance sampling I discuss in Section 2. For the most part, I consider this to be a strength, as it allows us to bring to bear advances in probabilistic inference for causal inference problems. However, approximate inference is a notoriously hard practical problem, and errors are often difficult to diagnose. While there is an active body of research on improving the practice of Bayesian inference workflows [46], it is far from complete.

With multiple formalisms in hand, each of which with limitations, we are left with the natural and obvious question, “when should we prefer one formalism over another?” The somewhat pithy answer is that one should use the Bayesian structural approach when one’s assumptions are not representable simply as the structure of a directed acyclic graph or not already covered by the small library of standard econometric designs. Somewhat less pithily, the Bayesian approach presented in this thesis provides both expressiveness and automation to causal reasoning, two characteristics that will become increasingly important as the scale and complexity of application domains continues to increase. For example, high-energy particle physics simulation models are clearly not representable as directed acyclic graphs, nor are they covered by any econometric methods, and yet scientists wish to use these simulation models as representations of causal knowledge in combination with data to yield novel causal conclusions. The Bayesian structural approach is perfectly compatible with these kinds of complex simulation models.

3.4 Related Work

The work in this thesis is certainly not the first to propose being Bayesian about causal inference; Rubin himself espoused the Bayesian approach to causal inference as far back as 1978 [112]. In fact, many recent papers in the modern literature on machine learning for causal inference could be equivalently framed as Bayesian structural causal inference as we do here, i.e. placing priors over probabilistic structural causal models [52, 78, 92, 128]. However, what these works do not address, and what we try to emphasize in this chapter, is the role of computational concepts of composition, abstraction, modularity, and reuse in representing and reasoning about causal assumptions expressed as probabilistic programs. As we saw in this chapter and as we'll continue to see in the remainder of this thesis, thinking of priors over structural causal models as code makes advanced applications of Bayesian nonparametric causal inference almost trivial.

Our work is also not alone in identifying the central role of probabilistic programming for causal reasoning and inference. For example, Omega [124] is a probabilistic programming language with first-class syntax for intervention. Where these contributions focus on the programming languages considerations to make interventions coherent and sound, the contributions in this thesis focuses on the practice of programming using these constructs to implement advanced variants of common causal designs that aren't supported by graph based methods. Other probabilistic programming languages have implemented interventions in various ways, including Pyro [16] and MultiVerse [99], although (unlike Omega) these languages do not provide a formal semantics.

3.5 Conclusion

In this chapter I illustrated the Bayesian structural approach to causal inference with a simple example of a linear model, and then extended that model to reflect assumptions necessary for common quasi-experimental designs. In doing so, I showed how representing models as code allows us to implement fairly diverse causal assumptions succinctly by changing only a few assignment statements. However, what should we do if we want to go beyond these simple linear examples, including Bayesian nonparametric priors over structural functions, or with to reason using experimental data? In the remainder of this thesis, I provide partial answers to these questions.

CHAPTER 4

HIERARCHICAL CAUSAL INFERENCE USING GAUSSIAN PROCESSES WITH STRUCTURED LATENT CONFOUNDERS

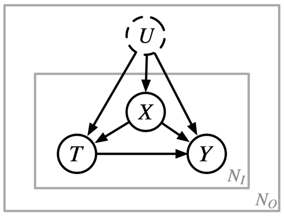
Multiple causal models can be *observationally equivalent*, i.e., they induce the same likelihoods for observed data, while producing different estimates of the effects of a particular intervention of interest. Distinguishing between causal models, and estimating the effects of interventions, typically requires untestable assumptions about causal structure.

One such common assumption is *unconfoundedness* [61], i.e., that there exist no latent variables that influence both treatment and outcome. This assumption enables the unique identification of interventional distributions from the joint distribution over observed variables [97] and reduces causal inference to probabilistic estimation. Unfortunately, assuming unconfoundedness is often unreasonable in real observational settings [117]. However, it may be more reasonable to assume unconfoundedness for a subset of data instances that are known to share a common structure.

For example, suppose a local school board proposes a new policy of holding back poor performing kindergarten students [57, 58] with the intention of increasing their future academic performance. To estimate the effect of this policy change, they gather data on student retention and education outcomes from a national database. Here, the unconfoundedness assumption is not justified, as the schools' retention policies are likely to be influenced by local economic conditions, which may also influence student outcomes through other causal mechanisms, such as the availability of educational resources. However, the assumption may be justified when considering only students

Symbol	Description	Entity
$\mathbf{U}_{o,:}$	Confounders	Object
$\mathbf{X}_{i,:}$	Covariates	Instance
t_i	Treatment	Instance
y_i	Outcome	Instance

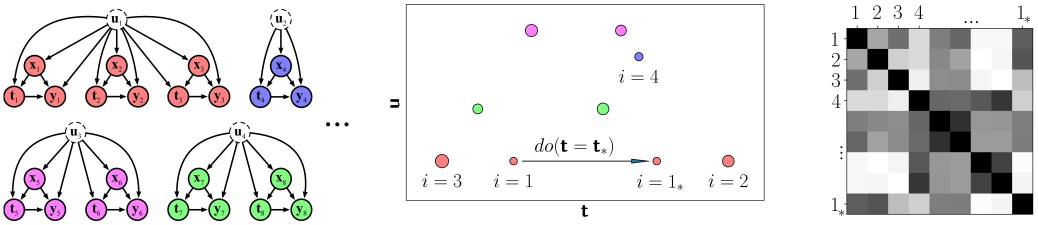
(a) Variable descriptions.



(b) Causal graph for GP-SLC.

$$f_u \sim GP(0, k_u) \quad f_x \sim GP(0, k_x)$$
$$f_t \sim GP(0, k_t) \quad f_y \sim GP(0, k_y)$$
$$\mathrm{U}_{o,j} = f_u(\epsilon_{u_{o,j}})$$
$$\mathrm{X}_{i,l} = f_{x_l}(\mathrm{u}_{o=Pa(i)}, \epsilon_{x_{i,l}})$$
$$\mathrm{t}_i = f_t(\mathbf{U}_{o=Pa(i),:}, \mathbf{X}_{i,:}, \epsilon_{t_i})$$
$$\mathrm{y}_i = f_y(\mathbf{U}_{o=Pa(i),:}, \mathbf{X}_{i,:}, \mathrm{t}_i, \epsilon_{y_i})$$

(c) Prior and causal functions for GP-SLC.


$\mathbf{u}_1$
$\mathbf{u}_2$
$\mathbf{u}_3$
$\mathbf{u}_4$
$\mathbf{x}_1$ $\mathbf{x}_2$ $\mathbf{x}_3$ $\mathbf{x}_4$ $\mathbf{x}_5$ $\mathbf{x}_6$ $\mathbf{x}_7$ $\mathbf{x}_8$
$\mathbf{t}_1$ $\mathbf{y}_1$ $\mathbf{t}_2$ $\mathbf{y}_2$ $\mathbf{t}_3$ $\mathbf{y}_3$ $\mathbf{t}_4$ $\mathbf{y}_4$
$\mathbf{t}_5$ $\mathbf{y}_5$ $\mathbf{t}_6$ $\mathbf{y}_6$ $\mathbf{t}_7$ $\mathbf{y}_7$ $\mathbf{t}_8$ $\mathbf{y}_8$
...
$\mathbf{u}$
$i = 4$
$do(\mathbf{t} = \mathbf{t}_*)$
$i = 3$
$i = 1$
$i = 1_*$
$i = 2$
$\mathbf{t}$
1 2 3 4 ... 1_*
1
2
3
4
⋮
1_*

(d) Example grounding of the structural causal model in (b) and (c). Latent confounders are shared within objects.

(e) Treatment, covariates, and inferred object-level confounders for instances in (d). Color = o. Size = $\mathbf{X}_{i,:}$.

(f) Kernel covariance matrix over observed (y_i) and counterfactual (y_{1,t_*}) outcomes for instances in (e). Dark > light.

Figure 4.1: **Model summary.** GP-SLC (a-c) is a Gaussian process model for causal inference in settings where object-level latent confounders, $\mathbf{U}$, influence instance-level observed covariates, $\mathbf{X}$, treatment, $\mathbf{t}$, and outcome, $\mathbf{y}$, random variables. For a given grounding (d), the outcome kernel function, k_y, applied to treatment, covariates, and inferred confounders (e) induces the covariance between observed and counterfactual outcomes (f). Instances belonging to the same object always have the same inferred latent $\mathbf{U}_{o,:}$. In this example, the counterfactual outcome $\mathrm{y}_1(t_*)$ has high covariance with factual outcomes y_1 and y_2. $\mathrm{y}_1(t_*)$ has low, but non-zero, covariance with y_4 because $\mathbf{U}_{Pa(1),:} \not\approx \mathbf{U}_{Pa(4),:}$, despite the fact that $\mathrm{t}_* \approx \mathrm{t}_4$ and $\mathbf{X}_{1,:} \approx \mathbf{X}_{4,:}$.

within a particular school, as this subset of students are similarly influenced by local economic conditions. In other words, statistical relationships within a school are less likely to be biased by latent confounders than are statistical relationships across the entire population.

In this chapter, I present Gaussian processes with structured latent confounders (GP-SLC), a novel Bayesian nonparametric approach to causal inference with hierarchically structured observational data. The key innovation behind GP-SLC is to place Gaussian process priors over functions in a hierarchical structural causal model, bringing the flexibility of Gaussian process models to a wide variety of practical causal inference techniques. GP-SLC naturally handles binary and continuous treatments and requires minimal assumptions about functional relationships between latent confounders, observed covariates, treatment, and outcomes. See Figure 4.1 for an overview on how GP-SLC estimates counterfactual outcomes from data.

4.1 Background

4.1.1 Object Conditioning

Recent work has studied how the analytical procedure of partitioning data based on a known object hierarchy (e.g. students belonging to the same school) relates to the syntax and semantics of causal graphical models [63]. This work concludes that conditioning on the identify of objects (referred to as *object conditioning*) is distinct from existing notions of conditioning on the values of variables. Importantly, object conditioning constrains a set of latent variables to be identical across a set of instances, but does not constrain the particular value of those variables. Furthermore, the statistical implications of object conditioning differ from those of variable conditioning in that object conditioning does not induce collider bias when variables on the object are caused jointly by treatment and outcome.

Partitioning hierarchical data in this way is the key analytical procedure for a variety of practical causal inference techniques, including within-subjects designs [77], difference-in-differences designs [117], longitudinal studies [76], twin studies [20], and multi-level-modeling [45]. As in the student retention example, these techniques take advantage of background knowledge about which instances (students) belong to which objects (schools) to mitigate the biasing effects of latent confounders. However, these methods typically rely on simple parametric assumptions, such as linear functional dependencies. These parametric assumptions are often unjustified in real domains, leading to poor estimates of causal effect.

We employ the idea of object conditioning directly in the GP-SLC model, constraining the joint distribution over individuals' latent confounders instead of treating object identity as a covariate in and of itself. By explicitly performing inference over object-level latent confounders, GP-SLC's estimates of counterfactual outcomes in one object are informed by observed outcomes in another. Sharing information between objects in this way is particularly valuable when each object contains few observed instances, as we show in Section 4.5.

4.1.2 Causal Inference with Latent Confounders

Latent confounders—unobserved variables that cause both treatment and outcome—bias estimates of treatment effect. However, this bias can be adjusted for with additional background knowledge, such as that a latent confounder influences an observed proxy variable [70, 84]. Similarly, recent work indicates that latent confounders can be adjusted for if they cause multiple candidate treatment variables [132].

GP-SLC is similar to these approaches, in that it leverages additional background knowledge to adjust for latent confounders. However, unlike prior work using generative models for causal inference with latent confounders, it leverages known hierarchical structure to identify causal effects.

4.1.3 Gaussian Process Models

As discussed in Chapter 2, Gaussian process models are a flexible technique for probabilistic modeling. Specifically, a Gaussian process is a distribution over deterministic functions, $\mathbf{y} = f(\mathbf{x})$, $f \sim GP(m, k)$, which is fully specified by its mean function, $m(\mathbf{x})$ and covariance function, $k(\mathbf{x}, \mathbf{x}')$, which we will refer to as the kernel function [103]. By definition, any finite collection of draws from a Gaussian process prior are jointly Gaussian distributed, $\mathbf{y} \sim \mathcal{N}(\mu, \Sigma)$, where $\mu_i = m(\mathrm{x}_i)$ and $\Sigma_{i,i'} = k(\mathrm{x}_i, \mathrm{x}_{i'})$. In this chapter, we denote such covariance matrices as $\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X})$. It is common to set the prior mean function to $m(\mathbf{x}) = 0$, which we do in GP-SLC.

4.2 Gaussian Processes with Structured Latent Confounders

Consider the common scenario where there are n_o object-level latent confounders ($\mathbf{U} \in \mathbb{R}^{n_o \times n_u}$) that influence n_i instances of observed treatment ($\mathbf{t} \in \mathbb{R}^{n_i}$), covariates ($\mathbf{X} \in \mathbb{R}^{n_i \times n_x}$), and outcomes ($\mathbf{y} \in \mathbb{R}^{n_i}$). We can describe this scenario as a structural causal model, where the particular functions relating $\mathbf{U}$, $\mathbf{X}$, $\mathbf{t}$, and $\mathbf{y}$ are given by the following for all $o \in [\![n_o]\!]$, $j \in [\![n_u]\!]$, $i \in [\![n_i]\!]$, $l \in [\![n_x]\!]$:

$$\begin{aligned}
\mathrm{U}_{o,j} &= f_{u_j}(\epsilon_{u_{o,j}}) \\
\mathrm{X}_{i,l} &= f_{x_l}(\mathrm{u}_{o=Pa(i)}, \epsilon_{x_{i,l}}) \\
\mathrm{t}_i &= f_t(\mathbf{U}_{o=Pa(i),:}, \mathbf{X}_{i,:}, \epsilon_{t_i}) \\
\mathrm{y}_i &= f_y(\mathbf{U}_{o=Pa(i),:}, \mathbf{X}_{i,:}, \mathrm{t}_i, \epsilon_{y_i})
\end{aligned} \tag{4.1}$$

If all instances belong to the same object ($n_o = 1$) the structural causal model in Equation 4.1 reduces to the standard propositional case and the latent $\mathbf{U}$ will not bias estimated counterfactual outcomes. However, if we wish to estimate counterfactual outcomes using instances from multiple objects ($n_i > n_o > 1$), $\mathbf{U}$'s influence on $\mathbf{t}$ and $\mathbf{y}$ would appear to render counterfactual queries nonparametrically unidentifiable [97]. However, the fact that the confounding variable is shared between multiple observed instances imposes additional restrictions on the structural causal model, i.e. multiple instances of $\mathbf{X}$, $\mathbf{t}$, and $\mathbf{y}$ are functions of the same confounder instances. Rather than be fully nonparametric, GP-SLC instead places a Gaussian process prior over each function in the structural causal model in Equation 4.1, with kernel functions k_x, k_t, and k_y respectively as follows:

$$f_{x_k} \sim GP(0, k_{x_k}) \quad f_t \sim GP(0, k_t) \quad f_y \sim GP(0, k_y). \tag{4.2}$$

The particular choice of each kernel function plays an important role in the prior over functions, and by extension the conditional distribution over counterfactual outcomes. We use a radial basis function (RBF) kernel with automatic relevance determination (ARD) [87] and additive Gaussian exogenous noise for each Gaussian process prior. Each kernel is parameterized by a set of kernel lengthscales, λ, scaling factors, σ^2, and exogenous noise variances σ^2_ϵ. We assume f_{u_l} is the identity function. We refer to the noise-free component of each kernel function as k', e.g. $k_t(x_i, x_{i'}) = k'_t(x_i, x_{i'}) + \sigma^2_{\epsilon_y}\delta_{i,i'}$, where $\sigma^2_{\epsilon_y}$ is the exogenous noise variance, $\delta_{i,i'}$ is the Dirac-delta function at $i' = i$, and k'_t is the ARD kernel. We omit kernel arguments for brevity, e.g. $k'_y([\mathbf{U}_{o,:}, \mathbf{X}_{i,:}, \mathrm{t}_i], [\mathbf{U}_{o',:}, \mathbf{X}_{i',:}, \mathrm{t}_{i'}])$ is represented as $k'_y(\cdot, \cdot)$. Specifically, each kernel is defined as follows, where $o = Pa(i)$ and $o' = Pa(i')$:

$$\begin{aligned} k'_{x_l}(\cdot,\cdot) &= \sigma^2_{x_l} \exp\left[-\sum_{j=1}^{n_u} \frac{(\mathrm{U}_{o,j} - \mathrm{U}_{o',j})^2}{\lambda_{ux_{j,l}}}\right] \\ k'_t(\cdot,\cdot) &= \sigma^2_t \exp\left[-\sum_{j=1}^{n_u} \frac{(\mathrm{U}_{o,j} - \mathrm{U}_{o',j})^2}{\lambda_{ut_j}} - \sum_{l=1}^{n_x} \frac{(\mathrm{X}_{i,l} - \mathrm{X}_{i',l})^2}{\lambda_{xt_l}}\right] \\ k'_y(\cdot,\cdot) &= \sigma^2_y \exp\left[-\sum_{j=1}^{n_u} \frac{(\mathrm{U}_{o,j} - \mathrm{U}_{o',j})^2}{\lambda_{uy_j}} - \sum_{l=1}^{n_x} \frac{(\mathrm{X}_{i,l} - \mathrm{X}_{i',l})^2}{\lambda_{xy_l}} - \frac{(\mathrm{t}_i - \mathrm{t}_{i'})^2}{\lambda_{ty}}\right] \end{aligned} \tag{4.3}$$

where λ is a lengthscale hyperparameter and defined for each dimension of corresponding variables. Here, each dimension of $\mathbf{X}$ is generated independently given $\mathbf{U}$, and k'_{x_l} refers to the kernel function for the lth dimension of x. Intuitively, each kernel lengthscale determines the relative strength of influence of each variable's structural function arguments. For example, if $\lambda_{ty} >> \lambda_{xy_1}$, the covariance between instances (or counterfactuals) with similar treatments will be greater than the covariance between instances with similar values of covariate the first covariate.

In addition to placing Gaussian process priors on the functions in the structural causal model in Equation 4.1, we also place inverse-gamma priors, $p(\theta) = \gamma^{-1}(\theta; \alpha_\theta, \beta_\theta)$ on each $\theta \in \Theta$, where Θ is the set of all kernel lengthscales, scaling factors, and

Algorithm 1 Individual Treatment Effect Estimation

```
procedure ITEE($\mathbf{t}_*, \mathbf{y}, \mathbf{t}, \mathbf{X}$)
    parameters: $\alpha_{\theta\in\Theta}, \beta_{\theta\in\Theta}$, prior hyperparameters; $\mathrm{n}_{\mathrm{Outer}}, \mathrm{n}_{\mathrm{MH}}, \mathrm{n}_{\mathrm{ES}}$ inference computation budget; $\mathrm{drift}_{\theta\in\Theta}$, Random walk proposal variance
    $\theta \sim \gamma^{-1}(\alpha_\theta, \beta_\theta), \forall \theta \in \Theta$    ▷ Kernel hyperparameter prior sample
    $\mathrm{U}_{o,j} \sim \mathcal{N}(0, \sigma^2_{\epsilon_{u_j}}), \forall o \in [\![n_o]\!], \forall j \in [\![n_u]\!]$    ▷ Confounder prior sample
    for $m = 1$ to $\mathrm{n}_{\mathrm{Outer}}$ do
        $\Theta \leftarrow$ HyperparameterUpdate(...)    ▷ Algorithm 2
        $\mathbf{U} \leftarrow$ ConfounderUpdate(...)    ▷ Algorithm 3
        $\mathbf{W}_{i,:} \leftarrow [\mathrm{t}_i, \mathrm{X}_{i,1}, \cdots, \mathrm{X}_{i,n_x}, \mathrm{U}_{o=pa(i),1}, \cdots, \mathrm{U}_{o=pa(i),n_u}], \forall i \in [\![n_i]\!]$
        $\mathbf{W}_{i,:,*} \leftarrow [\mathrm{t}_{i,*}, \mathrm{X}_{i,1}, \cdots, \mathrm{X}_{i,n_x}, \mathrm{U}_{o=pa(i),1}, \cdots, \mathrm{U}_{o=pa(i),n_u}], \forall i \in [\![n_i]\!]$
        $\mu_{ITE} \leftarrow (\boldsymbol{K'}(\mathbf{W}, \mathbf{W}_*)$ - $\boldsymbol{K'}(\mathbf{W}, \mathbf{W}))\boldsymbol{K}(\mathbf{W}, \mathbf{W})^{-1}\mathbf{y}$
        $ITE_{t_*,m} \sim \mathcal{N}(\mu_{ITE}, \Sigma_{ITE})$    ▷ See text for $\Sigma_{ITE}$
    end for
    return $ITE$
```

exogenous noise variances. In Section 4.3.1 we show how to perform approximate posterior inference on Θ.

4.2.1 Conditional Density

As f_y, f_t, and f_x are all drawn from Gaussian process priors, $p(\mathbf{y}|\mathbf{t}, \mathbf{X}, \mathbf{U}, \Theta)$, $p(\mathbf{t}|\mathbf{X}, \mathbf{U}, \Theta)$, and $p(\mathbf{X}_{:,l}|\mathbf{U}, \Theta)$ are all multivariate Gaussian distributed with mean zero and covariance given by their respective kernel covariance matrices. For example, $p(\mathbf{t}|\mathbf{X}, \mathbf{U}, \Theta) = \mathcal{N}(\mathbf{t}; 0, \mathbf{K}_t)$, where $\mathrm{K}_{t,i,i'} = k_t([\mathbf{U}_{o,:}, \mathbf{X}_{i,:}], [\mathbf{U}_{o',:}, \mathbf{X}_{i',:}])$. As $\mathrm{U}_{o,j}$ is given by the identity function of independent exogenous Gaussian noise, $p(\mathrm{U}_{o,j}|\Theta)$ $= \mathcal{N}(\mathrm{U}_{o,j}; 0, \sigma^2_{\epsilon_{u_j}})$. Therefore, the joint density is given by the following, which we use in Algorithms 2 and 3:

$$p(\mathbf{y}, \mathbf{t}, \mathbf{X}, \mathbf{U}, \Theta) = p(\mathbf{y}|\mathbf{t}, \mathbf{X}, \mathbf{U}, \Theta)p(\mathbf{X}|\mathbf{U}, \Theta)p(\mathbf{t}|\mathbf{X}, \mathbf{U}, \Theta)(\prod_{o=1}^{n_o}\prod_{j=1}^{n_u} p(\mathrm{U}_{o,j}|\Theta))p(\Theta). \tag{4.4}$$

By placing Gaussian process priors over each function in the hierarchical structural model, we encode our assumptions about which configurations of observed and latent variables are reasonable a-priori. Using a radial basis function kernel, we assume that if two objects have similar object-level latent confounders, they are likely to induce similar distributions over observed covariates, treatment, and outcome. Placing higher density on smooth structural causal functions in this way enables inference over object-level confounders.

4.3 Estimating Treatment Effects

In this section we describe how to estimate the *individual treatment effect*, $ITE_{i,t_*} = \mathrm{y}_i(\mathrm{t}_*) - \mathrm{y}_i$, the difference between observed and counterfactual outcomes for the ith instance. Standard aggregate measures of causal effect, such as the *sample average treatment effect*, $SATE_{t_*} = \frac{1}{n_i}\sum_{i=1}^{n_i} ITE_{i,t_*}$, can be derived from the individual treatment effect. We use ITE_{t_*} to denote the vector of individual treatment effects corresponding to the vector of counterfactuals $\mathbf{y}(\mathbf{t}_*)$.

First, note that when exogenous noise is additive in f_y, i.e $f_y(\mathbf{U}_{o=Pa(i),:}, \mathbf{X}_{i,:}, \mathrm{t}_i, \epsilon_{t_i}) = f'_y(\mathbf{U}_{o=Pa(i),:}, \mathbf{X}_{i,:}, \mathrm{t}_i) + g(\epsilon_{t_i})$, as in the GP-SLC model, individual treatment effect is given by the difference between noise-free functions $ITE_{i,t_*} = f'_y(\mathbf{U}_{o=Pa(i),:}, \mathbf{X}_{i,:}, \mathrm{t}_{i,*}) - f'_y(\mathbf{U}_{o=Pa(i),:}, \mathbf{X}_{i,:}, \mathrm{t}_i)$. We denote the outcome of these noise-free functions as $\mathrm{y}'_i(\mathbf{t}_*)$ and y'_i, and the vector of outcomes as $\mathbf{y}'(\mathbf{t}_*)$ and $\mathbf{y}'$ respectively.[1] As $\mathbf{U} \cup \mathbf{X}$ blocks all backdoor paths from $\mathbf{t}$ to $\mathbf{y}$, we have that the distribution over individual treatment effects is given by the following expression [97]:

$$\begin{aligned} p(ITE_{t_*}|\mathbf{y}, \mathbf{t}, \mathbf{X}) &= p(\mathbf{y}'(\mathbf{t}_*) - \mathbf{y}'|\mathbf{y}, \mathbf{t}, \mathbf{X}) \\ &= \int p(\mathbf{y}'(\mathbf{t}_*) - \mathbf{y}'|\mathbf{y}, \mathbf{t}, \mathbf{X}, \mathbf{U}, \Theta) p(\mathbf{U}, \Theta|\mathbf{y}, \mathbf{t}, \mathbf{X}) d\mathbf{U} d\Theta. \end{aligned} \tag{4.5}$$

Equation 4.5 directly informs our hybrid procedure for estimating counterfactual outcomes shown in Algorithm 1: (i) generate approximate samples from the posterior

[1] Noise-free prediction is often denoted as f in Gaussian process regression models. I avoid this notation to avoid confusion with functions in the structural causal model.

