EFFECTIVE, EFFICIENT, AND EXPLAINABLE LEARNING IN DATA-SCARCE AND NOISY DOMAINS

by

Bhagirath Athresh Karanam

APPROVED BY SUPERVISORY COMMITTEE:

Sriraam Natarajan, Chair

Vibhav Gogate

Rishabh Iyer

Nicholas Ruozzi

EFFECTIVE, EFFICIENT, AND EXPLAINABLE LEARNING IN DATA-SCARCE AND NOISY DOMAINS

by

BHAGIRATH ATHRESH KARANAM, B.Tech

DISSERTATION

Presented to the Faculty of

The University of Texas at Dallas

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY IN

COMPUTER SCIENCE

THE UNIVERSITY OF TEXAS AT DALLAS

April 2026

ACKNOWLEDGMENTS

I would like to begin by thanking my advisor, Prof. Sriraam Natarajan, for his unwavering support and mentorship. His integrity and dedication to the unadulterated pursuit of scholarly excellence have consistently inspired me to hold myself to a higher standard, and I will continue to strive toward the high bar he has set for the rest of my career and beyond. His infectious enthusiasm for the field and his humility as a learner have cemented my desire to be, above all, a lifelong student. I am grateful for his perspective on research, which encouraged me to read a wide range of papers deeply and critically. I am also grateful for the many opportunities he has provided over the years: arranging travel to conferences and summer workshops, introducing me to remarkable collaborators, and trusting me to review papers and organize tutorials. Prof. Natarajan, thank you for your patience. Most importantly, I am deeply grateful for your kindness and for always believing in me, even at times when I found it hard to believe in myself. I thank my committee members, Prof. Vibhav Gogate, Prof. Rishabh Iyer, and Prof. Nicholas Ruozzi, for their support and kind consideration throughout my graduate years. The courses I took with Prof. Gogate and Prof. Iyer are foundational to this dissertation, and I deeply admire their ability to explain complex topics with clarity and precision. In particular, I collaborated with Prof. Iyer on a paper that became one of the four primary chapters of this dissertation. Working with him taught me that asking the right questions is often harder and more important than finding the answers, an attitude that has since become central to the way I approach new problems.

I would like to thank my collaborators: Prof. David M. Haas, for his collaboration and medical expertise on the nuMoM2b project; Prof. Kristian Kersting, for his mentorship on two projects that inform the technical core of this dissertation, particularly sum-product networks; and Prof. Predrag Radivojac, for his mentorship while working on the nuMoM2b project, which became a central problem motivating most of the methods developed in this

dissertation. Members of the Starling Lab have been a constant source of collaboration, discussion, and support throughout my PhD journey. I thank Varun Balaji, Dr. Yuqiao Chen, Dr. Mayukh Das, Dr. Srijita Das, Dr. Devendra Dhami, Justin Dula, Fateme Golivand, Alexander Hayes, Dr. Navdeep Kaur, Dr. Harsha Kokel, Kavimayil Periyakoravampalayam Komarasam, Dr. Gautam Kunapuli, Dr. Saurabh Mathur, Nikhilesh Prabhakar, Dr. Nandini Ramanan, Dr. Brian Ricks, Sahil Sidheekh, Ranveer Singh, Michael (Mike) Skinner M.D., Pranuthi Tenali, Dr. Siwen Yan, and Yuxin Zi. In particular, I would like to thank Dr. Harsha Kokel, Dr. Saurabh Mathur, and Sahil Sidheekh for their collaboration and thoughtful discussions, many of which directly led to ideas that form a large portion of this dissertation. I greatly admire their collective work ethic. I would also like to thank my peers from outside the Starling Lab who collaborated with me on the iSPN and nuMoM2b projects: Hoyin Chu, Rashika Ramola, and Matej Zečević.

I owe an immeasurable debt of gratitude to my family, whose love and encouragement have been my foundation throughout this journey. To my wife, Sowmya: none of this would have been possible without you. You have been my steadiest source of strength through every late night, every moment of self-doubt, and every small victory along the way. Your patience, understanding, and unwavering belief in me carried me through the hardest stretches of graduate school. More importantly, I am also grateful to you for always laughing at my jokes, even at the ones that did not quite deserve it. To my sister, Manasvi, whose encouragement and warmth have meant more to me than she knows, and to my parents, Geeta and Gopal Rao Karanam, whose unconditional love shaped the person I am today: thank you. I am endlessly fortunate to have each of you in my corner. I also thank my friends from my undergraduate years, the ”3D lobby,” for making the journey far less lonely and a great deal more fun.

I acknowledge the grants that supported this research: NIH award R01HD101246 and ARO award W911NF2010224.

April 2026

EFFECTIVE, EFFICIENT, AND EXPLAINABLE LEARNING IN DATA-SCARCE AND NOISY DOMAINS

Bhagirath Athresh Karanam, PhD
The University of Texas at Dallas, 2026

Supervising Professor: Sriraam Natarajan, Chair

Deep learning has seen a lot of success over the recent years, in large part due to increasing dataset sizes. For example, the size of datasets used to train language models has doubled approximately every seven months over the last few years. A similar trend can be seen in domains such as computer vision and speech. Model sizes and the amount of computational power needed to train these models have followed a similar trend. However, many real-world datasets, especially in domains such as healthcare, social sciences, and psychology are not nearly as large, are harder to obtain, and are often noisy. This dissertation considers learning in data-scarce and noisy domains by directly addressing three tenets, namely effectiveness, efficiency, and explainability. By effectiveness we refer to learning models that faithfully capture the underlying distributions of complex real-world datasets. Efficiency refers to both sample efficiency and computational efficiency, thus resulting in algorithms that reduce the number of samples needed needed to faithfully capture complex distributions and doing so while using fewer computational resources. Finally, explainability refers to algorithmic generation of representations of machine learning models that are both human-interpretable and faithful to the original model. With this general direction in mind, this dissertation focuses on developing methods that are built upon one or more of these tenets. First, we build upon the extensive existing literature regarding incorporating expert knowledge

into machine learning systems, by proposing a unified framework for incorporating a wide variety of domain constraints that can be provided naturally by domain experts. Second, we propose a framework for improving the computational efficiency of training models suitable for domain adaptation through a data subset selection scheme. Third, we propose a method for extracting human-interpretable representations from a class of probabilistic models called probabilistic circuits. Fourth, we develop a simple yet effective method for extracting qualitative knowledge from learned probabilistic models to facilitate explainability, build trust among domain experts, and inform effective clinical decision-making.

TABLE OF CONTENTS

ACKNOWLEDGMENTS iv
ABSTRACT vii
LIST OF FIGURES xii
LIST OF TABLES xv
CHAPTER 1 INTRODUCTION 1
CHAPTER 2 BACKGROUND 6
2.1 Notation 6
2.2 Probabilistic Queries 7
2.3 Probabilistic Graphical Models and the Limits of Exact Inference 9
2.4 Probabilistic Circuits 10
2.4.1 Tractable Probabilistic Models 10
2.4.2 Formal Definition 12
2.4.3 Structural Properties and Tractability 13
2.4.4 Learning Probabilistic Circuits 14
2.5 Encoding Domain Knowledge 16
2.6 Causal Inference and Structural Causal Models 18
2.7 Supervised Domain Adaptation and Submodular Functions 20
CHAPTER 3 EXPLAINING PROBABILISTIC CIRCUITS 24
3.1 Introduction 25
3.2 Background 27
3.3 $\mathcal{EXSPN}$ Algorithm 30
3.4 Experimental Evaluation 35
3.5 Results 39
3.6 Discussion and Conclusion 41
CHAPTER 4 EXTRACTING QUALITATIVE KNOWLEDGE FROM PROBABILISTIC MODELS 42
4.1 Introduction 43
4.2 Background 43

4.3 Proposed Approach . . . 44
4.4 Learning Qualitative Influences for GDM Modeling . . . 47
CHAPTER 5 EFFICIENT DOMAIN ADAPTATION THROUGH DATA SUBSET SELECTION . . . 50
5.1 Introduction . . . 51
5.2 Methodology . . . 55
5.2.1 Connections to Previous Work . . . 58
5.3 Experimental Evaluation . . . 62
5.4 Discussion . . . 67
CHAPTER 6 HUMAN-ALLIED LEARNING OF PROBABILISTIC CIRCUITS . . . 68
6.1 Introduction . . . 69
6.2 Learning PCs with Domain Knowledge . . . 70
6.2.1 Encoding Knowledge as Constraints . . . 71
6.3 Defining the Penalty Function . . . 73
6.4 Experimental Evaluation . . . 75
6.4.1 Experimental Setup . . . 76
6.4.2 Results . . . 78
6.5 Discussion . . . 83
CHAPTER 7 LEARNING INTERVENTIONAL DISTRIBUTIONS USING PROBABILISTIC CIRCUITS . . . 86
7.1 Introduction . . . 87
7.2 Background and Related Work . . . 90
7.3 Interventional SPNs . . . 92
7.3.1 Adaptation to Causal Change . . . 92
7.3.2 Data Generating Process . . . 94
7.3.3 Introducing Interventional SPNs . . . 95
7.4 Experimental Results . . . 100
7.5 Conclusions . . . 104
CHAPTER 8 EXPLOITING DOMAIN KNOWLEDGE AS CAUSAL INDEPENDENCIES . . . 106
8.1 Introduction . . . 107

8.2 Data Description 108
8.3 Background: Knowledge-Guided Learning 109
8.3.1 Causal Independence 109
8.4 Causal Independencies with Qualitative Constraints for Modeling GDM . . . 112
8.4.1 Parameter Learning under Monotonicity Constraints 113
8.4.2 Derivation of the gradients of the log-likelihood term 115
8.4.3 Derivation of the gradients of the penalty term 116
8.5 Experimental Evaluation 117
8.6 Discussion 119
CHAPTER 9 CONCLUSION AND FUTURE WORK 121
REFERENCES 129
BIOGRAPHICAL SKETCH 144
CURRICULUM VITAE

LIST OF FIGURES

3.1 SPN that represents the joint distribution $P(X_1, X_2)$. Sum nodes are inscribed with "+", product nodes with "×" and leaf nodes with corresponding variable names(*best viewed in color).* 26

3.2 A sum-product network (left) and its corresponding CSI-Tree (right) 27

3.3 The sum-product network (left) trained on the synthetic data set and its corresponding CSI-tree extracted by $\mathcal{EXSPN}$ (right). Each edge in the CSI-tree corresponds to an edge from a sum node to a product node in the SPN. All other edges are collapsed. Clearly, the CSI-tree encodes all the CSIs represented in the sum-product network. . . 36

3.4 First two levels of the CSI-tree for the nuMoM2b dataset. Here, *oDM* is a boolean variable that represents whether or not the person has Gestational Diabetes. *Race* is a categorical variable having 8 categories namely, Non-Hispanic White (1), Non-Hispanic Black (2), Hispanic (3), American Indian (4), Asian (5), Native Hawaiian (6), Other (7), and Multiracial (8). *Smoked3Months* and *SmokedEver* are boolean variables representing tobacco consumption. 37

5.1 Illustration of ORIENT framework. **(1)** Given the target data D^t and source data D^s, a subset $A \subseteq D^s$ and model parameters θ are randomly initialized. **(2)** Model parameters θ are optimized for L epochs on the subset A with any gradient descent (GD) based method. **(3)** Subset A is updated using the SMI measure and current model parameters θ after every L^{th} epoch. **(4)** Model is trained for E epochs, with subset selection after every L interval. **(5)** The final model parameters are returned after E epochs. 52

5.2 Speed up vs prediction accuracy on three domains of Office31 dataset: Amazon (A), DSLR (D), and Webcam (W). X-axis represents the speed up by the model, i.e. the ratio of the time taken to train on complete source dataset (Full) to the time taken by the model. Y-axis represents the prediction accuracy of the model on the target domain. 57

5.3 GradCam (Selvaraju et al., 2017) activation maps of the models learned using d-SNE + Full and d-SNE + ORIENT (FLMI) on the Office-Home dataset with "Product" as the source domain and "Real World" as the target domain. As evidenced by class activation maps, the ORIENT framework enabled the model to learn more effective class discriminative features than Full data training. 62

5.4 Speed up achieved by combining d-SNE with ORIENT 63

5.5 Convergence curves on Office-Home for domain adaptation across Art (A), Clipart (C), Product (P), and Real World (R). The x-axis shows training time (hours), and the y-axis shows target-domain accuracy. 64

6.1 **3D Helix dataset**: Visualization of the 3D Helix dataset. The train dataset (a) consists of a single helix of length 2π with Gaussian noise added to it. The test (b) dataset consists of a helix of length 4π with Gaussian noise added to it. The two figures on the right show 1000 samples generated from *EinsumNet* without (c) and with (d) incorporating Generalization Constraint (GC). 76

6.2 **Learning Curves:** Mean train and validation log-likelihoods of *EinsumNet* trained with (in red) and without (in blue) incorporating Generalization Constraint (GC) on the three Set-MNIST datasets, across training epochs. The shaded regions denote the standard deviation across 3 independent trials. 79

6.3 **Qualitative Evaluation**: Visualization of randomly sampled datapoints generated by *EinsumNet* trained with and without the Generalization Constraint (GC) on Set-MNIST-Odd. 82

6.4 **Ablation Study:** Mean test log-likelihood of a *RatSPN* trained on the Set-MNIST-Even dataset, across varying (a) size of domain sets and (b) degrees of noise. Shaded regions represent the SD across 3 trials. 84

7.1 **Capturing interventional distributions using iSPN**. The interventional distributions for the ASIA data set using a causal Bayesian network (CBN, small-scale gold standard, gray bars) as well as an interventional SPN (iSPN) by intervening on *lung*. The iSPN is sensible to all the influences of the given intervention onto the system i.e., subsequent effects in the causal hierachy. (Best viewed in color.) 88

7.2 **An Overview of iSPN.** (a) The hidden process underlying the observable reality is modeled via a SCM that can be modified through interventions. The given SCM induces a causal graph and can generate data. An intervention can significantly alter the resulting data. (b) An over-parameterized density estimation framework using SPN is presented. The universal function approximator (here neural network) conditions on the mutilated causal graph and provides parameters to the SPN such that the given data's density can be modeled accordingly. The FA subsumes the *do*-operator. (c) Different causal queries are being presented. Furthermore, iSPN adapts to intervention-consequences. 93

7.3 **Mean running times in seconds until convergence (Causal Health)** for 50 full passes. More data set results are in the supplementary material. 100

7.4 **Generative Baselines.** A comparison to the ground-truth (via underlying SCM) and competing estimated distributions. Each row represents a data set and each column represents a variable for a given causal query. (Best in color.) 102

7.5 **Causal Baselines.** Different causal structures and corresponding causal effect estimation methods (CausalML, DoWhy) are being compared against iSPN. When confounding is present, then conditioning becomes different from intervening $p(Y \mid X) \neq p(Y \mid do(X))$ and iSPN correctly captures all evaluated cases. (* are analytical solutions, [1] differences of means for actual interventional distributions, Best viewed in color.) 104

7.6 **Adaptation to Different Interventions.** Training results for different kinds of interventions on the continuous CH data set. Left, the respective mean objective curves (log-likelihood), indicating consistent training and convergence for all three random seeds per configuration. Right, the (mean) density functions for two different interventions on H: Uniform $U(a, b)$ and Gamma $\Gamma(p, q)$ (other interventions shown in the supplementary). (Best viewed in color.) 105

8.1 Flowchart illustrating the process of selecting the cohorts for our experiments. The two sub-cohorts used in our experiments are indicated in green. 110

8.2 A belief network for multiple causes and a single effect (left) and Temporal interpretation of Independence of causal influence (right). 111

8.3 The Noisy-Or model 112

8.4 Flowchart for the Noisy-Or model construction process 118

8.5 Noisy-OR model used for the GDM dataset. Both QIs and causal independence knowledge are incorporated in this model. This representation shows that *PRS, Hist, PCOS, HiBP, Age and BMI* have a positive monotonic influence on GDM whereas *METs* have a negative monotonic influence. Additionally, all the risk factors are causally independent in this model. 118

8.6 The AUC-ROC scores for the Noisy OR model (NOR) as compared to the Gradient Boosted Trees model (GBT) with PRS (left) and without PRS (center). The AUC-ROC scores for the Noisy OR model (NOR) in the presence of PRS and Qualitative Influences (right). The bars show the mean score over 10 boostrap samples and the error bars show the standard deviation. 120

LIST OF TABLES

3.1 Summary statistics for the CSI rules extracted from SPNs by $\mathcal{EXSPN}$ and the association rules by Apriori algorithm. NP - # Product Nodes, NR - # Rules, MA - Mean Antecedent length, MC - the Mean Consequent length, and CR - Compression Ratio. 36

3.2 Summary statistics for the case where the instance function is inferred after training. The columns are the same as Table 3.1. 37

3.3 The Number of CSIs extracted by ExSPN from an SPN fit on each dataset (Total), The number of CSIs where at least 80% of the datapoints in the context matched with the CSIs from the Bayesian Network(BN) (Correct) and the Ratio of correct CSIs (Ratio). 38

3.4 Dataset details. 39

3.5 Hyperparameters(HP) 39

4.1 Comparision of QI from prior knowledge (PK), QuaKE and Data Alone. ✓/✗ represents that this relationship does/not exist respectively while **?** represents unknown influence. The three groups of rows show: (1) MI, (2) SI, and (3) sub-SI. Colors highlight rules recovered by QuaKE and show (a.) coherent with the PK and baseline (b.) contradicting the baseline (c.) coherent with baseline but contradicts the PK. 48

5.1 Instantiations of submodular mutual information functions (Kothawade et al., 2022). 55

5.2 Test accuracy for office-31 with SDA methods 63

5.3 Test accuracy for Office-Home with SDA methods 63

6.1 **Quantitative Evaluation:** Mean test log-likelihood of *RatSPN* trained with and without constraints on the Bayesian Network (BN), UCI Benchmark and Real-World (RW) datasets, ± standard deviation across 3 independent trials. 78

6.2 **Quantitative Evaluation**: Mean test log-likelihood of *RatSPN* and *EinsumNet* trained with and without the Generalization Constraint (GC) on the Helix and Set-MNIST datasets, ± standard deviation across 3 independent trials. 80

6.3 The test log-likelihoods for the RatSPN learned from data, with one form of knowledge and with two forms of knowledge averaged over 3 independent trials 81

7.1 **Jensen-Shannon-Divergence Evaluation of Estimated Interventional Distributions**. Numerical counterpart to Fig. 7.4, mean and standard deviation per $p(V_{j\setminus i} \mid do(V_i = U(V_i)))$ where U is the uniform distribution across all data sets. Lower = better. 100

CHAPTER 1

INTRODUCTION

Over the past decade, deep learning has transformed the landscape of AI, achieving remarkable success across a wide range of domains. Breakthroughs in natural language processing, computer vision, and speech recognition have been driven not only by algorithmic innovations, but also, and perhaps more decisively, by the unprecedented growth in the scale of available data. In particular, the datasets used to train modern language models have expanded at an extraordinary pace, reportedly doubling in size approximately every several months in recent years. Similar, albeit less dramatic, trends can be observed in computer vision and speech domains, where large-scale annotated datasets have become a central pillar of progress. This rapid expansion in data availability has been accompanied by a corresponding increase in model size and computational requirements, resulting in systems that are both highly expressive and resource-intensive.

While these developments have enabled state-of-the-art performance in many benchmark settings, they have also exposed a fundamental limitation of contemporary machine learning paradigms: *their reliance on large, high-quality datasets.* In contrast to the relatively controlled environments of benchmark datasets, many real-world applications, particularly those in healthcare, are characterized by limited and noisy data. In such settings, collecting large amounts of labeled data is often infeasible due to cost, privacy concerns, or logistical constraints. Moreover, the consequences of model failure in these domains can be severe, necessitating a higher standard of reliability, robustness, and interpretability. These challenges underscore the need for machine learning approaches that move beyond the prevailing "more data, more compute" paradigm and instead emphasize principled methods for (1) leveraging domain knowledge to improve model effectiveness and sample efficiency, (2) reducing the computational resources required so that they can be widely deployed by practitioners working

in resource-scarce environments, and (3) generating post-hoc explanations to improve trust among domain experts.

This dissertation is motivated by the premise that the successful deployment of machine learning systems in data-scarce and safety-critical domains hinges on three interrelated tenets: **effectiveness, efficiency, and explainability.** These are understood here as follows:

- *Effectiveness*, in this context, refers to the ability of a model to faithfully capture the underlying structure of complex, real-world data distributions, even when observations are scarce or noisy. Achieving such fidelity often requires incorporating inductive biases or domain knowledge that guide the learning process toward plausible solutions.

- *Efficiency* encompasses both sample efficiency and computational efficiency. That is, we seek methods that can learn accurate models from fewer data points while also reducing the computational resources required for training and inference. This is particularly important in practical settings where data collection is expensive and computational infrastructure may be limited.

- *Explainability*, in the context of this dissertation, refers to generating post-hoc explanations and extracting human-interpretable representations from learned models to improve transparency and build trust among domain experts.

Guided by these principles, this dissertation explores a collection of methods designed to address the challenges of learning in data-scarce and noisy domains. A common theme throughout this work is the integration of domain knowledge into the learning pipeline. While there exists a rich body of literature on incorporating expert knowledge into models, existing approaches are often tailored to specific types of constraints, require experts to express their knowledge in forms that are not intuitive, or are restricted to a small family of models. To address this limitation, this dissertation proposes a **unified framework for incorporating**

a diverse range of domain constraints, allowing experts to inject prior knowledge in an expressive and intuitive manner. This framework aims to bridge the gap between domain expertise and algorithmic design, enabling more effective learning in settings where data alone is insufficient.

In line with the tenet of efficiency, this dissertation investigates methods for improving the efficiency of model training, particularly in the context of domain adaptation, where data from a source domain is abundant but the target domain has only a few labeled examples. Modern machine learning systems are frequently deployed in environments that differ from the conditions under which they were trained, necessitating adaptation to new data distributions. However, retraining or fine-tuning models on large datasets can be computationally prohibitive. To address this challenge, this work introduces a **data subset selection framework** that identifies informative subsets of source data most relevant to the target domain, thereby reducing computational overhead while maintaining or even improving performance.