$\hat{\mathbf{U}}, \hat{\Theta} \sim p(\mathbf{U}, \Theta | \mathbf{y}, \mathbf{t}, \mathbf{X})$ and (ii) for each approximate posterior sample $\hat{\mathbf{U}}, \hat{\Theta}$ sample from the conditional distribution $p(\mathbf{y}'(\mathbf{t}_*) - \mathbf{y}' | \mathbf{y}, \mathbf{t}, \mathbf{X}, \hat{\mathbf{U}}, \hat{\Theta})$ in closed-form, taking advantage of Gaussian closure under conditioning and subtraction. As the posterior distribution $p(\mathbf{U}, \Theta | \mathbf{y}, \mathbf{t}, \mathbf{X})$ is intractable for non-trivial kernels, we turn to Monte Carlo approximate inference techniques.

4.3.1 Approximate Inference: Elliptical Slice and Metropolis-Hastings

Because we assume that our structural functions were drawn from Gaussian Processes, which provide a closed-form expression for the conditional density of the data, we are able to use standard likelihood-based approximate inference techniques. In our experiments, we approximate this posterior distribution using elliptical slice sampling [86] for the latent confounder, $\mathbf{U}$, and random walk Metropolis Hastings [53] on all kernel hyperparameters and exogenous noise variances, Θ. Pseudo-code implementations are presented in Algorithms 2 and 3.

4.3.2 Exact Inference: Gaussian Process Conditioning

To estimate $p(\mathbf{y}'(\mathbf{t}_*) - \mathbf{y}' | \mathbf{y}, \mathbf{t}, \mathbf{X}, \mathbf{U}, \Theta)$, we extend the Gaussian process model over in-sample and out-of-sample outcomes [103]. For compactness, we introduce the following shorthand:

$$\boldsymbol{W}_{i,:} = [\mathrm{t}_i, \mathrm{X}_{i,1}, \cdots, \mathrm{X}_{i,n_x}, \mathrm{U}_{o=pa(i),1}, \cdots, \mathrm{U}_{o=pa(i),n_u}]$$

$$\boldsymbol{W}_{*,i,:} = [\mathrm{t}_{i,*}, \mathrm{X}_{i,1}, \cdots, \mathrm{X}_{i,n_x}, \mathrm{U}_{o=pa(i),1}, \cdots, \mathrm{U}_{o=pa(i),n_u}]$$

The joint distribution over observed outcomes, $\mathbf{y}$, noise-free outcomes for each observed instance, $\mathbf{y}'$, and noise-free counterfactual outcomes, $\mathbf{y}'(\mathbf{t}_*)$, conditioned on observed treatments, $\mathbf{t}$, covariates, $\mathbf{X}$, inferred confounders, $\mathbf{U}$, and kernel hyperparameters, Θ, is Gaussian distributed as follows, where $\boldsymbol{K}(\mathbf{W}, \mathbf{W}) = \boldsymbol{K}'(\mathbf{W}, \mathbf{W}) + \sigma_y^2 \boldsymbol{I}_{n_i}$ and $\boldsymbol{K}'(\mathbf{W}, \mathbf{W})$ is the kernel matrix of k_y' given Θ:

$$\left(\begin{bmatrix}\mathbf{y}\\ \mathbf{y}'\\ \mathbf{y}'(\mathbf{t}_*)\end{bmatrix}|\mathbf{t},\mathbf{X},\mathbf{U},\Theta\right)\sim\mathcal{N}\left(0,\begin{bmatrix}\boldsymbol{K}(\mathbf{W},\mathbf{W}) & \boldsymbol{K}'(\mathbf{W},\mathbf{W}) & \boldsymbol{K}'(\mathbf{W},\mathbf{W}_*)\\ \boldsymbol{K}'(\mathbf{W},\mathbf{W}) & \boldsymbol{K}'(\mathbf{W},\mathbf{W}) & \boldsymbol{K}'(\mathbf{W},\mathbf{W}_*)\\ \boldsymbol{K}'(\mathbf{W}_*,\mathbf{W}) & \boldsymbol{K}'(\mathbf{W}_*,\mathbf{W}) & \boldsymbol{K}'(\mathbf{W}_*,\mathbf{W}_*)\end{bmatrix}\right) \tag{4.6}$$

Conditioning the joint distribution in Equation 4.6 on $\mathbf{y}$, we have the following:

$$\left(\begin{bmatrix}\mathbf{y}'\\ \mathbf{y}'(\mathbf{t}_*)\end{bmatrix}\middle|\mathbf{y},\mathbf{t},\mathbf{X},\mathbf{U},\Theta\right)\sim\mathcal{N}\left(\begin{bmatrix}\mu_1\\ \mu_2\end{bmatrix},\begin{bmatrix}\Sigma_{1,1} & \Sigma_{1,2}\\ \Sigma_{2,1} & \Sigma_{2,2}\end{bmatrix}\right)$$

where,

$$\begin{aligned}
\mu_1 &= \boldsymbol{K}'(\boldsymbol{W},\boldsymbol{W})\boldsymbol{K}(\boldsymbol{W},\boldsymbol{W})^{-1}\mathbf{y}\\
\mu_2 &= \boldsymbol{K}'(\boldsymbol{W},\boldsymbol{W}_*)\boldsymbol{K}(\boldsymbol{W},\boldsymbol{W})^{-1}\mathbf{y}\\
\Sigma_{1,1} &= \boldsymbol{K}'(\boldsymbol{W},\boldsymbol{W}) - \boldsymbol{K}'(\boldsymbol{W},\boldsymbol{W})\boldsymbol{K}(\boldsymbol{W},\boldsymbol{W})^{-1}\boldsymbol{K}'(\boldsymbol{W},\boldsymbol{W})\\
\Sigma_{1,2} &= \boldsymbol{K}'(\boldsymbol{W},\boldsymbol{W}_*) - \boldsymbol{K}'(\boldsymbol{W},\boldsymbol{W})\boldsymbol{K}(\boldsymbol{W},\boldsymbol{W})^{-1}\boldsymbol{K}'(\boldsymbol{W},\boldsymbol{W}_*)\\
\Sigma_{2,1} &= \boldsymbol{K}'(\boldsymbol{W}_*,\boldsymbol{W}) - \boldsymbol{K}'(\boldsymbol{W}_*,\boldsymbol{W})\boldsymbol{K}(\boldsymbol{W},\boldsymbol{W})^{-1}\boldsymbol{K}'(\boldsymbol{W},\boldsymbol{W})\\
\Sigma_{2,2} &= \boldsymbol{K}'(\boldsymbol{W}_*,\boldsymbol{W}_*) - \boldsymbol{K}'(\boldsymbol{W}_*,\boldsymbol{W})\boldsymbol{K}(\boldsymbol{W},\boldsymbol{W})^{-1}\boldsymbol{K}(\boldsymbol{W},\boldsymbol{W}_*)
\end{aligned}$$

Finally, as the difference of variables that are jointly Gaussian is Gaussian, we have that $(\mathbf{y}'(\mathbf{t}_*) - \mathbf{y}'|\mathbf{y},\mathbf{t},\mathbf{X},\mathbf{U},\Theta)$ is also Gaussian distributed:

$$\begin{aligned}
\mu_{ITE} &= \mu_2 - \mu_1\\
\Sigma_{ITE} &= \Sigma_{1,1} - \Sigma_{1,2} - \Sigma_{2,1} + \Sigma_{2,2}\\
(\mathbf{y}'(\mathbf{t}_*) - \mathbf{y}'|\mathbf{y},\mathbf{t},\mathbf{X},\mathbf{U},\Theta) &\sim \mathcal{N}(\mu_{ITE},\Sigma_{ITE})
\end{aligned} \tag{4.7}$$

Algorithm 2 Hyperparameter Update - Random Walk MH

```
procedure HyperparameterUpdate($\mathbf{y}, \mathbf{t}, \mathbf{X}, \mathbf{U}, \Theta$)
    parameters: $\alpha_{\theta\in\Theta}, \beta_{\theta\in\Theta}$, prior hyperparameters; $\mathrm{n}_{\mathrm{MH}}$ inference computation budget; $\mathrm{drift}_{\theta\in\Theta}$, Random walk proposal variance
    for $j = 1$ to $\mathrm{n}_{\mathrm{MH}}$ do
        for $\theta \in \Theta$ do
            $\alpha_{\theta'} \leftarrow \theta^2/\mathrm{drift}_\theta$
            $\beta_{\theta'} \leftarrow \theta(\alpha_{\theta'} - 1)$
            $\theta' \sim \gamma^{-1}(\alpha_{\theta'}, \beta_{\theta'})$    ▷ Sample $\alpha$ from an $\gamma^{-1}$ with mean $\theta$ and variance $\mathrm{drift}_\theta$
            $\alpha_\theta \leftarrow \theta'^2/\mathrm{drift}_\theta$    ▷ Compute inverse proposal distribution parameters
            $\beta_\theta \leftarrow \theta'(\alpha_\theta - 1)$    ▷ Compute inverse proposal distribution parameters
            $\Theta' \leftarrow \Theta \setminus \theta \cup \theta'$
            $a \leftarrow \frac{p(\mathbf{y}, \mathbf{t}, \mathbf{X}, \mathbf{U}, \Theta')}{p(\mathbf{y}, \mathbf{t}, \mathbf{X}, \mathbf{U}, \Theta)} \frac{\gamma^{-1}(\theta'; \alpha_{\theta'}, \beta_{\theta'})}{\gamma^{-1}(\theta; \alpha_\theta, \beta_\theta)}$    ▷ Compute MH acceptance probability
            $\eta \sim \mathrm{Uniform}(0, 1)$
            if $\eta > \min(a, 1)$ then
                $\Theta = \Theta'$    ▷ Accept proposal
            end if
        end for
    end for
    return $\Theta$
```

4.4 Asymptotic Posterior Consistency

In the special case where each kernel in the GP-SLC model is instead replaced with a linear kernel, $k(\mathbf{a}, \mathbf{a}) = \mathbf{a} \cdot \mathbf{a}^\top$, shared confounding among instances enables asymptotically consistent estimates of individual treatment effect. This is contrasted with the propositional setting (i.e. $n_o = n_i$) which does not lead to asymptotically consistent counterfactual estimation. Informally, a continuous random variable ψ is asymptotically consistent if its posterior $p(\psi|\mathbb{V})$ approaches a Dirac-delta distribution at some point ψ', regardless of the prior $p(\psi)$.

The analysis in this section follows the setup presented in D'Amour [30], with the inclusion of shared latent confounding among individual instances. We omit covariates $\mathbf{x}$ from this analysis and assume that $n_u = 1$ for brevity without loss of generality.

Algorithm 3 Confounder Update - Elliptical Slice Sampling

```
procedure ConfounderUpdate($\mathbf{y}, \mathbf{t}, \mathbf{X}, \mathbf{U}, \Theta$)
    parameters: prior hyperparameters; $\mathrm{n_{ES}}$ inference computation budget
    for $m = 1$ to $\mathrm{n_{ES}}$ do
        for $j = 1$ to $\mathrm{n_U}$ do
            done ← False
            $\nu \sim \mathcal{N}(0, \sigma^2_{\epsilon_{U_j}} \boldsymbol{I})$
            $y \sim \text{Uniform}(0, p(\mathbf{y}, t, \mathbf{X}, \mathbf{U}, \Theta))$
            $\phi \sim \text{Uniform}(0, 2\pi)$
            $[\phi_{\min}, \phi_{\max}] \leftarrow [\phi - 2\pi, \phi]$
            while not done do
                $\mathbf{U}'_{:,j} \leftarrow \mathbf{U}_{:,j} \cos\phi + \nu \sin\phi$
                if $p(\mathbf{y}, \mathbf{t}, \mathbf{X}, \mathbf{U}, \Theta) > y$ then
                    $\mathbf{U}_{:,j} \leftarrow \mathbf{U}'_{:,j}$
                    done ← True
                else
                    if $\phi < 0$ then $\phi_{\min} \leftarrow \phi$ else $\phi_{\max} \leftarrow \phi$
                    $\phi \sim \text{Uniform}(\phi_{\min}, \phi_{\max})$
                end if
            end while
        end for
    end for
    return $\mathbf{U}'$
```

4.4.1 Setup.

Assuming linear kernels and additive Gaussian exogenous noise, we can equivalently rewrite the GP-SLC model as follows. This equivalent structural causal model is parameterized by latent variables $\alpha, \beta, \tau \in \mathbb{R}$ and $\sigma_u^2, \sigma_t^2, \sigma_y^2 \in \mathbb{R}^+$.

$$\begin{aligned} \epsilon_{u_o} &\sim \mathcal{N}(0, \sigma_u^2) \quad \mathrm{u}_o = \epsilon_{u_o} \\ \epsilon_{t_i} &\sim \mathcal{N}(0, \sigma_t^2) \quad \mathrm{t}_i = \alpha \mathrm{u}_{o=Pa(i)} + \epsilon_{t_i} \\ \epsilon_{y_i} &\sim \mathcal{N}(0, \sigma_y^2) \quad \mathrm{y}_i = \beta \mathrm{t}_i + \tau \mathrm{u}_{o=Pa(i)} + \epsilon_{y_i}. \end{aligned} \tag{4.8}$$

In this setting, estimating individual treatment effect reduces to estimating β, as $\mathrm{y}_i(\mathrm{t}_*) - \mathrm{y}_i = \beta(\mathrm{t}_* - \mathrm{t}_i)$ for all $o \in [\![n_o]\!]$ and $i \in [\![n_i]\!]$.

Proposition 4.4.1. *When $n_o = n_i$, ITE$_{\mathrm{t}_*}$ is not asymptotically consistent* $\forall \mathrm{t}_* \in \mathbb{R}$.

For a detailed proof of Proposition 4.4.1, see Proposition 1 in D'Amour [30]. In summary, they show that given any set of latent parameters $\Theta = (\alpha, \beta, \tau, \sigma_u^2, \sigma_t^2, \sigma_y^2)$, there exists an alternative set of parameters Θ' such that $p(\mathbf{t}, \mathbf{y}|\Theta) = p(\mathbf{t}, \mathbf{y}|\Theta')$ and $\beta \neq \beta'$. In other words, the structural causal model forms a linear system of equations that is rank-deficient. The set of parameters that satisfy this condition construct an *ignorance region.*

Extending their results to the Bayesian setting, we have that for any two sets of parameters Θ and Θ' on the same ignorance region, the posterior odds ratio reduces to the prior odds ratio, $\frac{p(\Theta|\mathbf{t},\mathbf{y})}{p(\Theta'|\mathbf{t},\mathbf{y})} = \frac{p(\Theta)p(\mathbf{t},\mathbf{y}|\Theta)}{p(\Theta')p(\mathbf{t},\mathbf{y}|\Theta')} = \frac{p(\Theta)}{p(\Theta')}$. By definition, Θ is not asymptotically consistent, as the posterior $p(\Theta|\mathbf{t}, \mathbf{y})$ depends on the prior $p(\Theta)$. The problem of *asymptotic consistency* can be mitigated when $n_o < n_i$.

Theorem 4.4.2. *Assume there exists an object o that is the parent of n instances, $I' = \{i'_1, ..., i'_n\}$. Then ITE_{t_*} is asymptotically consistent as n approaches $\infty, \forall t_* \in \mathbb{R}$.*

Proof. For all $i' \in I'$, we have that $\mathrm{y}_{i'} = \beta \mathrm{t}_{i'} + C + \epsilon_{y_{i'}}$ for some constant $C \in \mathbb{R}$. Therefore, the covariance between $\mathbf{t}$ and $\mathbf{y}$ in I' is uniquely given by β, i.e. $cov(\mathrm{t}_{i' \in I'}, \mathrm{y}_{i' \in I'}) = \beta$. Estimating the covariance of a bivariate normal has a unique maximum likelihood solution. Therefore, by the Bernstein-von Mises Theorem [34] we have that the posterior over β, and thus ITE_{t_*}, is asymptotically consistent as n approach ∞. □

Theorem 4.4.3. *Assume there exist n objects, $\mathbb{O} = \{o_1, ..., o_n\}$, each of which are the unique parents of $k \geq 2$ instances $I'_o = \{i'_{o,1}, ..., i'_{o,k_o}\}$. Then ITE_{t_*} is asymptotically consistent as n approaches $\infty, \forall t_* \in \mathbb{R}$.*

Proof. For all $o \in \mathbb{O}$, $j \in [\![k_o]\!]$ let the $\mathrm{t}'_{i'_{o,j}}$ and $\mathrm{y}'_{i'_{o,j}}$ be treatment and outcome respectively normalized by the sample average over all instances that share a parent object. Specifically:

$$\mathrm{t}'_{i'_{o,j}} = \mathrm{t}_{i'_{o,j}} - \bar{\mathrm{t}}_o \tag{4.9}$$

$$\mathrm{y}'_{i'_{o,j}} = \mathrm{y}_{i'_{o,j}} - \bar{\mathrm{y}}_o \tag{4.10}$$

where,

$$\bar{\mathrm{t}}_o = \sum_{j=1}^{k_o} \mathrm{t}_{i'_{o,j}} / k_o \tag{4.11}$$

$$\bar{\mathrm{y}}_o = \sum_{j=1}^{k_o} \mathrm{y}_{i'_{o,j}} / k_o \tag{4.12}$$

Therefore, by the structural equation for t in Equation 4.8, we have the following:

$$\begin{aligned} \mathrm{t}'_{i'_{o,j}} &= \alpha \mathrm{u}_o + \epsilon_{t_{i'_{o,j}}} - \sum_{j=1}^{k_o} (\alpha \mathrm{u}_o + \epsilon_{t_{i'_{o,j}}}) / k_o && (4.13) \\ &= \epsilon_{t_{i'_{o,j}}} - \sum_{j=1}^{k_o} \epsilon_{t_{i'_{o,j}}} / k_o && (4.14) \end{aligned}$$

and similarly for y:

$$\begin{aligned} \mathrm{y}'_{i'_{o,j}} &= \beta(\alpha \mathrm{u}_o + \epsilon_{t_{i'_{o,j}}}) + \tau \mathrm{u}_o + \epsilon_{y_{i'_{o,j}}} - \sum_{j=1}^{k_o} (\beta(\alpha \mathrm{u}_o + \epsilon_{t_{i'_{o,j}}}) + \tau \mathrm{u}_o + \epsilon_{y_{i'_{o,j}}}) / k_o && (4.15) \\ &= \beta \mathrm{t}'_{i'_{o,j}} + \epsilon_{y_{i'_{o,j}}} - \sum_{i}^{k_o} \epsilon_{y_{i'_{o,j}}} / k_o && (4.16) \end{aligned}$$

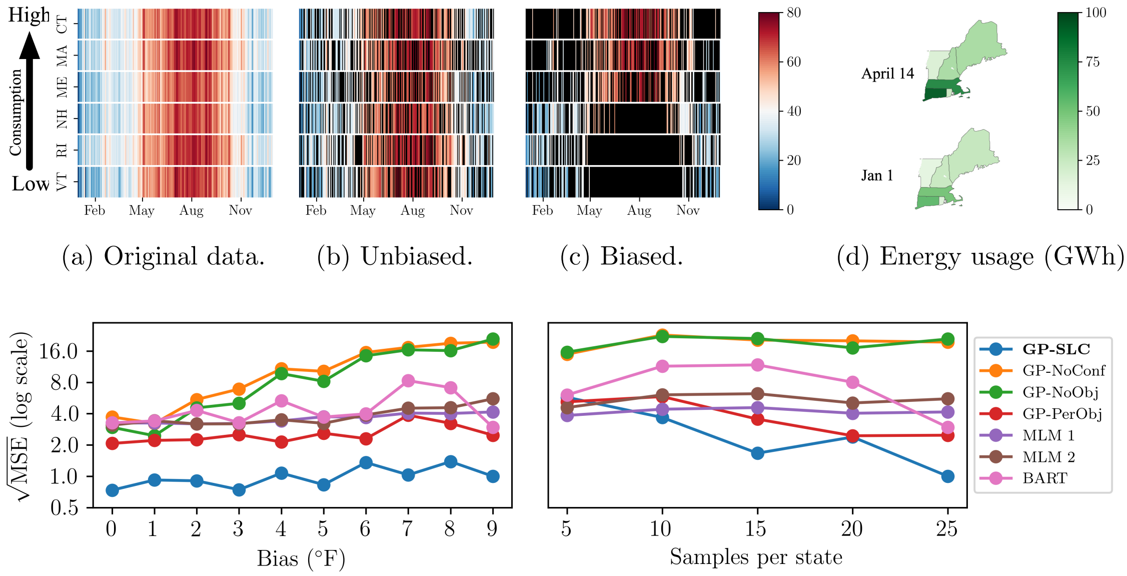
As $\epsilon_{y_{i'_{o,j}}}$ is independent of $\mathrm{t}'_{i'_{o,j}}$, we have that the covariance between $\mathrm{t}'_{i'_{o,j}}$ and $\mathrm{y}'_{i'_{o,j}}$ is equal to β. Therefore, the problem of estimating β reduces to estimating the covariance of a bivariate normal distribution, $p(\mathbf{t}', \mathbf{y}')$, which has a unique maximum likelihood solution. As in the proof of Theorem 4.4.2, by the Bernstein-von Mises Theorem [34] we have that the estimate of β, and thus ITE_{t_*}, is asymptotically consistent as n approach ∞. $\square$

4.5 Experiments

Unlike associational models, which can be evaluated using accuracy on held-out test data, causal models produce predictions about unobserved *counterfactual* distributions. As a result, effective evaluation of causal models requires different methods [47]. We evaluate the GP-SLC model using three benchmarks with known counterfactual outcomes. In Section 4.5.1, we evaluate GP-SLC using a fully synthetic hierarchical data generating process. In Section 4.5.2 we modify the Infant Health and Development Program (IHDP) benchmark [54] to include hierarchical structure and latent confounders. In Section 4.5.3 we introduce and evaluate on a new benchmark task for observational causal inference with hierarchical data, predicting the effect of changes in temperature on state-wide electric energy consumption in New England (NEEC).

We implement the GP-SLC model using Gen [28]. Except where otherwise specified we set $n_u = 3$ and $\alpha_\theta = \beta_\theta = 4$ for each inverse gamma prior over kernel hyperparameters and exogenous noise variance. We estimate individual treatment effects using Algorithm 1, with $\mathrm{n}_{\mathrm{Outer}} = 5000$, $\mathrm{n}_{\mathrm{MH}} = 3$, $\mathrm{n}_{\mathrm{ES}} = 5$, and $\mathrm{drift}_\theta = 0.5, \forall \theta \in \Theta$.

We compare the GP-SLC model against six baselines: a GP regression model that ignores latent confounding variables (GP-NoConf), a GP-SLC model where each instance is incorrectly assigned a single object (GP-NoObj), a seperate GP regression model for each object (GP-PerObj), Bayesian additive regression trees (BART) [54], a random slope and intercepts linear model (MLM 1), and a random intercepts linear model (MLM 2) [43]. The Gaussian process baselines are ablations of the full GP-SLC



(a) Original data. (b) Unbiased. (c) Biased. (d) Energy usage (GWh)

(e) Mean squared error in estimated sample average treatment effect.

Figure 4.2: **Process and results for New England energy consumption benchmark.** We sample hotter days with higher probability for states with higher daily energy consumption (a-d). Sampling in this way simulates confounding, creating an observational relationship (consumption is significantly higher in hotter days) that differs from the causal relationship (low or high temperature causes a moderate increase in energy consumption). GP-SLC (this chapter) produces accurate estimates of counterfactual outcomes, despite this confounding bias (e). For baselines that ignore hierarchical structure (GP-NoObj and GP-NoConf), accuracy decreases significantly with increasing confounding bias. Results are normalized by the $\sqrt{\text{MSE}}$ of the GP-SLC model with bias $= 9°F$ and 25 samples per state.

model, and use the same kernels, priors over hyperparameters, and inference scheme. The BART baseline uses the object identifier, o, as an additional covariate.