Finally, a significant focus of this dissertation is the development of methods for extracting interpretable representations from complex probabilistic models. In particular, we consider probabilistic circuits, a class of models that offer tractable inference while retaining expressive power (Choi et al., 2020; Sidheekh and Natarajan, 2024). Despite their favorable computational properties, the internal structure of probabilistic circuits can be difficult to interpret directly. To address this, we propose techniques for deriving **human-interpretable explanations** that faithfully reflect the behavior of the underlying model, enabling users to better understand and trust its predictions. Complementing these efforts, the dissertation also proposes methods for extracting qualitative knowledge from trained probabilistic models. Beyond quantitative predictions, such models often encode rich structural information about the relationships between variables. By systematically uncovering and presenting this information in an accessible form, we aim to facilitate communication between machine learning systems and domain experts, supporting decision-making in applications such as clinical practice where understanding the rationale behind a model's output is critical.

Taken together, the primary contributions of this dissertation advance the broader goal of making machine learning more applicable to real-world settings where data and compute are limited, domain expertise is invaluable, and transparency is essential. The dissertation additionally presents collaborative explorations in causal probabilistic learning that extend the themes of tractability and domain knowledge into the setting of causal inference. By emphasizing effectiveness, efficiency, and explainability throughout, this work advocates for a shift in perspective - from scaling up data and models to designing principled, expert-aware, resource-mindful, and transparent learning systems.

Organization

The remainder of this dissertation is organized as follows. Chapter 2 provides the necessary background on probabilistic circuits, tractable inference, submodular functions, and supervised domain adaptation that underlies the technical contributions of this work.

The next four chapters constitute the primary contributions of the dissertation, presented in an order that reflects the progression of the author's research. The first two chapters address the tenet of explainability. Chapter 3 introduces *CSI-trees*, a compact and human-interpretable representation of the context-specific independencies encoded in a learned sum-product network, along with an efficient algorithm for their extraction. Chapter 4 extends this line of inquiry by proposing QuaKE, a method for extracting qualitative influence statements – including monotonic and synergistic relationships between variables – directly from a learned probabilistic model, with application to a real-world gestational diabetes study. Together, these two chapters establish a vocabulary of interpretable knowledge structures that probabilistic models can encode and that domain experts can reason over. Chapter 5 then addresses the tenet of efficiency by introducing ORIENT, a data subset selection framework based on submodular mutual information that accelerates supervised domain adaptation by training on informative subsets of source data relevant to the target domain. Finally,

Chapter 6 addresses the tenet of effectiveness – and serves as the most comprehensive contribution of this dissertation – by proposing a unified framework for incorporating a wide variety of domain constraints, including the knowledge structures surfaced in the preceding chapters, into the parameter learning of probabilistic circuits.

The final two chapters present works in which the dissertation author served as a contributing rather than primary author – referred to as collaborative explorations throughout the remainder of this dissertation – that extend the themes of this dissertation into the setting of causal probabilistic learning. Chapter 7 introduces interventional sum-product networks (iSPNs), which learn to model interventional distributions by conditioning on the structure of a mutilated causal graph, connecting tractable probabilistic models with causal inference. Chapter 8 presents a complementary approach that combines causal independence assumptions and qualitative influence statements within a Noisy-Or probabilistic model for the prediction of gestational diabetes mellitus, demonstrating the practical value of structured domain knowledge in safety-critical clinical settings.

CHAPTER 2

BACKGROUND

This chapter introduces the background necessary to situate the contributions of this dissertation within the broader landscape of probabilistic machine learning. The ideas presented in the chapters that follow are built upon extensive work in machine learning and AI spanning several decades, and this chapter aims to reflect that foundation. We begin by establishing notation and motivating the class of probabilistic queries that are central to this work. We then review the treatment of these queries in probabilistic graphical models and explain the computational hardness results that motivate a fundamentally different modeling paradigm. We describe probabilistic circuits (Choi et al., 2020; Sidheekh and Natarajan, 2024) as a unified representation for computational graphs designed to enable tractable inference, trace the body of work that has converged on this representation, and describe the structural properties that underlie tractability. The chapter concludes with background on supervised domain adaptation and data subset selection, which underlies the efficiency contributions of the dissertation.

2.1 Notation

Throughout this dissertation, we consider a set of n random variables $\mathbf{X} = \{X_1, \ldots, X_n\}$ and a probability distribution $P(\mathbf{X})$ defined over them. We use bold letters to denote sets of variables (e.g., $\mathbf{X}$), uppercase letters to denote individual random variables (e.g., X_i), and bold lowercase letters to denote assignments or instantiations of variables (e.g., $\mathbf{x}$). For a subset $\mathbf{X}_e \subseteq \mathbf{X}$, we denote its instantiation as $\mathbf{x}_e$ and write $P(\mathbf{x}_e)$ as shorthand for $P(\mathbf{X}_e = \mathbf{x}_e)$. We use $\mathrm{Dom}(X_i)$ to denote the domain of variable X_i, and $\mathbf{X} \setminus \mathbf{X}_e$ to denote the complement of $\mathbf{X}_e$ in $\mathbf{X}$.

2.2 Probabilistic Queries

A central concern of this dissertation is the development of probabilistic models that can answer a well-defined class of queries both exactly and efficiently. Before formalizing these queries, we motivate their importance through a running example drawn from the nuMoM2b study (Haas et al., 2015), which recurs throughout this dissertation. The nuMoM2b study tracked pregnancies of over $10,000$ nulliparous women across eight clinical sites in the United States, collecting rich clinical, demographic, and genetic information with the goal of identifying early warning signs of adverse pregnancy outcomes (APOs) such as gestational diabetes mellitus (GDM), preeclampsia, and preterm birth.

Consider a probabilistic model learned over variables $\mathbf{X}$ representing clinical risk factors, including body mass index (BMI), maternal age (Age), family history of diabetes ($Hist$), physical activity level ($METs$), presence of polycystic ovary syndrome ($PCOS$), blood pressure ($HiBP$), and a polygenic risk score (PRS), along with the outcome variable GDM. A clinician or decision-support system working with such a model might need to answer several qualitatively distinct questions:

- **How likely is this particular patient profile?** Given a complete observation $\mathbf{x}$ over all variables for a patient, one might ask for the probability $P(\mathbf{x})$ of that joint configuration. This is useful, for example, in identifying anomalous or potentially erroneous clinical measurements.

- **What is the marginal risk of GDM given partial information?** In clinical practice, patient records are frequently incomplete. Given only a patient's BMI and Age with other factors unrecorded, a clinician might ask for the probability of GDM marginalized over all unobserved variables.

- **Given what we know about this patient, what is the probability of GDM?** Given observed values for a subset of risk factors, a decision-support system might

compute the conditional probability of a positive GDM outcome. This is the most directly actionable query for clinical use.

These three questions correspond to the evidential, marginal, and conditional query types defined below. It is instructive to contrast this with the predominant paradigm in modern deep learning, where models are trained discriminatively to predict Y from $\mathbf{X}$. A deep neural network trained for GDM classification can approximate the third type of query but cannot directly answer the first two: it does not model the joint distribution $P(\mathbf{X})$ and therefore has no mechanism for computing $P(\mathbf{x})$ or $P(\mathbf{x}_e)$ for arbitrary subsets $\mathbf{X}_e \subseteq \mathbf{X}$. In safety-critical domains such as healthcare, where understanding uncertainty over unobserved variables, detecting anomalous inputs, and reasoning about partial evidence are essential, this limitation has real consequences.

Formally, given $\mathbf{X} = \{X_1, \ldots, X_n\}$ and a joint distribution $P(\mathbf{X})$, we are interested in the following queries:

1. **Evidential Query (*EVI*)**: *EVI* computes the likelihood of a complete assignment $\mathbf{x}$ over $\mathbf{X}$, i.e., $P(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n)$, written $P(\mathbf{x})$ for succinctness.

2. **Marginal Query (*MAR*)**: *MAR* computes the probability of an assignment $\mathbf{x}_e$ over a partial subset of variables $\mathbf{X}_e \subseteq \mathbf{X}$, i.e., $P(\mathbf{x}_e) = \sum_{\mathbf{x} \setminus \mathbf{x}_e} P(\mathbf{x})$.

3. **Conditional Query (*CON*)**: Given disjoint subsets $\mathbf{X}_e, \mathbf{X}_q \subseteq \mathbf{X}$ with $\mathbf{X}_e \cap \mathbf{X}_q = \emptyset$, *CON* computes the conditional probability $P(\mathbf{x}_q \mid \mathbf{x}_e)$ of an assignment $\mathbf{x}_q$ given evidence $\mathbf{x}_e$.

Tractability. The above queries are in general computationally demanding to evaluate exactly. We formalize the notion of efficient exact inference using *tractability*: a class of queries $\mathcal{Q}$ is *tractable* on a class of probabilistic models $\mathcal{M}$ if and only if for any query $q \in \mathcal{Q}$ and model $m \in \mathcal{M}$, computing $q(m)$ exactly runs in time $O(\text{poly}(|m|))$, where $|m|$ denotes the size of the model representation.

2.3 Probabilistic Graphical Models and the Limits of Exact Inference

The dominant framework for reasoning about structured joint distributions over sets of random variables has long been that of *probabilistic graphical models* (PGMs) (Koller and Friedman, 2009). PGMs encode conditional independence relationships between variables using a graph structure, enabling compact representation of joint distributions and supporting principled probabilistic reasoning. The two most widely studied families are *Bayesian networks* (BNs) (Pearl, 1989), which use directed acyclic graphs to encode the factorization $P(\mathbf{X}) = \prod_i P(X_i \mid \text{pa}(X_i))$, and Markov random fields (Kindermann and Snell, 1980), which use undirected graphs to encode factorizations over clique potentials.

Despite their expressiveness and widespread adoption, PGMs face a fundamental computational barrier when answering the queries defined in Section 2.2. Exact computation of marginal probabilities in a Bayesian network is #P-complete in general (Cooper, 1990a; Roth, 1996). Informally, #P-completeness places the problem at least as hard as counting the number of solutions to an NP problem, placing exact marginal inference well beyond the reach of efficient algorithms in the worst case. Conditional queries, being ratios of marginals, inherit this hardness. Even obtaining approximations within a constant multiplicative factor of the marginal is NP-hard in general for BNs (Roth, 1996).

Classical exact inference algorithms such as the junction tree algorithm (Lauritzen and Spiegelhalter, 1988) and variable elimination can compute marginals exactly, but their complexity is exponential in the *treewidth* of the underlying graphical structure. Treewidth is a graph-theoretic quantity that measures, roughly, how far the graph is from being a tree. For the kinds of densely connected graphs that arise naturally in clinical settings such as nuMoM2b, where many risk factors are correlated with each other and with the outcome, treewidth grows rapidly, making exact inference computationally infeasible in practice. Approximate inference methods such as loopy belief propagation (Murphy et al., 1999) and Markov chain Monte Carlo (Hastings, 1970) sampling can scale to larger models

but provide no guarantees of accuracy. In safety-critical domains such as healthcare, where inaccurate probability estimates can directly influence clinical decisions, such approximations are difficult to justify.

These hardness results motivate a line of research into probabilistic models that achieve tractability not by approximation, but by design: by imposing structural constraints on the model representation that guarantee efficient exact inference. Probabilistic circuits have emerged as a unified representation for such models.

2.4 Probabilistic Circuits

Probabilistic circuits (PCs) are a class of tractable probabilistic models (Choi et al., 2020; Sidheekh and Natarajan, 2024; Van den Broeck et al., 2019) that represent joint distributions as computational graphs with explicitly designed structural properties that enable exact and efficient inference. PCs serve as a unified representation for a range of models developed over several decades in the machine learning and artificial intelligence communities, each of which can be understood as a special case or instantiation of this general framework. Before giving the formal definition, we briefly describe the body of work that has converged on the PC representation.

2.4.1 Tractable Probabilistic Models

Arithmetic Circuits. The foundational insight that probabilistic inference can be reduced to evaluating a polynomial over a computational graph was formalized by Darwiche (2003). An arithmetic circuit is a directed acyclic graph (DAG) whose leaves correspond to variables or parameters and whose internal nodes correspond to addition and multiplication operations. Darwiche (2003) showed that any BN can be compiled into an arithmetic circuit such that marginal and conditional queries can be answered in time linear in the circuit size, at the cost of a potentially exponential offline compilation step. This established a key insight: the

computational graph, rather than the graphical model itself, is the appropriate object for tractable inference. Follow-on work showed that different structural properties of the circuit correspond to different tractability guarantees (Choi and Darwiche, 2017; Rooshenas and Lowd, 2016; Ramanan et al., 2020).

Sum-Product Networks. Building on the arithmetic circuit framework, Poon and Domingos (2011) introduced sum-product networks (SPNs) as a class of arithmetic circuits that can be learned directly from data, without requiring compilation from an existing probabilistic model. SPNs are composed of sum and product nodes defined over tractable univariate distributions at the leaves, and satisfy the structural constraints of completeness and consistency, equivalent to smoothness and decomposability respectively, which together guarantee tractable computation of *EVI*, *MAR*, and *CON* queries. The SPN formulation prompted a substantial body of follow-on work, including structure learning algorithms (Gens and Domingos, 2013; Vergari et al., 2015; Rooshenas and Lowd, 2014, 2013), discriminative variants (Gens and Domingos, 2012), latent variable interpretations (Peharz et al., 2017), parameter learning methods (Zhao and Poupart, 2016), and a growing range of applications (Molina et al., 2018, 2017; Shao et al., 2020). Surveys of this literature are available in París et al. (2020) and Van den Broeck et al. (2019).

Probabilistic Sentential Decision Diagrams. Kisa et al. (2014) introduced probabilistic sentential decision diagrams (PSDDs), which extend sentential decision diagrams from the knowledge compilation literature to represent probability distributions. PSDDs are both deterministic and decomposable (defined formally below), enabling tractable computation of a rich class of queries. They are particularly well-suited to structured domains where the support of the distribution can be described by a logical formula, and have been applied to tasks requiring reasoning over complex combinatorial spaces.

Cutset Networks. Rahman et al. (2014b) introduced cutset networks (CNs), which combine the structure of Bayesian networks with the tractability of tree-structured distributions. A CN is defined by conditioning on a set of *cutset* variables that render the remaining

variables jointly independent, yielding a tree distribution that supports exact and efficient inference. CNs can be represented as deterministic, decomposable PCs, making them a special case within the broader framework. They have been extended to ensemble methods (Rahman and Gogate, 2016), compiled into latent-variable-free representations (Rahman et al., 2019), and applied to knowledge-intensive learning settings (Mathur et al., 2023).

The Probabilistic Circuits Framework. Choi et al. (2020) established probabilistic circuits as a unified framework subsuming arithmetic circuits, PSDDs, SPNs, cutset networks, and related models. This unification clarified that the tractability properties across these model families all derive from the same underlying structural constraints on the computational graph, and that different families represent different points in the expressiveness-tractability design space. The PC framework has since become the standard language of the tractable probabilistic models community, and is adopted throughout this dissertation. Recent work has continued to extend the framework, including connections to normalizing flows (Sidheekh et al., 2023), distillation from deep generative models (Liu et al., 2023), and scaling to larger architectures (Liu et al., 2023).

2.4.2 Formal Definition

A probabilistic circuit over variables $\mathbf{X}$ is a tuple $\mathcal{M} = \langle \mathcal{G}, \theta \rangle$, where $\mathcal{G}$ is a rooted DAG and θ are the parameters associated with its nodes. Each node N in $\mathcal{G}$ is associated with a *scope* $\mathbf{sc}(N) \subseteq \mathbf{X}$, defined as the set of variables appearing in the sub-circuit rooted at N. The circuit consists of three types of nodes:

Leaf nodes encode a tractable distribution over their scope $\mathbf{sc}(N) \subseteq \mathbf{X}$:

$$N(\mathbf{x}) = P_N(\mathbf{x}_{\mathbf{sc}(N)}).$$

Sum nodes compute a convex combination of their children's distributions, acting as mixture components:

$$N(\mathbf{x}_{\mathbf{sc}(N)}) = \sum_{N_i \in \mathrm{ch}(N)} w_i \, N_i(\mathbf{x}_{\mathbf{sc}(N_i)}), \quad w_i \geq 0, \ \textstyle\sum_i w_i = 1.$$

Product nodes compute the product of their children's distributions, encoding a factorization:

$$N(\mathbf{x}_{\mathbf{sc}(N)}) = \prod_{N_i \in \mathrm{ch}(N)} N_i(\mathbf{x}_{\mathbf{sc}(N_i)}).$$

The distribution encoded by $\mathcal{M}$ is the function computed at the root node:

$$P(\mathbf{x}; \mathcal{M}) = N_{\mathrm{root}}(\mathbf{x}).$$

2.4.3 Structural Properties and Tractability

Not all computational graphs of the form described above yield tractable inference. Tractability of *EVI*, *MAR*, and *CON* queries requires imposing structural constraints on $\mathcal{G}$. We now define these constraints formally.

Definition 2.4.1 (Smoothness)**.** A sum node N is *smooth* if all of its children are defined over the same scope: $\mathbf{sc}(N_i) = \mathbf{sc}(N_j)$ for all $N_i, N_j \in \mathrm{ch}(N)$. A PC is smooth if all of its sum nodes are smooth.

Definition 2.4.2 (Decomposability)**.** A product node N is *decomposable* if its children are defined over pairwise disjoint scopes: $\mathbf{sc}(N_i) \cap \mathbf{sc}(N_j) = \emptyset$ for all $N_i \neq N_j \in \mathrm{ch}(N)$. A PC is decomposable if all of its product nodes are decomposable.

Together, smoothness and decomposability guarantee that *EVI*, *MAR*, and *CON* can all be computed in time linear in the size of the circuit, i.e., $O(|\mathcal{M}|)$. Smoothness ensures that marginalization over a sum node's children can be performed efficiently, while decomposability ensures that the scopes of a product node's children partition its own scope, enabling factored computation. This establishes the central tractability-expressiveness trade-off within PCs: structural constraints reduce the set of representable distributions, but in exchange guarantee exact and efficient inference regardless of the complexity of the modeled distribution.

A further structural property, *determinism*, enables tractable computation of more complex queries.

Definition 2.4.3 (Determinism)**.** A sum node N is *deterministic* if for any assignment $\mathbf{x}$, at most one child $N_i \in \mathrm{ch}(N)$ has $N_i(\mathbf{x}) > 0$. A PC is deterministic if all of its sum nodes are deterministic.

Determinism, combined with decomposability, enables tractable MAP inference and supports a broader class of circuit operations (Vergari et al., 2021). PSDDs and cutset networks are examples of deterministic and decomposable PCs. While determinism is exploited in some of the works in this dissertation (notably in the context of cutset networks in Chapter 6), the primary focus is on smooth and decomposable PCs, for which *EVI*, *MAR*, and *CON* are tractable.

2.4.4 Learning Probabilistic Circuits

For completeness, we provide a brief overview of the parameter and structure learning tasks associated with PCs. For a more detailed discussion on this matter, including design extensions that go beyond the three types of nodes mentioned above, refer to the survey on this topic by Sidheekh and Natarajan (2024).

A PC $\mathcal{M} = \langle \mathcal{G}, \theta \rangle$ is characterized by both its structure $\mathcal{G}$ and its parameters θ. Learning a PC from data therefore involves two interrelated problems: *structure learning*, which determines the topology of the computational graph, and *parameter learning*, which estimates the weights of sum nodes and the parameters of leaf distributions given a fixed structure.

Parameter learning. Given a fixed structure $\mathcal{G}$, the parameters θ are typically estimated by maximizing the log-likelihood of training data $\mathcal{D}$:

$$\mathcal{L}(\mathcal{M}, \mathcal{D}) = \sum_{\mathbf{x} \in \mathcal{D}} \log P_{\mathcal{M}}(\mathbf{x}). \tag{2.1}$$

For smooth and decomposable PCs, this objective is differentiable with respect to θ, enabling gradient-based optimization. Closed-form updates via expectation-maximization (EM) are also available for certain PC architectures (Poon and Domingos, 2011; Peharz et al., 2017).

Structure learning. Learning the structure of a PC is considerably more challenging, as the space of valid computational graphs is combinatorially large. The LearnSPN algorithm (Gens and Domingos, 2013) uses a recursive partitioning strategy that alternates between splitting variables (creating product nodes) and clustering instances (creating sum nodes). Subsequent algorithms incorporated more principled independence tests (Rooshenas and Lowd, 2014) and discriminative objectives (Rooshenas and Lowd, 2016). More recently, random and fixed architectures such as Random And Tensorized SPNs (RAT-SPNs, Peharz et al. 2020a) and EinsumNetworks (EinsumNets, Peharz et al. 2020) have been proposed, foregoing structure learning in favor of expressive randomly initialized architectures trained end-to-end via gradient descent. These deep PC architectures scale to millions of parameters through GPU-accelerated computation and achieve strong empirical performance on density estimation benchmarks. They are the primary PC architectures used in the experiments of this dissertation.

PCs occupy a distinctive position in the landscape of probabilistic models. Like PGMs, they maintain explicit probabilistic semantics and support principled reasoning under uncertainty. Like deep generative models such as variational autoencoders (Kingma and Welling, 2014) and generative adversarial networks (Goodfellow et al., 2014), they are differentiable computational graphs that can be scaled through GPU-accelerated training. Unlike both, they provide guaranteed tractable exact inference for a rich class of queries by construction. Of course, this comes at the cost of a trade-off between expressiveness and tractability as a direct result of these structural constraints limiting the space of valid computational graphs. This combination of tractability, expressiveness, and learnability makes PCs particularly well-suited to safety-critical domains such as healthcare. The works in this dissertation build upon this foundation, extending PCs along the three axes of effectiveness, efficiency, and explainability described in Chapter 1.

2.5 Encoding Domain Knowledge

The idea of incorporating domain knowledge from experts directly into learning systems traces back to some of the earliest work in artificial intelligence. In his seminal 1959 paper, McCarthy (1960) described the *Advice Taker*, a hypothetical program that would accept declarative sentences as advice from a human and use them to improve its behavior. While largely a conceptual proposal, the Advice Taker articulated a vision that would take decades to realize: a learning system whose behavior can be shaped by human expertise expressed in a natural, explicit form rather than purely through data.