We use two evaluation metrics to evaluate GP-SLC and baselines, *mean squared error of the sample average treatment effect*, MSE $= \mathbb{E}_{t_*}[(SATE^*_{t_*} - SATE_{t_*})^2]$, and *precision in estimation of heterogeneous effect* [54], PEHE $= \mathbb{E}_{t_*}[\sum_i^{N_i}(ITE^*_{i,t_*} - ITE_{i,t_*})^2/N_i]$, where ITE^*_{i,t_*} and $SATE^*_{T_*}$ are the actual effects and ITE_{i,t_*} and $SATE_{t_*}$ are the predicted effects. For the synthetic benchmark, we average over 100 regular intervals between the 5th and 95th percentile of treatment assignment in the observational data. For the NEEC benchmark, we average over $\{30, 30.1, ..., 70°F\}$.

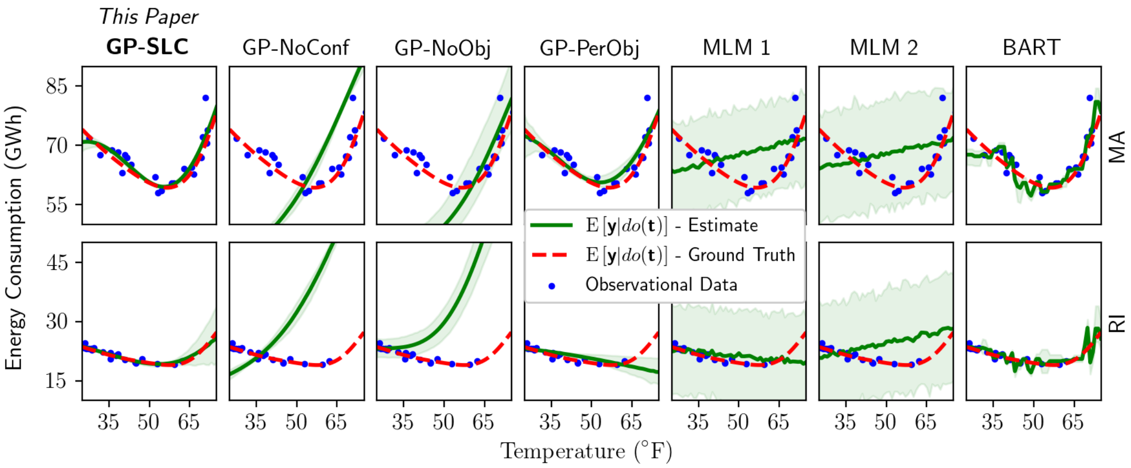


Figure 4.3: **Comparison among methods on the New England energy consumption benchmark.** Above are GP-SLC and all baselines' effect estimates on the NEEC benchmark with bias $= 9°F$ and 25 samples per state. Green shaded regions indicate 90% credible intervals. GP-SLC effectively recovers the effect of temperature on energy consumption, despite the latent confounding introduced by biased sampling. The best performing baseline, GP-PerObj, produces poor estimates of the effect of high temperatures in Rhode Island.

4.5.1 Synthetic Data

We evaluate GP-SLC and various baselines on two synthetic datasets with hierarchically structured latent confounders, one with additive and one with multiplicative treatment and outcome functions. Both synthetic datasets are generated using three dimensional object-level confounders for 20 objects, each of which contains 10 instances. Observed instance-level covariates are generated as a linear function of object-level Gaussian distributed latent confounders. Details for synthetic treatment and outcome functions are presented in the supplementary materials, and evaluation results are shown in Table 4.1. GP-SLC consistently matches and exceeds the counterfactual prediction performance of the six baselines on synthetic data. Baselines that ignore object structure (GP-NoConf and GP-NoObj) produce the least accurate counterfactual predictions.

In addition to the synthetic experiments presented in Table 4.1, we tested the behavior of GP-SLC using two alternative synthetic data generating processes. On the first, a linear structural data generating process with shared confounding, GP-SLC produces comparable estimates to the multi-level model baselines. On the second, in which each object shares a common effect of treatment and outcome rather than a common cause, GP-SLC is not susceptible to collider bias [14, 39]. This empirical finding is consistent with recent theory on object conditioning [63].

4.5.2 Infant Health and Development Program

The IHDP benchmark [54] uses real data for treatments (whether a child receives high-quality child care and home visits from a trained provider) and covariates (birth weight, head circumference, etc.) from the 1992 Infant Health and Development Program [101] with a synthetic nonlinear outcome function. We modify the IHDP benchmark to simulate hierarchically structured data by randomly duplicating 30% of the data instances and reassigning the duplicate's treatment assignment to be the opposite of the original instance. To introduce variation between duplicated instances, we add noise to each individuals' continuous covariates from a $\mathcal{N}(0, \sigma_j^2)$, where σ_j^2 is 5% of the jth covariate's marginal variance. We obscure the remaining 15 categorical covariates, representing object-level latent confounding. Even though the 15 categorical covariates are obscured from the GP-SLC model, they are identical across duplicates, unlike the observed covariates. We then generate observed and

Model	Additive		Multiplicative	
	$\sqrt{\text{PEHE}}$	$\sqrt{\text{MSE}}$	$\sqrt{\text{PEHE}}$	$\sqrt{\text{MSE}}$
GP-SLC	**1.0**	**1.0**	**1.0**	1.0
GP-NoConf	21.3	25.3	4.2	7.6
GP-NoObj	22.2	27.0	4.5	8.1
GP-PerObj	3.7	3.4	1.1	**0.9**
MLM1	1.2	1.02	2.4	2.9
MLM2	1.3	1.6	4.4	9.3
BART	8.5	10.7	2.6	4.3

Table 4.1: **Results on synthetic data with additive and multiplicative nonlinear treatment and outcome functions.** Scores are normalized by the score of GP-SLC. Lower is better.

counterfactual outcomes using the benchmark synthetic outcome function, applied to treatment, modified covariates, and latent confounders. In this setting, $Pa(i) = Pa(i')$ if instance i is a duplicate of instance i' or vice versa. Although each duplicate's treatment assignment is deterministic, the overall relationship between treatment and outcome is still confounded, as we only duplicate a subset of the original instances.

For the IHDP benchmark, which has binary treatment variables, we modify the GP-SLC model by replacing the expression $\mathrm{t}_i = f_t(\mathrm{u}_{o=Pa(i)}, \mathrm{x}_i, \epsilon_{t_i})$ with the expressions $\hat{\mathrm{t}}_i = f_{\hat{t}}(\mathbf{U}_{o=Pa(i),:}, \mathrm{X}_{i,:}, \epsilon_{\hat{t}_i})$ and $\mathrm{t}_i \sim \mathrm{Bernoulli}(\mathit{expit}(\hat{\mathrm{t}}_i))$. In this setting, we use elliptical slice sampling to approximate the latent logit probability of treatment, $\hat{\mathbf{t}}$.

Given the small size of each object, we omit the GP-PerObj baseline model from this evaluation. As the IHDP benchmark includes binary treatment variables we compared against four additional baselines: balanced linear regression (BalReg) and balanced neural nets (BALNN) [64], targeted maximum likelihood estimation with the superlearner (TMLE) [130], and inverse probability of treatment weighting with logistic regression (IPTW) [61].

Results of the IHDP evaluation are presented in Table 4.2. GP-SLC matches and exceeds the performance of other baselines when predicting the effect of assigning treatment to individuals who were previously untreated. In this setting, the linear models (MLM 1 and MLM 2) produce the least accurate counterfactual predictions.

4.5.3 New England Energy Consumption

We introduce a new benchmark for estimating heterogeneous effects in hierarchically structured settings, predicting the effect of changing temperature on state-wide electric energy consumption in New England. Unlike the evaluation in Section 4.5.2, which includes real treatments, covariates, and confounders and a synthetic outcome function, the New England energy consumption (NEEC) benchmark preserves outcome functions from real quasi-experimental data, and uses biased sampling to induce confounding.

Model	Control		Treated	
	$\sqrt{\text{PEHE}}$	$\sqrt{\text{MSE}}$	$\sqrt{\text{PEHE}}$	$\sqrt{\text{MSE}}$
GP-SLC	**1.0**	**1.0**	1.0	1.0
GP-NoConf	1.03	1.07	1.04	0.94
GP-NoObj	1.11	1.02	**0.82**	1.08
MLM1	68.3	33.2	106.7	1028.4
MLM2	73.3	389.1	45.8	63.2
BART	3.7	1.1	2.4	**0.33**
BALReg	5.1	82.7	1.9	0.5
BALNN	2.1	7.0	1.7	4.5
TMLE	n/a	209.8	n/a	12.2
IPTW	n/a	50.6	n/a	90.5

Table 4.2: **Results on the modified infant health and development program benchmark, shown separately for treated and untreated individuals.** Scores are normalized by the score of GP-SLC. TMLE and IPTW do not estimate individual treatment effects. Lower is better.

Specifically, we generate data for the NEEC benchmark task using the New England Independent Service Operator's public records on hourly dry-bulb temperature and state-wide energy consumption for the 2018 calendar year [40], which we then aggregate into daily averages.

While the marginal distribution over daily average temperature is nearly identical across states in the original dataset, the causal relationship between temperature and energy consumption differs across states, likely due to differences in population density, and commercial/industrial activity. To introduce confounding, we systematically sample days (instances) from states (objects) based on the state's typical energy consumption, including hotter days with higher probability for high consuming states. Specifically, we use importance resampling with a target distribution over Fahrenheit temperatures $T \sim \mathcal{N}(45 + bias \cdot s_o, 15)$, where $s_{CT} = 3, s_{MA} = 2, s_{ME} = 1, s_{NH} = -1, s_{RI} = -2, s_{VT} = -3$. An example of this sampling with bias = 9 is shown in Figure 4.2 (a-c). Biased sampling in this way introduces a statistical dependency across the dataset (consumption is significantly higher in hotter days), that differs from the causal relationship (low or high temperature causes a moderate increase in energy consumption). This approach of sampling quasi-experimental data to simulate confounding is an emerging standard in causal inference evaluation [47] although existing benchmarks are not hierarchically structured. Figure 4.2 (a-d) shows an example of this sampling process for the NEEC benchmark.

Sampling in this way does not provide instance-level counterfactual outcomes. Instead, we estimate the sample-average ground truth counterfactual outcome by fitting a Gaussian process regression model for each state, using treatments and outcomes from the entire calendar year.

Figure 4.2e shows the models' performances with varying degree of confounding and sample sizes, and Figure 4.3 shows the estimated and actual effect of temperature on electric energy consumption for two of the six states. Despite the induced confounding, GP-SLC consistently produces accurate estimates of causal effect. The baselines that ignore confounding (GP-NoConf and GP-NoObj) perform poorly as the degree of confounding increases, incorrectly attributing sample-wide association as indicative of causal effect. The linear multi-level models (MLM 1 and MLM 2) are not biased by confounding, but produce poor estimates due to their restrictive parametric assumptions. The remaining two baselines (GP-PerObj and BART) produce more accurate estimates than the other four baselines, but still overfit.

Model	CT	MA	ME	NH	RI	VT
GP-SLC	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	1.0
GP-NoConf	13.2	13	31.5	41.6	47.4	14.9
GP-NoObj	19.1	14	26.8	36.2	48	16.5
GP-PerObj	1.6	1.3	5.2	9.7	6.5	**0.7**
MLM1	6.9	5	25.0	5	5.1	**0.7**
MLM2	6.4	4.9	39.4	6.3	9.9	3.8
BART	4.1	2.1	13.3	3.6	3.3	2.4

Table 4.3: $\sqrt{\textbf{MSE}}$ **for the New England energy consumption benchmark, with bias** $= 9°F$ **and 25 samples per state.** Lower is better. Scores are normalized by GP-SLC's score for the same state.

4.5.4 Limitations

Despite the fact that GP-SLC produces state-of-the-art counterfactual predictions on most of our synthetic and semisynthetic benchmarks, it tends to underestimate the uncertainty in these estimates. In other words, the posterior density on the ground-truth counterfactual is sometimes low, despite the fact that the mean estimate is close to the ground-truth relative to the baselines. We suspect that this is partially attributable to inaccuracies resulting from our approximate inference procedure (Algorithms 2 and 3). Alternative approximate inference schemes, such as using our current approach as a rejuvenation move in a sequential Monte Carlo (SMC) algorithm [35], may resolve these inaccuracies. This kind of SMC-based inference procedure may also help GP-SLC scale to problems with more covariates and objects than we explore in this chapter.

Our empirical study focuses on data generating processes that satisfy GP-SLC's implicit semiparametric assumptions; (i) covariates for individuals belonging to the same object are marginally Gaussian distributed, and (ii) exogenous noise is additive and Gaussian. The effect of these modeling assumptions on counterfactual prediction and estimates of effect strength needs additional empirical characterization, ideally via large-scale synthetic experiments (where ground truth is known and robustness to modeling bias can be qualitatively studied).

4.6 Related Work

Leveraging hierarchical structure is well-established as a technique for adjusting for latent confounding [43, 45, 57]. Using Gaussian processes for causal inference is also well-established [2, 3, 116, 120, 143], as is the use of generative model approaches to adjust for latent confounders given restrictions on structure [78, 84, 128, 132]. To the best of our knowledge, GP-SLC is the first semiparametric generative modeling approach that leverages hierarchical structure to adjust for latent confounders.

GP-SLC is one of many recent techniques [64, 118] for estimating individual-level treatment effects. Prior work focuses on the propositional setting under *strong ignorability*, i.e. with no latent confounders. We focus on the hierarchical setting in which latent confounders are shared across multiple instances.

Recent work [116] has used Gaussian process models for causal inference in temporal settings, which assumes unconfoundedness and that the outcome is smooth with respect to time and covariates. GP-SLC allows for the existence of object-level latent confounders, and instead assumes that the outcome is smooth with respect to treatment assignment, covariates, and latent confounders. Longitudinal data analysis is closely related to the hierarchical settings we consider in this work: measurements (instances) of individuals (objects) are repeated over a period of time. Extending GP-SLC to the setting where latent confounders are not shared across instances, but instead change over time, is an exciting area of future work.

GP-SLC is most similar to recent work on multi-task GPs for causal inference [2], in that their approach also uses GP models to estimate individual treatment effects. However, GP-SLC: (i) handles hierarchical latent confounders by first performing inference over object-level latent variables; (ii) accounts for the covariance between noise-free factual and counterfactual outcomes (see Σ_{12} and Σ_{21}); and (iii) uses a Monte Carlo algorithm for inference that yields quantified uncertainty estimates. Their approach could be applied in hierarchical settings by treating the object identifier o as a categorical covariate and using a delta kernel to construct the outcome kernel covariance matrix. This is identical to the GP-PerObj baseline, except that GP-PerObj does not share inferred kernel hyperparameters across objects.

4.7 Conclusions

This chapter presents GP-SLC, a Gaussian process model for causal inference with hierarchically structured latent confounders. In Section 4.5, we show that, compared to widely used alternatives, GP-SLC produces more accurate estimates of causal effect in realistic sparse observational settings where strong prior knowledge about structure can inform causal estimates. The hierarchical structure we exploit in this chapter is one of many kinds of structural background knowledge that could improve causal estimates, and developing techniques to exploit such knowledge is an important area of future work. Extending GP-SLC to handle large observational datasets [23, 100] or to leverage experimental evidence [137] are also exciting areas of future work.

CHAPTER 5

MULTI-SOURCE EXPERIMENTAL DATA

In this chapter, we explore a new approach to implementing Bayesian causal inference based on probabilistic programming, inspired by Bayesian synthesis [114]. Probabilistic programming languages enable users to compactly specify probabilistic models in code. Some languages, like Stan [24], have syntax that closely resembles the statistical notation often used in the literature to define probabilistic models: a list of equations of the form $x \sim \dots$. Others, like Gen [28], allow users to include arbitrary program control flow in their models; a model is represented by a program that simulates stochastically from a distribution. In this chapter, we represent hypothesized causal models explaining some phenomenon as programs in *MiniStan*, a simple probabilistic programming language designed to resemble Stan (Figure 5.1). Then, we use a more expressive probabilistic programming language, Gen, to encode a prior and likelihood over MiniStan programs, and to do inference. The Gen model (i) stochastically generates MiniStan programs to encode a prior distribution over causal model structures and parameters, (ii) programmatically edits the generated MiniStan programs to reflect interventions and experimental conditions, then (iii) interprets the MiniStan programs to generate observational and experimental data. We can then use Gen's inference programming and conditioning features to condition the entire process on actual observational and experimental data, and to obtain posterior samples of the MiniStan code defining the original observational model—that is, to perform both structure learning and parameter estimation.

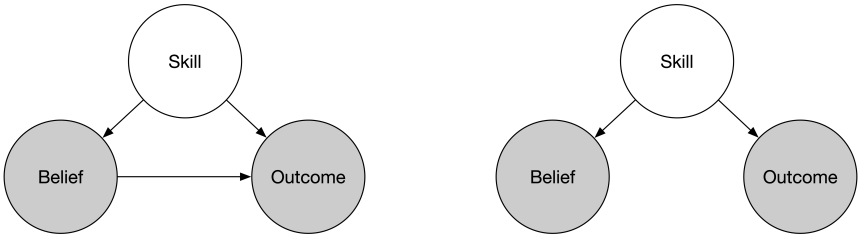
Causal models are typically structured as a set of autonomous components [4, 50, 94], such that interventions in the system can be accurately represented in the model as an alteration of a small number of model components, and all other model components (and the causal relationships among them) remain unchanged. In the formalism of causal graphical models, interventions are typically expressed using the do-operator [94], which fixes the value of one random variable and removes the influence of its parents. However, many realistic interventions are not accurately represented by this particular variety of model alteration [38, 68, 119]. For example, realistic interventions might best be represented by altering the functional form of a particular dependence, enabling or disabling specific causes, or enacting complex combinations of these interventions. This chapter demonstrates interventions represented as modifications of probabilistic program source code and shows how this representation enables the Bayesian synthesis approach to handle a broad class of experimental data.

5.1 A Conceptual Example

Consider the task of inferring whether a student's belief in her ability is causal for success at a research project. Observational data on student belief and student success alone are insufficient to answer this question, due to the confounding effect of skill (see Figures 5.2a and 5.2b).

$P \to S \mid S;\ P$	Programs
$S \to x = E \mid x \sim D$	Statements
$D \to \texttt{normal}(E, E) \mid \texttt{uniform}(E, E) \mid \texttt{bernoulli}(E)$	Distributions
$E \in$ deterministic Julia expressions	
$x \in$ Julia variable identifiers	

Figure 5.1: **Grammar of MiniStan**



(a) “Belief and skill matter” CGM.

(b) “Only skill matters” CGM.

```
quote
  s ∼ normal(mu_s, sigma_s)
  b ∼ normal(s, sigma_b)
  logit_o = s * lambda_so + b * lambda_bo
  o ∼ bernoulli(1/(1+exp(-logit_o)))
end
```

(c) “Belief and skill matter” model as source code.

```
quote
  s ∼ normal(mu_s, sigma_s)
  b ∼ normal(s, sigma_b)
  logit_o = s * lambda_so
  o ∼ bernoulli(1/(1+exp(-logit_o)))
end
```

(d) “Only skill matters” model as source code.

Figure 5.2: **A conceptual example combining structure learning and parameter estimation.**

We can imagine multiple types of experiments that would enable effective causal inference despite the confounding effect of skill. For example, an advisor could encourage a student, shifting her belief in her ability (but not increasing her skill). An advisor could also administer an assessment on the key skills needed for the project, before the student attempts it, and look at the results. Unfortunately, although this might reveal the true skill level to the advisor, this might also change the student’s belief in her own ability to succeed. Hypothetically, one can imagine a miracle pill that modifies one’s confidence to a fixed value, without changing anything else. Each of these experiments corresponds to a different modification to the source code from Figures 5.2c and 5.2d. Examples of these modifications are shown in Figures 5.3a-f.

This chapter shows how to formalize this example, using probabilistic programs that generate, edit, and interpret the source code of causal models. It also presents results from an implementation in the Gen probabilistic programming language, demonstrating the utility of incorporating diverse sources of experimental data.

5.2 Priors on Causal Models

To compute the posterior distribution over the two candidate causal models, we first specify a prior distribution over a set of global latent variables. One of these variables, *edge*, determines whether *Belief* influences *Outcome*.

$$\begin{array}{ccc} \mu_s \sim Normal(0,1) & \sigma_s \sim Uniform(0,1) & \sigma_b \sim Uniform(0,1) \\ \lambda_{so} \sim Uniform(0,1) & \lambda_{bo} \sim Uniform(0,1) & edge \sim Bernoulli(0.5) \end{array}$$

In the Bayesian synthesis framework, a prior distribution over causal models is a stochastic procedure generating programs in a domain specific language (Figure 5.5). The grammar for our simple domain specific language, MiniStan, is presented in Figure 5.1.

```
quote
  s ~ normal(mu_s, sigma_s)
  b = 5
  logit_o = s * lambda_so + b * lambda_bo
  o ~ bernoulli(1/(1+exp(-logit_o)))
end
```

(a) “Belief and skill matter” with belief pill.

```
quote
  s ~ normal(mu_s, sigma_s)
  b = 5
  logit_o = s * lambda_so
  o ~ bernoulli(1/(1+exp(-logit_o)))
end
```

(b) “Only skill matters” with belief pill.

```
quote
  s ~ normal(mu_s, sigma_s)
  b ~ normal(s + 3, sigma_b)
  logit_o = s * lambda_so + b * lambda_bo
  o ~ bernoulli(1/(1+exp(-logit_o)))
end
```

(c) “Belief and skill matter” with encouragement design.

```
quote
  s ~ normal(mu_s, sigma_s)
  b ~ normal(s + 3, sigma_b)
  logit_o = s * lambda_so
  o ~ bernoulli(1/(1+exp(-logit_o)))
end
```

(d) “Only skill matters” with encouragement design.

```
quote
  s ~ normal(mu_s + 2, sigma_s)
  b ~ normal(s, sigma_b / 100)
  logit_o = s * lambda_so + b * lambda_bo
  o ~ bernoulli(1/(1+exp(-logit_o)))
end
```

(e) “Belief and skill matter” with assessment.

```
quote
  s ~ normal(mu_s + 2, sigma_s)
  b ~ normal(s, sigma_b / 100)
  logit_o = s * lambda_so
  o ~ bernoulli(1/(1+exp(-logit_o)))
end
```

(f) “Only skill matters” with assessment.

Figure 5.3: **Interventions expressed as MiniStan source code transformations.**

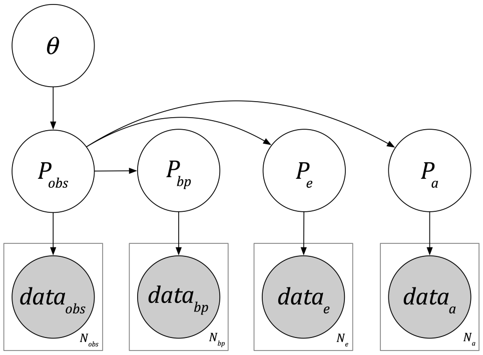


Figure 5.4: **Graphical meta-model for the Bayesian synthesis approach to causal structure and parameter learning.** A set of global parameters θ determine the source code of the observational causal program P_{obs}, which is modified via code-editing intervention functions to induce experimental causal programs for the belief-pill (P_{bp}) encouragement design (P_e), and the assessment (P_a) interventions. The code for each program is run through an interpreter, which generates (observational or experimental) data. The likelihoods of the various kinds of data under the different interpreted programs can be used to infer the posterior distribution over θ, and therefore over the observational causal program P_{obs}.

5.3 Likelihoods for Experiments

To incorporate experimental evidence of various forms, the Bayesian synthesis approach requires an intervention library which consists of a set of code-editing functions that modify causal model programs in the domain specific language. For the conceptual example, our intervention library contains three interventions: (i) an atomic intervention, which applies the do-operator; (ii) a shift intervention, which changes the mean of a distribution by a fixed increment; and (iii) a variance-scaling intervention, which modifies the variance of a random variable assumed to be drawn from a normal distribution. In principle, an intervention library could contain arbitrary rules for modifying causal model source code, including changing the underlying distribution for a random variable or adding variables (latent or observed) that didn't exist in the observational model.