A concrete realization of this vision came through the line of work on knowledge-based neural networks. Towell and Shavlik (1994) proposed Knowledge-Based Artificial Neural Networks (KBANNs), which initialize a neural network's topology and weights directly from a set of domain-specific rules expressed as propositional Horn clauses. The approach inserts knowledge as a structural prior into the network before any data-driven training begins, allowing the network to refine its behavior in regions where data is abundant while remaining constrained by expert knowledge elsewhere. This framework demonstrated that symbolic knowledge and learning from data need not be treated as competing paradigms, but could instead be productively combined. Subsequent work extended this idea to support vector machines (Fung et al., 2002), to relational and statistical learning (Odom et al., 2015), and to gradient-boosted models (Kokel et al., 2020a), showing that the integration of domain knowledge is a broadly applicable principle across model families.

Alongside this work on knowledge-based learning, a parallel thread developed within the probabilistic modeling community focused on expressing domain knowledge through *qualitative influence statements*, a formalism due to Wellman (1990a). A qualitative influence statement characterizes the directional effect of one variable on another without requiring a precise quantitative specification. Wellman's framework of *qualitative probabilistic networks* (QPNs) formalized monotonic and synergistic influences as qualitative constraints on a

probabilistic graphical model, enabling a form of principled qualitative reasoning under uncertainty. This was an important insight: domain experts can often confidently state that one variable increases or decreases another without being able to specify exact conditional probability tables.

Altendorf et al. (2005) operationalized this idea for parameter learning in Bayesian networks. They showed that *ceteris paribus* monotonicity constraints, which state that increasing the value of one variable increases the probability of another while holding all other parents fixed, can be incorporated as inequality constraints during maximum likelihood parameter estimation. Their experiments demonstrated that even a small amount of such qualitative knowledge can substantially improve parameter estimation when data is sparse, precisely the setting where purely data-driven approaches struggle most. This work established the practical utility of qualitative influence statements as a form of domain knowledge for probabilistic model learning.

The framework of qualitative influences was subsequently adopted and extended in several directions. Yang and Natarajan (2013a) combined monotonicity and synergy constraints with causal independence assumptions for parameter learning in graphical models, showing that the two types of knowledge are complementary and that their combination yields particularly robust models in clinical domains. Feelders and van der Gaag (2005) extended the treatment of qualitative influences to include context-specific variants, where a monotonic relationship between two variables holds only under particular assignments to a third variable. Kokel et al. (2020a) brought qualitative influence statements into the gradient-boosted learning setting through the KiGB algorithm, which encodes soft monotonicity constraints as a regularization term and applies them to learn decision trees and gradient boosted models in noisy, sparse domains. In the clinical setting, Mathur et al. (2023) demonstrated that monotonicity constraints can improve the learning of cutset networks from small datasets, reinforcing the practical value of this form of domain knowledge.

This dissertation adopts the framework of qualitative influence statements as its primary mechanism for encoding domain knowledge into probabilistic circuits. Specifically, Chapter 6 proposes a unified framework that subsumes monotonic influence statements, synergistic influence statements, context-specific independencies, and several other forms of domain knowledge as equality and inequality constraints on tractable queries of a PC, and shows how these constraints can be incorporated into PC parameter learning via a penalized likelihood objective. Chapter 4 addresses the complementary problem of *extracting* qualitative influence statements from a learned PC, enabling expert validation and transparent communication of what a model has learned. Chapter 8 applies qualitative influence statements in conjunction with causal independence assumptions within a Noisy-Or model for clinical prediction of gestational diabetes, illustrating the practical value of this form of domain knowledge in a safety-critical setting.

2.6 Causal Inference and Structural Causal Models

The probabilistic queries introduced in Section 2.2 - evidence, marginal, and conditional - are defined with respect to a single probability distribution $P(\mathbf{X})$ learned from observed data. This is the natural habitat of probabilistic circuits and, indeed, of most statistical machine learning. Answering these queries well requires a model that faithfully captures the joint distribution of the variables of interest, and the tractable inference enabled by PCs makes them especially suited to this task. However, there is a class of questions that observational distributions are fundamentally incapable of answering, irrespective of how faithfully they are estimated or how much data is available. These are *causal* questions. In a clinical context, knowing that higher BMI is associated with increased risk of gestational diabetes - a *MAR* or *CON* query on $P(\mathbf{X})$ - does not tell us what would happen to a patient's diabetes risk if their BMI were reduced through intervention. The two questions look superficially similar but answer fundamentally different questions: the first concerns a pattern in the observational

data, while the second concerns the behavior of the data-generating mechanism under an external manipulation. Treating the former as evidence for the latter can therefore lead to incorrect estimates of intervention effects, since observational associations may be induced or distorted by confounding, selection bias, or other statistical dependencies that would not persist under intervention.

Pearl (2000, 2019) formalized this distinction through the *Pearl Causal Hierarchy* (PCH), a three-level stratification of the questions one can ask about a system. The first level, $\mathcal{L}_1$, is the level of *association*: queries of the form $P(Y \mid X = x)$, asking what one observes about a variable of interest Y when another variable X takes value x in the data. Models covered in all the chapters in this dissertation except for chapter 7 operate at this level. The second level, $\mathcal{L}_2$, is the level of *intervention*: queries of the form $P(Y \mid do(X = x))$, asking what one would observe if X were set to x by an external action, regardless of its natural causes. The *do*-operator (Pearl, 2009) distinguishes an imposed manipulation from a passive observation and is the formal language of experimental reasoning. The third level, $\mathcal{L}_3$, is the level of *counterfactuals*: queries of the form $P(Y_{X=x} \mid X = x', Y = y')$, asking what Y would have been, had X been x, given that we actually observed $X = x'$ and $Y = y'$. A fundamental result of the PCH is that no amount of data at level $\mathcal{L}_1$ is sufficient to answer a query at $\mathcal{L}_2$ or $\mathcal{L}_3$ without additional causal assumptions about the data-generating process (Pearl, 2019).

The formal language for expressing those causal assumptions is a *Structural Causal Model* (SCM) (Pearl, 2009; Peters et al., 2017). An SCM $\mathfrak{C} := (\mathbf{S}, P_{\mathbf{N}})$ consists of a set of d structural equations

$$X_i := f_i(\text{pa}(X_i), N_i), \quad i = 1, \ldots, d, \tag{2.2}$$

where $\text{pa}(X_i)$ denotes the causal parents of X_i in the induced causal graph $G(\mathfrak{C})$, and $P_{\mathbf{N}}$ is a joint distribution over mutually independent exogenous noise variables $N_1, \ldots, N_d$. Each structural equation encodes the causal mechanism by which X_i is generated from its causes and the associated noise. An intervention $do(X_k = a)$ replaces the structural equation for X_k with

the constant assignment $X_k := a$, severing its dependence on its natural parents and producing a *mutilated* causal graph $\hat{G}$. The resulting interventional distribution $P(\mathbf{X} \mid do(X_k = a))$ factorizes over the mutilated graph, a result known as the *truncated factorization* (Pearl, 2009). Counterfactual queries go a step further: they require reasoning about *what would have happened* in an alternate world where X was set differently, conditioned on what was actually observed in the real world, and require the full SCM rather than just the induced distribution.

Viewed through the lens of the probabilistic queries in Section 2.2, an SCM makes it possible to pose *CON* queries not only on the observational distribution $P(\mathbf{X})$ but also on the interventional distribution $P(\mathbf{X} \mid do(\mathbf{X_k} = \mathbf{x_k}))$ for any intervention on a subset of k variables, $\mathbf{X_k} \subseteq \mathbf{X}$. For instance, an interventional *MAR* query $P(X_i \mid do(\mathbf{X_k} = \mathbf{x_k}))$ asks for the marginal distribution of a variable X_i under an externally imposed manipulation of $\mathbf{X_k}$, which is directly relevant to questions like the effect of a clinical intervention on a target outcome. Answering such queries tractably from data is considerably harder than their observational counterparts: standard PC learning operates at $\mathcal{L}_1$ and has no mechanism for *do*-reasoning. The collaborative explorations in this dissertation address this challenge from two complementary angles. Chapter 7 introduces interventional SPNs (iSPNs), which learn to model interventional distributions directly by conditioning on a mutilated causal graph, placing them firmly at $\mathcal{L}_2$ of the PCH. Chapter 8 takes a different approach, using causal independence assumptions to constrain the parameterization of a Noisy-Or model, exploiting causal structure to make learning feasible when data is scarce, while still operating at $\mathcal{L}_\infty$.

2.7 Supervised Domain Adaptation and Submodular Functions

The preceding sections address what a probabilistic model can represent, encode, and reason tractably about. The tenet of *efficiency*, embodied by Chapter 5, turns to a more operational question: *how can such a model be trained effectively and efficiently when the data available for*

training comes from a distribution that differs from the one the model will serve in practice? Answering this question requires both a formal account of distribution shift between domains and a principled method for identifying, from a large source dataset, the subset of examples most relevant to the target. This section introduces the two technical ideas that underpin Chapter 5: supervised domain adaptation, which formalizes the notion of distribution shift between a source and a target domain, and submodular functions, which provide the discrete optimization framework for efficient and theoretically grounded data subset selection.

Supervised Domain Adaptation. In many real-world settings, labeled data is abundant in a related *source* domain $\tau^s = \{X^s, Y^s\}$ but scarce in the *target* domain of interest $\tau^t = \{X^t, Y^t\}$. A natural strategy is to pretrain a model on the source domain and fine-tune it on the target domain (Girshick et al., 2014; Yosinski et al., 2014; Chu et al., 2016). *Supervised domain adaptation* (SDA) formalizes this setting by assuming that a small number of labeled examples from the target domain are available during training. SDA methods differ in how they characterize the distribution shift between domains. In this dissertation, we focus on *covariate shift* (Shimodaira, 2000), the setting in which the marginal input distributions differ across domains ($P(X^s) \neq P(X^t)$) but the conditional label distribution remains invariant ($P(Y^s \mid X^s) = P(Y^t \mid X^t)$). A broad family of SDA methods addresses covariate shift by learning a shared latent representation that is domain-invariant while remaining discriminative, with different approaches varying in their choice of alignment objective (Motiian et al., 017a,b; Xu et al., 2019; Morsing et al., 2020; Hedegaard et al., 2021). Chapter 5 discusses these methods and their relationship to the approach proposed in this dissertation in detail.

Submodular functions. Submodular functions play a central role in combinatorial optimization and have emerged as a powerful tool for modeling diversity, coverage, and information in machine learning. A set function $f : 2^V \to \mathbb{R}$ defined on a finite ground set V is said to be

submodular if it satisfies the diminishing returns property: for all $A \subseteq B \subseteq V$ and $x \in V \setminus B$,

$$f(A \cup \{x\}) - f(A) \geq f(B \cup \{x\}) - f(B). \tag{2.3}$$

Intuitively, the marginal gain of adding an element decreases as the context grows. Submodular functions are often assumed to be monotone (i.e., $f(A) \leq f(B)$ for $A \subseteq B$) and normalized (i.e., $f(\emptyset) = 0$), though these properties are not strictly required.

A key advantage of submodular functions is that they admit efficient approximation algorithms for optimization. In particular, the problem of maximizing a monotone submodular function under a cardinality constraint,

$$\max_{|S| \leq k} f(S), \tag{2.4}$$

can be approximately solved using a simple greedy algorithm with a $(1 - 1/e)$ optimality guarantee, where $S \subseteq V$. This result makes submodular functions especially attractive for large-scale data subset selection, where exact optimization is computationally infeasible.

In the context of machine learning, submodular functions have been widely used as *data acquisition functions* to select informative and representative subsets of data. Common formulations include facility location functions, coverage functions, and graph-based objectives, which aim to balance representativeness and diversity. These objectives are particularly useful in settings such as active learning, core-set selection, and data summarization, where labeling or processing the entire dataset is expensive.

Submodular Mutual Information. A principled way to quantify the usefulness of a subset relative to a target set is through the *submodular mutual information* (SMI). Given two subsets $A, B \subseteq V$, the SMI induced by a submodular function f is defined as

$$I_f(A; B) = f(A) + f(B) - f(A \cup B). \tag{2.5}$$

This formulation mirrors classical mutual information but replaces entropy with a general submodular function. When f is chosen appropriately (e.g., as a facility location or graph cut function), $I_f(A; B)$ captures the similarity or shared information between subsets A and B.

In data subset selection, SMI provides a natural objective for selecting a subset $S \subseteq V$ that is maximally informative about a target set T, which may represent unlabeled data, a validation set, or a distribution of interest:

$$\max_{|S| \leq k} I_f(S; T). \tag{2.6}$$

This objective encourages the selected subset S to cover the important characteristics of T while avoiding redundancy, owing to the submodular structure of f. Importantly, when f is monotone submodular, the SMI objective is also submodular in S, allowing the use of efficient greedy algorithms with theoretical guarantees. Different choices of f lead to different notions of coverage: facility location functions model representativeness via pairwise similarities, graph cut functions emphasize diversity, and log-determinant functions connect SMI to determinantal point processes and capture higher-order structure. Chapter 5 instantiates this framework for supervised domain adaptation, using gradient-space similarities to define f and the labeled target data as the query set T.

CHAPTER 3

EXPLAINING PROBABILISTIC CIRCUITS

The preceding chapter introduced the background machinery of probabilistic circuits and established the kinds of queries they can answer tractably. A natural question follows immediately: once a PC has been learned from data, what has it actually learned? A model that can answer probabilistic queries accurately is useful, but in safety-critical domains such as clinical medicine, accuracy alone is insufficient. Practitioners and domain experts need to understand the structure captured by the model, to scrutinize whether that structure is consistent with prior knowledge, and ultimately to decide whether to trust its outputs. This chapter addresses that challenge by introducing a method for extracting human-interpretable representations from learned sum-product networks, and in doing so represents the first contribution of this dissertation to the tenet of explainability.

The central contribution of this chapter is EXSPN, an algorithm for extracting *CSI-trees* from a learned SPN. A CSI-tree is a compact, decision-tree-style representation of the context-specific independence (CSI) relations encoded in a learned network. Unlike global conditional independence statements, CSI relations hold only within specific variable contexts and therefore carry richer and more granular information about the model's internal structure. The EXSPN algorithm traverses the sum and product nodes of a trained SPN to identify and organize these relations into a readable form, making explicit what the model has implicitly learned about the conditional relationships between variables. This form of explanation is *structural*: it surfaces the independence geometry of the model itself, rather than describing how the model's outputs vary as inputs change.

The nuMoM2b gestational diabetes dataset, introduced as a running example in Chapter 2, serves as the primary real-world evaluation in this chapter. The extracted CSI-trees were reviewed and validated by Dr. David Haas, a co-investigator on the nuMoM2b study, providing external grounding for the interpretability claims. This collaboration establishes a pattern

that recurs in subsequent chapters: domain expert validation is not incidental but is treated as a first-class component of the evaluation.

This chapter also lays essential groundwork for the dissertation's later arc. The CSI relations that $\mathcal{EXSPN}$ extracts are precisely the knowledge structures that Chapter 4 will complement from a different angle, and, more importantly, that Chapter 6 will formalize as differentiable constraints in a unified knowledge-intensive learning framework. The vocabulary of interpretable knowledge structures developed across Chapters 3 and 4 thus finds its full payoff in Chapter 6, where those structures are used not to explain what a model has learned, but to guide what it learns. The reader may find it useful to hold this forward connection in mind while reading the present chapter.

3.1 Introduction

In this work we tackle the challenging task of generating human-interpretable representations from PCs that are faithful to the original model. In particular, we pose the following question – *can SPNs with their multiple layers be explained using existing tools inside probabilistic modeling?* To achieve this, we move beyond the traditional notions of conditional independencies that can be read off an SPN and instead focus on context-specific independencies (CSI, (Boutilier et al., 1996)). CSIs provide a more in-depth look at the relationships between two variables when affecting the third variable. For instance, in a gestational diabetes prediction task ((Karanam et al., 2021)), one could state that gestational diabetes and education are conditionally independent given the age. This allows the care provider/physician to develop a good interventional treatment plan. Recent work on developing interventions given a learned SPN ((Zečević et al., 2021)) demonstrates the potential of such TDPMs and our work goes in the same direction by identifying CSIs that could potentially aid the expert in identifying appropriate interventions.

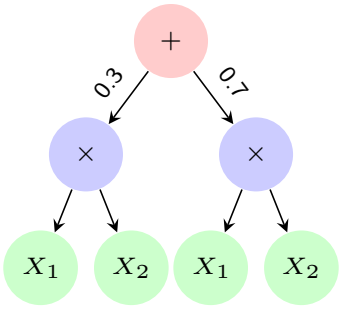


Figure 3.1: SPN that represents the joint distribution $P(X_1, X_2)$. Sum nodes are inscribed with "+", product nodes with "×" and leaf nodes with corresponding variable names(*best viewed in color).*

Specifically, we define the notion of a CSI-tree that is used as a *visual tool to explain SPNs*. We present an algorithm ($\mathcal{EXSPN}$) that grows a CSI-tree iteratively. We show clearly that the constructed CSI-tree can recover the full SPN structure. Once a tree is constructed, we then approximate the CSIs by learning a supervised model to fit the CSIs. The resulting feature importances can then be used to further approximate the tree. Our evaluations against association rule mining clearly demonstrate that the recovered CSI-trees indeed are shorter and more interpretable.

We make the following key contributions: (1) We develop CSI-trees that focus on the explanation. These trees are built on the earlier successes inside graphical models and we use them in the context of TDPMs. (2) We develop an iterative procedure that constructs a CSI-tree given an SPN. The resulting tree is a complete representation of the original SPN. We present an approximation heuristic that compresses these trees further to enhance the explainability of the model. One key aspect of $\mathcal{EXSPN}$ is that it is **independent** of the underlying SPN learning algorithm. Any SPN that is complete and consistent (as defined in the next section) can be used as input for $\mathcal{EXSPN}$. [1] (3) We perform extensive experiments on synthetic, multiple standard data sets and a real clinical data set. Our

[1]Code is available at `https://github.com/saurabhmathur96/ExSPN-SPFlow`

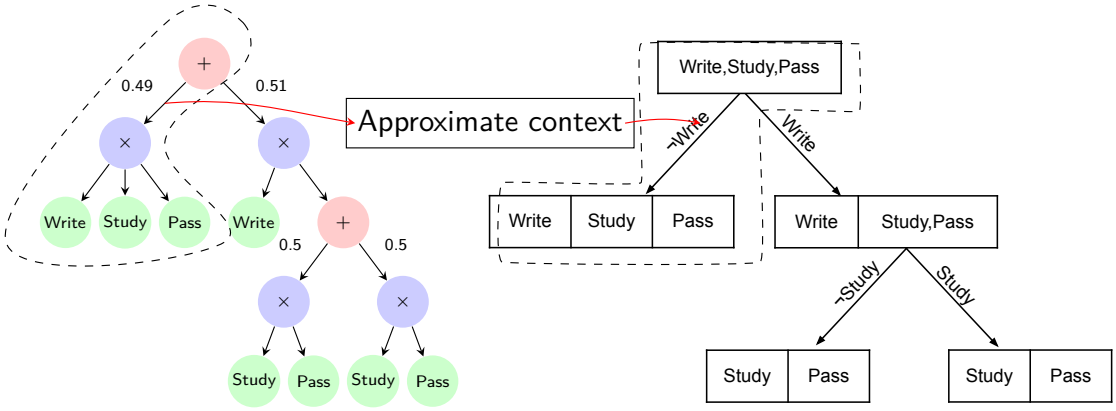


Figure 3.2: A sum-product network (left) and its corresponding CSI-Tree (right)

evaluation demonstrates that the final model induces smaller rules/models compared to the original SPN.

3.2 Background

Before we outline our procedure for explaining SPNs, we briefly explain the notion of Context-specific independence(CSI) ((Boutilier et al., 1996)). CSI is a generalization of the concept of statistical independence of random variables. CSI-relations have been studied extensively over the last three decades ((Nyman et al., 2014; Tikka et al., 2019; Nyman et al., 2016)). They can be used to speed-up probabilistic inference, improve structure learning and explain graphical models that are learnt from data. We propose **a novel algorithm to extract these CSI-relations from SPNs** and empirically demonstrate that the extracted CSI-relations are interpretable. CSIs can be directly extracted from the data by computing conditional probabilities ((Shen et al., 2020)) with context using parameterized functions such as neural networks((Kingma and Welling, 2013)). While the literature on rule learning is vast ((Fürnkranz and Kliegr, 2015)), we note that our objective is *explaining SPNs* and not *rule learning*. Additionally, we note that the relationship between SPNs and BNs ((Zhao et al., 2015a)), and that between SPNs and multi-layer perceptrons

((Vergari et al., 2019)) is well established. However, unlike our work, these methods do not explicitly attempt to explain SPNs through a compact representation of CSIs. The algorithm to convert SPNs to BNs presented by Zhao et. al.((Zhao et al., 2015a)) introduces unobserved hidden variables into the BN. It generates a BN with a directed bipartite structure with a layer of hidden variables pointing to a layer of observable variables. In this case, the BN associates each sum node in the SPN to a hidden variable in BN. We argue that the introduction of these hidden variables renders the resulting BN uninterpretable.

To construct meaningful explanations, we define a compact and interpretable representation of CSIs called **CSI-tree**.

Definition 3.2.1. (CSI-tree): A CSI-tree τ is a 4-tuple $(G, \psi_{csi}, \chi, \zeta)$ where G is a tree with a set of variables $V \subseteq \mathbf{X}$, *scope function* ψ_{csi}, *partition function* χ and edge labels ζ.

Definition 3.2.2. (Partition function): A partition function $\chi_\psi : N \rightarrow 2^{|\psi(N)|}$ is a mapping from a node to a set C of disjoint sets $P_i \subseteq \psi(N)$ under a given scope function ψ such that $\cup_{i \in |C|} P_i = \psi(N)$.