These interventions can be freely composed to represent a diverse set of experimental scenarios. We demonstrate this compositionality in the "assessment" experiment, which is composed of a shift intervention (a student's skill may improve if she has to

```
@gen function generate_causal_model()
  mu_s = @trace(normal(0, 1), :mu_s)
  sigma_s = @trace(uniform(0, 1), :sigma_s)
  sigma_b = @trace(uniform(0, 1), :sigma_b)
  lambda_so = @trace(uniform(0, 1), :so_weight)
  lambda_bo = @trace(uniform(0, 1), :bo_weight)
  edge = @trace(bernoulli(0.5), :edge)

  if edge
    logit_o_expr = quote s * $so_weight + b * $bo_weight end
  else
    logit_o_expr = quote s * $so_weight end
  end

  causal_model = quote
    s ~ normal($mu_s, $sigma_s)
    b ~ normal(s, $sigma_b)
    logit_o = $logit_o_expr
    o ~ bernoulli(1/(1+exp(-logit_o)))
  end
  return causal_model
end
```

(a)

```
quote
  s ~ normal(0.237, 0.449)
  b ~ normal(s, 0.913)
  logit_o = s * 0.137 + b * 0.852
  o ~ bernoulli(1/(1 + exp(-logit_o)))
end
```

```
quote
  s ~ normal(-0.592, 0.302)
  b ~ normal(s, 0.724)
  logit_o = s * 0.503 + b * 0.491
  o ~ bernoulli(1/(1 + exp(-logit_o)))
end
```

```
quote
  s ~ normal(1.892, 0.108)
  b ~ normal(s, 0.301)
  logit_o = s * 0.542
  o ~ bernoulli(1/(1 + exp(-logit_o)))
end
```

(b)

```
1  @gen function generate_data(NObs, NBeliefPill, NEncouragement, NAssessment)
2    observational_model = @trace(generate_causal_model())
3    belief_pill_model = applyDoIntervention(observational_model, :b, 5)
4    encouragement_model = applyShiftIntervention(observational_model, :b, 3)
5    assessment_model = applyVarianceScalingIntervention(applyShiftIntervention(observational_model, :s, 2),
6                                                        :b, 1/100)
7
8    observational_data = @trace(interpretMiniStan(observational_model, n_runs=NObs), :obs)
9    belief_pill_data = @trace(interpretMiniStan(belief_pill_model, n_runs=NBeliefPill), :belief_pill)
10   encouragement_data = @trace(interpretMiniStan(encouragement_model, n_runs=NEncouragement), :encouragement)
11   assessment_data = @trace(interpretMiniStan(assessment_model, n_runs=NAssessment, :assessment)
12 end
```

(c)

Figure 5.5: **Gen implementation of causal inference via Bayesian synthesis.** The `generate_causal_model` Gen program (a) encodes a prior distribution over MiniStan models; (b) shows three samples from this prior. The `generate_data` Gen program (c) encodes the likelihood: it samples a possible causal model from the prior (line 2), modifies it to obtain MiniStan code representing experimental conditions (lines 3-6), then simulates observational and experimental data by running the MiniStan programs (lines 8-11). The interpreter is itself a Gen probabilistic program.

take a test) and a variance-scaling intervention (a student's belief in her ability has less noise after taking a test).

When interpreted, a causal program in MiniStan represents a likelihood function over observational data. To compute the likelihood of experimental data, we sim-

```julia
function applyDoIntervention(program, var, newValue)
  walk(program) do expr
    @match expr begin
      :($x = $val)  && if x == var end => :($var = $newValue)
      :($x ~ $dist) && if x == var end => :($var = $newValue)
      _ => expr
    end
  end
end
```

```julia
function applyShiftIntervention(program, var, shiftValue)
  walk(program) do expr
    @match expr begin
      :($x ~ normal($mean, $std)) && if x == var end => :($x ~ normal($mean + $shiftValue, $std))
      :($x ~ uniform($a, $b)) && if x == var end => :($x ~ uniform($a + $shiftValue, $b + $shiftValue))
      :($x = $value) && if x == var end => :($x = $value + $shiftValue)
      _ => expr
    end
  end
end
```

Figure 5.6: **Julia implementation of the atomic ("do") intervention and the shift intervention.** Rather than perform graph operations such as removing edges, an atomic intervention on a program walks the program's code and replaces any expression that assigns `var` with a new expression, implementing the intervention (`var = newValue`). The shift intervention walks the program's code and adds `shiftvalue` to the mean argument for the normal distribution, the lower and upper bound arguments for the uniform distribution, and the value of any deterministic assignment.

ply modify the causal program using the intervention library before subsequently interpreting the modified program.

5.4 Inference

We demonstrate the utility of this approach by performing approximate posterior inference over synthesized causal model programs from our conceptual example. In this example we: (i) generate a MiniStan program from the prior, (ii) generate a set of observational and experimental data from the interpreted MiniStan program, and (iii) perform approximate posterior inference over synthesized causal models using sequential Monte Carlo [35] with Metropolis Hastings rejuvenation. We generated ten individuals' skill, belief, and outcome for each of the four observational and experimental settings from a single causal model where $\mu_s = -0.013, \sigma_s = 0.776, \sigma_b = 0.646, \lambda_{so} = 0.734, \lambda_{bo} = 0.717$, and $edge = True$.

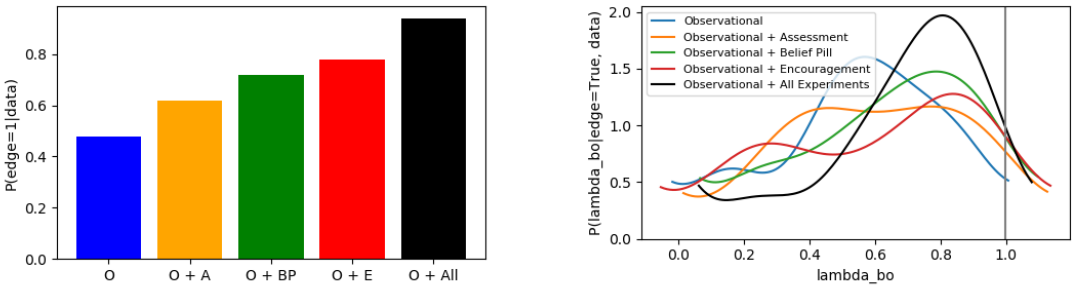


Figure 5.7: **Posterior probability of the existence and strength of causal dependence between a student's belief and her subsequent outcome.** The vertical gray line is the actual value for `lambda_bo`.

Using only observational data, the posterior probability of the edge variable is low. This may be because the data can be explained only by appealing to skill, and this simpler model could lead to a higher marginal probability than one which introduces a new parameter (`lambda_bo`). (This phenomenon is sometimes called "Bayesian Occam's Razor".) However, as we incorporate additional experimental evidence the posterior probability of the edge increases. Similarly, the posterior distribution over λ_{bo}, the effect of belief on outcome, concentrates around the true value as we leverage experimental evidence.

5.5 Discussion

The Bayesian synthesis approach we have outlined in this chapter provides several advantages over alternative approaches to structure discovery and parameter estimation in causal modeling: (i) an explicit characterization of uncertainty over model structures; (ii) a principled way to model diverse interventions; and (iii) a formalization that can be re-used in diverse problems, with varying degrees of prior knowledge, without requiring practitioners to design custom inferences for each use case.

Although this example uses parametric causal models, it is conceptually straightforward to use Gaussian processes and/or Dirichlet process mixture models for the functional forms of causal relationships [114]. It may thus be fruitful to develop Bayesian variants of existing non-parametric techniques for causal inference [62, 78].

The results reported here were obtained using vanilla sequential Monte Carlo over the joint space of model structure, parameters, and the latent variables in each observation or experiment. In order for this approach to scale to complex models, hierarchical priors over models, and large datasets, we expect more powerful techniques will be necessary. However, the Gen platform provides programmable inference constructs [28], including hybrids of Hamiltonian Monte Carlo [37] and Metropolis-Adjusted Langevin [108] approaches with sequential Monte Carlo [35], that could potentially address some of these scaling challenges.

5.6 Related Work

Probabilistic programs are often used to represent causal processes [48]. Some languages, such as Omega [124], make this causal interpretation explicit, including a semantics for interventional and counterfactual reasoning. It would be interesting to consider whether the framework we present here, which considers interventions to be arbitrary code-editing procedures, could also be usefully applied to counterfactual reasoning problems.

Incorporating experimental evidence for structure learning and parameter estimation can be thought of as the inner loop of an optimal experimental design procedure. Probabilistic programs have been used to automate this search over experiments [91], seeking to maximize the expected information gain over some query given new evidence. In that work, experiments are modeled as arguments to a probabilistic program. Our approach instead describes an experiment as a modification of MiniStan programs, enabling a clean abstraction between the specification of causal models (or distributions over causal models) and interventions that modify those models.

Improving methodology for combining observational and experimental evidence has far-reaching implications for a wide variety of scientific disciplines, and has received significant attention in the graph-based causal inference literature. For example, extensions of the do-calculus have been developed to incorporate experiments expressed as atomic interventions given a known causal graphical model structure [74]. Recent extensions of existing graph-based structure discovery algorithms have been made to incorporate atomic interventions [133] and imperfect interventions [141]. Our work proposes characterizing imperfect interventions as code-editors acting on probabilistic programs; this representation enables us to perform posterior inference (with uncertainty estimates) over both structure and model parameters.

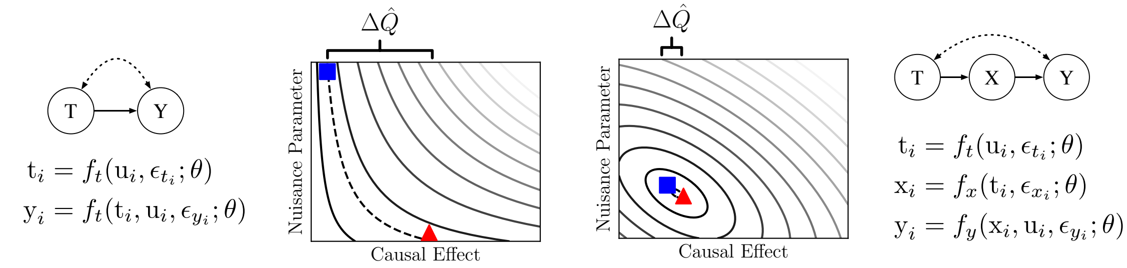
CHAPTER 6

SBI: A SIMULATION-BASED TEST OF IDENTIFIABILITY FOR BAYESIAN STRUCTURAL CAUSAL INFERENCE

Drawing causal conclusions from data requires assumptions about underlying causal mechanisms [97]. Consequently, it is important to determine when these assumptions are sufficient to answer a causal query, i.e. whether the query is *identifiable.* Existing computational methods, such as the do-calculus, can rigorously determine identifiability from graph structure alone [59, 93], however, graph structure alone can be incomplete in some cases. For example, instrumental variable designs require an assumption of monotonicity or linearity [26], within-subjects designs require an assumption that latent confounders are shared across units [43, 77], and regression discontinuity designs violate positivity, an assumption required by the do-calculus [73].

A growing body of causal inference research employs assumptions that go beyond graph structure. For example, researchers in causal machine learning [9, 52], and in hierarchical probabilistic modeling approaches to causal inference [21, 78, 128, 138], have achieved promising results. Some of these techniques can be expressed as priors over structural causal models, and implemented as probabilistic programs.

Unfortunately, it is difficult to apply either analytical or graphical techniques to determine the identifiability of complex Bayesian approaches to causal inference. As a result, these approaches can produce inaccurate effect estimates even with infinite data [30, 107]. This chapter introduces new automated techniques that can improve the rigor of causal inferences by providing simulation-based tests of identifiability (SBI).


T
Y
$t_i = f_t(u_i, \epsilon_{t_i}; \theta)$
$y_i = f_t(t_i, u_i, \epsilon_{y_i}; \theta)$
$\Delta\hat{Q}$
Nuisance Parameter
Causal Effect
$\Delta\hat{Q}$
Nuisance Parameter
Causal Effect
T
X
Y
$t_i = f_t(u_i, \epsilon_{t_i}; \theta)$
$x_i = f_x(t_i, \epsilon_{x_i}; \theta)$
$y_i = f_y(x_i, u_i, \epsilon_{y_i}; \theta)$

(a) Non-identifiable causal model and likelihood (b) Identifiable causal model and likelihood

Figure 6.1: **Overview of simulation-based identifiability.** Simulation-based identifiability (SBI) recasts causal identifiability as an optimization problem that seeks to maximize the data likelihood *and* the distance, $\Delta\hat{Q}$, between the effect estimates induced by two sets of parameters, $\theta^{(1)}$ and $\theta^{(2)}$. When causal effects are not identifiable (a) SBI discovers maximum likelihood parameters (blue and red) that estimate different causal effects. When causal effects are identifiable (b) the two models converge to the same effect estimates.

SBI is compatible with any prior over structural causal models that: (i) can be used to sample data; and (ii) induces a differentiable likelihood function. The key innovation is to reduce causal identification to an optimization procedure that maximizes the likelihood of two sets of parameters while also maximizing the distance between their causal effect estimates. If the optimal solution is two sets of parameters that agree on effect estimates, then the effect is identifiable. See Figure 6.1 for intuition.

In Section 6.3, we prove that SBI is asymptotically sound and complete, assuming certain (strong) regularity conditions. In Section 6.4, we show that SBI is broadly applicable by presenting a suite of compatible benchmarks reflecting common graph-based and quasi-experimental designs. We show empirically that SBI correctly determines whether average treatment effects are identifiable for all fourteen benchmarks. Finally, we use SBI to extract quantitative insight about Gaussian process regression discontinuity designs.

6.1 Related Work

Our work is not the first to automate identification for causal inference. Symbolic methods for observational [59, 93] and experimental [74] data determine whether queries are nonparametrically identifiable using graph structure alone. Similar methods have been developed for linear models [17, 69]. When applicable, symbolic methods like the do-calculus should be the de-facto choice, as they have strong theoretical guarantees, are computationally efficient, and require minimal ancillary assumptions. However, these approaches are inconclusive for more flexible parameterizations, such as those using Gaussian processes, or models employing non-graphical assumptions, such as within-subjects designs. These methods (and SBI) do not attempt to test whether a set of assumptions are satisfied given a particular dataset. Instead, they test whether assumptions are sufficient to uniquely determine a causal effect from (yet unseen) data.

Similar approaches for determining identifiability have been developed in other fields, such as neuroscience [129] and dynamical systems [105], by searching for likelihood equivalent parameters using gradient-based search. SBI differs from these approaches in two important ways. First, SBI uses a particle-based objective function to search for likelihood equivalent models globally, rather than locally near a single maximum likelihood solution. Second, SBI's objective function searches for models that estimate different causal effects, not only different parameters. This distinction means that SBI can correctly determine identifiability even when effects are composed of many parameters (e.g. see Section 6.3.2). It is well known that queries can be identified even in settings where individual parameters cannot [97]. Optimization techniques have been used to bound counterfactual queries [11, 126, 142] or for neural-causal models [139], but do not support user-specified parametric assumptions.

Bayesian priors over parametric structural causal models can be implemented in probabilistic programming languages [48, 83], which provide a syntax for expressing

Design	**Description**	**Source**
Unconfounded	No latent variables influence both treatment, $\mathbf{t}$, and outcome, $\mathbf{y}$.	[93]
Confounded	A latent confounder, $\mathbf{u}$, influences both $\mathbf{t}$ and $\mathbf{y}$.	[93]
Backdoor	An observed confounder, $\mathbf{x}$, influences $\mathbf{t}$ and $\mathbf{y}$.	[93]
Frontdoor	$\mathbf{u}$ influences $\mathbf{t}$ and $\mathbf{y}$, but does not influence a mediator, $\mathbf{x}$.	[93]
Instrumental variable	$\mathbf{u}$ influences $\mathbf{t}$ and $\mathbf{y}$. An observed instrument, $\mathbf{x}$, influences $\mathbf{t}$, does not influence $\mathbf{y}$ except through $\mathbf{t}$, and is not influenced by $\mathbf{u}$.	[8]
Within subjects	Each instance of $\mathbf{u}$ influences multiple instances of $\mathbf{t}$ and $\mathbf{y}$.	[36]
Regression discontinuity	An observed confounder, $\mathbf{x}$, influences $\mathbf{t}$ and $\mathbf{y}$. $\mathbf{t}$ is fully determined by $\mathbf{x}$ being above or below a known threshold.	[111]

Table 6.1: **Description of quasi-experimental designs benchmarks.** Of these seven standard causal designs, instrumental variable, within subjects, and regression discontinuity designs require assumptions that go beyond graph-structure. Parameterized versions of all seven designs can be represented as probabilistic programs, and can thus be tested using simulation-based identifiability.

probabilistic models as code. Many of these languages support automatic differentiation and gradient-based optimization [16, 24, 28, 31], providing the necessary utilities for our optimization-based approach. While some languages contain an explicit representation of interventions [16, 99, 124, 137], none currently address causal identifiability.

6.2 Identifiability in Bayesian Causal Inference

In this work we are interested in understanding the key asymptotic properties of the posterior distribution over causal effects, $p(Q|\mathbb{V})$, namely whether posterior mass concentrates around the true causal effect assymptotically. In other words, can the causal effect be identified from data? We define η-identifiability in this setting as follows:

Definition 6.2.1. ***η-identifiability.*** *Let $(\tilde{\mathbb{F}}, \tilde{\mathbb{U}})$ be a set of structural functions and latent confounders in the support of the prior, $p(\mathbb{F}, \mathbb{U})$. Then, a causal query, Q, is η-identifiable given $(\tilde{\mathbb{F}}, \tilde{\mathbb{U}})$ if for a dataset of n instances, $\tilde{\mathbb{V}} \sim p(\mathbb{V}|\tilde{\mathbb{F}}, \tilde{\mathbb{U}})$,*

$P(|\tilde{Q} - Q| \leq \eta|\tilde{\mathbb{V}}) \to 1$ for some $\eta \in \mathbb{R}^+$ almost surely as $n \to \infty$, where $\tilde{Q}$ is the causal effect induced by $(\tilde{\mathbb{F}}, \tilde{\mathbb{U}}, \tilde{\mathbb{V}})$.[1]

Even though Definition 6.2.1 is given in terms of an intractable posterior distribution, determining whether a causal effect is η-identifiable does not require computation or approximation of the posterior directly. Instead, we show that a causal query is η-identifiable if and only if there do not exist a set of maximum likelihood structural functions and latent confounder in the support of the prior, $(\mathbb{F}', \mathbb{U}')$, that induce causal effects that differ from $(\tilde{\mathbb{F}}, \tilde{\mathbb{U}})$ by more than η.

First, we prove a lemma that the likehood ratio uniformly converges to 0 or 1 asymptotically for any pair of SCMs.

Lemma 6.2.1. *For all $(\tilde{\mathbb{F}}, \tilde{\mathbb{U}}), (\mathbb{F}', \mathbb{U}')$ in the support of $p(\mathbb{F}, \mathbb{U})$, $p(\tilde{\mathbb{V}}|\mathbb{F}', \mathbb{U}')/p(\tilde{\mathbb{V}}|\tilde{\mathbb{F}}, \tilde{\mathbb{U}})$ converges uniformly to 0 or 1 almost surely as $n \to \infty$, where $\tilde{\mathbb{V}} \sim p(\mathbb{V}|\tilde{\mathbb{F}}, \tilde{\mathbb{U}})$.*

Proof. Let $r_i(\mathbb{F}', \mathbb{U}') := p(\tilde{\mathbb{V}}_i|\mathbb{F}', \mathbb{U}')/p(\tilde{\mathbb{V}}_i|\tilde{\mathbb{F}}, \tilde{\mathbb{U}}) \leq 1$ for a single data instance $\tilde{\mathbb{V}}_i$. As each element of ϵ is assumed to be independent and identically distributed, then $r_i(\mathbb{F}', \mathbb{U}') = r_j(\mathbb{F}', \mathbb{U}') = r(\mathbb{F}', \mathbb{U}')$ for all $i, j \in [\![n]\!]$. Therefore, $\mathbb{E}[p(\tilde{\mathbb{V}}|\mathbb{F}', \mathbb{U}')/p(\tilde{\mathbb{V}}|\tilde{\mathbb{F}}, \tilde{\mathbb{U}})] = r(\mathbb{F}', \mathbb{U}')^n$ for n i.i.d data instances. As $0 \leq r(\mathbb{F}', \mathbb{U}') \leq 1$, $r(\mathbb{F}', \mathbb{U}')^n \to 0$ or 1 uniformly as $n \to \infty$. By the weak law of large numbers, we have that $p(\tilde{\mathbb{V}}|\mathbb{F}', \mathbb{U}')/p(\tilde{\mathbb{V}}|\tilde{\mathbb{F}}, \tilde{\mathbb{U}}) \to 0$ or 1 almost surely for all $(\tilde{\mathbb{F}}, \tilde{\mathbb{U}}), (\mathbb{F}', \mathbb{U}')$ in the support of $p(\mathbb{F}, \mathbb{U})$ as $n \to \infty$. □

Theorem 6.2.2. *Q is η-identifiable given $(\tilde{\mathbb{F}}, \tilde{\mathbb{U}})$ if and only if for a dataset of n instances, $\tilde{\mathbb{V}} \sim p(\mathbb{V}|\tilde{\mathbb{F}}, \tilde{\mathbb{U}})$, there does not exist an $(\mathbb{F}', \mathbb{U}')$ such that $p(\tilde{\mathbb{V}}|\mathbb{F}', \mathbb{U}') = p(\tilde{\mathbb{V}}|\tilde{\mathbb{F}}, \tilde{\mathbb{U}})$, $|\tilde{Q} - Q'| > \eta$, and $p(\mathbb{F}', \mathbb{U}')/p(\tilde{\mathbb{F}}, \tilde{\mathbb{U}}) > 0$ almost surely as $n \to \infty$. Here, $\tilde{Q}$ and Q' are the causal effects induced by $(\tilde{\mathbb{F}}, \tilde{\mathbb{U}}, \tilde{\mathbb{V}})$ and $(\mathbb{F}', \mathbb{U}', \tilde{\mathbb{V}})$ respectively.*

Proof. Let $\mathcal{A}'$ and $\tilde{\mathcal{A}}$ be the set of $(\mathbb{F}, \mathbb{U})$ that induce the same effect as $(\mathbb{F}', \mathbb{U}')$ and $(\tilde{\mathbb{F}}, \tilde{\mathbb{U}})$ respectively and let $\mathbb{L}$ be the set of $(\mathbb{F}, \mathbb{U})$ that maximize the likelihood of the

[1] Importantly, $p(\tilde{Q}|\tilde{\mathbb{V}})$ marginalizes over $(\mathbb{F}, \mathbb{U})$, and does not condition on the "known" $(\tilde{\mathbb{F}}, \tilde{\mathbb{U}})$.

data asymptotically, i.e. $\{(\mathbb{F},\mathbb{U}) \in \text{supp}(p(\mathbb{F},\mathbb{U})) : \frac{p(\tilde{\mathbb{V}}|\mathbb{F},\mathbb{U})}{p(\tilde{\mathbb{V}}|\tilde{\mathbb{F}},\tilde{\mathbb{U}})} \to 1 \text{ as } n \to \infty\}$. To show that Q is η-identifiable only if there does not exist such an $(\mathbb{F}',\mathbb{U}')$, we have that for all $(\mathbb{F}',\mathbb{U}')$ in the support of $p(\mathbb{F},\mathbb{U}))$:

$$\begin{aligned}
\lim_{n\to\infty} p(Q'|\tilde{\mathbb{V}}) &= \lim_{n\to\infty} \frac{1}{p(\tilde{\mathbb{V}})} \int_{(\mathbb{F},\mathbb{U})\in\mathcal{A}'} p(\tilde{\mathbb{V}}|\mathbb{F},\mathbb{U}) p(\mathbb{F},\mathbb{U}) d\mathbb{F} d\mathbb{U} \\
&= \lim_{n\to\infty} \frac{p(\tilde{\mathbb{V}}|\tilde{\mathbb{F}},\tilde{\mathbb{U}})}{p(\tilde{\mathbb{V}})} \int_{(\mathbb{F},\mathbb{U})\in\mathcal{A}'} \frac{p(\tilde{\mathbb{V}}|\mathbb{F},\mathbb{U})}{p(\tilde{\mathbb{V}}|\tilde{\mathbb{F}},\tilde{\mathbb{U}})} p(\mathbb{F},\mathbb{U}) d\mathbb{F} d\mathbb{U} \\
&= \left(\lim_{n\to\infty} \frac{p(\tilde{\mathbb{V}}|\tilde{\mathbb{F}},\tilde{\mathbb{U}})}{p(\tilde{\mathbb{V}})}\right) \int_{(\mathbb{F},\mathbb{U})\in\mathcal{A}'} \lim_{n\to\infty} \frac{p(\tilde{\mathbb{V}}|\mathbb{F},\mathbb{U})}{p(\tilde{\mathbb{V}}|\tilde{\mathbb{F}},\tilde{\mathbb{U}})} p(\mathbb{F},\mathbb{U}) d\mathbb{F} d\mathbb{U} \\
&= \left(\lim_{n\to\infty} \frac{p(\tilde{\mathbb{V}}|\tilde{\mathbb{F}},\tilde{\mathbb{U}})}{p(\tilde{\mathbb{V}})}\right) \int_{(\mathbb{F},\mathbb{U})\in\mathcal{A}'\cap\mathbb{L}} \lim_{n\to\infty} p(\mathbb{F},\mathbb{U}) d\mathbb{F} d\mathbb{U}
\end{aligned}$$

Here, the limit can be moved inside the integrand by the bounded convergence theorem, as $\frac{p(\tilde{\mathbb{V}}|\mathbb{F},\mathbb{U})}{p(\tilde{\mathbb{V}}|\tilde{\mathbb{F}},\tilde{\mathbb{U}})} p(\mathbb{F},\mathbb{U})$ converges uniformly to $p(\mathbb{F},\mathbb{U})$ or 0. Therefore, we have that:

$$\lim_{n\to\infty} \frac{p(Q'|\tilde{\mathbb{V}})}{p(\tilde{Q}|\tilde{\mathbb{V}})} = \frac{\int_{(\mathbb{F},\mathbb{U})\in\mathcal{A}'\cap\mathbb{L}} \lim_{n\to\infty} p(\mathbb{F},\mathbb{U}) d\mathbb{F} d\mathbb{U}}{\int_{(\mathbb{F},\mathbb{U})\in\tilde{\mathcal{A}}\cap\mathbb{L}} \lim_{n\to\infty} p(\mathbb{F},\mathbb{U}) d\mathbb{F} d\mathbb{U}} > 0 \text{ if and only if } \mathcal{A}'\cap\mathbb{L} \neq \emptyset$$

Therefore, if there exists an $(\mathbb{F}',\mathbb{U}') \in \mathcal{A}'\cap\mathbb{L}$ such that $|Q' - \tilde{Q}| > \eta$, then Q is not η-identifiable. If no such $(\mathbb{F}',\mathbb{U}')$ exists, then Q is η-identifiable. □

Definition 6.2.1 describes identifiability with respect to a single instantiation, $(\tilde{\mathbb{F}},\tilde{\mathbb{U}})$. Instead, we would like to make statements about whether causal effects can be uniquely identified with high probability across SCMs sampled from the prior. Let $\text{ID}(\tilde{\mathbb{F}},\tilde{\mathbb{U}},\eta)$ be a function that returns 1 if Q is η-identifiable given $(\tilde{\mathbb{F}},\tilde{\mathbb{U}})$ under Definition 6.2.1, and 0 otherwise. Then, we define (ζ,η)-identifiability as follows:

Definition 6.2.2. (ζ, η)***-identifiability.*** *For some* $0 \leq \zeta \leq 1$, $\eta \in \mathbb{R}^+$, *a causal query,* Q, *is* (ζ, η)*-identifiable given a prior distribution* $p(\mathbb{F}, \mathbb{U})$ *if the probability that* Q *is* η*-identifiable given a* $(\tilde{\mathbb{F}}, \tilde{\mathbb{U}}) \sim p(\mathbb{F}, \mathbb{U})$ *is greater than or equal to* ζ, *i.e.* $\zeta \leq \int ID(\tilde{\mathbb{F}}, \tilde{\mathbb{U}}, \eta)\, dp(\tilde{\mathbb{F}}, \tilde{\mathbb{U}})$.[2]

6.2.1 Example: Confounded Linear Model

Here, we illustrate the Bayesian approach with a linear parametric example over observed $\mathbb{V} = \{\mathbf{t}, \mathbf{y}\}$ and latent $\mathbb{U} = \{\mathbf{u}\}$ and $\mathbb{X} = \{\epsilon_t, \epsilon_y\}$, which is a simplified version of the example in Section 5 of [30]. This example corresponds to the graphical structure shown in Figure 6.1a. Here, the structural causal model is parameterized by $\theta = \{\gamma, \beta, \alpha, \sigma_u^2, \sigma_t^2, \sigma_y^2\}$. Assume the following parameterized SCM:

$$\begin{aligned} &\mathrm{u}_i \sim \mathcal{N}(0, \sigma_u^2) && \mathrm{t}_i = \gamma \mathrm{u}_i + \epsilon_{\mathrm{t}_i} && \mathrm{y}_i = \beta \mathrm{t}_i + \alpha \mathrm{u}_i + \epsilon_{\mathrm{y}_i} \\ &\epsilon_{\mathrm{t}_i} \sim \mathcal{N}(0, \sigma_t^2) && \epsilon_{\mathrm{y}_i} \sim \mathcal{N}(0, \sigma_y^2) && \end{aligned}$$

Let our causal query again be the sample average treatment effect (SATE), i.e. $Q(\mathbf{y}, \mathbf{y}(\boldsymbol{t})) = \beta(\boldsymbol{t} - \sum_{i=1}^{n} \mathrm{t}_i)$. In this setting, estimating the causal effect reduces to estimating β. As shown in [30], for all $\tilde{\mathbb{V}}$ in the support of $p(\mathbb{V})$ there exists a set of parameters Θ such that for all $\theta^{(1)}, \theta^{(2)} \in \Theta$, $\beta^{(1)} \neq \beta^{(2)}$ and $p(\tilde{\mathbb{V}}|\theta^{(1)}) = p(\tilde{\mathbb{V}}|\theta^{(2)})$. In summary, the induced linear system of equations relating parameters to the observable covariance between $\mathbf{t}$ and $\mathbf{y}$ is rank deficient, leading to non-uniqueness of the maximum likelihood solution. This implies that the posterior odds ratio, $p(\beta^{(1)}|\tilde{\mathbb{V}})/p(\beta^{(2)}|\tilde{\mathbb{V}})$, reduces to the prior odds ratio, $p(\beta^{(1)})/p(\beta^{(2)})$, regardless of n [138]. It follows straightforwardly that given any non-degenerate prior, $p(\theta)$, Q is therefore not (ζ, η)-

[2] The standard Definition 3.2.4 in (Pearl, 2009) is equivalent to our Definition 6.2.2 with $\eta = 0, \zeta = 1$.

Algorithm 4 Simulation-Based Identifiability (SBI)

```
procedure SBI($p(\mathbb{F}, \mathbb{U}, \mathbb{V}), Q, \eta, \zeta$)
    parameters: $m$, SCM samples; $n$; dataset size; $k$, data samples; $\lambda$, repulsion strength,
        $\alpha$, significance level
    for $i = 1$ to $m$ do
        $\tilde{\mathbb{F}}_i, \tilde{\mathbb{U}}_i \sim p(\mathbb{F}, \mathbb{U})$      ▷ Sample structural functions and confounder instances.
        for $j = 1$ to $k$ do
            $\tilde{\mathbb{V}}_{i,j} \sim p(\mathbb{V} | \tilde{\mathbb{F}}_i, \tilde{\mathbb{U}}_i)$      ▷ Sample $n$ observations from the $i$'th SCM.
            $\Delta\hat{Q}_{i,j} \leftarrow$ Optimize $\mathcal{L}(\cdot, \tilde{\mathbb{V}}_{i,j}; \lambda)$
                                        ▷ Using stochastic gradient descent. See Equation 6.1
        end for
        $\hat{\mu}_i \leftarrow \sum_{j=1}^{k} \Delta\hat{Q}_{i,j}/k$      ▷ Compute sample mean for $i$'th SCM
        $\hat{S}_i \leftarrow \sum_{i=1}^{k} (\hat{\mu}_i - \Delta\hat{Q}_{i,j})^2/(k-1)$      ▷ Compute sample variance for $i$'th SCM
    end for
    $l_0 \leftarrow \max_{\zeta' \in [0,\zeta]} \sum_{i=1}^{m} \log(\mathcal{N}(\hat{\mu}_i; \min(\hat{\mu}_i, \eta), \hat{S}_i/k)\zeta' + \mathcal{N}(\hat{\mu}_i; \max(\hat{\mu}_i, \eta), \hat{S}_i/k)(1-\zeta'))$
                                        ▷ Null log likelihood
    $l_a \leftarrow \max_{\zeta' \in [0,1]} \sum_{i=1}^{m} \log(\mathcal{N}(\hat{\mu}_i; \min(\hat{\mu}_i, \eta), \hat{S}_i/k)\zeta' + \mathcal{N}(\hat{\mu}_i; \max(\hat{\mu}_i, \eta), \hat{S}_i/k)(1-\zeta'))$
                                        ▷ Alternative log likelihood
    return TRUE if $\chi^2(2(l_a - l_0); 1) < \alpha$ else return FALSE
                                        ▷ Chi-squared test of statistical significance
```

identifiable for any $0 \leq \zeta \leq 1$, $\eta \in \mathbb{R}^+$. This conclusion agrees with known parametric identification results [97].

6.3 Simulation-Based Identifiability

For the linear example in Section 6.2.1, we were able to relate the observable covariance between $\mathbf{t}$ and $\mathbf{y}$ to the latent parameters θ algebraically. However, it is not clear how we might derive similar results for nonlinear structural functions in general. Instead, we propose an approach for determining causal identifiability using a particle-based optimization scheme which we call simulation-based identifiability (SBI). In summary, SBI uses gradient-based search to find two sets of maximum likelihood structural functions and latent confounders in the support of $p(\mathbb{F}, \mathbb{U})$, $(\mathbb{F}^{(1)}, \mathbb{U}^{(1)})$ and $(\mathbb{F}^{(2)}, \mathbb{U}^{(2)})$, that induce different causal effects, $Q^{(1)}$ and $Q^{(2)}$, respectively. Let $\lambda \in \mathcal{R}^+$

be a hyperparameter and $\Delta Q \coloneqq |Q^{(1)} - Q^{(2)}|$. Then, consider the following objective function:

$$\mathcal{L}(\underbrace{\mathbb{F}^{(1)}, \mathbb{U}^{(1)}}_{\text{SCM 1}}, \underbrace{\mathbb{F}^{(2)}, \mathbb{U}^{(2)}}_{\text{SCM 2}}, \underbrace{\tilde{\mathbb{V}}}_{\text{Data}}; \lambda) = \underbrace{\log p(\tilde{\mathbb{V}}|\mathbb{F}^{(1)}, \mathbb{U}^{(1)})}_{\text{SCM 1 log likelihood}} + \underbrace{\log p(\tilde{\mathbb{V}}|\mathbb{F}^{(2)}, \mathbb{U}^{(2)})}_{\text{SCM 2 log likelihood}} + \underbrace{\lambda \Delta Q}_{\text{Repulsion}} \tag{6.1}$$

Let $\hat{\mathbb{F}}^{(1)}, \hat{\mathbb{U}}^{(1)}, \hat{\mathbb{F}}^{(2)}, \hat{\mathbb{U}}^{(2)}$ denote a solution that maximizes $\mathcal{L}$, and let $\Delta\hat{Q}$ be the corresponding optimal ΔQ.

To prove that SBI is asymptotically sound and complete we first prove that the optimal solutions to $\mathcal{L}$ are almost surely maximum likelihood solutions, and that $\hat{Q}$ is almost surely the maximum distance between causal effects among maximum likelihood solutions. Recall that $(\hat{\mathbb{F}}^{(1)}, \hat{\mathbb{U}}^{(1)})$ and $(\hat{\mathbb{F}}^{(2)}, \hat{\mathbb{U}}^{(2)})$ are solutions that maximize $\mathcal{L}$.

Lemma 6.3.1. *For a dataset of n instances $\tilde{\mathbb{V}} \sim p(\tilde{\mathbb{V}}|\tilde{\mathbb{F}}, \tilde{\mathbb{U}})$, $p(\tilde{\mathbb{V}}|\hat{\mathbb{F}}^{(1)}, \hat{\mathbb{U}}^{(1)})$ and $p(\tilde{\mathbb{V}}|\hat{\mathbb{F}}^{(2)}, \hat{\mathbb{U}}^{(2)})$ converge to $p(\tilde{\mathbb{V}}|\tilde{\mathbb{F}}, \tilde{\mathbb{U}})$ almost surely as $n \to \infty$.*

Proof. Without loss of generality, toward a contradiction assume that $p(\tilde{\mathbb{V}}|\hat{\mathbb{F}}^{(1)}, \hat{\mathbb{U}}^{(1)}) \not\to p(\tilde{\mathbb{V}}|\tilde{\mathbb{F}}, \tilde{\mathbb{U}})$ as $n \to \infty$. Therefore, by Lemma 6.2.1 we have that $\frac{p(\tilde{\mathbb{V}}|\hat{\mathbb{F}}^{(1)}, \hat{\mathbb{U}}^{(1)})}{p(\tilde{\mathbb{V}}|\tilde{\mathbb{F}}, \tilde{\mathbb{U}})} \to 0$ as $n \to \infty$. Therefore:

$$\mathcal{L}(\hat{\mathbb{F}}^{(1)}, \hat{\mathbb{U}}^{(1)}, \hat{\mathbb{F}}^{(2)}, \hat{\mathbb{U}}^{(2)}) \geq \mathcal{L}(\tilde{\mathbb{F}}, \tilde{\mathbb{U}}, \tilde{\mathbb{F}}, \tilde{\mathbb{U}}) \tag{6.2}$$

$$\log p(\tilde{\mathbb{V}}|\hat{\mathbb{F}}^{(1)}, \hat{\mathbb{U}}^{(1)}) + \log p(\tilde{\mathbb{V}}|\hat{\mathbb{F}}^{(2)}, \hat{\mathbb{U}}^{(2)}) + \lambda\Delta\hat{Q} \geq 2\log p(\tilde{\mathbb{V}}|\tilde{\mathbb{F}}, \tilde{\mathbb{U}}) + \lambda|\tilde{Q} - \tilde{Q}| \tag{6.3}$$

$$\geq 2\log p(\tilde{\mathbb{V}}|\tilde{\mathbb{F}}, \tilde{\mathbb{U}}) \tag{6.4}$$

Or equivalently, as $n \to \infty$:

$$
\begin{aligned}
0 &\leq \log p(\tilde{\mathbb{V}}|\hat{\mathbb{F}}^{(1)}, \hat{\mathbb{U}}^{(1)}) + \log p(\tilde{\mathbb{V}}|\hat{\mathbb{F}}^{(2)}, \hat{\mathbb{U}}^{(2)}) + \lambda \Delta \hat{Q} - 2 \log p(\tilde{\mathbb{V}}|\tilde{\mathbb{F}}, \tilde{\mathbb{U}}) && (6.5)\\
&\leq \log \frac{p(\tilde{\mathbb{V}}|\hat{\mathbb{F}}^{(1)}, \hat{\mathbb{U}}^{(1)})}{p(\tilde{\mathbb{V}}|\tilde{\mathbb{F}}, \tilde{\mathbb{U}})} + \log \frac{p(\tilde{\mathbb{V}}|\hat{\mathbb{F}}^{(2)}, \hat{\mathbb{U}}^{(2)})}{p(\tilde{\mathbb{V}}|\tilde{\mathbb{F}}, \tilde{\mathbb{U}})} + \lambda \Delta \hat{Q} && (6.6)\\
&\leq \log(0) + \log \frac{p(\tilde{\mathbb{V}}|\hat{\mathbb{F}}^{(2)}, \hat{\mathbb{U}}^{(2)})}{p(\tilde{\mathbb{V}}|\tilde{\mathbb{F}}, \tilde{\mathbb{U}})} + \lambda \Delta \hat{Q} = -\infty && (6.7)
\end{aligned}
$$

which is a contradiction. □

Lemma 6.3.2. *For a dataset of n instances $\tilde{\mathbb{V}} \sim p(\tilde{\mathbb{V}}|\tilde{\mathbb{F}}, \tilde{\mathbb{U}})$, $\Delta \hat{Q} \rightarrow \max_{(\mathbb{F}^{(1)},\mathbb{U}^{(1)}),(\mathbb{F}^{(2)},\mathbb{U}^{(2)}) \in \mathbb{L}} \Delta Q$ almost surely as $n \rightarrow \infty$.*

Proof. Toward a contradiction assume that there exists some $(\mathbb{F}'^{(1)}, \mathbb{U}'^{(1)}, \mathbb{F}'^{(2)}, \mathbb{U}'^{(2)})$ such that $\mathcal{L}(\mathbb{F}'^{(1)}, \mathbb{U}'^{(1)}, \mathbb{F}'^{(2)}, \mathbb{U}'^{(2)}) \leq \mathcal{L}(\hat{\mathbb{F}}^{(1)}, \hat{\mathbb{U}}^{(1)}, \hat{\mathbb{F}}^{(2)}, \hat{\mathbb{U}}^{(2)})$ and $\Delta Q' > \Delta \hat{Q}$. By Lemmas 6.2.1 and 6.3.1, we have that as $n \rightarrow \infty$, $p(\tilde{\mathbb{V}}|\mathbb{F}'^{(1)}, \mathbb{U}'^{(1)}) = p(\tilde{\mathbb{V}}|\mathbb{F}'^{(2)}, \mathbb{U}'^{(2)}) = p(\tilde{\mathbb{V}}|\hat{\mathbb{F}}^{(1)}, \hat{\mathbb{U}}^{(1)}) = p(\tilde{\mathbb{V}}|\hat{\mathbb{F}}^{(2)}, \hat{\mathbb{U}}^{(2)}) = p(\tilde{\mathbb{V}}|\tilde{\mathbb{F}}, \tilde{\mathbb{U}})$. Therefore, by definition of $\mathcal{L}$, we have the following as $n \rightarrow \infty$:

$$
\begin{aligned}
\mathcal{L}(\mathbb{F}'^{(1)}, \mathbb{U}'^{(1)}, \mathbb{F}'^{(2)}, \mathbb{U}'^{(2)}) &\leq \mathcal{L}(\hat{\mathbb{F}}^{(1)}, \hat{\mathbb{U}}^{(1)}, \hat{\mathbb{F}}^{(2)}, \hat{\mathbb{U}}^{(2)}) && (6.8)\\
2 \log p(\tilde{\mathbb{V}}|\tilde{\mathbb{F}}, \tilde{\mathbb{U}}) + \lambda \Delta Q' &\leq 2 \log p(\tilde{\mathbb{V}}|\tilde{\mathbb{F}}, \tilde{\mathbb{U}}) + \lambda \Delta \hat{Q} && (6.9)\\
\Delta Q' &\leq \Delta \hat{Q} && (6.10)
\end{aligned}
$$

which is a contradiction. □

The following asymptotic theorems hold for any $\lambda \in \mathbb{R}^+$ and bounded Q:

Theorem 6.3.3. *A causal query Q is η-identifiable given $(\tilde{\mathbb{F}}, \tilde{\mathbb{U}})$ for a dataset of n instances, $\tilde{\mathbb{V}} \sim p(\mathbb{V}|\tilde{\mathbb{F}}, \tilde{\mathbb{U}})$, if $\Delta \hat{Q} \leq 2\eta$ and only if $\Delta \hat{Q} \leq \eta$ almost surely as $n \rightarrow \infty$.*

Proof. By Lemma 6.3.1 we have that $(\hat{\mathbb{F}}^{(1)}, \hat{\mathbb{U}}^{(1)})$ and $(\hat{\mathbb{F}}^{(2)}, \hat{\mathbb{U}}^{(2)})$ are in $\mathbb{L}$, i.e. the set of functions that maximize the log likelihood of the data asymptotically. Therefore, if $|\hat{Q}^{(1)} - \hat{Q}^{(2)}| > 2\eta$, then at least one of $(\hat{\mathbb{F}}^{(1)}, \hat{\mathbb{U}}^{(1)})$ or $(\hat{\mathbb{F}}^{(2)}, \hat{\mathbb{U}}^{(2)})$ are a $(\mathbb{F}', \mathbb{U}')$ that satisfy Theorem 6.2.2. By Lemma 6.3.2 we have that $|\hat{Q}^{(1)} - \hat{Q}^{(2)}|$ maximizes the distance between induced causal effects. Therefore, if $|\hat{Q}^{(1)} - \hat{Q}^{(2)}| < \eta$ as $n \to \infty$, no such $(\mathbb{F}', \mathbb{U}')$ exists. Note that if $\eta < |\hat{Q}^{(1)} - \hat{Q}^{(2)}| < 2\eta$ we can not conclude whether Q is η-identifiable, as the true causal effect $\tilde{Q}$ may be within η of either or neither of $\hat{Q}^{(1)}$ or $\hat{Q}^{(2)}$. □

Theorem 6.3.4. *A causal query Q is (ζ, η)-identifiable given a prior $p(\mathbb{F}, \mathbb{U})$ for m samples of functions and confounders, $\tilde{\mathbb{F}}_i, \tilde{\mathbb{U}}_i \sim p(\mathbb{F}, \mathbb{U})$, and m datasets of n instances, $\tilde{\mathbb{V}}_i \sim p(\mathbb{V}|\tilde{\mathbb{F}}_i, \tilde{\mathbb{U}}_i)$, if $\zeta < \sum_{i=1}^{m} \mathbb{1}_{\Delta\hat{Q}_i > 2\eta}$ and only if $\zeta < \sum_{i=1}^{m} \mathbb{1}_{\Delta\hat{Q}_i > \eta}$ almost surely as $n, m \to \infty$.*

Proof. Theorem 6.3.4 follows directly from the weak law of large numbers applied to the results of Theorem 6.3.3. □

6.3.1 Likelihood Ratio Test

Theorems 6.3.3 and 6.3.4 provide necessary and sufficient conditions for determining identifiability in the limit of infinite simulations given exact solutions to $\mathcal{L}$. However, given finite n and m and approximate solutions to $\mathcal{L}$, $\Delta\hat{Q}$ may be large even if the query is identifiable. To address the problem of finite n and m we propose a likelihood ratio hypothesis test using gradient-based approximate solutions to $\mathcal{L}$. The details of this procedure are shown in Algorithm 4, which works as follows. Repeatedly sample a set of functions and latent confounders, $(\tilde{\mathbb{F}}, \tilde{\mathbb{U}})$, from the prior. For each, repeatedly sample a set of observations, $\tilde{\mathbb{V}}$, and optimize $\mathcal{L}$ jointly for two SCMs, resulting in an approximately optimal $\Delta\hat{Q}$ for the simulated data. Then, apply a likelihood ratio test to determine if the distance between particles is statistically significantly greater than η with probability ζ. For finite k, where the central limit theorem does not provide an exact description of the distribution of the sample mean $\hat{\mu}_i$, this procedure is best described as an approximate test.

Recall that $\text{ID}(\tilde{\mathbb{F}}, \tilde{\mathbb{U}}, \eta)$ is a function that returns 1 if Q is η-identifiable given $(\tilde{\mathbb{F}}, \tilde{\mathbb{U}})$ under Definition 6.2.1, and 0 otherwise. Additionally, recall that $\hat{\mu}_i$ is the sample-averaged $\Delta\hat{Q}$ across k datasets drawn from $p(\mathbb{V}|\tilde{\mathbb{F}}_i, \tilde{\mathbb{U}}_i)$ with n instances.

Let ζ' be the true (unknown) probability that $\text{ID}(\tilde{\mathbb{F}}, \tilde{\mathbb{U}}, \eta) = 1$ for $(\tilde{\mathbb{F}}, \tilde{\mathbb{U}}) \sim p(\mathbb{F}, \mathbb{U})$, let H_o be the null hypothesis that Q is not (ζ, η)-identifiable, i.e. $\zeta' < \zeta$, H_a be the alternative hypothesis that Q is (ζ, η)-identifiable, i.e. $\zeta' \geq \zeta$, and let $\text{ID}_{\eta,i}$ be shorthand for $\text{ID}(\tilde{\mathbb{F}}_i, \tilde{\mathbb{U}}_i, \eta)$.

To construct a likelihood ratio test, we evaluate the maximum of the log data likelihood (here over observed data $\hat{\mu}_i$) in the set of parameters in the null hypothesis, denoted l_0, and given the full union of parameters in the null and alternative hypotheses, denoted l_a. If the difference between these two quantities is significantly large, i.e. $\chi^2(2(l_a - l_0); 1) < \alpha$, then we reject the null hypothesis. Intuitively, this test fails to reject the null if adding additional degrees of freedom to the parameter space (here by allowing $\zeta < \zeta' < 1$) does not substantially change the maximum of the likelihood.