Each node, N, of a CSI-tree has a scope $\psi_{csi}(N) \subseteq \mathbf{X}$. The partition function divides the variables within the scope of each node into disjoint subsets such that the union of these subsets is $\psi_{csi}(N)$. The edges are labeled with a conjunction ζ over a subset of $\mathbf{X}$. An example of an edge label is $(X_i \geq 0.5 \wedge X_j \leq 1)$. Essentially, the edge label narrows the scope of the context from parent to child node.

Example CSI-tree: The right side of the Figure 3.2 shows a CSI-tree defined over the binary variables $\langle Write, Study, Pass \rangle$. The set of variables inscribed within each node represent the scope of that node. For example, the scope of the root node is $\langle Write, Study, Pass \rangle$. The edge label *confines the context* by conditioning on a set of variables (on a singleton set in this example). The variables in the scope of a node that are conditionally independent when conditioned on the context are separated into subsets (denoted by a vertical bar). It is the

output of the partition function for that node. For example, the right child of the root node specifies a partition $\langle\langle Write\rangle, \langle Study, Pass\rangle\rangle$ of the scope of that node.

The left child of the root node induces a CSI-relation $Write \perp\!\!\!\perp Study \perp\!\!\!\perp Pass|\neg Write$. Similarly, the right child of the root node induces two CSI-relations $Write \perp\!\!\!\perp Study|Write$ and $Write \perp\!\!\!\perp Pass|Write$. To summarize, the subsets of variables within the scope of a node, as defined by the partition function, are independent of each other when conditioned on the proposition specified in the label of the edge connecting that node to its parent. Note that *reading this CSI-tree is significantly easier than a SPN* on the left, due to the internal nodes entirely comprising of observed variables, and thus allows for a more explainable and interpretable representation. Now, we formally define our goal:

Given: SPN $S = (G, \psi, w, \theta)$, data $\mathcal{D}$

To Do: Extract CSI-tree $\tau = (G_{csi}, \psi_{csi}, \chi, \zeta)$

The left side of Figure 3.2 shows an SPN learnt over the variables $Write, Study, Pass$. The root node N_0 is a sum node, implying that $Write, Study, Pass$ are not independent of each other in the context of the entire dataset. Now, consider the left child of the root node N_1, which is a product node with three leaf nodes as its children. It implies that $Write \perp\!\!\!\perp Study \perp\!\!\!\perp Pass|\phi(N_1)$. In other words, the $Write, Study, Pass$ are **independent of each other in the context of** $\phi(N_1)$. While the context of $\phi(N_1)$ accurately explains the conditional independence induced by the product node, it requires $2^{|\psi(N_1)|}$ parameters to fully parameterize the context which renders the CSI specified under $\phi(N_1)$ uninterpretable. Therefore, we need to approximate this context in order to ensure interpretability of these CSI-relations. Additionally, while the instantiation of a subset of $\psi(N_1)$ that is consistent with $\phi(N_1)$ can be used to define the context for a particular CSI, this method would be highly sensitive to noise.

To address these issues, we propose to **first learn a discriminative model** f in the supervised learning paradigm to predict if a data point belongs to $\phi(N_1)$ and use the notion of **feature importance to approximate the context**. We identify a set $I \subseteq \psi(N_1)$

Algorithm 1: $\mathcal{EXSPN}$

input : $\mathcal{D}$, $\mathbf{X}$, $S = (G = (\mathbf{N}, \mathbf{E}), \psi, w, \theta)$, λ
output : CSI-tree $\tau = (G_{csi}, \psi_{csi}, \chi, \zeta)$

Convert S to a normal-spn S_{normal}
Infer ϕ using alg. 2
Initialize $G_{csi} = G_{normal}, \psi_{csi} = \psi_{normal}$,
$\tau = (G_{csi}, \psi_{csi}, \chi, \zeta)$
st = [G_{csi}.root]
Add $\psi(G_{csi}.root)$ to $\chi(G_{csi}.root)$
while *st is not empty* **do**
    $N_{current} = st.pop()$
    if $N_{current}$ *is not the root node* **then**
        if $N_{current}$ *is a leaf node* **then**
            Add $\psi(N_{current})$ to $\chi(pa(N_{current}))$
        end
        if $N_{current}$ *is a sum node* **then**
            Add $\psi(N_{current})$ to $\chi(pa(N_{current}))$
            Connect $pa(N_{current})$ and each $ch(N_{current})$
            Replicate sub-SPN rooted at $ch(N_{current})$ for each $|pa(ch(N_{current}))|$ to ensure G_{csi} has a tree structure
            Delete $N_{current}$ from G_{csi}
        end
    end
    Add $ch(N_{current})$ to st
end
$\tau = ComputeLabels(\tau, \mathcal{D}, \phi, \lambda)$
return τ

of the most important features as determined by the feature importance scores for $\psi(N_1)$ w.r.t f and approximate the context to a proposition of the form $\wedge_{i \in I}$. For this example, we choose f to be a *decision tree* and the *mean decrease in impurity* as the measure of feature importance and obtain $(\neg Write)$ as the approximated context. So, the original CSI $Write \perp\!\!\!\perp Study \perp\!\!\!\perp Pass | \phi(N_1)$ is approximated to $Write \perp\!\!\!\perp Study \perp\!\!\!\perp Pass | \neg Write$.

3.3 $\mathcal{EXSPN}$ Algorithm

For simplicity, we present our approach with binary variables. However, in our experiments, we demonstrate that our method **can be applied to continuous and multi-class discrete**

variables as well. We now outline our approach to infer CSI-trees from SPNs. The steps involved in converting an SPN into a CSI-tree are: (1) Convert S into a normal-SPN S_{normal}. (2) Infer the instance function ϕ. (3) Create a CSI-tree $\tau_{unlabeled}$ with no edge labels using ϕ. (4) Compute edge labels for $\tau_{unlabeled}$, and create CSI-tree τ. (5) Optionally, compress τ into $\tau_{compressed}$ by pruning τ.

Algorithm 1 presents $\mathcal{EXSPN}$: **Ex**plaining **S**um-**P**roduct **N**etworks. It infers a CSI-tree τ, given an SPN S, data $\mathcal{D}$, set of random variables $\mathbf{X}$ and feature importance score threshold λ.

(**Step 1:**) Convert S to a normal-spn $S_{normal} = (G_{normal}, \psi_{normal}, w_{normal}, \theta_{normal})$

[**line 1**] using the conversion scheme proposed by Zhao et. al. ((Zhao et al., 2015a)). Any arbitrary SPN S can be converted into a normal SPN S_{normal} that represents the same joint probability over variables and $|S_{normal}| = \mathcal{O}(|S|^2)$.

(**Step 2:**) It then infers the instance function for S_{normal}[**line 2**] using algorithm 2. This algorithm is similar to the algorithm proposed in (Poon and Domingos, 2011) for approximating most probable explanation(MPE) inference in arbitrary SPNs. For each instance $\mathcal{D}_j \in \mathcal{D}$ it first performs an upward pass from leaf nodes to the root node and computes $S_i^{max}(\mathcal{D}_j)$ for each node N_i as follows:

- if N_i is a sum node, then, $S_i^{max}(\mathcal{D}_j) = \max_{k \in ch(N_i)} w_{ik}.S_j^{max}(\mathcal{D}_j)$
- otherwise $S_i^{max}(\mathcal{D}_j) = S_i(\mathcal{D}_j)$

Then the algorithm backtracks from the root to the leaves, appending $\mathcal{D}_j$ to the instance function associated with each child of a sum node that led to $S_i^{max}(\mathcal{D}_j)$ and to the instance function associated with all product nodes along the path from the root to a leaf node.

(**Step 3:**) Next, it performs a depth-first search(DFS) on the computational graph G_{normal} associated with S_{normal} and constructs the tree-structured graph G_{csi} associated with τ [**lines 4-19**]. G_{csi} is initialized with G and the scope of root of G_{csi}, $\psi(G_{csi}.root)$ is added to the

Algorithm 2: Infer Instance Function

input : $\mathcal{D}$, $\mathbf{X}$, normal SPN S
output : Instance function ϕ

for *Each instance* $\mathcal{D}_j \in \mathcal{D}$ **do**
    Perform upward pass for $\mathcal{D}$
    Compute $S_N^{max}(\mathcal{D}_j)$ for each node N
    Perform a downward pass
    Append $\mathcal{D}_j$ to the $\phi(N_{ch})$ of a child N_{ch} of a sum node that led to $S^{max}(\mathcal{D}_j)$
    Append $\mathcal{D}_j$ to $\phi(N_{product})$ for all product nodes $N_{product}$ along the path from the root to a leaf node
end
return ϕ

value of the partition function associated with τ[**line 5**]. $\mathcal{EXSPN}$ maintains a stack st that is initialized with the root of G_{csi}[**line 4**]. The current node selected in DFS, $N_{current}$, is popped from the stack st[**line 7**]. It then considers three cases: 1. $N_{current}$ is a leaf node 2. $N_{current}$ is a sum node 3. $N_{current}$ is a product node[**lines 9-16**]. If $N_{current}$ is a leaf node, its scope $\psi(N_{current})$ is added to the partition function of its parent node $\chi(pa(N_{current}))$[**lines 9-10**]. If $N_{current}$ is a sum node, its scope $\psi(N_{current})$ is added to the partition function of its parent node $\chi(pa(N_{current}))$, edges are added to G_{csi} to connect the parent of $N_{current}$, $pa(N_{current})$, to each child of $N_{current}$, and deleting $N_{current}$ and all edges e in G_{csi} of the form $e(\alpha, N_{current})$ or $e(N_{current}, \alpha)$[**line 12-16**]. If $N_{current}$ is a product node, it is ignored. It then continues DFS over G_{csi} by adding all the children of $N_{current}$ to st[**line 19**]. Note that the CSI-tree τ is unlabeled.

(**Step 4:**) Algorithm 3 in the Appendix presents the procedure to train a model f, compute a set of important features I, and finally compute the edge labels ζ. For each edge $e(N_{from}, N_{to})$, it first computes class labels y to indicate if an instance $\mathcal{D}_j \in \phi(N_{from})$ belongs to $\phi(N_{to})$. Then it computes a set of important features I for f using a suitable feature importance computation technique based on the choice of f and a threshold for feature importance score λ. The edge label for e is the conjunction of the features in I.

Algorithm 3: ComputeLabels

input : $\tau = (G, \psi, \chi, NULL), \mathcal{D}, \phi, \lambda$
output : Labeled CSI-tree $\tau = (G, \psi, \chi, \zeta)$

for *Each edge* $e(N_{from}, N_{to})$ *in* G **do**
    Compute class labels y
    $f = TrainModel(\mathcal{D}, \phi(N_{from}), \mathbf{y})$
    Compute a set of important features I for f
    $\zeta(e) = \wedge_{i \in I} i$ thresholded by λ
end
return $\tau = (G, \psi, \chi, \zeta)$

(**Step 5:**) The labeled CSI-tree τ created in the previous step might be too large in some cases and the CSIs induced by τ at a product node N may be supported by a small number of examples given by $|\phi(N)|$. To avoid these two issues, we propose deleting the sub-tree induced by a product node N for which $|\phi(N)| < min_{instances}$. This heuristic significantly reduces the size of the CSI-tree while retaining CSIs induced by product nodes closer to the root node of the SPN, as demonstrated in our experiments.

Computational Complexity: The computational complexity of $\mathcal{EXSPN}$ algorithm depends on the complexity of each of its constituent 5 steps. The size ($\#nodes + \#edges$) of the normal-SPN S' obtained from the original SPN S has a space-complexity of $O(|S|^2)$. This conversion can be done is time linear in the size of the normal-SPN ((Zhao et al., 2015a)). The generation of instance function for the CSI-tree involves MAX inference for each of $\mathcal{D}$ data points which can be performed in $O(|\mathcal{D}||S'|)$ ((Poon and Domingos, 2011)). The generation of unlabeled CSI-tree has a time-complexity of $O(|S'|)$. Generating edge labels involves training a discriminative model which can be performed in $O(|\mathcal{D}||\theta_D|)$ for a universal approximator such as neural network with parameters θ_D. Here $|N_{sum}|$ is the number of sum nodes in the network and $|\mathbf{X}|$ is the number of variables. Finally the compression can be performed in time linear in the size of the network. This gives us an *overall time complexity of* $O(max(|\theta|, |X|)|S|^2|\mathcal{D}|)$.

Algorithm 4: RetrieveSPN

```
input  : CSI-tree τ = (G_csi, ψ_csi, χ, ζ)
output : G_normal, ψ_normal
initialize: G_normal = G_csi, ψ_normal = ψ_csi
st = [G_normal.root]
while st is not empty do
    N_current = st.pop()
    if N_current has not been visited and is not the root node then
        if |χ(N_current)| = |ψ_csi(N_current)| then
            Replace N_current with N_p
            Append |χ(N_current)| leaf nodes
        if |χ(N_current)| ≠ |ψ_csi(N_current)| then
            for k in χ(N_current) do
                if |k| = 1 then
                    Append N_l s.t. ψ(N_l) = k
                if |k| ≠ 1 then
                    Append N_s s.t. ψ(N_s) = k for c in ch(N_current) do
                        Append N_p to N_s if ψ_csi(c) = k
                    end
            end
    Add ch(N_current) to st
end
return G_normal, ψ_normal
```

Properties of CSI-tree: $\tau = (G_{csi}, \psi_{csi}, \chi, \zeta)$, corresponding to an SPN $S, \psi, w, \theta)$ obtained through $\mathcal{EXSPN}$ has the following properties: (1) Num nodes in G_{csi} = num product nodes in G_{normal} + 1. (2) The context induced by a product node N_p in S requires $2^{|\psi(N_p)|}$ parameters to be sufficiently expressed, while the approximate context from $\mathcal{EXSPN}$ has $< |\psi(N_p)|$ parameters. Additionally, the structure of a tree-structured normal-SPN can be retrieved from its corresponding CSI-tree in time linear in the size of the SPN.

Theorem 3.3.1. *The CSI-tree τ, inferred from an SPN, S_{normal}, using $\mathcal{EXSPN}$ can infer G'_{normal}, and ψ'_{normal} which encodes the same CSIs as S_{normal}.*

Proof. Algorithm 4 that retrieves G_{normal} and ψ_{normal} associated with S_{normal}, given τ provides the proof. It performs DFS over G_{normal} and identifies two main cases of $N_{current}$: **1.** A node where the length of the partition function $\chi(N_{current})$ is equal to length of the scope function

$N_{current}$[**line 6**] **2.** A node where that's not the case[**line 10**]. In the first case, it replaces $N_{current}$ with a product node N_p such that $\psi(N_p) = \psi_{csi}(N_{current})$ and adds $|\chi(N_{current})|$ number of leaf nodes. In the second case, it iterates over the subsets in $\chi(N_{current})$, adds a leaf node if that subset k is a singleton set and adds an intermediate sum node N_s for all other children associated with a subset k[**lines 10-17**]. Then, it returns computational graph G and scope function ψ. Although this algorithm retrieves only G_{normal} and ψ_{normal} which define the structure of the SPN, the parameters of the SPN w_{normal} and θ can also be retrieved by modifying $\mathcal{EXSPN}$ to produce a CSI-tree that stores these parameters in the leaf nodes of the CSI-tree alongside the scope ψ_{csi} and the partition χ. This modification is trivial and does not improve the interpretability. Hence, we do not consider it. □

3.4 Experimental Evaluation

We explicitly answer the following questions: (**Q1: Correctness**) Does $\mathcal{EXSPN}$ recover all the CSIs encoded in an SPN? (**Q2: Compression**) Can the CSIs be compressed further? (**Q3. Baseline**) How do the CSIs extracted using $\mathcal{EXSPN}$ compare with a strong rule learner? (**Q4. Real data**) Does $\mathcal{EXSPN}$ extract reasonable CSIs in a real clinical study? **System:** We implemented $\mathcal{EXSPN}$ using SPFlow library ((Molina et al., 2019)). We assume that the instance function is computed during the training process. For experiments where the instance function is computed separately after training, see Table 3.2. Since Decision Trees can be represented as a set of decision rules, we used the Classification and Regression Trees (CART, (Breiman et al., 1984)) algorithm as the explainable function approximator. We used scikit-learn's DecisionTreeClassifier ((Pedregosa et al., 2011)) to implement CART. The hyperparameter configuration of these algorithms is shown in Table 3.5.

Baseline: To evaluate the CSIs extracted by $\mathcal{EXSPN}$, we compared them with the association rules mined using the Apriori algorithm ((Agrawal et al., 1994)). We used the Mlxtend library ((Raschka, 2018)) to implement this baseline. Since the Apriori algorithm

Table 3.1: Summary statistics for the CSI rules extracted from SPNs by $\mathcal{EXSPN}$ and the association rules by Apriori algorithm. NP - # Product Nodes, NR - # Rules, MA - Mean Antecedent length, MC - the Mean Consequent length, and CR - Compression Ratio.

	SPN	All CSIs			Reduced CSIs				Association Rules		
Dataset	NP	NR	MA	MC	NR	MA	MC	CR	NR	MA	MC
Synthetic	7	7	2.29	2.57	**3**	1.33	2.67	2.33	12	1.25	1.25
Mushroom	39	39	5.90	8.54	**14**	4.79	7.93	2.79	10704	2.87	2.43
Plants	342	342	9.60	9.61	**23**	6.22	7.09	14.87	1043	1.72	1.40
NLTCS	74	74	9.84	3.32	**19**	6.32	4.05	3.89	165	1.96	1.28
MSNBC	8	**8**	4.12	5.75	**8**	4.12	5.75	1.00	16	2.38	1.00
Abalone	194	194	11.31	7.00	**4**	4.25	2.00	48.50	730	2.15	1.71
Adult	263	263	14.49	4.02	**19**	7.37	2.74	13.84	917	2.24	1.72
Wine quality	236	236	12.45	6.76	**5**	3.60	2.60	47.20	337	1.99	1.56
Car	18	18	5.22	2.50	**14**	5.21	2.64	1.29	19	1.58	1.00
Yeast	181	181	16.20	3.26	**10**	7.90	2.30	18.10	50	1.52	1.52
nuMoM2b	104	104	10.60	2.33	**31**	6.55	2.19	3.35	21	1.29	1.14

requires binary features, we discretized the continuous variables in the datasets into 5 categories and one-hot encoded the categorical variables.

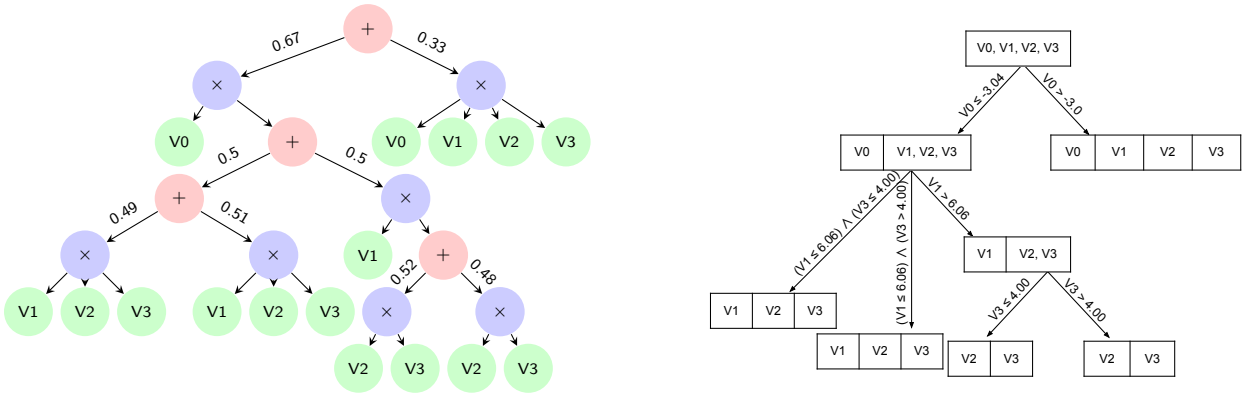


Figure 3.3: The sum-product network (left) trained on the synthetic data set and its corresponding CSI-tree extracted by $\mathcal{EXSPN}$ (right). Each edge in the CSI-tree corresponds to an edge from a sum node to a product node in the SPN. All other edges are collapsed. Clearly, the CSI-tree encodes all the CSIs represented in the sum-product network.

Datasets: We evaluated $\mathcal{EXSPN}$ on 11 datasets – one synthetic, 9 benchmark, and one **real clinical study**. We generated the synthetic dataset by sampling 10,000 instances each from 3 multivariate Gaussians. We used 8 datasets from the UCI repository and

Table 3.2: Summary statistics for the case where the instance function is inferred after training. The columns are the same as Table 3.1.