The following expression gives the maximum of the likelihood for the parameters in the null hypothesis. Here, the likelihood is given with respect to parameters $\theta = \{\zeta', \bar{\mu}_{\text{ID},1}, ..., \bar{\mu}_{\text{ID},k}, \bar{\mu}_{\text{nID},1}, ..., \bar{\mu}_{\text{nID},k}\}$. The space of parameters under the null, Θ_0, is defined such that $0 < \zeta' < \zeta$, $\bar{\mu}_{\text{ID},1}, ..., \bar{\mu}_{\text{ID},k}$ are in $[0, \eta]$, and $\bar{\mu}_{\text{nID},1}, ..., \bar{\mu}_{\text{nID},k}$ are in (η, ∞). Here, $\bar{\mu}_{\text{ID},i}$ and $\bar{\mu}_{\text{nID},i}$ represent the true (unknown) centers for $\hat{Q}$ for the i'th SCM when $\text{ID}_{\eta,i} = 1$ or 0 respectively. The space of parameters under the alternative hypothesis, Θ_a, is identical, except that $\zeta < \zeta' < 1$.

$$\begin{aligned}
l_0 &:= \max_{\theta\in\Theta_0} \log p(\hat{\mu}_1, ..., \hat{\mu}_k|\theta) \\
&= \max_{\theta\in\Theta_0} \log \prod_{i=1}^{m} p(\hat{\mu}_i|\theta) \\
&= \max_{\theta\in\Theta_0} \sum_{i=1}^{m} \log(p(\hat{\mu}_i|\mathrm{ID}_{\eta,i} = 1, \theta)p(\mathrm{ID}_{\eta,i} = 1|\theta) + p(\hat{\mu}_i|\mathrm{ID}_{\eta,i} = 0, \theta)p(\mathrm{ID}_{\eta,i} = 0|\theta)) \\
&= \max_{\theta\in\Theta_0} \sum_{i=1}^{m} \log(\mathcal{N}(\hat{\mu}_i; \bar{\mu}_{\mathrm{ID},i}, \Sigma_i)p(\mathrm{ID}_{\eta,i} = 1|\zeta') + \mathcal{N}(\hat{\mu}_i; \bar{\mu}_{\mathrm{nID},i}, \Sigma_i)p(\mathrm{ID}_{\eta,i} = 0|\zeta')) \\
&= \max_{\zeta'\in[0,\zeta]} \sum_{i=1}^{m} \log(\mathcal{N}(\hat{\mu}_i; \min(\hat{\mu}_i, \eta), \Sigma_i)\zeta' + \mathcal{N}(\hat{\mu}_i; \max(\hat{\mu}_i, \eta), \Sigma_i)(1 - \zeta'))
\end{aligned} \tag{6.11}$$

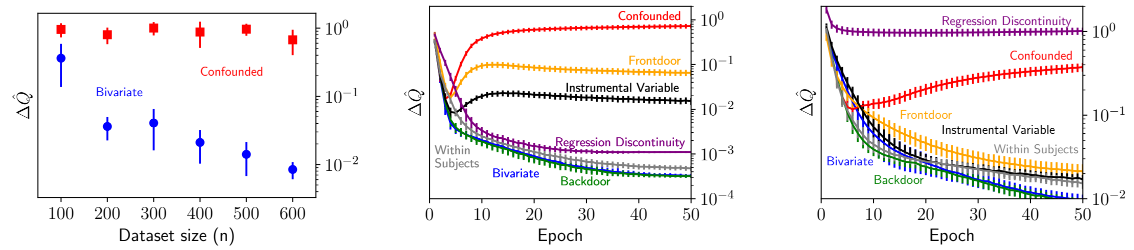
Note that the maximum likelihood value of $\bar{\mu}_{\mathrm{ID},i}$ and $\bar{\mu}_{\mathrm{ID},i}$ is given by the closest value to $\hat{\mu}$ in their respective set of possible assignments, resulting in the $\min(\hat{\mu}_i, \eta)$ and $\max(\hat{\mu}_i, \eta), \Sigma_i)$ expressions in the final equation above. By a similar argument, l_a is given by the following expression.

$$\begin{aligned}
l_a &:= \max_{\theta\in\Theta_0\cup\Theta_a} \log p(\hat{\mu}_1, ..., \hat{\mu}_k|\theta) \\
&= \max_{\zeta'\in[0,1]} \sum_{i=1}^{m} \log(\mathcal{N}(\hat{\mu}_i; \min(\hat{\mu}_i, \eta), \Sigma_i)\zeta' + \mathcal{N}(\hat{\mu}_i; \max(\hat{\mu}_i, \eta), \Sigma_i)(1 - \zeta'))
\end{aligned} \tag{6.12}$$

Theorem 6.3.5. *For convex $\mathcal{L}$, Algorithm 4 approaches the most powerful exact test with significance α as $n, k \to \infty$.*

Proof. Theorem 6.3.5 follows directly from the Neyman-Pearson lemma [90] and Theorem 6.3.4. □

While gradient-based optimization is not guaranteed to escape local optima, our many experiments in Section 6.4 suggest that SBI is robust even when $\mathcal{L}$ is non-convex and for finite n, m, and k. SBI correctly determines identifiability for all six of our latent variable model benchmarks, which we strongly suspect all have non-convex likelihoods. We believe that approximate solutions to $\mathcal{L}$ are reliable in practice for two



(a) Gaussian Process (b) Linear Training Curves (c) GP Training Curves

Figure 6.2: **Summaries of particle-based optimization.** As the simulated dataset size increases the difference between effect estimates of the two particles ($\Delta\hat{Q}$) remains large for the confounded Gaussian process model (a), indicating that the model is not identifiable. Without confounding however, the optimized particles converge to the same causal effect. Using gradient-based optimization, SBI is able to discover likelihood equivalent causal models when they exist that induce different effects for linear (b) and Gaussian process (c) models.

reasons. First, SBI aggregates $m \cdot k$ independent runs of gradient-based optimization on simulated data in its statistical test. For example, even though 14 of the 5000 trajectories had $\Delta\hat{Q} > \eta$, SBI concluded that SATE for the linear IV benchmark is identifiable. Second, SBI uses stochastic gradients and modern optimizers (e.g., Adam) that are known to escape local optima in non-convex high-dimensional settings.

While the choice of repulsion strength, λ, does not influence our asymptotic results, this is not generally the case for any finite n. In our experiments in Section 6.4, we find that even small values of λ produce large $\Delta\hat{Q}$ for non-identifiable models.

6.3.2 Example: Confounded Gaussian Process

Let us again consider the confounded model in Section 6.2.1, instead assuming that the function $\mathrm{y}_i = f(\mathrm{t}_i, \mathrm{u}_i, \epsilon_{\mathrm{y}_i})$ is drawn from the following Gaussian process prior over $\mathrm{y}_i = \mu_y(\mathrm{t}_i, \mathrm{u}_i) + \sigma_y^2 \epsilon_{\mathrm{y}_i}$, where $D \in \mathbb{N}$, $\boldsymbol{\mu}_y = [\mu_y(\mathrm{t}_1, \mathrm{u}_1), ..., \mu_y(\mathrm{t}_n, \mathrm{u}_n)]$, and $\mathbb{W} = \{\sigma_y^2, \mathrm{w}_0, \mathrm{w}_{1,1}, ..., \mathrm{w}_{4,1}, ..., \mathrm{w}_{1,D}, ..., \mathrm{w}_{4,D}\}$:

$$\mu_y(\mathrm{t}_i, \mathrm{u}_i) = \mathrm{w}_0 + \sum_{d=1}^{D} \mathrm{w}_{1,d} \sin(d\mathrm{t}_i) + \mathrm{w}_{2,d} \cos(d\mathrm{t}_i) + \mathrm{w}_{3,d} \sin(d\mathrm{u}_i) + \mathrm{w}_{4,d} \cos(d\mathrm{u}_i) \quad (6.13)$$

This Gaussian process model is known as the Fourier model, where the choice of D and the prior $p(\boldsymbol{W})$ dictate the characteristics of the sampled functions [104]. In this and all subsequent experiments we set $D = 10$, $\mathrm{w}_0 \sim \mathcal{N}(0, 1)$, and $\mathrm{w}_{1,d}, \mathrm{w}_{2,d}, \mathrm{w}_{3,d}, \mathrm{w}_{4,d} \overset{iid}{\sim} \mathcal{N}(0, 1/d^2)$. This choice of prior results in relatively smooth functions, as the weights on higher-order terms are typically close to 0. Again, let the causal query, Q, be the sample average treatment effect with the intervention $do(\mathrm{t}_i = t')$. Then the log likelihood and the difference between causal effects are given by the following:

$$\log p(\mathbb{V}|\mathbb{F}, \mathbb{U}) = \log \mathcal{N}(\mathbf{t}; \gamma \mathbf{u}, \sigma_t^2 \boldsymbol{I}) + \log \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_y, \sigma_y^2 \boldsymbol{I}) \tag{6.14}$$

$$\Delta Q = \sum_{d=1}^{D} |\mathrm{w}_{1,d}^{(1)} - \mathrm{w}_{1,d}^{(2)}| \sin(dt') + |\mathrm{w}_{2,d}^{(1)} - \mathrm{w}_{2,d}^{(2)}| \cos(dt') \tag{6.15}$$

Given this expressions for the log likelihood and the causal query in terms of parameters, θ, and latent confounders, $\mathbb{U}$, we can now compute the partial derivative of the particle-based objective function, $\frac{\partial}{\partial s}\mathcal{L} = \frac{\partial}{\partial s} \log p(\tilde{\mathbb{V}}|\mathbb{F}^{(1)}, \mathbb{U}^{(1)}) + \frac{\partial}{\partial s} \log p(\tilde{\mathbb{V}}|\mathbb{F}^{(2)}, \mathbb{U}^{(2)}) + \lambda \frac{\partial}{\partial s}\Delta Q$ with respect to all $s \in \theta \cup \mathbb{U}$. Given an expression for each partial derivative, we can then apply standard gradient-descent algorithms to determine identifiability using SBI. Without loss of generality, the derivative of the repulsion term with respect to s for $\mathbb{F}^{(1)}$, $\mathbb{U}^{(1)}$ is given by the following:

$$\frac{\partial}{\partial s}\Delta Q = \begin{cases} \dfrac{\mathrm{w}_{1,d}^{(1)} - \mathrm{w}_{1,d}^{(2)}}{|\mathrm{w}_{1,d}^{(1)} - \mathrm{w}_{1,d}^{(2)}|} \sin(dt') & s = \mathrm{w}_{1,d}^{(1)} \\ \dfrac{\mathrm{w}_{2,d}^{(1)} - \mathrm{w}_{2,d}^{(2)}}{|\mathrm{w}_{1,d}^{(1)} - \mathrm{w}_{2,d}^{(2)}|} \cos(dt') & s = \mathrm{w}_{2,d}^{(1)} \\ 0 & \text{otherwise} \end{cases} \tag{6.16}$$

For the derivative of the log density we expand on standard identities of Gaussians, where $L_{\mathbb{V}}$, L_t, and L_y are shorthand for $\log p(\mathbb{V}|\mathbb{F}^{(1)}, \mathbb{U}^{(1)})$, $\log p(\mathbf{t}|\gamma\mathbf{u}, \sigma_t^2\boldsymbol{I})$, and $\log \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_y, \sigma_y^2\boldsymbol{I})$ respectively:

$$\frac{\partial L_{\mathbb{V}}}{\partial s} = \frac{\partial L_t}{\partial s} + \frac{\partial L_y}{\partial s} \tag{6.17}$$

$$\frac{\partial L_t}{\partial s} = \frac{1}{\sigma_t^2}\sum_{i=1}^{n}(\mathrm{t}_i - \gamma\mathrm{u}_i)\frac{\partial\gamma\mathrm{u}_i}{\partial s} - \frac{\partial\sigma_t^2}{\partial s}\frac{1}{2\sigma_t^2}\left(n - \frac{1}{\sigma_t^2}\right)\sum_{i=1}^{n}(\mathrm{t}_i - \gamma\mathrm{u}_i)^2 \tag{6.18}$$

$$\frac{\partial L_y}{\partial s} = \frac{1}{\sigma_y^2}\sum_{i=1}^{n}(\mathrm{y}_i - \mu_y(\mathrm{t}_i, \mathrm{u}_i))\frac{\partial\mu_y(\mathrm{t}_i, \mathrm{u}_i)}{\partial s} - \frac{\partial\sigma_y^2}{\partial s}\frac{1}{2\sigma_y^2}\left(n - \frac{1}{\sigma_y^2}\right)\sum_{i=1}^{n}(\mathrm{y}_i - \mu_t(\mathrm{t}_i, \mathrm{u}_i))^2 \tag{6.19}$$

$$\frac{\partial\gamma\mathrm{u}_i}{\partial s} = \begin{cases} \mathrm{u}_i & s = \gamma \\ \gamma & s = \mathrm{u}_i \\ 0 & \text{otherwise} \end{cases} \qquad \frac{\partial\sigma_t^2}{\partial s} = \begin{cases} 1 & s = \sigma_t^2 \\ 0 & \text{otherwise} \end{cases}$$

$$\frac{\partial\sigma_t^2}{\partial s} = \begin{cases} 1 & s = \sigma_t^2 \\ 0 & \text{otherwise} \end{cases} \qquad \frac{\partial\mu_t(\mathrm{t}_i, \mathrm{u}_i)}{\partial s} = \begin{cases} 1 & s = w_0 \\ \sin(d\mathrm{t}_i) & s = w_{1,d} \\ \cos(d\mathrm{t}_i) & s = w_{2,d} \\ \sin(d\mathrm{u}_i) & s = w_{3,d} \\ \cos(d\mathrm{u}_i) & s = w_{4,d} \\ d(w_{3,d}\cos(d\mathrm{t}_i) \\ \quad - w_{4,d}\sin(d\mathrm{t}_i)) & s = \mathrm{u}_i \\ 0 & \text{otherwise} \end{cases}$$

Note that although deriving these gradients is cumbersome and error-prone in general, it can be easily automated using standard automatic differentiation procedures.

Design	Prior	$\Delta\hat{Q}_{\text{SBI}}$	$\Delta\hat{Q}_{\text{PL}}$	$\Delta\hat{Q}_{\text{MH}}$	ID_{Truth}	ID_{SBI}	ID_{PL}	ID_{MH}	ID_{DAG}
Unconfounded	Linear	.00 ± .00	.11 ± .01	.12 ± .03	✓	✓	✗	✗	✓
	GP	.01 ± .00	.34 ± .02	.26 ± .06	✓	✓	✗	✗	✓
Confounded	Linear	.83 ± .15	1.8 ± .34	.14 ± .03	✗	✗	✗	✗	✗
	GP	.50 ± .28	.73 ± .14	.38 ± .08	✗	✗	✗	✗	✗
Backdoor	Linear	.00 ± .00	.10 ± .01	.11 ± .02	✓	✓	✗	✗	✓
	GP	.01 ± .00	.28 ± .02	.26 ± .05	✓	✓	✗	✗	✓
Frontdoor	Linear	.06 ± .04	.17 ± .05	.37 ± .13	✓	✓	✗	✗	✓
	GP	.02 ± .01	.20 ± .09	.34 ± .17	✓	✓	✗	✗	✓
Instrumental variable	Linear	.01 ± .01	.05 ± .01	.13 ± .06	✓	✓	✓	✗	✓
	GP	.01 ± .00	.37 ± .03	.40 ± .08	✓	✓	✗	✗	✗
Within subject	Linear	.00 ± .00	.10 ± .01	.14 ± .03	✓	✓	✗	✗	✗
	GP	.01 ± .01	.39 ± .04	.26 ± .06	✓	✓	✗	✗	✗
Regression discontinuity	Linear	.00 ± .00	.16 ± .01	.21 ± .03	✓	✓	✗	✗	✓
	GP	1.1 ± .12	1.1 ± .09	.82 ± .1	✗	✗	✗	✗	✓

Table 6.2: **Empirical results on quasi-experimental design benchmarks.** Simulation-based identifiability (this chapter) correctly determines the identifiability of sample average treatment effects for all fourteen of the benchmark linear and Gaussian process (GP) quasi-experimental designs. Lower $\Delta\hat{Q}$ implies identifiability. The columns labeled ID show whether SBI and the baselines determine the design to be statistically significantly identifiable. Neither of the profile likelihood (PL) or the Metropolis Hastings (MH) baselines consistently determine identifiability. The column labeled ID_{DAG} presents the results of the do-calculus [93] for GP benchmarks, and IC [69] for linear benchmarks applied (incorrectly) to the underlying causal graphs, despite the fact that they do not account for all of the parametric restrictions. This comparison is only to illustrate the effect of parametric restrictions on identifiability.

Figure 6.2a shows the results of Algorithm 4 with this prior over structural causal models using the Adam gradient descent algorithm [67] to optimize $\mathcal{L}$. Unlike the unconfounded model, which is identical except that $\mathbb{U}$ has been omitted, we conclude that the confounded model is not identifiable. We expand on these examples in Section 6.4.

6.4 Experiments

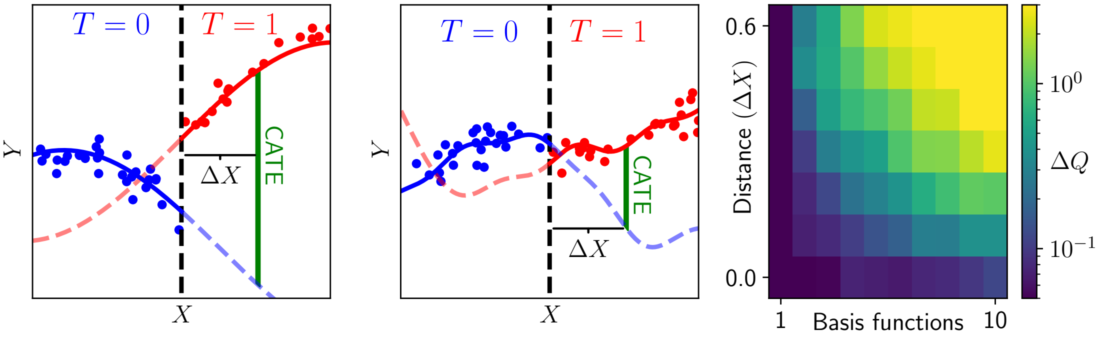
We evaluated SBI on a benchmark suite of priors reflecting seven standard causal designs which are summarized in Table 6.1; unconfounded regression, confounded regression, backdoor adjusted, frontdoor adjusted, instrumental variable, within-subjects, and regression discontinuity designs. For each of these seven benchmarks we tested SBI using a linear parameterization (e.g. Section 6.2.1) as well as a parameterization where the outcome function is replaced with a finite dimensional Gaussian process (e.g. Section 6.3.2). Additional experimental details and descriptions of each prior are provided in Section A.2.

We compared SBI against two baselines, one which seeks to approximate the full posterior directly using a Metropolis-Hastings based inference procedure (MH), and one which uses a variation of profile likelihood (PL) identification [105], which alternates between parameter perturbations and maximum-likelihood optimization. We implemented Algorithm 4, all designs, and the baselines using Gen. Using $m = 100$, $n = 1000$, $k = 50$, $\lambda = 1$, $\eta = 0.1$, $\zeta = 0.8$, and $\alpha = 0.05$, SBI correctly determines the identifiability of all designs, performing significantly better than the two baselines. As we formalized in Section 6.3, if $\Delta\hat{Q}$ is close to 0 then the causal query is identifiable.

Our experiments demonstrate that SBI agrees with the do-calculus in settings where graph structure alone is sufficient, and produces correct identification results for designs that previously required custom identification proofs. Finally, we present the first known identification results for Gaussian process quasi-experimental designs, demonstrating agreement with widely held intuition. See Table 6.2 for a summary of SATE identification results.

6.4.1 Causal Graphical Models.

In addition to the unconfounded and confounded regression designs presented in Section 6.3.2, we evaluated SBI on two models that are covered by the do-calculus, backdoor-adjusted and frontdoor-adjusted designs. Backdoor-adjusted designs represent settings where all of the random variables that confound the relationship between treatment and outcome are observed, blocking all *backdoor* paths. Unlike backdoor-adjusted designs, frontdoor-adjusted designs can include latent confounding



(a) 1 basis function

(b) 10 basis functions

(c) Identifiability heatmap

Figure 6.3: **Quantitative insight for conditional average treatment effects.** SBI provides novel and intuitive identification results for the Gaussian process regression discontinuity design benchmark. These results (c) show that conditional average treatment effects (CATE) becomes *less identifiable* as we condition on covariates further from the discontinuity ($\Delta X > 0$) and for less smooth outcome functions, i.e. increasing the number of basis functions (a, b).

between treatment and outcome, as long as there exists an observed mediator that is not confounded, as in Figure 6.1b. Despite this latent confounding, average treatment effects are nonparametrically identifiable [97].

6.4.2 Linear Quasi-Experimental Designs.

Instrumental variable designs differ from the confounded design in that an observed variable, known as the instrument, influences the treatment. Two conditions must be satisfied to enable identification: (i) the instrument and the treatment must not be confounded; and (ii) all influence from the instrument to the outcome is mediated through the treatment. While these assumptions can be expressed graphically, additional parametric assumptions are needed for effects to be identifiable [97]. For example, if exogenous noise is additive, then the average treatment effect is identifiable [52].

Within-subjects designs involve hierarchically structured data in which individual instances (e.g., students) are affiliated with one of several objects (e.g., schools). Treatment effects for these kinds of settings can be identified even if treatment and outcome are confounded, as long as confounders are shared across all instances belonging to the same object [138]. These designs can be described as the family of structural causal models; $\mathrm{t}_i = f_t(\mathrm{u}_{o(i)}, \epsilon_{t_i})$, $\mathrm{y}_i = f_t(\mathrm{t}_i, \mathrm{u}_{o(i)}, \epsilon_{y_i})$, where $\mathrm{u}_{o(i)}$ refers to the shared value of the latent confounder corresponding to instance i. Hierarchically structured confounding is applicable to a wide variety of common causal designs [63]: including twin studies [20], difference-in-differences designs [117], and multi-level-modeling [43].

Regression discontinuity designs are quasi-experimental designs in which the treatment depends on a particular observed covariate being above or below a known threshold. We consider a sharp deterministic discontinuity, i.e. $\mathrm{t}_i = 1$ if $\mathrm{x}_i > 0$, and $\mathrm{t}_i = 0$ otherwise. These regression discontinuity designs correspond to the family of structural causal models $\mathrm{x}_i = f_x(\epsilon_{x_i})$, $\mathrm{t}_i = \mathbb{1}_{\mathrm{x}_i > 0}$, and $\mathrm{y}_i = f_t(\mathrm{t}_i, \mathrm{x}_i, \epsilon_{\mathrm{t}_i})$. Even though all confounders are observed, the deterministic relationship between $\mathbf{x}$ and $\mathbf{t}$ violates the positivity assumption, which is a necessary assumption for the do-calculus to be sound [93]. Here, average treatment effects are identifiable for linear models, but not nonparametrically.

6.4.3 Gaussian Process Quasi-Experimental Designs.

We used SBI to determine the previously unknown identifiability of Gaussian process versions of quasi-experimental designs. By assuming a particular kernel we place an inductive bias on the class of structural functions, which could in principle enable identification. SBI instead confirms that the identifiability of these Gaussian process models agrees with the literature on nonparametric identification.

We also evaluated SBI on the conditional average treatment effect (CATE) for a Gaussian process version of the regression discontinuity design. For nonlinear outcome functions, such as our Gaussian process, observations in one region of $\mathbf{x}$ provide only partial information about counterfactuals in another. For example, in

Figure 6.3b the outcome function for untreated individuals ($T = 0$) to the right of the discontinuity (dashed blue curve) is only one of many that are compatible with observed data. Therefore, we should expect that CATE is more ambiguous further from the discontinuity and for less smooth functions. SBI's results in Figure 6.3c agree with this intuition, demonstrating that ΔQ increases as we condition on covariates further from the discontinuity and as we increase the number of basis functions.

6.5 Discussion

In this chapter we demonstrated how SBI can be used to test the identifiability of Bayesian models for causal inference. While determining identifiability is particularly salient in these causal settings, it can also be valuable in non-causal settings as a part of a holistic modeling workflow [46], supplementing other introspection tools such as simulation-based calibration [123].

In addition to determining identifiability, SBI can be used as a kind of sensitivity analysis [42, 66, 109], bounding the range of causal effects that are likelihood equivalent. Our regression discontinuity design results shown in Figure 6.3c emphasize this capability, showing that irreducible uncertainty in effect estimates increases with increasing distance from the discontinuity and with less smooth outcome functions.

Our benchmarks encode strong parametric assumptions about latent confounders and exogenous noise. If desired, users may represent broader uncertainty using hyperpriors. To demonstrate this, we ran a version of the confounded GP model with additional hyperpriors over the mean and variance of $\mathbf{u}$. See the supplementary materials for details. As another example, one could relax additive noise assumptions using Bayesian versions of invertible neural networks [32], which satisfy SBI's requirements that the likelihood be differentiable and that counterfactual outcomes (and thus Q) are fully determined by $(\mathbb{F}, \mathbb{U}, \mathbb{V})$.

SBI builds on a long history of optimization-focused machine learning research. Reducing identifiability to optimization in this way provides a path towards reasoning about Bayesian models for causal inference at previously unattainable scales. However, this reduction means that SBI's conclusions are dictated by the performance of an approximate global optimization method. Formally quantifying the implications of this approximation error, and extending SBI to discrete combinatorial causal models (e.g. causal discovery) are important areas of future work.

CHAPTER 7

CONCLUSION

In this Chapter I summarize the contributions made in this thesis, and explore opportunities for future work.

7.1 Summary of Contributions

In this thesis, I presented the Bayesian structural approach to causal inference, and discussed how it could be realized using probabilistic programming languages.

In Chapter 3, I presented a concise mathematical description of the Bayesian structural approach to causal inference and showed how a linear example could be implemented as a probabilistic program. I illustrated how various modeling choices influence our ability to draw causal conclusions and how these modeling choices can be straightforwardly reflected in the source code of probabilistic programs.

In Chapter 4, I presented an advanced application of the Bayesian structural approach to causal inference, combining richly structured assumptions about how latent confounders are shared between observed data instances with flexible Bayesan nonparameteric Gaussian process priors over structural functions. I showed empirically that this model, GP-SLC, achieves state-of-the-art effect estimation on a collection of synthetic and semi-synthetic benchmarks.

In Chapter 5, I presented a simple extension of the Bayesian structural approach to causal inference for reasoning with a combination of observational and experimental data. Building on the insight from Chapter 3 that programs are a compact and convenient representation for causal assumptions, I showed how composing syntactic program transformations with a causal language interpreter embedded in a probabilistic programming language makes this expanded capability remarkably straightforward.

In Chapter 6, I presented Simulation-Based Identifiability (SBI), an approach for determining if a Bayesian structural causal model yields unique causal conclusions asymptotically given data. SBI reduces the problem of causal identifiability to a particle-based optimization problem, which can be solved approximately with gradient-based search. I proved that SBI is asymptotically sound and complete in the limit of infinite simulations and exact solutions to the particle-based optimization problem. For the case with finite simulations, I presented a custom likelihood ratio hypothesis test and proved that it is the most powerful exact test in the sample limit. Finally, on an extensive suite of linear and Gaussian process benchmarks I show that SBI correctly determines the identifiability for seven graph-based and econometric causal designs. No other automated method provides such coverage.

7.1.1 Key Claims Restated

Taken together, these contributions provide evidence for five key claims about the Bayesian structural approach to causal inference, first introduced in Chapter 1. I restate those claims here. For details on how the specific chapters provide evidence for each claim see Section 1.1.2.

Claim 1. *The Bayesian structural approach provides an expressive substrate for representing practical assumptions for causal inference that can not be expressed using graph structure alone.*

Claim 2. *A large and diverse collection of qualitative findings scattered throughout the causal inference literature emerge as a consequence of the Bayesian structural approach to causal inference.*

Claim 3. *The Bayesian structural approach can be used to represent broad uncertainty over structural functions, and to learn complex nonlinear dependencies from data.*

Claim 4. *The Bayesian structural approach can provide valuable insight into causal inference problems even without exact probabilistic inference, which is NP-hard in general.*

Claim 5. *The Bayesian structural approach provides a computational foundation on which a software engineering discipline of causal inference can be constructed; enabling modular, composable, and extensible causal inference software artifacts.*

7.2 Future Work

In this thesis, I presented the Bayesian structural approach to causal inference with probabilistic programming. In doing so I contributed: (i) advanced Bayesian nonparameteric models for causal inference in richly structured domains [134]; (ii) methods for combining observational and experimental data [137]; and (iii) a general meta-reasoning technique for determining whether causal assumptions are sufficient for drawing causal conclusions [135]. While these serve as important steps towards the larger vision of fully featured causal probabilistic programming languages, as well as software engineering disciplines of causal probabilistic programming, more work is needed. In this final section, I discuss some exciting opportunities for future work in this area. These opportunities fall into two distinct categories: (i) language design; and (ii) applications.

7.2.1 Causal Probabilistic Programming Language Design

Automating Program Transformations. In Chapter 3, I showed how the Gen probabilistic programming language could be used to represent Bayesian structural causal models by composing a prior over probabilistic structural causal models with an intervention program transformation to induce a joint distribution over factual and counterfactual outcomes, and thus also over causal queries of interest. However, in doing so I omitted important details about how to implement such an intervention, as well as how to automate the reparameterization and inversion logic (see Section 3.2.8) necessary to compute probability densities and their gradients that are often used by approximate inference algorithms. In Chapter 5, I demonstrated one approach to implementing such an intervention transformation using syntax rewriting in a restricted domain specific language for causal inference. In our restricted language with no recursion, loops, or other control, this was adequate, because any intervened random variables could be transformed statically before the causal program was interpreted by the embedded causal program interpreter. However, with more expressive control flow constructs in the causal language, interventions will need to be applied dynamically as the causal program is evaluated. Omega [124] is able to achieve the kind of expressiveness we are interested in by lazily evaluating sampling statements for random variables, but does so at the cost of losing tractable density evaluation. Design a causal probabilistic programming language that support interventions in dynamic models and leads to tractable inference is an exciting areas of future work.

Black-Box Abstractions for Causal Inference In Chapter 4 I showed how to combine Bayesian nonparametric models such as Gaussian processes with rich causal assumptions about how confounders are shared among instances. While these are appealing model families, they pose somewhat of a problem for programming languages that automate intervention program transformations. In our linear example, and in the examples in Chapter 5, the probabilistic program implementation directly translates to a straightforward intervention semantics. To implement an intervention we simply apply the structural functions with new arguments, keeping exogenous noise between factual and counterfactual worlds fixed. However, when using Gaussian processes the code implementation obscures the interventional semantics, as implementations of Gaussian processes rely on an equivalence between (i) sampling a structural function from a prior and then applying that structural function to its arguments with the tractable alternative of (ii) jointly sampling a collection of structural function outputs from a multivariate Gaussian. In fact, the custom derivations of specialized distributions over counterfactuals in Section 4.3.2 were necessary exactly because we don't have access to structural functions, and thus could not intervene on them directly.

This example illustrates the somewhat awkward reality that interventions on some models (such as the linear examples in Chapter 3) can be fully automated and require no additional user-specified information, while others (such as the Gaussian process models in Chapter 4) require user input. Therefore, if we want to permit these kinds of Gaussian process models in a general purpose causal probabilistic programming language, what interfaces or programming abstractions must we expose to enable it? In other words, what is the causal analog to Gen's generative function interface [28] that will enable arbitrary causal model composition?

Beyond the somewhat niche area of Bayesian nonparametrics, this question ("What are the necessary abstractions for causal inference?") is generally important as causal modeling continues to become more collaborative. With such an abstraction in hand, distinct users could implement causal models in different host languages and with different implementation strategies, while still enabling the kinds of composition we expect in large-scale software engineering efforts.

7.2.2 Applications.

Dynamical Systems and Differential Equations. Throughout this thesis, I have emphasized the expressive ability of the Bayesian structural approach to bring the practice of causal inference closer to the needs of working scientists, policymakers, or other analysts. However, I have thusfar omitted a substantial class of causal models, those defined in terms of a sequence of (potentially stochastic) differential equations. For many scientific endeavors, especially the physical sciences, differential equations are a pervasive representation for mechanistic knowledge. While causal inference with dynamical systems is an active area of research [18, 19, 110], the Bayesian approach has promise for reasons we have discussed throughout this thesis.

Structure Learning and Program Synthesis. In Chapter 5, I presented approaches that enabled observational and experimental data to be combined to infer whether a causal dependency exists, demonstrating a particularly simple form of structure learning using the Bayesian structural approach and probabilistic programming. Scaling this problem up to realistic settings with many candidate structures is an exciting, and daunting technical challenge. Doing so will inevitably require more advanced inference algorithms, as the vanilla sequential Monte Carlo algorithm we applied in Chapter 5 is unlikely to scale to the super-exponential collection of discrete structures in a typical structure learning problem, let alone one with more expressive programming constructs like loops and conditional branching. Future work could explore how to leverage recent innovations in gradient-based optimization of directed acyclic graph structures using continuous relaxations of acyclicity constraints [144], although doing so is not trivial.

Hybrid Models. Perhaps the most exciting forward-looking opportunity for Bayesian structural causal modeling with probabilistic programming is the ability to compose multiple models from multiple stakeholders and experts, and yield causal inferences that no individual model could provide. In fact, this setting where multiple seemingly disparate models must be composed likely represents the majority of scientific modeling efforts; unfortunately our existing formal computational machinery is ill equipped for these realistic and important settings. Large-scale climate simulation, for example, clearly requires a symphony of models at different time scales, spatial resolutions, and levels of abstraction to yield actionable inferences. Our existing computational tools for causal inference provide little insight in these rich settings. However, with substantial efforts, the Bayesian structural approach might be able to fill in the gaps.

APPENDIX

ADDITIONAL EXPERIMENTAL DETAIL

A.1 Causal Inference Using Gaussian Processes with Structured Latent Confounding

In this section we provide additional detail on the baseline methods used in our experiments in Chapter 4.

A.1.1 Baselines

The following structural equations summarize the data generating process for the synthetic baselines:

$$\begin{aligned}
\mathrm{W}_{j,k} &\sim \mathcal{N}(0,1) \ \text{ for } j,k \in [\![3]\!] \\
\mathrm{U}_{o,j} &\sim \mathcal{N}(0,0.5) \text{ for } o \in [\![n_o]\!], j \in [\![3]\!] \\
\mathrm{X}_{i,j} &= \mathbf{W}_{j,:}^{\top} \cdot \mathbf{U}_{o=pa(i),:} + \boldsymbol{\epsilon}_{x_i} \text{ where } \boldsymbol{\epsilon}_{x_i} \sim \mathcal{N}(0, 0.5\boldsymbol{I}_3) \text{ for } i \in [\![n_i]\!] \\
\mathrm{t}_i &= g_t(\mathbf{X}_{i,:}, \mathbf{U}_{o=pa(i),:}) + \boldsymbol{\epsilon}_{t_i} \text{ where } \boldsymbol{\epsilon}_{t_i} \sim \mathcal{N}(0,0.5) \text{ for } i \in [\![n_i]\!] \\
\mathrm{y}_i &= g_y(\mathrm{t}_i, \mathbf{X}_{i,:}, \mathbf{U}_{o=pa(i),:}) + \boldsymbol{\epsilon}_{y_i} \text{ where } \boldsymbol{\epsilon}_{y_i} \sim \mathcal{N}(0,0.5) \text{ for } i \in [\![n_i]\!]
\end{aligned}$$

First, we draw $\mathbf{U}$ from a multivariate Gaussian distribution. Then, we generate covariates $\mathbf{X}$ as linear combinations of $\mathbf{U}$ with additive exogenous noise. We generate treatments $\mathbf{t}$ as a function (g_t) of $\mathbf{X}$ and $\mathbf{U}$ with additive noise. Finally, we generate outcome $\mathbf{y}$ as a function (g_y) of $\mathbf{X}$, $\mathbf{t}$, and $\mathbf{U}$ with additive noise. For multi-dimensional variables, $\mathbf{X}$ and $\mathbf{U}$, we first apply the nonlinear function to each dimension of $\mathbf{X}$ and $\mathbf{U}$, then we aggregate them by summing across dimensions. The nonlinear treatment and outcome functions are shown in Table A.1.

	$g_t(\mathbf{X}_{i,:}, \mathbf{U}_{o=Pa(i),:})$	$g_y(\mathrm{t}_i, \mathbf{X}_{i,:}, \mathbf{U}_{o=Pa(i),:})$
Add	$\sum_j \mathrm{X}_{i,j}\ \sin(\mathrm{X}_{i,j}) - \sum_j \mathrm{U}_{o,j}\ \sin(\mathrm{U}_{o,j})$	$\mathrm{t}_i \sin(2\mathrm{t}_i) + \sum_j \mathrm{X}_{i,j}\ \sin(\mathrm{X}_{i,j}) + 3\sum_j \mathrm{U}_{o,j}\ \sin(\mathrm{U}_{o,j})$
Mult	$\frac{1}{10}(\sum_j \mathrm{X}_{i,j}\ \sin(\mathrm{X}_{i,j}))(\sum_j \mathrm{U}_{o,j}\ \sin(\mathrm{U}_{o,j}))$	$\frac{1}{10}(\mathrm{t}_i \sin(2\mathrm{t}_i))(\sum_{j=1} \mathrm{X}_{i,j}\ \sin(\mathrm{X}_{i,j}))(\sum_j \mathrm{U}_{o,j}\ \sin(\mathrm{U}_{o,j}))$

Table A.1: **The functional form of t and y for 2 synthetic datasets with continuous treatments and nonlinear outcome functions.**

A.2 Simulation Based Identifiability

In this section we provide additional detail for the linear and Gaussian process experiments in Chapter 6. For each of the seven designs we assume that treatment, $\mathbf{t}$, outcome, $\mathbf{y}$, and where applicable covariates, $\mathbf{x}$, and instruments, I, are observed. All other random variables are latent.

For all of the experiments we used the Adam [67] algorithm to optimize $\mathcal{L}$. We ran Adam with $\alpha = 0.01$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$ for fifty epochs with a minibatch size of ten instances. For all of the linear parametric experiments, we assume that each function $V_i = f_V(Pa(V)_i, \epsilon_{V_i}) = \beta \cdot Pa(V)_i + \epsilon_{V_i}$, where each element of β is drawn from a normal prior. Here, $Pa(V)_i$ refers to the vector of all latent and observed arguments in the structural function, f_V. All of the experiments, including the Gaussian process models, assume that exogenous noise is normally distributed and additive. For all of the Gaussian process experiments, we assume that each outcome function $\mathrm{y}_i = f_t(Pa(Y), \epsilon_{y_i})$ is drawn from the same Gaussian process prior described in Section 6.3.2.

A.2.1 Linear Structural Causal Models

Here, we provide the full prior over linear structural causal models for each of the seven designs in Table 1.

Unconfounded Regression.

$$\beta_y \sim \mathcal{N}(1, 0.3) \qquad \log(\sigma_t^2) \sim \mathcal{N}(\text{-}1, 0.3) \qquad \log(\sigma_y^2) \sim \mathcal{N}(\text{-}3, 0.3)$$

$$\epsilon_{t_i} \overset{iid}{\sim} \mathcal{N}(0, \sigma_t^2) \qquad \epsilon_{y_i} \overset{iid}{\sim} \mathcal{N}(0, \sigma_y^2) \qquad \mathrm{t}_i = \epsilon_{t_i}$$

$$\mathrm{y}_i = \beta_y \cdot \mathrm{t}_i + \epsilon_{y_i}$$

Confounded Regression.

$$\beta_t \sim \mathcal{N}(.5, 0.3) \qquad \beta_y \sim \mathcal{N}([1, .5]^\top, 0.3\boldsymbol{I}) \qquad \log(\sigma_t^2) \sim \mathcal{N}(\text{-}1, 0.3)$$

$$\log(\sigma_y^2) \sim \mathcal{N}(\text{-}3, 0.3) \qquad \mathrm{u}_i \overset{iid}{\sim} \mathcal{N}(0, 0.3) \qquad \epsilon_{t_i} \overset{iid}{\sim} \mathcal{N}(0, \sigma_t^2)$$

$$\epsilon_{y_i} \overset{iid}{\sim} \mathcal{N}(0, \sigma_y^2) \qquad \mathrm{t}_i = \beta_t \cdot \mathrm{u}_i + \epsilon_{t_i} \qquad \mathrm{y}_i = \beta_y \cdot [\mathrm{t}_i, \mathrm{u}_i] + \epsilon_{y_i}$$

Backdoor Adjusted.

$$\beta_t \sim \mathcal{N}(.5, 0.3) \qquad \beta_y \sim \mathcal{N}([1, .5]^\top, 0.3\boldsymbol{I}) \qquad \log(\sigma_x^2) \sim \mathcal{N}(\text{-}2, 0.3)$$

$$\log(\sigma_t^2) \sim \mathcal{N}(\text{-}1, 0.3) \qquad \log(\sigma_y^2) \sim \mathcal{N}(\text{-}3, 0.3) \qquad \epsilon_{x_i} \overset{iid}{\sim} \mathcal{N}(0, \sigma_x^2)$$

$$\epsilon_{t_i} \overset{iid}{\sim} \mathcal{N}(0, \sigma_t^2) \qquad \epsilon_{y_i} \overset{iid}{\sim} \mathcal{N}(0, \sigma_y^2) \qquad \mathrm{x}_i = \epsilon_{x_i}$$

$$\mathrm{t}_i = \beta_t \cdot \mathrm{x}_i + \epsilon_{t_i} \qquad \mathrm{y}_i = \beta_y \cdot [\mathrm{t}_i, \mathrm{x}_i] + \epsilon_{y_i}$$

Frontdoor Adjusted.

$$\beta_t \sim \mathcal{N}(.5, 0.3) \qquad \beta_x \sim \mathcal{N}(1, 0.3) \qquad \beta_y \sim \mathcal{N}([1, .5]^\top, 0.3\boldsymbol{I})$$

$$\log(\sigma_t^2) \sim \mathcal{N}(\text{-}2, 0.3) \qquad \log(\sigma_x^2) \sim \mathcal{N}(\text{-}1, 0.3) \qquad \log(\sigma_y^2) \sim \mathcal{N}(\text{-}3, 0.3)$$

$$\epsilon_{t_i} \overset{iid}{\sim} \mathcal{N}(0, \sigma_t^2) \qquad \epsilon_{x_i} \overset{iid}{\sim} \mathcal{N}(0, \sigma_x^2) \qquad \epsilon_{y_i} \overset{iid}{\sim} \mathcal{N}(0, \sigma_y^2)$$

$$\mathrm{u}_i \overset{iid}{\sim} \mathcal{N}(0, 0.3) \qquad \mathrm{t}_i = \beta_t \cdot \mathrm{u}_i + \epsilon_{t_i} \qquad \mathrm{x}_i = \beta_x \cdot \mathrm{t}_i + \epsilon_{x_i}$$

$$\mathrm{y}_i = \beta_y \cdot [\mathrm{x}_i, \mathrm{u}_i] + \epsilon_{y_i}$$

Instrumental Variable.

$$\begin{array}{ccc}
\beta_t \sim \mathcal{N}([2, .5]^\top, 0.3\boldsymbol{I}) & \beta_y \sim \mathcal{N}([1, .5]^\top, 0.3\boldsymbol{I}) & \log(\sigma_x^2) \sim \mathcal{N}(0, 0.3) \\
\log(\sigma_t^2) \sim \mathcal{N}(\text{-}1, 0.3) & \log(\sigma_y^2) \sim \mathcal{N}(\text{-}3, 0.3) & \epsilon_{x_i} \overset{iid}{\sim} \mathcal{N}(0, \sigma_x^2) \\
\epsilon_{t_i} \overset{iid}{\sim} \mathcal{N}(0, \sigma_t^2) & \epsilon_{y_i} \overset{iid}{\sim} \mathcal{N}(0, \sigma_y^2) & \mathrm{u}_i \overset{iid}{\sim} \mathcal{N}(0, 0.3) \\
\mathrm{x}_i = \epsilon_{x_i} & \mathrm{t}_i = \beta_t \cdot [I_i, \mathrm{u}_i] + \epsilon_{t_i} & \mathrm{y}_i = \beta_y \cdot [\mathrm{t}_i, \mathrm{u}_i] + \epsilon_{y_i}
\end{array}$$

Within Subjects. Here, $\mathrm{u}_{o(i)}$ refers to the shared value of the latent confounder, u_o, associated with instance i. For these experiments, we assume that each object instance, o, is shared between 25 instances of treatment and outcome.

$$\begin{array}{ccc}
\beta_t \sim \mathcal{N}(.5, 0.3) & \beta_y \sim \mathcal{N}([1, .5]^\top, 0.3\boldsymbol{I}) & \log(\sigma_t^2) \sim \mathcal{N}(\text{-}1, 0.3) \\
\log(\sigma_y^2) \sim \mathcal{N}(\text{-}3, 0.3) & \mathrm{u}_o \overset{iid}{\sim} \mathcal{N}(0, 0.3) & \epsilon_{t_i} \overset{iid}{\sim} \mathcal{N}(0, \sigma_t^2) \\
\epsilon_{y_i} \overset{iid}{\sim} \mathcal{N}(0, \sigma_y^2) & \mathrm{t}_i = \beta_t \cdot \mathrm{u}_{o(i)} + \epsilon_{t_i} & \mathrm{y}_i = \beta_y \cdot [\mathrm{t}_i, \mathrm{u}_{o(i)}] + \epsilon_{y_i}
\end{array}$$

Regression Discontinuity Design.

$$\begin{array}{ccc}
\beta_y \sim \mathcal{N}([0.5, 0.5, \text{-}0.5]^\top, 0.3\boldsymbol{I}) & \log(\sigma_x^2) \sim \mathcal{N}(\text{-}1, 0.3) & \log(\sigma_y^2) \sim \mathcal{N}(\text{-}3, 0.3) \\
\epsilon_{x_i} \overset{iid}{\sim} \mathcal{N}(0, \sigma_x^2) & \epsilon_{y_i} \overset{iid}{\sim} \mathcal{N}(0, \sigma_y^2) & \mathrm{x}_i = \epsilon_{x_i} \\
\mathrm{t}_i = \mathbb{1}_{\mathrm{x}_i > 0} & \mathrm{y}_i = \beta_y \cdot [\mathrm{x}_i, \mathrm{t}_i, 1 - \mathrm{t}_i] + \epsilon_{y_i} &
\end{array}$$

A.2.2 Gaussian Process Structural Causal Models

For each of the experiments using Gaussian process priors over structural causal models we use the same prior over linear structural causal models for all functions except the outcome function f_t, which is drawn from the Gaussian process prior described in Section 6.3.2.

A.2.3 Additional Baseline Details

For our profile likelihood baseline identification method, we used an approach based on profile likelihood identification [105]. For each model the baseline is identical to the SBI in all respects, except that it uses only a single particle with no repulsion term. Instead, to traverse the likelihood surface the baseline first performs 100 epochs of the Adam optimization method using the gradient of the log-likelihood to find a single maximum likelihood solution. Then, for each parameter $s \in \theta$, we increment the parameter by a small amount $s \leftarrow s + \Delta s$ and then again run the Adam optimization method using the gradient of the log-likelihood with respect to all parameters except for s for 100 steps. In our experiments we use $\Delta s = 0.01$ for all parameters. We report the range over estimated causal effects after repeating this procedure 100 times for all $s \in \theta$. Intuitively, if the likelihood surface is on a *ridge* of equivalent maximum likelihood models then alternating between perturbations and optimization will find other locations on that maximum likelihood surface. We discuss limitations of this kind of approach in Section 6.1, and show empirically that SBI outperforms it in Section 6.4.

For our Metropolis Hastings baseline identification method, we used a combination of standard inference procedures to approximate the posterior $p(Q|\tilde{V})$ directly. This inference procedure involved alternating between 10 steps of random walk Metropolis Hastings on each $s \in \theta$ and 10 steps of elliptical slice sampling on $\mathbf{u}$ (when applicable) a total of 100 times. To compensate for the additional computational costs of this sampling-based approximate inference procedure, we reduced the number of instances, (n), to 250 for this baseline. In addition, we eliminated the first 25 sets of 10 Metropolis-Hastings and elliptical slice moves as a *burn-in*.

A.2.4 Hyperprior Demonstration

As we discussed in Section 6.5, we ran an additional experiment to demonstrate the use of hyperpriors to represent broader uncertainty. In this experiment, each $\mathrm{u}_i \sim \mathcal{N}(\mathrm{u}_{\mathrm{mean}}, \mathrm{u}_{\mathrm{var}})$, and $\mathrm{u}_{\mathrm{mean}} \sim \mathcal{N}(0, 1)$, $\log(\mathrm{u}_{\mathrm{var}}) \sim \mathcal{N}(0, 1)$. SBI correctly determined that the SATE is not identifable with $\Delta\hat{Q}_{\mathrm{SBI}} = 0.55 \pm 0.36$.

BIBLIOGRAPHY

[1] Abelson, Harold, and Sussman, Gerald Jay. *Structure and interpretation of computer programs.* The MIT Press, 1996.

[2] Alaa, Ahmed, and van der Schaar, Mihaela. Bayesian inference of individualized treatment effects using multi-task Gaussian processes. In *Advances in Neural Information Processing Systems* (2017), pp. 3424–3432.

[3] Alaa, Ahmed, and van der Schaar, Mihaela. Bayesian nonparametric causal inference: Information rates and learning algorithms. *IEEE Journal of Selected Topics in Signal Processing 12*, 5 (2018), 1031–1046.

[4] Aldrich, John. Autonomy. *Oxford Economic Papers 41*, 1 (1989), 15–34.

[5] Andrieu, Christophe, Lee, Anthony, and Livingstone, Sam. A general perspective on the metropolis-hastings kernel. *arXiv preprint arXiv:2012.14881* (2020).

[6] Andrieu, Christophe, and Roberts, Gareth O. The pseudo-marginal approach for efficient monte carlo computations. *The Annals of Statistics 37*, 2 (2009), 697–725.

[7] Angrist, Joshua D. Lifetime earnings and the vietnam era draft lottery: evidence from social security administrative records. *The american economic review* (1990), 313–336.

[8] Angrist, Joshua D, Imbens, Guido W, and Rubin, Donald B. Identification of causal effects using instrumental variables. *Journal of the American statistical Association 91*, 434 (1996), 444–455.

[9] Athey, Susan, and Imbens, Guido W. Identification and inference in nonlinear difference-in-differences models. *Econometrica 74*, 2 (2006), 431–497.

[10] Aumann, Robert J. Borel structures for function spaces. *Illinois Journal of Mathematics 5*, 4 (1961), 614–630.

[11] Balke, Alexander, and Pearl, Judea. Counterfactual probabilities: Computational methods, bounds and applications. In *Uncertainty Proceedings 1994*. Elsevier, 1994, pp. 46–54.

[12] Bareinboim, Elias, Correa, Juan D, Ibeling, Duligur, and Icard, Thomas. On pearl's hierarchy and the foundations of causal inference. *ACM Special Volume in Honor of Judea Pearl (provisional title) 2*, 3 (2020), 4.