	SPN	All CSIs			Reduced CSIs			
Dataset	NP	NR	MA	MC	NR	MA	MC	CR
Synthetic	7	7	2.29	2.57	3	1.33	2.67	2.33
Mushroom	39	39	5.90	8.54	13	5.08	8.38	3.00
Plants	342	342	8.33	9.61	32	4.62	6.72	10.69
NLTCS	74	74	9.74	3.32	19	6.32	4.05	3.89
MSNBC	8	8	4.12	5.75	8	4.12	5.75	1.00
Abalone	194	157	9.61	6.85	8	5.75	2.00	19.63
Adult	263	244	11.27	3.89	12	6.17	3.08	20.33
Wine	236	235	9.18	6.78	6	3.50	2.50	39.17
Car	18	18	5.22	2.50	14	5.21	2.64	1.29
Yeast	181	181	13.06	3.26	10	6.60	2.30	18.10
nuMoM2b	104	98	9.64	2.29	35	6.97	2.17	2.80

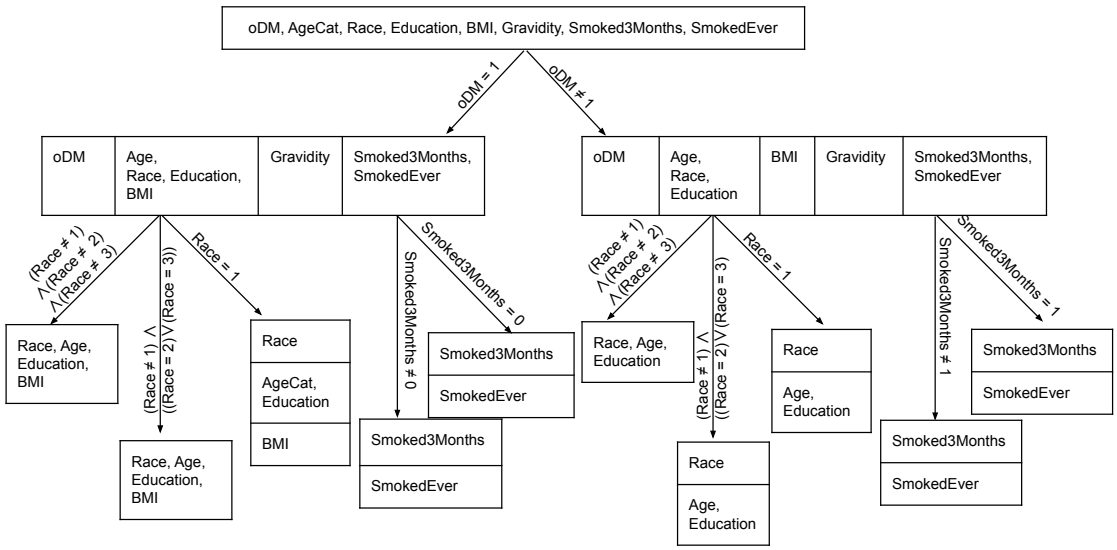


Figure 3.4: First two levels of the CSI-tree for the nuMoM2b dataset. Here, *oDM* is a boolean variable that represents whether or not the person has Gestational Diabetes. *Race* is a categorical variable having 8 categories namely, Non-Hispanic White (1), Non-Hispanic Black (2), Hispanic (3), American Indian (4), Asian (5), Native Hawaiian (6), Other (7), and Multiracial (8). *Smoked3Months* and *SmokedEver* are boolean variables representing tobacco consumption.

the National Long Term Care Survey (NLTCS, (Lowd and Davis, 2010)) data from CMU StatLib (`http://lib.stat.cmu.edu/datasets/`). The 8 datasets from the UCI machine learning repository were Mushroom, Plants, MSNBC, Abalone, Adult, Wine quality, Car,

Table 3.3: The Number of CSIs extracted by ExSPN from an SPN fit on each dataset (Total), The number of CSIs where at least 80% of the datapoints in the context matched with the CSIs from the Bayesian Network(BN) (Correct) and the Ratio of correct CSIs (Ratio).

Dataset	Total	Correct	Ratio
Earthquake	12	10	0.83
Cancer	12	10	0.83
Asia	25	22	0.88

and Yeast. In MSNBC and Plants, we used the rows which had at least two items present. For Wine quality, we concatenated the red wine and white wine tables. In addition, we used **Nulliparous Pregnancy Outcomes Study: Monitoring Mothers-to-Be** (nuMoM2b, (Haas et al., 2015)) study data. Our subset has 8 variables - *oDM, Age, Race, Education, BMI, Gravidity, Smoked3Months,and SmokedEver*. Of these, oDM is the boolean representing Gestational Diabetes (0/1). We split each dataset into a train set having 75% of the examples and a test set having the remaining 25%. To ensure a balanced split, we stratified the splits on the target variable for the classification datasets. Table 3.4, in the appendix. summarizes the datasets.

Metrics: We defined the following metrics on the CSIs - min_precision (mp), min_recall (mr), and n_instances (ni). mp and mr of a CSI are the minimum values of precision and recall respectively for each of the decision rules that approximate the context. ni of a CSI is the number of training instances in that context. We used thresholds on these metrics to obtain a reduced set of CSIs (0.7 for mp and mr, and $5 \times$ `min_instances_slice` for ni). We quantified this reduction as the *Compression Ratio (CR)*, which is the fraction of the total number of CSI rules in reduced set to the total number of rules. To compare association rules, we use mean antecedent length (mean $|A|$) and mean consequent length (mean $|C|$).

Table 3.4: Dataset details.

Dataset	Type	$\|\mathbf{X}\|$	Train	Test
Synthetic	Continuous	4	22,500	7,500
Mushroom	Categorical	23	4,233	1,411
Plants	Binary	70	17,411	5,804
NLTCS	Binary	16	16,180	5,394
MSNBC	Binary	17	291,325	97,109
Abalone	Mixed	9	3,132	1,045
Adult	Mixed	15	33,916	11,306
Wine quality	Continuous	12	4,872	1,625
Car	Categorical	7	1,296	432
Yeast	Mixed	9	1,113	371
nuMoM2b	Categorical	8	8,832	388

Table 3.5: Hyperparameters(HP)

Component	HP	Value
SPN	rows	GMM
	mis	1% of $\|Train\|$
		5% of $\|Train\|$
		1% of $\|Train\|$
DT	max_depth	2
	mid	0.1
	class_weight	balanced
$\mathcal{EXSPN}$	min_precision	0.7
	min_recall	0.7
	n_instances	$5 \times mis$
		mis (Synthetic)

3.5 Results

(Q1: Correctness) Table 3.1 summarizes the SPNs, the full set of CSI rules extracted by $\mathcal{EXSPN}$, and the reduced set of CSI rules. For each dataset in the table, the number of CSIs extracted by $\mathcal{EXSPN}$(NR) **is exactly equal to** the number of product nodes of the SPN(NP). Figure 3.3 shows the SPN learnt from the synthetic data and the CSI-tree extracted from that SPN using $\mathcal{EXSPN}$. Clearly, the CSI-tree recovers all the CSIs

encoded in the synthetic data. We further quantify how well the combination of SPN and $\mathcal{EXSPN}$ approximates the ground truth CSIs using data samples from BNs. Since the data is sampled from a BN, the ground truth CSIs can be obtained by converting the conditional distributions $P(X_i|X \setminus X_i) \forall X_i \in X$ to *t*ree-*s*tructured *c*onditional *p*robability *d*istributions (Tree-CPDs). We use the ground truth CSIs to evaluate the CSIs extracted by $\mathcal{EXSPN}$. Table 3.3 summarizes the results of this evaluation. The ratios of the CSIs extracted by $\mathcal{EXSPN}$ to the ground truth is significantly high. Thus, Q1 is answered affirmatively.

(Q2: Compression) We can also infer from Table 3.1 that filtering the CSI rules using min_precision, min_recall and n_instances results in high compression ratios for **all but the MSNBC dataset**. This is because the MSNBC dataset already had 8 rules and all of the rules satisfied the threshold conditions. Hence, Q2 is answered strongly affirmatively.

(Q3. Baseline) Table 3.1 summarizes the association rules extracted from the data using the Apriori algorithm, and the mean confidence of the rules on the test set. Comparing the number of rules and the mean antecedent and consequent length from Table 3.1 allows us to answer Q3. We can infer that while the CSI rules extracted by $\mathcal{EXSPN}$ are longer than the rules extracted by the Apriori algorithm, the set of CSI rules is much smaller.

(Q4. Are the explanations correct?) Figure 3.4 shows the first two levels of the CSI-tree extracted by $\mathcal{EXSPN}$ from the nuMoM2b dataset. The first split of the CSI-tree is on the target variable oDM. While the BMI variable is independent of other variables when $oDM \neq 1$, it is dependent on $Age, Race, Education$ for the case when $oDM = 1$. These **independencies are validated by our domain expert, Dr. David Haas**. This clearly demonstrates the potential of explaining a joint model such as SPN in a real clinically relevant domain.

3.6 Discussion and Conclusion

We considered the challenging problem of explaining SPNs by defining a CSI-tree that captures the CSIs that exist in the data. We presented an iterative procedure for inducing the CSI-tree from a learned SPN by approximating the context induced by a product node using supervised learning. We then presented an algorithm for recovering an SPN that encodes the same CSIs as the original SPN from the CSI-tree thus establishing the correctness of the conversion. Our experiments in synthetic, benchmarks and a real clinical study demonstrate the effectiveness of the approach by identifying the correct CSIs from the data. As far as we are aware, this is the first work on explaining joint distributions using the lens of CSI. Validating our method on more relevant clinical studies, allowing for domain experts to interact with our learned model, extending the algorithm to work on the broader class of distributions in general and TDPMs in particular, including more type of explanations, and finally, scaling the algorithm to large number of features remain interesting directions.

CHAPTER 4

EXTRACTING QUALITATIVE KNOWLEDGE FROM PROBABILISTIC MODELS

Chapter 3 addressed explainability by looking inward: it asked what independence structure a trained SPN has encoded, and surfaced that structure as CSI-trees. This chapter asks a complementary question, one that looks outward at the model's behavior rather than inward at its architecture. Given a learned probabilistic model, what can be said about the qualitative relationships between its variables? Specifically, does the model reflect the intuition that increasing one risk factor tends to raise or lower the probability of an outcome, or that two risk factors together exert an influence that neither exerts alone? This chapter introduces QuaKE, a method for extracting qualitative influence statements of precisely this kind, and constitutes the second and final contribution of this dissertation to the tenet of explainability.

QuaKE targets two categories of qualitative influence: monotonic influence statements, which capture directed relationships between individual variables and an outcome, and synergistic influence statements, which capture interaction effects between pairs of variables. The method is applied to the nuMoM2b gestational diabetes dataset and validated by Dr. David Haas, continuing the pattern of expert-grounded evaluation established in Chapter 3. The technical details of both influence types and the clinical setup are developed in the sections that follow.

Taken together, Chapters 3 and 4 establish two distinct but complementary modes of explanation: structural explanation through CSI-trees, and behavioral explanation through qualitative influence statements. Both modes surface knowledge structures that are, notably, among the constraint types formally incorporated into the unified learning framework of Chapter 6. The explainability arc thus closes with a forward implication: the representations that make a model interpretable are also the representations that make domain knowledge

injectable. The next chapter departs from this thread to address the efficiency tenet through a data subset selection approach for supervised domain adaptation, before the dissertation returns to probabilistic circuits and the effectiveness tenet in its capstone contribution.

4.1 Introduction

The nuMoM2bstudy (Haas et al., 2015) aims to identify early warning signs of adverse pregnancy outcomes, design interventions, and assist with decision-making. Since 2010, eight research sites in the United States followed up with women throughout their pregnancies - collecting routine clinical information, exercise data, and food they ate. Using this data, we consider learning to explain the relationship between gestational diabetes mellitus (GDM) and some common risk factors.

A common way to employ knowledge in machine learning and AI is via the use of qualitative relationships that express how changes in a (subset of) feature(s)/risk factor(s) affect the target. These rules were mainly used as "inductive bias" apriori to learning since they are both intuitive and natural in many domains. We address the challenging "reverse task". Can we extract these rules from data? To this effect, in the context of nuMoM2b, we propose a two step process. First we learn a joint probability distribution over all the variables including the target (GDM). In the second step, the constraints are extracted by reasoning over this joint probability distribution. We demonstrate in our experiments that such an approach yields rules that are both intuitive and valid (as validated by our clinical expert). We first explain these constraints before outlining our approach and presenting our learned rules.

4.2 Background

A qualitative influence (QI) statement outlines how a change in one or more factor(s) would influence another factor (Wellman, 1990b). We focus on two types of QI: *Monotonicity and*

Synergy (Altendorf et al., 2005; Kokel et al., 2020b; Yang and Natarajan, 2013b). *Monotonicity* represents a direct relationship between two variables: "As BMI increases, neck circumference increases" indicates that the probability of greater neck circumference increases with increase in BMI. Specifically, a *monotonic influence* (MI) of variable X on variable Y, denoted by $X \prec^{M+} Y$ (or its inverse $X \prec^{M-} Y$), indicates that higher values of X stochastically result in higher (or lower) values of Y. *Synergy* represents interactions among influences. Two variables synergistically influence a third if their joint influence is greater than their separate, statistically independent influences. Synergy can capture influences like "Increase in BMI increases the risk of high blood pressure in patients with family history of hypertension more than patients without family history." Formally, a *synergistic influence* (SI) of two variables A and B on variable Y, denoted by $A, B \prec^{S+} Y$, indicates that increasing the value of A has greater effect on Y for higher value of B than the lower value of B. Both A and B should necessarily have same monotonic relationship with Y.[1] Similarly, a *sub-synergistic influence* (sub-SI), denoted by $A, B \prec^{S-} Y$, indicates that while A and B have increasing monotonic influence on Y, the joint influence is lesser than their separate, statistically independent influence.

4.3 Proposed Approach

Given: A data set $\mathcal{D}$ consisting of examples in the form of risk factors $\mathbf{x}$ and binary target Y (in this case: GDM).

To Do: Learn a set of QIs that explain the effect of $\mathbf{x}$ on Y.

We use X_a to denote the a^{th} variable in the feature set $\mathbf{x}$. x_a^i denotes a particular value of variable X_a and $|X_a|$ denotes the number of discrete values X_a takes. We assume that the joint distribution (P) over the set of random variables $\mathbf{x}$ is known (we learn this joint

[1]Without loss of generality, assume the variables in synergistic relation have monotonically increasing impact.

distribution in our empirical evaluation using a causal learning algorithm). For brevity, we restrict the description of our method to extracting positive MIs and SIs, $\prec^{M+}$ and $\prec^{S+}$. The *degree of monotonic influence*, δ_a, of $X_a \in \boldsymbol{X}$ on Y is defined as

$$\delta_a = I_{(C_a>0)} \cdot \sum_j \sum_{j'>j} \sum_k \frac{P(Y \leq k|X_a = x_a^j) - P(Y \leq k|X_a = x_a^{j'})}{|X_a|} \tag{4.1}$$

where,

$$C_a = \prod_j \prod_{j'>j} \prod_k \max(P(Y \leq k|X_a = x_a^j) - P(Y \leq k|X_a = x_a^{j'}) + \epsilon_m, 0) \tag{4.2}$$

For monotonicity to hold, we require $P(Y \leq k|X_a = x_a^j) + \epsilon_m \geq P(Y \leq k|X_a = x_a^{j'})$ for all pairs of configurations of $X_a, (j, j')$ with $j' > j$ at any given threshold value k. Here the monotonic slack ϵ_m allows violating a constraint within a chosen margin. The degree of MI, δ_a, in Equation 4.1 measures the cumulative difference in the probability that the target variable Y is less than a threshold k given X_a at two different values x_a^j and $x_a^{j'}$.

We extend the concept of degree of MI to SI by conditioning on a pair of variables instead of a single variable. First, consider the difference in the effect of changing X_a from x_a^i to $x_a^{i'}$ on Y under the context of two different values of X_b (x_b^j and $x_b^{j'}$). We define this as

$$\phi_{a,b}^{i,i',j,j'} = \sum_k P(Y \leq k|X_a = x_a^i, X_b = x_b^j) - P(Y \leq k|X_a = x_a^{i'}, X_b = x_b^j) - P(Y \leq k|X_a = x_a^i, X_b = x_b^{j'}) + P(Y \leq k|X_a = x_a^{i'}, X_b = x_b^{j'})$$

For synergy to hold, we require $\phi_{a,b}^{i,i',j,j'} + \epsilon_s$ to be non-negative for all $i' > i$ and $j' > j$. Where ϵ_s is the synergistic slack. We define the *degree of synergistic influence*, $\delta_{a,b}$, of variables $X_a \in \boldsymbol{X}$ and $X_b \in \boldsymbol{X}$ on $Y \in \boldsymbol{X}$ as the cumulative difference in degrees of context specific influence of X_a on Y in the context of X_b. It is given by

$$\delta_{a,b} = I_{(C_{a,b}>0)} \cdot \sum_i \sum_{i'>i} \sum_j \sum_{j'>j} \frac{\phi_{a,b}^{i,i',j,j'}}{|X_a| \cdot |X_b|} \tag{4.3}$$

Algorithm 5: QuaKE

input : Probabilistic model P, Target variable Y, Features $\mathbf{X}$, monotonic slack ϵ_m, synergistic slack ϵ_s, monotonic threshold T_m, synergistic threshold T_s

output : Rules $\boldsymbol{R}$

initialize: $\boldsymbol{R} \leftarrow \emptyset$
for $a \leftarrow 0$ **to** $(|\boldsymbol{X}| - 1)$ **do**
  compute δ_a using Eq. 4.1
  if $\delta_a \geq T_m$ **then**
    $\boldsymbol{R} \leftarrow (X_{a\prec}^{M+}Y) \cup \boldsymbol{R}$
  end
  for $b \leftarrow a + 1$ **to** $(|\boldsymbol{X}| - 1)$ **do**
    compute $\delta_{a,b}$ using Eq. 4.3
    if $\delta_{a,b} \geq T_s$ **then**
      $\boldsymbol{R} \leftarrow (X_a, X_{b\prec}^{S+}Y) \cup \boldsymbol{R}$
    end
  end
end
`(Similarly for decreasing cases))`
return $\boldsymbol{R}$

where,

$$C_{a,b} = \prod_{i} \prod_{i'>i} \prod_{j} \prod_{j'>j} \max(\phi_{a,b}^{i,i',j,j'} + \epsilon_s, 0) \tag{4.4}$$

We employ both definitions to learn QIs in Algorithm 5, Qualitative Knowledge Extraction (QuaKE). The algorithm assumes the existence of a joint distribution (Pearl, 1988) over ordinal features, which we learn using a causal probabilistic learning algorithm (PC) (Spirtes and Glymour, 1991; Colombo and Maathuis, 2014). We chose PC algorithm to verify our hypothesis that the use of a causal model will yield causally interpretable qualitative relationships. We calculate the degree of MI of every variable $X_a \in \boldsymbol{X}$ on Y and SI of every pair of variables $X_a, X_b \in \boldsymbol{X}$ on Y. The MI rules $X_{a\prec}^{M+}Y$ are extracted if their corresponding degree of MI δ_a are above a pre-defined threshold T_m. Similarly, the synergistic rules $X_a, X_{b\prec}^{S+}Y$ are extracted if their corresponding degree of SI $\delta_{a,b}$ are above a pre-defined threshold T_s.

4.4 Learning Qualitative Influences for GDM Modeling

The nuMoM2b study tracked pregnancies of $10,037$ women near 8 sites in the United States. We excluded 817 cases where women were already diagnosed with diabetes; and we evaluate our proposed method for extracting QIs using 8 features[2] of the remaining $9,220$ women. 7 features had inherent ordering of categories whereas *Race* had no obvious ordering. We use an ordering based on previous studies (Hedderson et al., 2010) on the effect of *Race* on *GDM*. We pose and answer the following questions: **(Q1)** Does QuaKE extract high-quality rules that align with background knowledge in this domain? **(Q2)** Does QuaKE help uncover QI statements in cases where prior knowledge is uncertain?

We compare learned rules with those from our clinical expert, *Dr. Haas*. W.r.t GDM, these could either be increasing, decreasing, no effect, or unknown. Since Algorithm 5 assumes a complete joint distribution P is available, we consider two factorizations of P. The first learns a causal model (Colombo and Maathuis, 2014) and the other (baseline) estimates the probabilities directly from data. Alternative baselines might have included rules extracted from decision trees, rule mining, or Bayesian rule learning—but each induce conjunctive rules of the form $(x_1 \wedge x_2 \wedge ... \wedge x_n) \implies y$, making their exact connection to the QI statements tenuous.

All rules are presented in Table 4.1. The "Prior" knowledge refers to the rules provided by our expert. We compare these to the rules extracted by QuaKE and baseline (Data Alone). QuaKE's precision compared to expert advice is 0.923 ± 0; whereas the precision of our unstructured baseline is 0.636 ± 0. Precision of each method was consistent across five stratified cross validation folds. This affirms **Q1**: QuaKE can extract high-quality rules aligning with prior knowledge.

[2]Refer to the supplementary material for details on the data and features: https://starling-lab.github.io/papers/QuaKE/

Table 4.1: Comparision of QI from prior knowledge (PK), QuaKE and Data Alone. ✓/✗ represents that this relationship does/not exist respectively while **?** represents unknown influence. The three groups of rows show: (1) MI, (2) SI, and (3) sub-SI. Colors highlight rules recovered by QuaKE and show (a.) coherent with the PK and baseline (b.) contradicting the baseline (c.) coherent with baseline but contradicts the PK.

Rule	Prior Knowledge	QuaKE	Data Alone
$BMI \overset{M+}{\prec} GDM$	✓	✓	✓
$Age \overset{M+}{\prec} GDM$	✓	✓	✓
$Race \overset{M+}{\prec} GDM$	✓	✓	✗
$Education \overset{M+}{\prec} GDM$	✓	✓	✗
$Gravidity \overset{M+}{\prec} GDM$	✓	✓	✗
$Smoked3months \overset{M+}{\prec} GDM$	✓	✗	✗
$SmokedEver \overset{M+}{\prec} GDM$	✓	✗	✗
$Age, BMI \overset{S+}{\prec} GDM$	✓	✓	✓
$Age, Smoked3months \overset{S+}{\prec} GDM$	✓	✓	✓
$BMI, SmokedEver \overset{S+}{\prec} GDM$	✓	✓	✓
$Education, Smoked3months \overset{S+}{\prec} GDM$	?	✓	✓
$BMI, Gravidity \overset{S+}{\prec} GDM$	✓	✓	✗
$BMI, Smoked3months \overset{S+}{\prec} GDM$	✓	✗	✓
$Age, SmokedEver \overset{S+}{\prec} GDM$	✓	✗	✗
$BMI, Education \overset{S+}{\prec} GDM$	✗	✓	✓
$Education, SmokedEver \overset{S+}{\prec} GDM$	?	✗	✗
$Age, Education \overset{S-}{\prec} GDM$	✓	✓	✓
$BMI, Smoked3months \overset{S-}{\prec} GDM$	✗	✓	✗
$Age, SmokedEver \overset{S-}{\prec} GDM$	✗	✗	✓
$BMI, Gravidity \overset{S-}{\prec} GDM$	✗	✗	✓
$Gravidity, SmokedEver \overset{S-}{\prec} GDM$	✗	✗	✓
$Education, SmokedEver \overset{S-}{\prec} GDM$	?	✗	✓
$Age, Gravidity \overset{S-}{\prec} GDM$	✓	✗	✗

Since we have formalized degree of the QIs in Equations 4.1 and 4.3, we can analyze rules that were highly uncertain according to the prior knowledge. Two of the synergistic relations involving smoking and education had an unknown effect with relation to GDM. $Education, Smoked3months \overset{S+}{\prec} GDM$ was a high-confidence rule extracted by QuaKE and the baseline. We speculate that this could be either due to the high correlation between *Education* and *Age*, or related to an unobserved relationship between education and socioeconomic

status. Note that both these results are especially interesting since we found only a weak monotonic relationship between smoking and GDM more generally. We use this to answer **Q2**: our approach can identify potentially interesting cases where prior knowledge is uncertain.

Discussion and Conclusion: We considered the problem of learning interpretable and explainable qualitative rules for modeling GDM. To this effect, we learned a causal (probabilistic) model and recovered the knowledge by applying the rules. Our results indicate that most of our rules are in line with the prior knowledge of our expert and some interesting influence relationships appear that are worth investigating. Incorporating richer domain knowledge, automatically refining the rules, identifying broader relationships and scaling to larger feature sets are interesting future research directions.

CHAPTER 5

EFFICIENT DOMAIN ADAPTATION THROUGH DATA SUBSET SELECTION

The preceding two chapters developed methods for explaining what a probabilistic circuit has learned, surfacing its internal independence structure and its qualitative behavioral relationships in forms that domain experts can read and evaluate. This chapter departs from that thread in terms of application domain and model class, and it is worth noting that departure directly before motivating the applicability of this work to the types of clinical settings seen in the rest of the chapters. Consider a probabilistic model trained on nuMoM2b data collected across eight clinical sites in the United States, where the population is predominantly nulliparous women with a particular demographic profile. Deploying that model in a different country – where the age distribution, race composition, and prevalence of relevant risk factors may differ substantially – is precisely a supervised domain adaptation problem: the source and target distributions are mismatched, labeled target data is scarce, and naive retraining on the full source dataset is both expensive and potentially counterproductive. This is exactly the setting that ORIENT is designed to address. The empirical evaluation in this chapter uses standard computer vision benchmarks rather than clinical data, but the efficiency gains it demonstrates are directly relevant to deployment scenarios of this kind, and the framework itself is model-agnostic and domain-agnostic.

The technical contribution is ORIENT, a subset selection framework that addresses the computational overhead of supervised domain adaptation (SDA) by identifying, from a large pool of source data, the subset most relevant to the target domain. The background established in Chapter 2 – specifically the formalism of SDA and the submodular mutual information (SMI) framework – provides the necessary technical foundation, and the chapter builds directly on that groundwork without re-deriving it. ORIENT is agnostic to the choice of SDA loss and can therefore be combined with any gradient-based SDA method, a generality

that distinguishes it from prior subset selection approaches tailored to specific objectives. Empirically, it achieves roughly a threefold reduction in training time while maintaining or improving classification performance on standard domain adaptation benchmarks.

The efficiency gains demonstrated here foreshadow a theme that runs through the contribution of Chapter 6: that simplicity in objective design, when grounded in sound theory, need not sacrifice effectiveness. The following chapter returns to probabilistic circuits and to the question of how domain knowledge can be incorporated into their learning, drawing on the constraint vocabulary established in Chapters 3 and 4 to motivate a unified and practically grounded framework.

5.1 Introduction

The recent success of deep learning frameworks in applications such as image classification (Ciresan et al., 2012), speech recognition (Hershey et al., 2010), and object detection (Geirhos et al., 2018) stems primarily from the availability of large amounts of labeled data. However, due to enormous labeling costs and the need for specialists in specific domains such as medical imaging, it is not always possible to obtain vast amounts of labeled data in all situations. On the contrary, training deep models on limited amounts of data can lead to poor performance due to overfitting (Arpit et al., 2017). Consequently, where obtaining large amounts of labeled data is difficult for the target domain, a closely related domain (source domain) is used to train the model. However, this may result in a deep model with suboptimal performance on the target domain as a result of the distribution shift (Shimodaira, 2000; Ben-David et al., 2010; Ben-David et al., 2006; Torralba and Efros, 2011), i.e., change of data distribution from source domains to target domains. A change in the data distribution often renders the features learned by the model on the source domain irrelevant for the target domain.

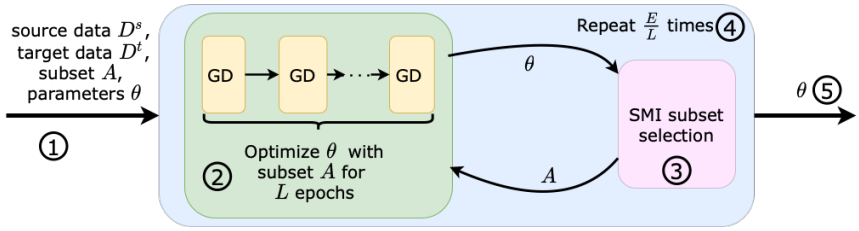


Figure 5.1: Illustration of ORIENT framework. **(1)** Given the target data D^t and source data D^s, a subset $A \subseteq D^s$ and model parameters θ are randomly initialized. **(2)** Model parameters θ are optimized for L epochs on the subset A with any gradient descent (GD) based method. **(3)** Subset A is updated using the SMI measure and current model parameters θ after every L^{th} epoch. **(4)** Model is trained for E epochs, with subset selection after every L interval. **(5)** The final model parameters are returned after E epochs.

To address the problem of distribution shift, many domain adaptation (Wang and Deng, 2018; Patel et al., 2015) and domain generalization (Muandet et al., 2013) techniques have been proposed in recent years. Domain adaptation methods assume that some target domain information is available (usually target data), whereas domain generalization methods do not. Domain adaptation (DA) methods can be categorized into unsupervised (Gong et al., 2012; Ganin and Lempitsky, 2015; Liu and Tuzel, 2016; Tzeng et al., 2017) (using unlabeled target data), semi-supervised (Guo and Xiao, 2012; Yao et al., 2015; Saito et al., 2019; Singh, 2021) (using labeled and unlabeled target data), and supervised (Motiian et al., 017a,b; Xu et al., 2019; Morsing et al., 2020; Hedegaard et al., 2021) (using labeled target data) methods. UDA methods that do not require any labeled target data assume access to large volumes of unlabeled target data. When providing the same amount of target data, SDA methods are more effective than UDA methods (Motiian et al., 017a). As it is not difficult to obtain a few samples (as less as 2 examples per class) of labeled target data in practice, SDA methods are an attractive approach in scenarios with limited target data. Hence, in this work we are explicitly focused on the SDA setting. Recently, various effective SDA methods are proposed (Motiian et al., 017a; Xu et al., 2019; Morsing et al., 2020). However, they are usually compute-intensive. For example, using one of the state-of-the-art SDA method, *d*-SNE (Xu et al., 2019), to train a ResNet50 model (He et al., 2016) on the

Office-Home dataset (Venkateswara et al., 2017) with 3022 samples of the source data and 338 samples a target data for 300 epochs takes > 18 hours using a `GTX 1080 Ti` GPU. To put that in perspective, training a ResNet50 model on the larger CIFAR100 dataset with 45000 training samples for 300 epochs on the same machine for standard cross-entropy loss takes only 14 hours. The increase in training time not only increases energy consumption and CO2 emissions but also restricts the usefulness of SDA methods in resource-constrained environments. We address this problem by seeking an answer to the following question: ***Can we improve the efficiency of SDA methods by training on subsets of source data to ensure faster adaptation and reduction in training time?***

To this end, we propose ORIENT, a subset selection framework that utilizes Submodular Mutual Information (SMI) measures (Iyer et al., 2022; Gupta and Levin, 2020) to select subsets of source data that are similar to the target data for training. Our key insight is that by training the model on data points that are similar to the target data, we can speed up the training. The ORIENT framework, illustrated in Figure 5.1, complements existing SDA methods and can be effectively combined with any of them, enabling us to achieve a reduction in training times, energy costs, and CO2 emissions.

Subset Selection methods: Submodular functions (Lovász, 1982; Fujishige, 2005) provide an effective way of selecting relevant subsets of data by acting as data acquisition functions, as applied in various settings (Bilmes, 2022). For example, it has been employed for efficient training in speech recognition (Wei et al., 014a,b), machine translation (Kirchhoff and Bilmes, 2014), computer vision (Kaushal et al., 2019), supervised classification (Mirzasoleiman et al., 2020; Killamsetty et al., 021a,b), semi-supervised classification (Killamsetty et al., 021d), active learning methods (Wei et al., 2015; Sener and Savarese, 2018; Ash et al., 2020; Killamsetty et al., 021a), and hyper-parameter optimization (Killamsetty et al., 2022). Another widely used method for selecting subsets is coreset construction. The coresets (Feldman, 2020) are weighted subsets of the data that approximate the semantic characteristics of the entire

Algorithm 6: ORIENT

Input: source domain data D^s, query data D^t, batch size b, total epochs E, subset selection interval L, SMI function I_f.

Output: Final model parameters θ

$A \leftarrow \text{RandomSubset}(D^s)$ ▷ initial model parameters
θ ▷ randomly intialize model parameters
for *epoch* e *in* E **do**
  $\theta \leftarrow \text{mini-batch-gd}(\theta, \mathcal{L}(A))$ ▷ mini-batch gradient descent
  if $e \mod L == 0$ **then**
    $\boldsymbol{\chi}^s \leftarrow \nabla_\theta \mathcal{L}(D^s)$ ▷ compute gradients for source data
    $\boldsymbol{\chi}^t \leftarrow \nabla_\theta \mathcal{L}(D^t)$ ▷ compute gradients for query data
    $S \leftarrow \text{SIMILARITY}(\boldsymbol{\chi}^s, \boldsymbol{\chi}^t)$ ▷ compute the similarity matrix
    Instantiate I_f with S
    $A \leftarrow \underset{A \subseteq D^s, |A| \leq b}{\arg\max}\ I_f(A; D^t)$ ▷ update subset A
  end
end

dataset (e.g., loss, marginal probability, etc.). A number of recent coreset selection-based methods (Mirzasoleiman et al., 2020; Killamsetty et al., 021a,b,d) have demonstrated the promise to efficiently and robustly train deep models. Coresets are used for other deep learning applications like continual learning, active learning and data summarization (Borsos et al., 2021; Tiwari et al., 2021).

Subset selection plays a significant role in robust learning under realistic scenarios, such as imbalances or rare classes, out-of-distribution(OOD) data, or redundancy. Axelrod et al. (2011) propose a simple cross-entropy based data selection method for effective domain adaptation in statistical machine learning. Kothawade et al. (2021) employed SMI measures for active learning in realistic scenarios tackling imbalance, OOD, and redundancy. PRISM (Kothawade et al., 2022) employed SMI measures for targeted data summarization. GLISTER (Killamsetty et al., 021a) and GRADMATCH (Killamsetty et al., 021b) show that using a clean held-out validation set with SMI mitigates the label noise and class imbalance in the training set. Mirzasoleiman et al. (2020) showed the data samples with clean labels cluster

Table 5.1: Instantiations of submodular mutual information functions (Kothawade et al., 2022).

Name	$f(A)$	$I_f(A; D^t)$
FLMI	$\sum_{i \in D^t} \max_{j \in A} S_{ij}$	$\sum_{i \in D^t} \max_{j \in A} S_{ij} + \eta \sum_{i \in A} \max_{j \in D^t} S_{ij}$
LOGDETMI	$\log\det(S_A)$	$\log\det(S_A) - \log\det(S_A - \eta^2 S_{A,D^t} S_{D^t}^{-1} S_{A,D^t}^T)$
GCMI	$\sum_{i \in A, j \in D^s} S_{ij} - \lambda \sum_{i,j \in A} S_{ij}$	$2\lambda \sum_{i \in A, j \in D^t} S_{ij}$

together, whereas the ones with noisy labels spread out in the gradient space, thus making k-medoid clustering an effective method for reducing noise in labeled data. Furthermore, Killamsetty et al. (021d) tackles OOD and imbalance in the unlabeled data in semi-supervised learning setting through data subset selection.

5.2 Methodology

For our SDA setting, we assume a small amount of labeled data (as less as 2 examples per class) is available from the target domain τ^t and sufficient labeled data is available from the source domain τ^s. We denote the source dataset as $D^s = (x_i, y_i)_{i=1}^{|D^S|}$ and the target dataset as $D^t = (x_i, y_i)_{i=1}^{|D^t|}$. The source dataset is used for training the model $\mathcal{M}$ using an SDA loss function $\mathcal{L}$.

First, we extend the SMI measure to incorporate data subsets from different domains. For a source domain subset $A \subseteq D^s$ and the target domain data set D^t, we denote SMI measure as $I_f(A; D^t)$. Essentially, assuming that both the domains form a set $\mathcal{D}$, and $A \subseteq D^s \subset \mathcal{D}$ and $D^t \subset \mathcal{D}$. With S denoting the similarity matrix defined over a subset A and D^t, Table 5.1 presents the mathematical expression of the SMI functions for different instantiations of submodular functions (f) (Kothawade et al., 2022). We defer the readers to Kothawade et al. (2022) for more details. Note that the FLMI, GCMI and COM only need the pairwise

similarities of the data points in A and D^t. Consequently, the similarity kernel is of size $|A| \times |D^t|$, making it very efficient to optimize. For efficient SDA, we want to select a subset A of the source domain τ^s such that training a model on A results in a classifier that is proficient on the target domain τ^t. To this effect, we use the gradients $\boldsymbol{\chi}$ of the current model to represent the data points and use it to compute the similarity matrix. Specifically, we define the pairwise similarity between two data points as the cosine similarity between the gradients,

$$S_{ij} = \frac{\chi_i \, \chi_i}{|\chi_i| \, |\chi_j|} \tag{5.1}$$

$$\text{where, } \chi_i = \nabla_\theta \mathcal{L}(x_i, y_i) \text{ and } \chi_j = \nabla_\theta \mathcal{L}(x_j, y_j)$$

Given a set size b, we select a *diverse* subset A of the source data D^s that is *similar* to the target data D^t by maximizing the following SMI measure,

$$\arg \max_{A \subseteq D^s, |A| \leq b} I_f(A; D^t) \tag{5.2}$$

$I_f(A; D^t)$ is an instance of cardinality constrained monotone submodular maximization for all SMI functions except for LogDetMI. Therefore, we can achieve a $1 - \frac{1}{e}$ approximation using the lazy greedy algorithm (Minoux, 1978). Even though LogDetMI is not submodular, previous works (Kothawade et al., 2022, 2021) have reported good empirical performance using the lazy greedy algorithm for maximization of the LogDetMI function. Following the footsteps of Kothawade et al. (2022, 2021), we also use the lazy greedy algorithm to maximize the LogDetMI function. However, as the number of parameters in the deep learning model can be extremely high, our gradients χ can be very high dimensional. Following the success of targeted subset selection approaches, GLISTER (Killamsetty et al., 021a), SIMILAR (Kothawade et al., 2021), CRAIG (Mirzasoleiman et al., 2020) and BADGE (Ash et al., 2020), we circumvent this problem by using last-layer gradient approximations to

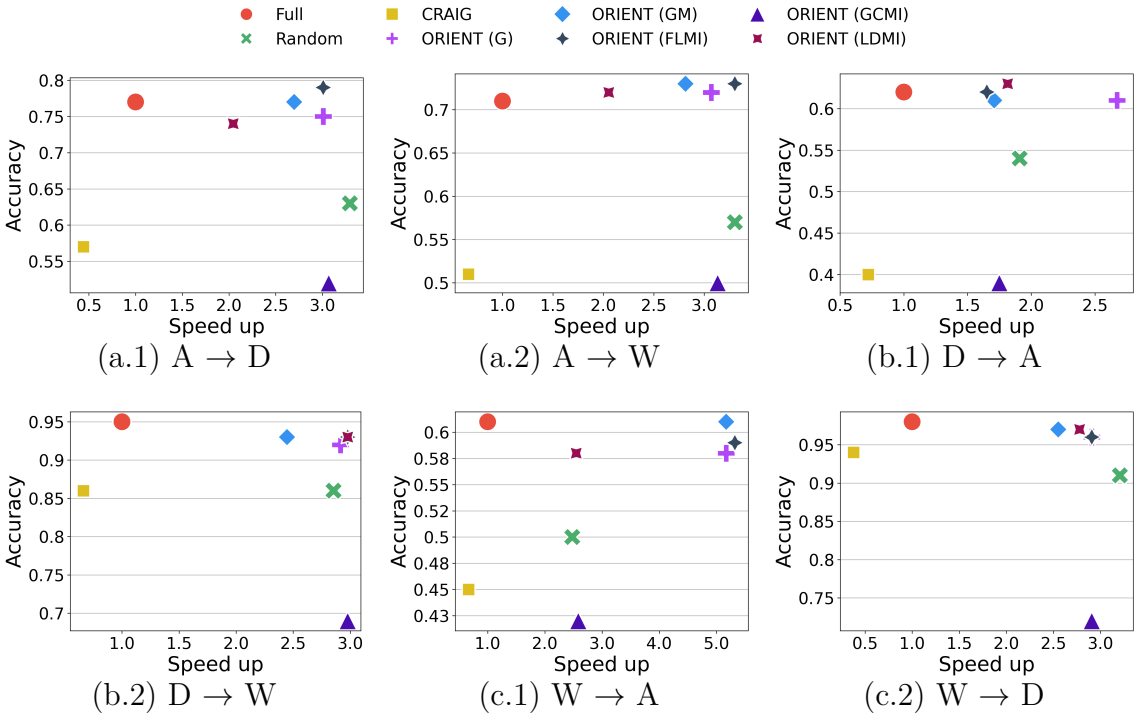


Figure 5.2: Speed up vs prediction accuracy on three domains of Office31 dataset: Amazon (A), DSLR (D), and Webcam (W). X-axis represents the speed up by the model, i.e. the ratio of the time taken to train on complete source dataset (Full) to the time taken by the model. Y-axis represents the prediction accuracy of the model on the target domain.

represent the data point in the similarity matrix S. Further, we select a new subset every L epochs as the model is trained, and the subsets chosen are adapted accordingly.

Algorithm: We present the complete training procedure of ORIENT in Algorithm 6. We first randomly initialize the training subset A and the model parameters θ in **line 1** and **2**, respectively. For each epoch, we update the model parameters by optimizing a gradient-descent (GD) based loss $\mathcal{L}$ on the current training subset A in **line 4**. After a fixed interval of L epochs, we update the subset A (**line 5–11**). To do this, we compute the gradients $\boldsymbol{\chi}^s$ and $\boldsymbol{\chi}^t$ for all the data points in D^s and D^t, in **line 6** and **7**, respectively. Next, we use the gradients to compute the similarity matrix S in **line 8**, as described in Equation

5.1. In **line 9** we instantiate the SMI function I_f with S and update the subset A in **line 10** using Equation 5.2. Our implementation is made available online[1].

Figure 5.1 illustrates the ORIENT workflow. The general framework of ORIENT can incorporate any gradient decent (GD) based SDA technique. It can train any model with loss function $\mathcal{L}$ by using a subset of the source data, selected every L epochs. In our experiments, we demonstrate the effectiveness of our method in conjunction with standard cross-entropy loss as well as two state-of-the-art domain adaptation losses, CCSA (Motiian et al., 017a) and d-SNE (Xu et al., 2019).

5.2.1 Connections to Previous Work

ORIENT generalizes three approaches, GLISTER (Killamsetty et al., 021a), GRADMATCH (Killamsetty et al., 021b), and a slightly adapted version of CRAIG (Mirzasoleiman et al., 2020), which were initially proposed for efficient and robust subset selection. Specifically, these subset selection approaches maximize a submodular function corresponding to a different instantiations of SMI functions for subset selection. Following theorems summarize the connection of SMI functions with these subset selection approaches.

Theorem 5.2.1. *When the outer level loss of the discrete bi-level optimization problem of GLISTER is hinge loss, logistic loss, and perceptron loss, then the optimization problem becomes an instance of maximization of Concave over Modular (COM) SMI function.*

Proof. Given, training loss L_T, validation loss L_V, training set $\mathcal{D}$, subset $\mathcal{S}$, subset size k, and validation set $\mathcal{V}$, the objective of GLISTER can be written as follows:

$$\begin{aligned} &\min_{\mathcal{S}\subseteq\mathcal{D},|\mathcal{D}|=k} L_V(\theta^*, \mathcal{V}) \\ &\text{where } \theta^* = \arg\min_{\theta} L_T(\theta, \mathcal{S}) \end{aligned} \tag{5.3}$$

[1]Code: https://github.com/athresh/orient

Loss on the subset denoted by $L_T(\theta, \mathcal{S})$ is the summation of the losses of individual data samples in the subset. i.e., $L_T(\theta, \mathcal{S}) = \sum_{(x,y)\in\mathcal{S}} L_T(\theta, x, y)$. Using the one-step gradient approximation of GLISTER, the above optimization problem can be written as:

$$\begin{aligned} &\min_{\mathcal{S}\subseteq\mathcal{D},|\mathcal{D}|=k} L_V(\theta - \eta\nabla_\theta L_T(\theta, \mathcal{S}), \mathcal{V}) \\ &\min_{\mathcal{S}\subseteq\mathcal{D},|\mathcal{D}|=k} L_V(\theta - \eta \sum_{(x,y)\in\mathcal{S}} \nabla_\theta L_T(\theta, x, y), \mathcal{V}) \end{aligned} \tag{5.4}$$

where η is the learning rate.