[13] Bengio, Yoshua, Goodfellow, Ian, and Courville, Aaron. *Deep learning*, vol. 1. MIT press Cambridge, MA, USA, 2017.

[14] Berkson, Joseph. Limitations of the application of fourfold table analysis to hospital data. *Biometrics Bulletin 2*, 3 (1946), 47–53.

[15] Bezanson, Jeff, Edelman, Alan, Karpinski, Stefan, and Shah, Viral B. Julia: A fresh approach to numerical computing. *SIAM review 59*, 1 (2017), 65–98.

[16] Bingham, Eli, Chen, Jonathan P., Jankowiak, Martin, Obermeyer, Fritz, Pradhan, Neeraj, Karaletsos, Theofanis, Singh, Rohit, Szerlip, Paul, Horsfall, Paul, and Goodman, Noah D. Pyro: Deep Universal Probabilistic Programming. *Journal of Machine Learning Research* (2018).

[17] Bollen, Kenneth A. Structural equation models. *Encyclopedia of biostatistics 7* (2005).

[18] Bongers, Stephan, Blom, Tineke, and Mooij, Joris M. Causal modeling of dynamical systems. *arXiv preprint arXiv:1803.08784* (2018).

[19] Bongers, Stephan, Forré, Patrick, Peters, Jonas, and Mooij, Joris M. Foundations of structural causal models with cycles and latent variables. *The Annals of Statistics 49*, 5 (2021), 2885–2915.

[20] Boomsma, Dorret, Busjahn, Andreas, and Peltonen, Leena. Classical twin studies and beyond. *Nature Reviews Genetics 3*, 11 (2002), 872–882.

[21] Branson, Zach, Rischard, Maxime, Bornn, Luke, and Miratrix, Luke W. A nonparametric bayesian methodology for regression discontinuity designs. *Journal of Statistical Planning and Inference 202* (2019), 14–30.

[22] Campbell, Donald T, and Stanley, Julian C. *Experimental and quasi-experimental designs for research.* Ravenio books, 2015.

[23] Cao, Yanshuai. *Scaling Gaussian Processes.* PhD thesis, University of Toronto (Canada), 2018.

[24] Carpenter, Bob, Gelman, Andrew, Hoffman, Matthew D, Lee, Daniel, Goodrich, Ben, Betancourt, Michael, Brubaker, Marcus, Guo, Jiqiang, Li, Peter, and Riddell, Allen. Stan: A probabilistic programming language. *Journal of statistical software 76*, 1 (2017).

[25] Cinelli, Carlos, Forney, Andrew, and Pearl, Judea. A crash course in good and bad controls. *Sociological Methods & Research* (2021), 00491241221099552.

[26] Cragg, John G, and Donald, Stephen G. Testing identifiability and specification in instrumental variable models. *Econometric Theory* (1993), 222–240.

[27] Cusumano-Towner, Marco, Lew, Alexander K, and Mansinghka, Vikash K. Automating involutive mcmc using probabilistic and differentiable programming. *arXiv preprint arXiv:2007.09871* (2020).

[28] Cusumano-Towner, Marco F, Saad, Feras A, Lew, Alexander K, and Mansinghka, Vikash K. Gen: A general-purpose probabilistic programming system with programmable inference. In *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation* (2019), ACM, pp. 221–236.

[29] Cusumano-Towner, Marco Francis. *Gen: a high-level programming platform for probabilistic inference.* PhD thesis, Massachusetts Institute of Technology, 2020.

[30] D'Amour, Alexander. On multi-cause approaches to causal inference with unobserved counfounding: Two cautionary failure cases and a promising alternative. In *The 22nd International Conference on Artificial Intelligence and Statistics* (2019), pp. 3478–3486.

[31] Dillon, Joshua V, Langmore, Ian, Tran, Dustin, Brevdo, Eugene, Vasudevan, Srinivas, Moore, Dave, Patton, Brian, Alemi, Alex, Hoffman, Matt, and Saurous, Rif A. Tensorflow distributions. *arXiv preprint arXiv:1711.10604* (2017).

[32] Dinh, Laurent, Sohl-Dickstein, Jascha, and Bengio, Samy. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803* (2016).

[33] Domke, Justin, and Sheldon, Daniel R. Divide and couple: Using monte carlo variational objectives for posterior approximation. *Advances in neural information processing systems 32* (2019).

[34] Doob, Joseph L. Application of the theory of martingales. *Le calcul des probabilites et ses applications* (1949), 23–27.

[35] Doucet, Arnaud, Godsill, Simon, and Andrieu, Christophe. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and computing 10*, 3 (2000), 197–208.

[36] Draper, David. Inference and hierarchical modeling in the social sciences. *Journal of Educational and Behavioral Statistics 20*, 2 (1995), 115–147.

[37] Duane, Simon, Kennedy, Anthony D, Pendleton, Brian J, and Roweth, Duncan. Hybrid monte carlo. *Physics letters B 195*, 2 (1987), 216–222.

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

[39] Elwert, Felix, and Winship, Christopher. Endogenous selection bias: The problem of conditioning on a collider variable. *Annual review of sociology 40* (2014), 31–53.

[40] England, ISO New. Energy, load, and demand reports. https://www.iso-ne.com/isoexpress/web/reports/load-and-demand/-/tree/zone-info, 2018.

[41] Finke, Axel. *On extended state-space constructions for Monte Carlo methods.* PhD thesis, University of Warwick, 2015.

[42] Franks, AlexanderM, D'Amour, Alexander, and Feller, Avi. Flexible sensitivity analysis for observational studies without observable implications. *Journal of the American Statistical Association* (2019).

[43] Gelman, Andrew. Multilevel (hierarchical) modeling: What it can and cannot do. *Technometrics 48*, 3 (2006), 432–435.

[44] Gelman, Andrew, Carlin, John B, Stern, Hal S, and Rubin, Donald B. *Bayesian data analysis.* Chapman and Hall/CRC, 1995.

[45] Gelman, Andrew, and Hill, Jennifer. *Data Analysis Using Regression and Multilevel/Hierarchical Models.* Cambridge University Press, 2006.

[46] Gelman, Andrew, Vehtari, Aki, Simpson, Daniel, Margossian, Charles C, Carpenter, Bob, Yao, Yuling, Kennedy, Lauren, Gabry, Jonah, Bürkner, Paul-Christian, and Modrák, Martin. Bayesian workflow. *arXiv preprint arXiv:2011.01808* (2020).

[47] Gentzel, Amanda, Garant, Dan, and Jensen, David. The case for evaluating causal models using interventional measures and empirical data. In *Advances in Neural Information Processing Systems* (2019), pp. 11717–11727.

[48] Goodman, Noah D, Mansinghka, Vikash K, Roy, Daniel, Bonawitz, Keith, and Tenenbaum, Joshua B. Church: a language for generative models. In *Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence* (2008), pp. 220–229.

[49] Gothoskar, Nishad, Cusumano-Towner, Marco, Zinberg, Ben, Ghavamizadeh, Matin, Pollok, Falk, Garrett, Austin, Tenenbaum, Josh, Gutfreund, Dan, and Mansinghka, Vikash. 3dp3: 3d scene perception via probabilistic programming. *Advances in Neural Information Processing Systems 34* (2021), 9600–9612.

[50] Haavelmo, Trygve. The probability approach in econometrics. *Econometrica: Journal of the Econometric Society* (1944), iii–115.

[51] Halpern, Joseph Y. *Actual causality.* MiT Press, 2016.

[52] Hartford, Jason, Lewis, Greg, Leyton-Brown, Kevin, and Taddy, Matt. Deep iv: A flexible approach for counterfactual prediction. In *International Conference on Machine Learning* (2017), PMLR, pp. 1414–1423.

[53] Hastings, Keith. *Monte Carlo Sampling Methods Using Markov Chains and Their Applications.* Oxford University Press, 1970.

[54] Hill, Jennifer. Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics 20*, 1 (2011), 217–240.

[55] Hoffman, Matthew D, Gelman, Andrew, et al. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *J. Mach. Learn. Res. 15*, 1 (2014), 1593–1623.

[56] Hong, Guanglei. *Causal inference for multi-level observational data with application to kindergarten retention.* University of Michigan, 2004.

[57] Hong, Guanglei, and Raudenbush, Stephen. Evaluating kindergarten retention policy. *Journal of the American Statistical Association 101*, 475 (2006), 901–910.

[58] Hong, Guanglei, and Yu, Bing. Effects of kindergarten retention on children's social-emotional development: An application of propensity score method to multivariate, multilevel data. *Developmental Psychology 44*, 2 (2008), 407.

[59] Huang, Yimin, and Valtorta, Marco. Pearl's calculus of intervention is complete. In *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence* (2006), pp. 217–224.

[60] Ibeling, Duligur, and Icard, Thomas. On open-universe causal reasoning. In *Uncertainty in Artificial Intelligence* (2020), PMLR, pp. 1233–1243.

[61] Imbens, Guido, and Rubin, Donald. *Causal Inference in Statistics, Social, and Biomedical Sciences.* Cambridge University Press, 2015.

[62] Imbens, Guido W. Nonparametric estimation of average treatment effects under exogeneity: A review. *Review of Economics and statistics 86*, 1 (2004), 4–29.

[63] Jensen, David, Burroni, Javier, and Rattigan, Matthew. Object conditioning for causal inference. In *Uncertainty in Artificial Intelligence* (2020), PMLR, pp. 1072–1082.

[64] Johansson, Fredrik, Shalit, Uri, and Sontag, David. Learning representations for counterfactual inference. In *Proceedings of The 33rd International Conference on Machine Learning* (New York, New York, USA, 20–22 Jun 2016), vol. 48 of *Proceedings of Machine Learning Research*, PMLR, pp. 3020–3029.

[65] Johnson, Brittany, Bartola, Jesse, Angell, Rico, Keith, Katherine, Witty, Sam, Giguere, Stephen J, and Brun, Yuriy. Fairkit, fairkit, on the wall, who's the fairest of them all? supporting data scientists in training fair models. *arXiv preprint arXiv:2012.09951* (2020).

[66] Kallus, Nathan, Mao, Xiaojie, and Zhou, Angela. Interval estimation of individual-level causal effects under unobserved confounding. In *The 22nd International Conference on Artificial Intelligence and Statistics* (2019), PMLR, pp. 2281–2290.

[67] Kingma, Diederik P, and Ba, Jimmy. Adam: A method for stochastic optimization. In *ICLR (Poster)* (2015).

[68] Korb, Kevin B, Hope, Lucas R, Nicholson, Ann E, and Axnick, Karl. Varieties of causal intervention. In *Pacific Rim International Conference on Artificial Intelligence* (2004), Springer, pp. 322–331.

[69] Kumor, Daniel, Chen, Bryant, and Bareinboim, Elias. Efficient identification in linear structural causal models with instrumental cutsets. In *Advances in Neural Information Processing Systems* (2019), pp. 12477–12486.

[70] Kuroki, Manabu, and Pearl, Judea. Measurement bias and effect restoration in causal inference. *Biometrika 101*, 2 (2014), 423–437.

[71] Lawrence, Neil. Gaussian process latent variable models for visualisation of high dimensional data. In *Advances in Neural Information Processing Systems* (2004), pp. 329–336.

[72] Le, Tuan Anh, Baydin, Atilim Gunes, and Wood, Frank. Inference compilation and universal probabilistic programming. In *Artificial Intelligence and Statistics* (2017), PMLR, pp. 1338–1348.

[73] Lee, David S, and Lemieux, Thomas. Regression discontinuity designs in economics. *Journal of economic literature 48*, 2 (2010), 281–355.

[74] Lee, Sanghack, Correa, Juan D, and Bareinboim, Elias. General identifiability with arbitrary surrogate experiments. In *Uncertainty in Artificial Intelligence* (2020), PMLR, pp. 389–398.

[75] Lew, Alexander K, Cusumano-Towner, Marco, and Mansinghka, Vikash. Recursive monte carlo and variational inference with auxiliary variables. In *The 38th Conference on Uncertainty in Artificial Intelligence* (2022).

[76] Liang, Kung-Yee, and Zeger, Scott L. Longitudinal data analysis using generalized linear models. *Biometrika 73*, 1 (1986), 13–22.

[77] Loftus, Geoffrey, and Masson, Michael. Using confidence intervals in within-subject designs. *Psychonomic Bulletin & Review 1*, 4 (1994), 476–490.

[78] Louizos, Christos, Shalit, Uri, Mooij, Joris, Sontag, David, Zemel, Richard, and Welling, Max. Causal effect inference with deep latent-variable models. In *Advances in Neural Information Processing Systems* (2017), pp. 6446–6456.

[79] Lousdal, Mette Lise. An introduction to instrumental variable assumptions, validation and estimation. *Emerging themes in epidemiology 15*, 1 (2018), 1.

[80] Maclaren, Oliver J, and Nicholson, Ruanui. What can be estimated? identifiability, estimability, causal inference and ill-posed inverse problems. *arXiv preprint arXiv:1904.02826* (2019).

[81] Maier, Marc. Causal discovery for relational domains: Representation, reasoning, and learning. *UMass PhD Dissertation* (2014).

[82] Malinsky, Daniel, Shpitser, Ilya, and Richardson, Thomas. A potential outcomes calculus for identifying conditional path-specific effects. In *The 22nd International Conference on Artificial Intelligence and Statistics* (2019), PMLR, pp. 3080–3088.

[83] Mansinghka, Vikash, Selsam, Daniel, and Perov, Yura. Venture: a higher-order probabilistic programming platform with programmable inference. *arXiv preprint arXiv:1404.0099* (2014).

[84] Miao, Wang, Geng, Zhi, and Tchetgen Tchetgen, Eric. Identifying causal effects with proxy variables of an unmeasured confounder. *Biometrika 105*, 4 (08 2018), 987–993.

[85] Murphy, Kevin P. *Machine learning: a probabilistic perspective.* MIT Press, 2012.

[86] Murray, Iain, Adams, Ryan, and MacKay, David. Elliptical slice sampling. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (2010), JMLR Workshop and Conference Proceedings, pp. 541–548.

[87] Neal, Radford. *Bayesian Learning for Neural Networks*, vol. 118. Springer Science & Business Media, 2012.

[88] Neal, Radford M, et al. Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo 2*, 11 (2011), 2.

[89] Neklyudov, Kirill, Welling, Max, Egorov, Evgenii, and Vetrov, Dmitry. Involutive mcmc: a unifying framework. In *International Conference on Machine Learning* (2020), PMLR, pp. 7273–7282.

[90] Neyman, Jerzy, and Pearson, Egon Sharpe. Ix. on the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character 231*, 694-706 (1933), 289–337.

[91] Ouyang, Long, Tessler, Michael Henry, Ly, Daniel, and Goodman, Noah. Practical optimal experiment design with probabilistic programs. *arXiv preprint arXiv:1608.05046* (2016).

[92] Pawlowski, Nick, Coelho de Castro, Daniel, and Glocker, Ben. Deep structural causal models for tractable counterfactual inference. *Advances in Neural Information Processing Systems 33* (2020), 857–869.

[93] Pearl, Judea. Causal diagrams for empirical research. *Biometrika 82*, 4 (1995), 669–688.

[94] Pearl, Judea. *Causality: models, reasoning and inference*, vol. 29. Springer, 2000.

[95] Pearl, Judea. Bayesianism and causality, or, why i am only a half-bayesian. In *Foundations of bayesianism*. Springer, 2001, pp. 19–36.

[96] Pearl, Judea. Causal inference in statistics: An overview. *Statistics surveys 3* (2009), 96–146.

[97] Pearl, Judea. *Causality: Models, Reasoning and Inference*, 2nd ed. Cambridge University Press, New York, NY, USA, 2009.

[98] Pearl, Judea. Interpretation and identification of causal mediation. *Psychological methods 19*, 4 (2014), 459.

[99] Perov, Yura, Graham, Logan, Gourgoulias, Kostis, Richens, Jonathan, Lee, Ciaran, Baker, Adam, and Johri, Saurabh. Multiverse: causal reasoning using importance sampling in probabilistic programming. In *Symposium on advances in approximate bayesian inference* (2020), PMLR, pp. 1–36.

[100] Quiñonero-Candela, Joaquin, and Rasmussen, Carl Edward. A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research 6*, Dec (2005), 1939–1959.

[101] Ramey, Craig T, Bryant, Donna M, Wasik, Barbara H, Sparling, Joseph J, Fendt, Kaye H, and La Vange, Lisa M. Infant health and development program for low birth weight, premature infants: Program elements, family participation, and child intelligence. *Pediatrics 89*, 3 (1992), 454–465.

[102] Ranganath, Rajesh, Gerrish, Sean, and Blei, David. Black box variational inference. In *Artificial intelligence and statistics* (2014), PMLR, pp. 814–822.

[103] Rasmussen, Carl. Gaussian processes in machine learning. In *Summer School on Machine Learning* (2003), Springer, pp. 63–71.

[104] Rasmussen, Carl Edward, and Ghahramani, Zoubin. Occam's razor. *Advances in neural information processing systems* (2001), 294–300.

[105] Raue, Andreas, Kreutz, Clemens, Maiwald, Thomas, Bachmann, Julie, Schilling, Marcel, Klingmüller, Ursula, and Timmer, Jens. Structural and practical identifiability analysis of partially observed dynamical models by exploiting the profile likelihood. *Bioinformatics 25*, 15 (2009), 1923–1929.

[106] Richardson, Thomas S, and Robins, James M. Single world intervention graphs (swigs): A unification of the counterfactual and graphical approaches to causality. *Center for the Statistics and the Social Sciences, University of Washington Series. Working Paper 128*, 30 (2013), 2013.

[107] Rissanen, Severi, and Marttinen, Pekka. A critical look at the consistency of causal estimation with deep latent variable models. *Advances in Neural Information Processing Systems 34* (2021).

[108] Roberts, Gareth O, Tweedie, Richard L, et al. Exponential convergence of langevin distributions and their discrete approximations. *Bernoulli 2*, 4 (1996), 341–363.

[109] Robins, James M, Rotnitzky, Andrea, and Scharfstein, Daniel O. Sensitivity analysis for selection bias and unmeasured confounding in missing data and causal inference models. In *Statistical models in epidemiology, the environment, and clinical trials*. Springer, 2000, pp. 1–94.

[110] Rubenstein, Paul K, Bongers, Stephan, Schölkopf, Bernhard, and Mooij, Joris M. From deterministic odes to dynamic structural causal models. *arXiv preprint arXiv:1608.08028* (2016).

[111] Rubin, Donald B. Assignment to treatment group on the basis of a covariate. *Journal of educational Statistics 2*, 1 (1977), 1–26.

[112] Rubin, Donald B. Bayesian inference for causal effects: The role of randomization. *The Annals of statistics* (1978), 34–58.

[113] Rubin, Donald B. Using the sir algorithm to simulate posterior distributions. *Bayesian statistics 3* (1988), 395–402.

[114] Saad, Feras A, Cusumano-Towner, Marco F, Schaechtle, Ulrich, Rinard, Martin C, and Mansinghka, Vikash K. Bayesian synthesis of probabilistic programs for automatic data modeling. *Proceedings of the ACM on Programming Languages 3*, POPL (2019), 37.

[115] Salimans, Tim, Kingma, Diederik, and Welling, Max. Markov chain monte carlo and variational inference: Bridging the gap. In *International Conference on Machine Learning* (2015), PMLR, pp. 1218–1226.

[116] Schulam, Peter, and Saria, Suchi. Reliable decision support using counterfactual models. In *Advances in Neural Information Processing Systems* (2017), pp. 1697–1708.

[117] Shadish, William, Clark, Margaret, and Steiner, Peter. Can nonrandomized experiments yield accurate answers? a randomized experiment comparing random and nonrandom assignments. *Journal of the American Statistical Association 103*, 484 (2008), 1334–1344.

[118] Shalit, Uri, Johansson, Fredrik, and Sontag, David. Estimating individual treatment effect: Generalization bounds and algorithms. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70* (2017), ICML'17, JMLR.org, p. 3076–3085.

[119] Sherman, Eli, and Shpitser, Ilya. Intervening on network ties. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence* (2019).

[120] Silva, Ricardo, and Gramacy, Robert B. Gaussian process structural equation models with latent variables. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence* (2010), pp. 537–545.

[121] Snelson, Edward, Ghahramani, Zoubin, and Rasmussen, Carl E. Warped gaussian processes. In *Advances in neural information processing systems* (2004), pp. 337–344.

[122] Swanson, Sonja A, Hernán, Miguel A, Miller, Matthew, Robins, James M, and Richardson, Thomas S. Partial identification of the average treatment effect using instrumental variables: review of methods for binary instruments, treatments, and outcomes. *Journal of the American Statistical Association 113*, 522 (2018), 933–947.

[123] Talts, Sean, Betancourt, Michael, Simpson, Daniel, Vehtari, Aki, and Gelman, Andrew. Validating bayesian inference algorithms with simulation-based calibration. *arXiv preprint arXiv:1804.06788* (2018).

[124] Tavares, Zenna, Koppel, James, Zhang, Xin, Das, Ria, and Solar-Lezama, Armando. A language for counterfactual generative models. In *International Conference on Machine Learning* (2021), PMLR, pp. 10173–10182.

[125] Thistlethwaite, Donald L, and Campbell, Donald T. Regression-discontinuity analysis: An alternative to the ex post facto experiment. *Journal of Educational psychology 51*, 6 (1960), 309.

[126] Tian, Jin, and Pearl, Judea. Probabilities of causation: Bounds and identification. *Annals of Mathematics and Artificial Intelligence 28*, 1 (2000), 287–313.

[127] Titsias, Michalis, and Lawrence, Neil. Bayesian gaussian process latent variable model. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* (2010), pp. 844–851.

[128] Tran, Dustin, and Blei, David M. Implicit causal models for genome-wide association studies. In *International Conference on Learning Representations* (2018).

[129] Valdes-Sosa, Pedro A, Roebroeck, Alard, Daunizeau, Jean, and Friston, Karl. Effective connectivity: influence, causality and biophysical modeling. *Neuroimage 58*, 2 (2011), 339–361.

[130] Van der Laan, Mark J, Polley, Eric C, and Hubbard, Alan E. Super learner. *Statistical applications in genetics and molecular biology 6*, 1 (2007).

[131] Wainwright, Martin J, Jordan, Michael I, et al. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning 1*, 1–2 (2008), 1–305.

[132] Wang, Yixin, and Blei, David M. The blessings of multiple causes. *Journal of the American Statistical Association* (2019), 1–71.

[133] Wang, Yuhao, Solus, Liam, Yang, Karren, and Uhler, Caroline. Permutation-based causal inference algorithms with interventions. In *Advances in Neural Information Processing Systems* (2017), pp. 5822–5831.

[134] Witty, Sam, and Jensen, David. Causal graphs vs. causal programs: The case of conditional branching. In *Proceedings of the First Conference on Probabilistic Programming* (2018).

[135] Witty, Sam, Jensen, David, and Mansinghka, Vikash. A simulation-based test of identifiability for bayesian causal inference, 2022.

[136] Witty, Sam, Lee, Jun K, Tosch, Emma, Atrey, Akanksha, Clary, Kaleigh, Littman, Michael L, and Jensen, David. Measuring and characterizing generalization in deep reinforcement learning. *Applied AI Letters 2*, 4 (2021), e45.

[137] *Witty, Sam, *Lew, Alexander, Jensen, David, and Mansinghka, Vikash. Bayesian causal inference via probabilistic program synthesis. In *Proceedings of the Second Conference on Probabilistic Programming* (2020).

[138] Witty, Sam, Takatsu, Kenta, Jensen, David, and Mansinghka, Vikash. Causal inference using gaussian processes with structured latent confounders. In *International Conference on Machine Learning* (2020), PMLR, pp. 10313–10323.

[139] Xia, Kevin, Lee, Kai-Zhan, Bengio, Yoshua, and Bareinboim, Elias. The causal-neural connection: Expressiveness, learnability, and inference. *Advances in Neural Information Processing Systems 34* (2021).

[140] Yalburgi, Sharan, Freer, Cameron, Quinn, Jameson, Weiner, Veronica, Witty, Sam, and Mansinghka, Vikash. Assessing inference quality for probabilistic programs using multivariate simulation based calibration. In *Proceedings of the Third Conference on Probabilistic Programming* (2021).

[141] Yang, Karren, Katcoff, Abigail, and Uhler, Caroline. Characterizing and learning equivalence classes of causal dags under interventions. In *International Conference on Machine Learning* (2018), PMLR, pp. 5541–5550.

[142] Zhang, Junzhe, Tian, Jin, and Bareinboim, Elias. Partial counterfactual identification from observational and experimental data. *arXiv preprint arXiv:2110.05690* (2021).

[143] Zhang, Kun, Schölkopf, Bernhard, and Janzing, Dominik. Invariant Gaussian process latent variable models and application in causal discovery. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence (UAI)* (2010).

[144] Zheng, Xun, Aragam, Bryon, Ravikumar, Pradeep K, and Xing, Eric P. Dags with no tears: Continuous optimization for structure learning. *Advances in Neural Information Processing Systems 31* (2018).