We can convert the above minimization problem to a maximization problem as following:

$$\max_{\mathcal{S}\subseteq\mathcal{D},|\mathcal{D}|=k} -L_V(\theta - \eta \sum_{(x,y)\in\mathcal{S}} \nabla_\theta L_T(\theta, x, y), \mathcal{V}) \tag{5.5}$$

Using the Proof of Theorem-1 in GLISTER (Killamsetty et al., 021a), the optimization problem in Equation 5.5 for different validation losses can be written as follows:

When L_V is hinge loss or perceptron loss, the optimization problem of GLISTER is:

$$\begin{aligned} &\max_{\mathcal{S}\subseteq\mathcal{D},|\mathcal{D}|=k} \sum_{i=1}^{|\mathcal{V}|} \min(0, C_i + \sum_{j\in\mathcal{S}} \hat{g}_{ij}) \\ &\text{where } \sum_{j\in\mathcal{S}} \hat{g}_{ij} \geq 0 \end{aligned} \tag{5.6}$$

We can write the above optimization problem as,

$$\max_{\mathcal{S}\subseteq\mathcal{D},|\mathcal{D}|=k} |\mathcal{V}|C_i + \sum_{i=1}^{|\mathcal{V}|} \min(-C_i, \sum_{j\in\mathcal{S}} \hat{g}_{ij}) \tag{5.7}$$

$$\max_{\mathcal{S}\subseteq\mathcal{D},|\mathcal{D}|=k} \sum_{i=1}^{|\mathcal{V}|} \min(-C_i, \sum_{j\in\mathcal{S}} \hat{g}_{ij}) \tag{5.8}$$

Note that in the above equation, $\min(C, x)$ is a concave function in x and $\sum_{j\in\mathcal{S}} \hat{g}_{ij}$ is a non-negative modular function. Hence, the above formulation is submodular and is an instance of concave over modular function.

When L_V is logistic loss, the optimization problem of GLISTER is:

$$\max_{\mathcal{S}\subseteq\mathcal{D},|\mathcal{D}|=k} \sum_{i=1}^{|\mathcal{V}|} C - \log\left(1 + C_i \exp\left(\alpha \sum_{j\in\mathcal{S}} \hat{g}_{ij}\right)\right) \tag{5.9}$$

Note that in the above equation, $-\log(1 + C\exp -x)$ is concave in x and $\sum_{j\in\mathcal{S}} \hat{g}_{ij}$ is modular. Hence, the above formulation of set function is submodular and is an instance of concave over modular function. In our setting, we use labeled target data D^t as the validation set. Furthermore, the above given formulations of GLISTER corresponds to instantiation of maximization of COM MI function $(\eta \sum_{i\in A} \psi(\sum_{j\in D^t} S_{ij}) + \sum_{j\in D^t} \psi(\sum_{i\in A} S_{ij}))$ with $\eta = 0$.

□

Theorem 5.2.2. *When the optimization problem of GRADMATCH with equal sample weights is used to match gradients of a held-out validation set, it becomes an instance of maximization of summation of GCMI and a diversity function.*

Proof. Given, training loss L_T, validation loss L_V, training set $\mathcal{D}$, and validation set $\mathcal{V}$. Let us denote loss of i^{th} sample in the training dataset as $L_T^i(\theta)$ and the loss of j^{th} sample in the validation dataset as $L_V^j(\theta)$.

The objective of GRADMATCH with equal sample weights can be written as follows:

$$\min_{\mathcal{S}\subseteq\mathcal{D},|\mathcal{S}|=k} \left\| \frac{1}{k} \sum_{i\in\mathcal{S}} \nabla_\theta L_T^i(\theta_t) - \frac{1}{|\mathcal{V}|} \sum_{j\in\mathcal{V}} \nabla_\theta L_V^j(\theta) \right\|^2 \tag{5.10}$$

Without the loss of generality, we assumed sample weights to be 1 in the above equation.

We can convert this to a maximization problem as follows:

$$\max_{\mathcal{S}\subseteq\mathcal{D},|\mathcal{S}|=k} - \left\| \frac{1}{k} \sum_{i\in\mathcal{S}} \nabla_\theta L_T^i(\theta_t) - \frac{1}{|\mathcal{V}|} \sum_{j\in\mathcal{V}} \nabla_\theta L_V^j(\theta) \right\|^2 \tag{5.11}$$

$$\max_{\mathcal{S}\subseteq\mathcal{D},|\mathcal{S}|=k} \frac{-1}{k^2} \left\| \sum_{i\in\mathcal{S}} \nabla_\theta L_T^i(\theta) \right\|^2 + \frac{-1}{|\mathcal{V}|^2} \left\| \sum_{j\in\mathcal{V}} \nabla_\theta L_V^j(\theta_t) \right\|^2 + 2\frac{1}{k|\mathcal{V}|} \sum_{i\in\mathcal{S},j\in\mathcal{V}} \nabla_\theta L_T^i(\theta)^T \cdot \nabla_\theta L_V^j(\theta) \tag{5.12}$$

In the above equation, the second term is not dependent on $\mathcal{S}$ and can be ignored during optimization. Following which the above optimization problem can be written as:

$$\max_{\mathcal{S}\subseteq\mathcal{D},|\mathcal{S}|=k} \frac{-1}{k^2}\left\|\sum_{i\in\mathcal{S}}\nabla_\theta L_T^i(\theta)\right\|^2 + 2\frac{1}{k|\mathcal{V}|}\sum_{i\in\mathcal{S},j\in\mathcal{V}}\nabla_\theta L_T^i(\theta)^T\cdot\nabla_\theta L_V^j(\theta) \tag{5.13}$$

Note that in our setting, labeled target data D^t is used as the validation set. In the above equation, the second term corresponds to GCMI function($2\lambda\sum_{i\in A,j\in D^t} S_{ij}$) with $\lambda = 1$.

Expanding on the first term we have,

$$\frac{-1}{k^2}\left\|\sum_{i\in\mathcal{S}}\nabla_\theta L_T^i(\theta)\right\|^2 = \frac{-1}{k^2}\sum_{i\in\mathcal{S}}\left\|\nabla_\theta L_T^i(\theta)\right\|^2 - \frac{2}{k^2}\sum_{i,j\in\mathcal{S}|i\neq j}\nabla_\theta L_T^i(\theta)^T\cdot\nabla_\theta L_T^j(\theta) \tag{5.14}$$

The function given in Equation 5.14 is a diversity function.

Hence, the optimization problem of GradMatch with equal sample weights when matched with validation set is an instance of maximization of summation of the GCMI function and diversity function.

□

Theorem 5.2.3. *When the optimization problem of Craig is adapted to match gradients of a held-out validation set, it becomes an instance of maximization of the FLMI function with* $\eta = 0$.

Proof. Given, training loss L_T, validation loss L_V, training set $\mathcal{D}$, and validation set $\mathcal{V}$. Let us denote the loss of i^{th} sample in the training dataset as $L_T^i(\theta)$ and the loss of j^{th} sample in the validation dataset as $L_V^j(\theta)$.

The objective of Craig matching gradients of a held-out validation set is as follows:

$$\max_{\mathcal{S}\subseteq\mathcal{D},|\mathcal{S}|=k}\sum_{i\in\mathcal{V}}\max_{j\in\mathcal{S}}\nabla_\theta L_V^i(\theta)^T\cdot\nabla_\theta L_T^j(\theta) \tag{5.15}$$

(a) d-SNE + Full

(b) d-SNE + ORIENT (FLMI)

(c) d-SNE + Full

(d) d-SNE + ORIENT (FLMI)

Figure 5.3: GradCam (Selvaraju et al., 2017) activation maps of the models learned using d-SNE + Full and d-SNE + ORIENT (FLMI) on the Office-Home dataset with "Product" as the source domain and "Real World" as the target domain. As evidenced by class activation maps, the ORIENT framework enabled the model to learn more effective class discriminative features than Full data training.

Note that in our setting, labeled target data D^t is used as the validation set. The set function in the above equation corresponds to FLMI function$(\sum_{i \in D^t} \max_{j \in A} S_{ij} + \eta \sum_{i \in A} \max_{j \in D^t} S_{ij})$ with $\eta = 0$.

Hence, the optimization problem of CRAIG adapted to match gradients of a held-out validation set is an instance of maximization of the FLMI function with $\eta = 0$. □

5.3 Experimental Evaluation

Our experiments explicitly aim at answering the following questions,

Q1: Does the proposed ORIENT approach substantially reduce the training time while maintaining comparable performance to training on complete source dataset?

Q2: Can the proposed ORIENT approach augment the existing domain adaptation approaches?

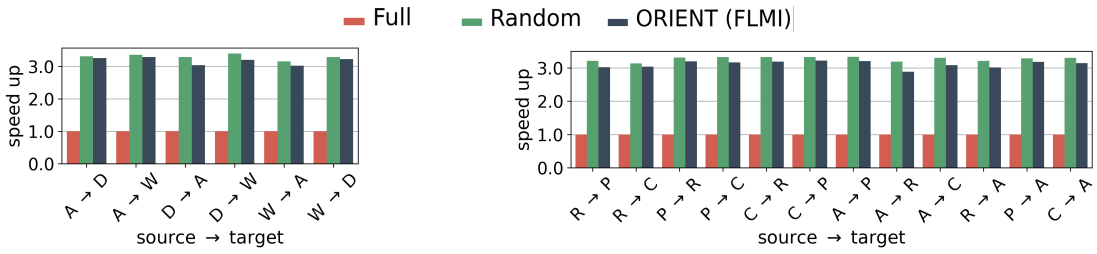
We evaluate our proposed approach on two domain adaptation datasets: Office-31 (Saenko et al., 2010) and Office-Home (Venkateswara et al., 2017). Office-31 dataset consists of 4110 images of 31 object categories from three different domains: Amazon (A), DSLR (D), and Webcam (W). For our experiments we used all the images from the source domain in the training set (D^s) and two examples of each category in the query set (D^t). Office-Home

Table 5.2: Test accuracy for office-31 with SDA methods

		A → D	A → W	D → A	D → W	W → A	W → D
CCSA	Full	0.78	0.72	0.55	**0.93**	0.55	**0.97**
	Random	0.76	0.72	0.54	0.81	0.54	0.92
	ORIENT	0.77	**0.76**	0.55	0.89	0.55	**0.96**
d-SNE	Full	0.77	0.69	0.53	**0.93**	0.54	**0.98**
	Random	0.76	0.68	0.53	0.86	0.53	0.94
	ORIENT	0.78	0.71	**0.55**	0.90	0.56	**0.97**

Table 5.3: Test accuracy for Office-Home with SDA methods

		R → P	R → C	P → R	P → C	C → R	C → P	A → P	A → R	A → C	R → A	P → A	C → A
CCSA	Full	0.74	**0.55**	0.62	0.46	0.56	0.67	0.70	**0.64**	0.48	0.57	**0.49**	0.41
	Random	0.71	0.47	0.62	0.45	0.54	0.66	0.69	0.57	0.48	0.5	0.47	0.45
	ORIENT	**0.78**	**0.54**	**0.65**	**0.5**	**0.61**	**0.71**	0.71	**0.65**	**0.51**	**0.59**	**0.54**	**0.47**
d-SNE	Full	0.77	**0.53**	**0.62**	**0.50**	**0.60**	0.71	**0.72**	**0.63**	0.49	**0.52**	0.44	0.40
	Random	0.75	0.50	0.60	0.45	0.57	0.69	0.68	0.59	0.49	0.46	0.43	0.40
	ORIENT	0.77	**0.52**	**0.63**	**0.50**	**0.60**	0.71	**0.71**	0.61	0.51	**0.52**	0.44	0.42



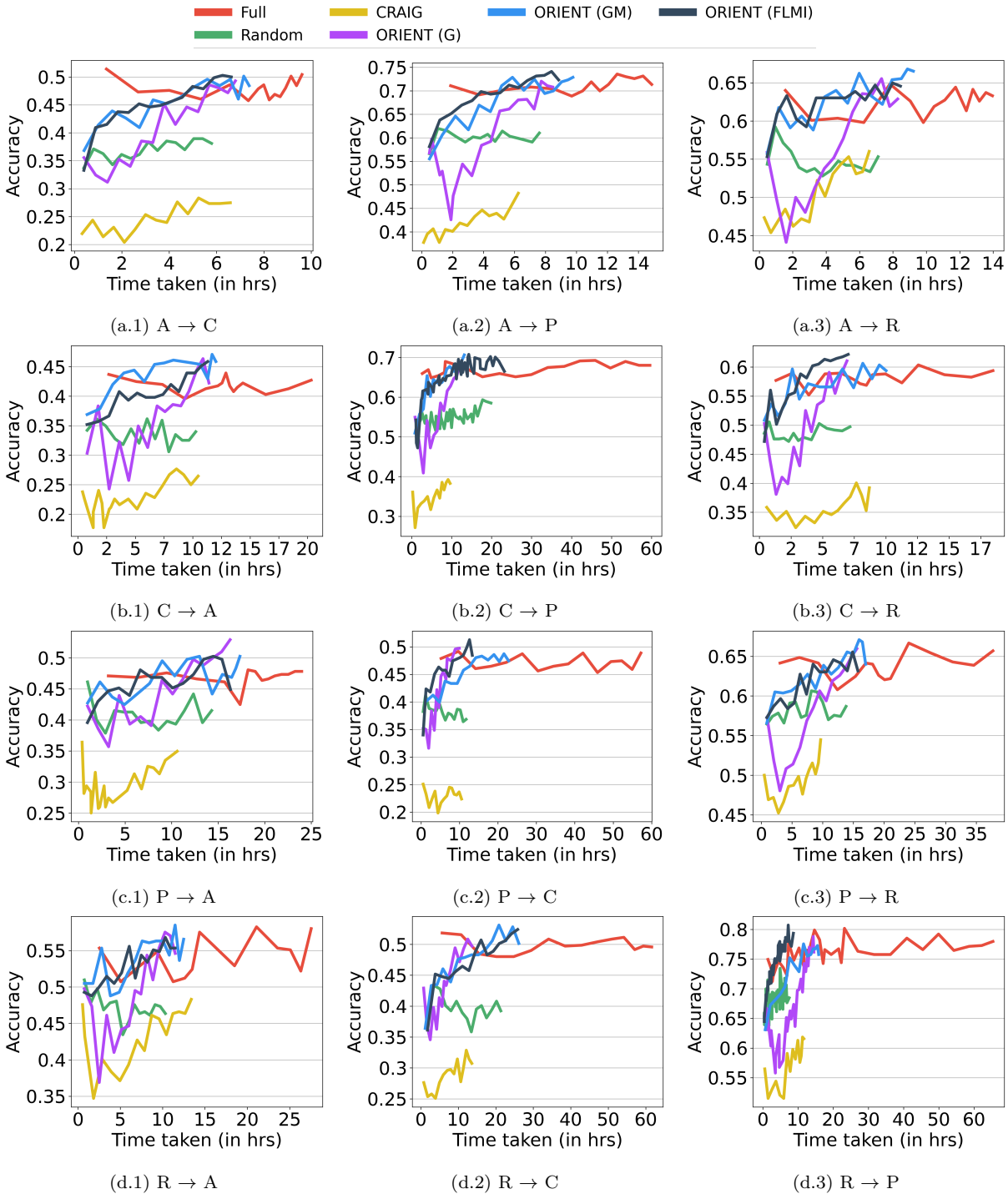
(a) Office-31 dataset

(b) Office-Home dataset

Figure 5.4: Speed up achieved by combining d-SNE with ORIENT

dataset consists of 10812 images of 65 object categories from four different domains: Art (A), Clipart (C), Product (P), and Real-world (R). Here, the target data (D^t) consists of about 20% of the images from the target domain. We use $L = 20$, that is, we sample the subset after every 20 epochs and use $b = 0.3\%$, that is, we sample 30% of the source domain for training.

To answer the first question, we use ORIENT to train a ResNet50 architecture using the stochastic gradient descent (SGD) algorithm with the momentum of 0.9 and the weight



(a.1) A → C (a.2) A → P (a.3) A → R

(b.1) C → A (b.2) C → P (b.3) C → R

(c.1) P → A (c.2) P → C (c.3) P → R

(d.1) R → A (d.2) R → C (d.3) R → P

Figure 5.5: Convergence curves on Office-Home for domain adaptation across Art (A), Clipart (C), Product (P), and Real World (R). The x-axis shows training time (hours), and the y-axis shows target-domain accuracy.

decay ratio of $5e4$. We compare our approach against three baselines: **Full**—training on the source data and the target data, i.e. $D^s \cup D^t$; **Random**—a random subset of the source dataset and the target set, i.e. $R \cup D^t, R = \text{RandomSubset}(D^s)$; and **CRAIG**—a coreset of the source dataset, i.e coreset(D^t), without any information of the target dataset. We use standard categorical cross-entropy loss function (for multi-class classification) and five different instantiations of the submodular mutual information function: Facility Location Mutual Information (**ORIENT (FLMI)**), Graph Cut Mutual Information (**ORIENT (GCMI)**), Log Determinant Mutual Information (**ORIENT (LDMI)**), GLISTER (**ORIENT (G)**), and GradMatch (**ORIENT (GM)**).

Figure 5.2 presents the scatter plot of the speed up against the prediction accuracy of Office-31 dataset using cross-entropy loss.. In all our experiments, we use 0.3 fraction of the source data as the subset and see $> 2.5\times$ speed up. Figure 5.5 presents the convergence curves of the Office-Home dataset using cross-entropy loss. It is evident from these charts that our proposed approach ORIENT can achieve better or comparable performance to the Full training in significantly less amount of time, indicating better efficiency. This analysis helps us answer our Q1. ORIENT reduces the training time substantially, the speed up achieved is reciprocal to the fraction of the subset used, without trading the performance. Rather, in some combinations, we observe that ORIENT outperforms Full.

Figure 5.4 presents the bar plots of the speed up when ORIENT(FLMI) is used in conjunction with d-SNE loss function on Office-31 and Office-Home datasets. Here, we see a consistent $3\times$ speed up as compared to full model training with d-SNE loss. These bar plots demonstrate that our proposed approach ORIENT can augment existing domain adaptation approaches and substantially reduce the training times. Additionally, Tables 5.2 and 5.3 present the test accuracy while using d-SNE and CCSA loss in conjunction with ORIENT(FLMI) on Office-31 and Office-Home datasets, respectively. It's evident that augmenting existing domain adaptation approaches with ORIENT achieves better or

comparable performance to the Full training. These two observations help us answer our Q2 in the affirmative.

Figure 5.3 presents the GradCam (Selvaraju et al., 2017) class-activation maps of trained models on the Office-Home dataset ($P \rightarrow R$ setting) using d-SNE loss for both Full and ORIENT (FLMI). These activation maps show that the model trained with the ORIENT framework learned better class discrimination features than the model trained with Full. This might explain why the ORIENT framework performs better sometimes than Full.

Comparison of different SMI functions: Domain adaptation results on Office31 (5.2) and OfficeHome (5.5) datasets show that ORIENT(GCMI) performs suboptimally compared to ORIENT using other SMI functions in terms of target domain accuracy. Although ORIENT(LDMI) achieves reasonable target domain accuracy, it is computationally expensive and does not achieve the best performance-speedup trade-off. In comparison to the ORIENT using remaining SMI functions, i.e., ORIENT(GM), ORIENT(G), and ORIENT(FLMI), ORIENT(FLMI) consistently achieves the best performance versus speed-up trade-off.

To summarize, our experiments on two adaptation datasets suggest that our proposed approach ORIENT can substantially reduce the training time while maintaining comparable performance to training on complete source data. Additionally, our experiments using d-SNE and CCSA loss functions suggest that ORIENT can be used in conjunction with existing SDA methods to achieve significant speed ups in training time, while maintaining or improving the prediction accuracy. In particular, two instantiations of our proposed approach, ORIENT(FLMI) and ORIENT(GM) consistently achieve comparable or better performance compared to Full training while being $\sim 3\times$ faster to train.

5.4 Discussion

We introduce ORIENT, a subset selection framework based on SMI functions for supervised domain adaptation. The submodularity of SMI functions allows us to use scalable greedy algorithms to select the data subsets efficiently. In addition, we demonstrate how ORIENT is a unified framework that integrates previous approaches based on subset selection for robust learning. Empirically, we show that ORIENT is very effective for SDA. Specifically, it achieves $\sim 3\times$ speed up over existing SDA approaches like d-SNE and CCSA while achieving comparable or better performance. Our findings confirm that ORIENT has a significant social impact by making existing SDA algorithms significantly faster and more energy-efficient, reducing CO2 emissions and energy consumption incurred during training, thus contributing to a Green AI (Schwartz et al., 2020). The main limitation of our work is that although ORIENT significantly reduces the training time, it requires more memory to store the similarity kernel required for subset selection. This makes running ORIENT harder on devices with low memory.

CHAPTER 6

HUMAN-ALLIED LEARNING OF PROBABILISTIC CIRCUITS

The three tenets that organize this dissertation – effectiveness, efficiency, and explainability – are not binary properties that a piece of work either has or lacks. They are better understood as continuous dimensions along which any contribution can be situated to varying degrees. The preceding chapters each engage primarily with one dimension: Chapters 3 and 4 with explainability, Chapter 5 with computational efficiency. This chapter comes closest among the primary contributions to moving along all three simultaneously, and is in that sense the natural capstone of the dissertation.

The primary tenet is effectiveness. By incorporating domain knowledge as differentiable constraints into the parameter learning of probabilistic circuits, the framework proposed here consistently improves model performance in the data-scarce and noisy settings that motivate this dissertation. The connection to efficiency is secondary but empirically grounded: domain knowledge functions as a strong inductive bias, and the experimental results suggest that in the right scenario a single well-chosen constraint can compensate for a meaningful reduction in training data. This is sample efficiency not by algorithmic design but as an emergent property of knowledge-intensive learning, and it is offered here as an interpretation of the results rather than a primary claim. The connection to explainability is more modest still, but worth naming. The framework does not generate interpretable representations in the manner of Chapters 3 and 4. What it does offer is *verifiability*: a domain expert who provides a constraint can subsequently confirm whether the learned model respects it, which is a meaningful, if partial, step toward the kind of trust that deployment in safety-critical settings requires. In a domain such as clinical medicine, a model that provably adheres to an expert's stated belief about the relationship between BMI and gestational diabetes risk is more *auditable* - and therefore more deployable - than one that merely performs well on held-out data.

The chapter is also the most substantial primary contribution of the dissertation in scope. It formalizes seven distinct constraint types, several of which subsume the knowledge structures surfaced by the preceding explainability chapters as special cases, and demonstrates the framework across two PC architectures, multiple synthetic and benchmark datasets, and the nuMoM2b clinical study. The two chapters that follow present collaborative explorations that extend the themes of tractability and domain knowledge into the causal setting, complementing the associational perspective that has been the primary focus of this dissertation.

6.1 Introduction

Purely data-driven PCs often struggle in data-scarce and noisy domains, exhibiting overfitting and reduced generalizability. While traditional techniques like regularization can address overfitting to some extent, they often do so by introducing strong and sometimes arbitrary assumptions about the data distribution (Domingos, 1999). This can limit the model's ability to capture the true underlying relationships between variables.

Incorporating domain knowledge and expert insights into PC learning offers a compelling solution to these problems. Domain experts can provide invaluable insights that go beyond the raw data, grounded in experience and contextual understanding. This knowledge can not only mitigate the risk of overfitting without sacrificing expressive power but also tailor models to specific domain constraints. For instance, fairness and privileged information might be crucial factors in real-world decision-making, requiring models to incorporate such constraints. Knowledge-intensive learning strategies have demonstrably enhanced both discriminative and generative models in various contexts (Kokel et al., 2020a; Odom et al., 2015; Altendorf et al., 2005; de Campos et al., 2008; Yang and Natarajan, 2013a; Mathur et al., 2023). These models perform more reliably in scenarios with limited or suboptimal data. However, the concept of integrating human expertise and domain knowledge remains *relatively unexplored in the context of the general class of PCs.*

This chapter addresses this problem by introducing a framework for incorporating domain knowledge into PC learning. We first develop a unified framework that allows encoding *various types of knowledge* as probabilistic domain constraints. A key aspect of our work is that we can model any constraint **as long as its violation can be measured by a differentiable function**. We then demonstrate how **some** of these constraints, including equality constraints for generalization and privileged information, and inequality constraints for handling class imbalance, monotonicity, and variable interactions, can be seamlessly integrated into PC learning. Our method can be interpreted as **frustratingly easy** as with Shi et al. (2021), however, it works well and is generalizable, precisely why simpler learning methods should be favored over unnecessarily complex ones (Rudin et al., 2024; Rudin, 2019). This approach makes PCs more adaptable to real-world scenarios with limited data and specific modeling requirements. The key tenets of our framework are (1) Expressivity - allowing incorporation of a wide variety of well-known domain knowledge, (2) Effectiveness - consistently improving model performance by leveraging the domain knowledge and (3) Simplicity - allowing domain experts to easily quantify their domain knowledge through equality and inequality constraints.

We make the following key contributions:(1) We propose a unified framework that subsumes several types of domain knowledge and allows encoding them as domain constraints and demonstrate six particular instantiations of these constraints. (2) We propose an augmented objective function that effectively incorporates these domain constraints and can be optimized through generic parameter learning algorithms for PCs. (3) We experimentally validate the added utility of our approach on several benchmark and real-world scenarios where data is limited but knowledge is abundant.

6.2 Learning PCs with Domain Knowledge

We aim to solve the following problem:

Given: Dataset $\mathcal{D}$ over random variables $\mathbf{X}$, and a specification of domain knowledge $\mathcal{K}$.

To Do: Learn a probabilistic circuit $\mathcal{M}$ that accurately models $P(\mathbf{X})$.

Mathematically, the above problem can be expressed as the following constrained optimization:

$$\arg\max_{\mathcal{M}} \mathcal{L}(\mathcal{M}, \mathcal{D}) \text{ s.t. } \mathcal{M} \text{ does not violate } \mathcal{K} \tag{6.1}$$

where $\mathcal{L}(\mathcal{M}, \mathcal{D})$ is the log-likelihood of $\mathcal{D}$ with respect to $\mathcal{D}$ and is given by the following equation

$$\mathcal{L}(\mathcal{M}, \mathcal{D}) = \sum_{\mathbf{x} \in \mathcal{D}} \log P_{\mathcal{M}}(\mathbf{X} = \mathbf{x}) \tag{6.2}$$

This formulation reduces to standard maximum likelihood estimation when there are no domain constraints ($\mathcal{K} = \emptyset$,). Similarly, including a simple constraint on model complexity ($\mathcal{K}$ ="$\mathcal{M}$ is not too complex.") recovers maximum likelihood estimation with a basic regularization term. We first present an encoding scheme to encode commonly used forms of knowledge as constraints on the PC and then we present an algorithm that learns a PC using these constraints.

6.2.1 Encoding Knowledge as Constraints

We now demonstrate that commonly used forms of knowledge can be represented as either equality or inequality constraints on marginal and conditional queries on the PC. Specifically, we consider generalization, monotonicity, context-specific independence, class imbalance, synergy, and privileged information. We formally define equality and inequality constraints as follows:

Definition 6.2.1. An equality constraint $\langle f, g, \mathcal{S} \rangle$ states that $f(\mathbf{x}) = g(\mathbf{x}') \forall \mathbf{x}, \mathbf{x}' \in \mathcal{S}$, where $f(\mathbf{x})$ and $g(\mathbf{x})$ are tractable, differentiable functions of the PC and $\mathcal{S} \subseteq \mathbf{X}^2$ is a set of pairs of data points.

Definition 6.2.2. An inequality constraint $\langle f, g, \mathcal{S} \rangle$ states that $f(\mathbf{x}) > g(\mathbf{x}') \forall \mathbf{x}, \mathbf{x}' \in \mathcal{S}$, where $f(\mathbf{x})$ and $g(\mathbf{x})$ are tractable, differentiable functions of the PC and $\mathcal{S} \subseteq \mathbf{X}^2$ is a set of pairs of data points.

We use the case of learning a PC to model the risk of Gestational Diabetes (GD) using data from a multi-center clinical study as a running example. Ideally, such a PC should capture the relationships between various risk factors such as age, race, BMI, family history, genetic predisposition, and exercise level, by modeling their joint distribution. The following forms of domain knowledge can be used for learning:

1. **Generalization constraints (GC).** These constraints capture inherent symmetries in the data distribution, such as exchangeability (Lüdtke et al., 2022) and permutation invariance(Zaheer et al., 2017). For instance, subjects from the same center might have similar distributions due to study design. We can express this as:

$$P(\mathbf{x}) = P(\mathbf{x}'), \quad \forall (\mathbf{x}, \mathbf{x}') \in \mathcal{D}^2 \text{ s.t. } sim(\mathbf{x}, \mathbf{x}')$$

   where $sim(\mathbf{x}, \mathbf{x}')$ indicates if $\mathbf{x}$ and $\mathbf{x}'$ are similar based on predefined criteria (e.g., belonging to the same center).

2. **Context-Specific Independence relations (CSI).** These relations are a generalization of conditional independence relationships between variables. For example, BMI might be independent of age only for patients with GD. We can represent this as:

$$P(x_i \mid \mathbf{x}_k) = P(x_i \mid x_j, \mathbf{x}_k), \quad \forall \mathbf{x} \in \text{Dom}(\mathbf{X}) \text{ s.t. } \mathbf{x}_k = c$$

   where $\mathbf{x}_k = c$ denotes a specific value assignment to variables in $\mathbf{X}_k$, defining the context (e.g., having GD) where independence between X_i and X_j holds.

3. **Preference constraints (PF)**. Medical experts might provide a set of probabilistic logical conditions indicative of high GD risk. We denote this set as $R = \{(r_1, p_1), \ldots, (r_M, p_M)\}$,

where each pair (r, p) consists of a logical condition r over variables $\mathbf{X}$ and its associated probability p. These constraints can be expressed as

$$P(x_i = 1 \mid \mathbf{x}_{-i}) = p \quad \forall \mathbf{x} \models r,\ \forall (r, p) \in R \tag{6.3}$$

4. **Class-imbalance tradeoff (CT)** In GD screening, minimizing false negatives is crucial since patients predicted as low risk might not undergo further clinical testing. We can express such false negative constraints as

$$t_0 > P(X_i = 0 \mid \mathbf{x}_{-i}), \quad \forall \mathbf{x} \in \mathcal{D} \text{ s.t. } x_i = 1$$

where t_0 is the threshold such that $P(X_i = 0 \mid x_{-i}) > t_0$ is predicted as negative. These constraints ensure that the model is cautious in predicting a negative outcome, thus reducing the risk of false negatives. Constraints to minimize false positives can be formulated similarly.

5. **Monotonic Influence Statements (MISs)** The statements express positive or negative monotonic relationships between pairs of variables. For example, the risk of GD might increase with an increase in age. We can represent positive monotonic influence of X_j on X_i ($X_j \prec^{M+} X_i$) as:

$$P(X_i \leq x_i \mid x_j) > P(X_i \leq x_i' \mid x_j')$$
$$\forall \mathbf{x}, \mathbf{x}' \text{ s.t. } x_i = x_i',\ x_j' > x_j$$

(6) Synergistic influence statements can be encoded similar to monotonicities while (7) Privileged information can be encoded similar to CSIs. We defer the details about both of these constraints to the supplementary material.

6.3 Defining the Penalty Function

Having demonstrated that **seven commonly used forms of knowledge** can be represented as equality or inequality constraints on marginal and conditional queries on the PC, we

now present a learning algorithm that uses these constraints to learn PCs. To formalize the constrained optimization problem, we define the notion of a *penalty function* that quantifies the severity of domain knowledge violation in the distribution induced by the PC.

Definition 6.3.1. A function $\zeta : M \mapsto \mathbb{R}_0^+$ is a penalty function corresponding to domain knowledge $\mathcal{K}$ if it maps a PC $\mathcal{M} \in M$ to a non-negative real number $\zeta(\mathcal{M})$ that quantifies the extent of the violation of knowledge $\mathcal{K}$ such that

1. $\zeta(\mathcal{M}) = 0 \iff \mathcal{M}$ does not violate $\mathcal{K}$.
2. $\zeta(\mathcal{M}) < \zeta(\mathcal{M}') \iff \mathcal{M}$ violates $\mathcal{K}$ to a lesser extent than $\mathcal{M}'$.
3. The total violation in a model $\mathcal{M}$ due to two penalty functions ζ and ζ' is $\zeta(\mathcal{M}) + \zeta'(\mathcal{M})$

Using this notation, we can rewrite the optimization problem in equation (6.1) as

$$\arg\max_{\mathcal{M}} \mathcal{L}(\mathcal{M}, \mathcal{D}) \text{ s.t. } \zeta(\mathcal{M}) = 0 \tag{6.4}$$

where ζ measures the model's deviation from domain knowledge $\mathcal{K}$. For computational reasons, we focus on learning the model parameters for a PC with a given structure ($\mathcal{G}$). The optimization problem becomes finding PC parameters (θ) that maximize the likelihood while adhering to the domain knowledge:

$$\arg\max_{\theta} \mathcal{L}(\langle \mathcal{G}, \theta \rangle, \mathcal{D}) \text{ s.t. } \zeta(\langle \mathcal{G}, \theta \rangle) = 0 \tag{6.5}$$

However, since domain knowledge elicited from experts might not be perfectly consistent or fully compatible with the given PC structure, equation (6.4) might not have any feasible solution. We address this by using the penalty method to find a solution closest to the feasible region (Luenberger and Ye, 2016; Mathur et al., 2023). Algorithm 1 presents the parameter learning procedure, which solves a series of optimizations of the form

$$\theta_t^* = \arg\max_{\theta} \mathcal{L}(\langle \mathcal{G}, \theta \rangle, \mathcal{D}) - \lambda_t \zeta(\langle \mathcal{G}, \theta \rangle) \tag{6.6}$$

where θ_t^* is the solution of the tth optimization and λ_t is the penalty weight in the tth optimization. The initial value of the penalty weight, λ_0 is set to 0 (corresponding to the maximum likelihood solution), and the value of λ_t for each subsequent optimization problem is increased using the penalty control parameter γ and is set to γ^{t-1}. Each optimization where $t > 0$ is initialized with the solution from the previous problem θ_{t-1}. Note that while the objective function appears simple (as a function of likelihood with penalty), this has been widely used for structure learning in graphical models (Schwarz, 1978), and more recently, even in domain adaptation. Our empirical evaluation is yet-another-proof that one does not necessarily need complex objective functions if the simpler ones are both effective and by definition, efficient.

6.4 Experimental Evaluation

Our framework is generic and agnostic to the type of PC used. Thus, to empirically validate the effectiveness of incorporating domain constraints we consider two different instantiations of deep PCs - (i) *RatSPN* (Peharz et al., 2020a) and (ii) *EinsumNet* (Peharz et al., 2020). We implement both models with and without domain constraints for a comparative analysis. We design experiments over various datasets and scenarios to answer the following questions empirically:

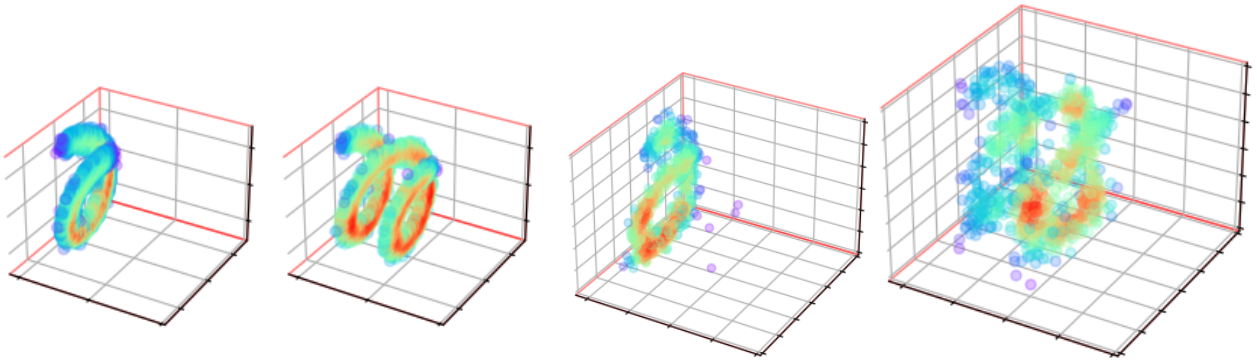
(**Q1**) Can domain knowledge be incorporated faithfully into the parameter learning of PCs?

(**Q2**) Does incorporating knowledge improve the generalization performance of PCs?

(**Q3**) Can the proposed framework exploit domain knowledge in heterogenous forms to learn more accurate PCs?

(**Q4**) How sensitive is the approach to the hyperparameters governing the size of domain sets and penalty weight? Further, is it robust to noisy or redundant advice?

(**Q5**) Does the proposed method learn more accurate models on real-world data?



(a) Train (b) Test (c) *EinsumNet* (d) *EinsumNet+GC*

Figure 6.1: **3D Helix dataset**: Visualization of the 3D Helix dataset. The train dataset (a) consists of a single helix of length 2π with Gaussian noise added to it. The test (b) dataset consists of a helix of length 4π with Gaussian noise added to it. The two figures on the right show 1000 samples generated from *EinsumNet* without (c) and with (d) incorporating Generalization Constraint (*GC*).

6.4.1 Experimental Setup

To answer these questions, we compare our knowledge-intensive learning method to purely data-driven approaches. We defer further implementation details to the supplementary material[1]. We used three types of data sets for our experiments: synthetic, benchmark, and real-world clinical data.

Synthetic data sets We used two types of synthetic data sets: four Bayesian Network (BN)-based datasets and a manifold-based data set. We derived data sets from 4 BNs: Earthquake (Korb and Nicholson, 2010a), Asia (Lauritzen, 1988), Sachs (Sachs et al., 2005), and Survey (Scutari and Denis, 2021). For each BN, we constructed a data set by sampling 100 data points. To simulate perfect domain knowledge, we then extracted two conditional independence relations from each BN. These relations were then translated into CSI constraints.

We used a 3D Helix Manifold to construct the second type of synthetic data set (Figure 6.1) (Sidheekh et al., 2022). This dataset presents difficulties due to its intricate spiral

[1]The code is available at `https://github.com/athresh/unified-constraints-pc/`

structure. However, its inherent symmetry can be exploited to define a generalization constraint (GC). We encoded the helical symmetry by specifying that points separated by 2π along the x-axis and lying on a helix manifold of unit radius have similar density. We use 100 pairs of datapoints of the form $((x, sin(x), cos(x)), (x + 2\pi, sin(x + 2\pi), cos(x + 2\pi)))$.

Benchmark data sets We used two kinds of benchmark data sets – four UCI benchmark data sets and two point cloud-based data sets. We used 4 data sets from the UCI Machine Learning repository, namely, breast-cancer, diabetes, thyroid, and heart-disease. We used the monotonic influence statements that were used in prior work by Yang and Natarajan (2013a) as domain knowledge for these data sets.

We used the MNIST (Deng, 2012) and fashion-MNIST image datasets to construct point cloud representations following Zhang et al. (2019) (Zhang et al., 2019). We transformed each image into a set-based representation by sampling the locations of a small set of foreground pixels. The resulting 2D point cloud representation inherently exhibits permutation invariance – a symmetry difficult to model even for advanced generative models such as Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) (Li et al., 2019; Kim et al., 2021). We refer to these datasets as full set data (e.g., Set-MNIST-Full). Additionally, to simulate a data-scarce scenario, we further divided the MNIST dataset into two subsets: (i) Set-MNIST-Even comprising only even digits and (ii) Set-MNIST-Odd comprising only odd digits. We encoded permutation invariance as a GC by specifying the domain set criteria as $sim(\mathbf{x}, \mathbf{x}') = \mathbb{I}[\mathbf{x}' = \pi(\mathbf{x})]$, where $\pi(\mathbf{x})$ represents a permutation of $\mathbf{x}$. In practice, we defined the GC domain set by taking $\gamma_{size} = 2$ permutations for each sample in the dataset.

Real-World Clinical Data To evaluate the performance of our framework on real-world data, we used data from the Nulliparous Pregnancy Outcomes Study: Monitoring Mothers-to-Be (nuMoM2b, (Haas et al., 2015)). We considered two subsets of the data, numom2b-a focusing on a single Adverse Pregnancy Outcome (APO) namely Gestational diabetes (GD), and numom2b-b focusing on three different APOs namely New Hypertension (NewHTN), Pre-eclampsia (PreEc), and Pre-term Birth (PTB).

Table 6.1: **Quantitative Evaluation:** Mean test log-likelihood of *RatSPN* trained with and without constraints on the Bayesian Network (BN), UCI Benchmark and Real-World (RW) datasets, ± standard deviation across 3 independent trials.

		RatSPN	*RatSPN+Constraint*
BN	asia	-483.3 ± 4.1	$\mathbf{-313.2 \pm 3.9}$
	sachs	-1097.5 ± 8.8	$\mathbf{-861.2 \pm 8.7}$
	survey	-611.7 ± 7.2	$\mathbf{-476.6 \pm 6.6}$
	earthquake	-272.0 ± 2.4	$\mathbf{-121.8 \pm 2.1}$
UCI	breast-cancer	-2110.8 ± 15.6	$\mathbf{-1271.5 \pm 14.6}$
	diabetes	-7010.3 ± 31.0	$\mathbf{-5070.3 \pm 481.8}$
	thyroid	-351.5 ± 6.1	$\mathbf{-200.5 \pm 23.2}$
	heart-disease	-931.7 ± 15.0	$\mathbf{-739.8 \pm 7.2}$
RW	numom2b-a	-14573.9 ± 69.9	$\mathbf{-7288.2 \pm 1.6}$

The first data set consists of 3,657 subjects of white, European ancestry that have data for GD and its seven risk factors: Age, BMI at the start of pregnancy, presence of Polycystic Ovary Syndrome (PCOS), physical activity (METs), presence of High Blood Pressure (HiBP), family history of diabetes (Hist), and a polygenic risk score (PRS) genetic predisposition to diabetes. All risk factors except physical activity (METs) are known to positively monotonically influence the risk of GDM, while physical activity is known to have a negative monotonic influence. The second data set consists of 9,368 subjects of diverse racial and ethnic backgrounds that have data for three APOs: NewHTN, PreEc, PTB, and four risk factors: BMI, Age, Hist, and HiBP.

6.4.2 Results

(**Q1**) **Faithful Incorporation of Domain Constraints.**

Table 6.1 (rows 1-3) presents the test log-likelihood scores for RatSPN models trained on the BN-based data sets with and without domain knowledge encoded as CSIs. Notably, the degree of constraint violation for RatSPN models incorporating domain constraints remained consistently below 0.0001 across all datasets. This confirms our

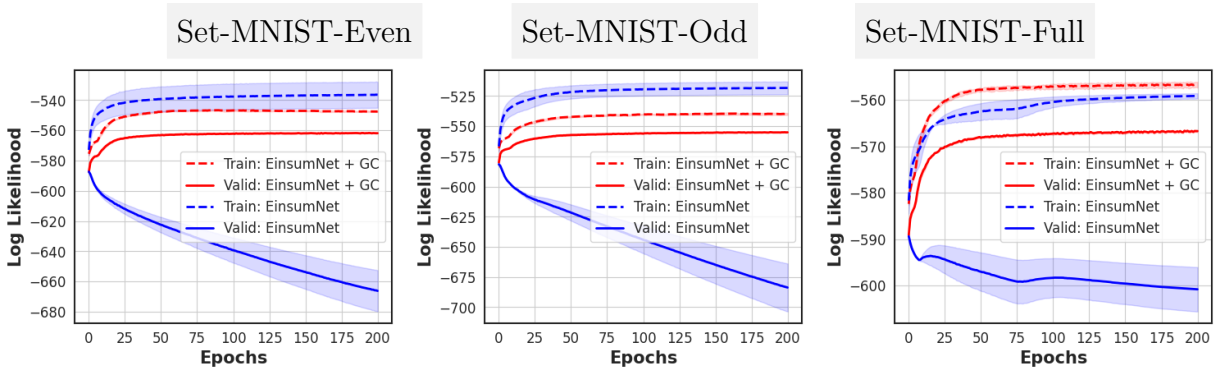


Figure 6.2: **Learning Curves:** Mean train and validation log-likelihoods of *EinsumNet* trained with (in red) and without (in blue) incorporating Generalization Constraint (*GC*) on the three Set-MNIST datasets, across training epochs. The shaded regions denote the standard deviation across 3 independent trials.

framework's ability to faithfully integrate valid domain knowledge. Moreover, RatSPN models trained with domain constraints performed better than those trained solely on data.

(**Q2**) **Improvement in Generalization Performance.**

We studied the effect on generalization performance due to two kinds of domain knowledge – generalization constraints (GC) and monotonic influence statements.

We used the 3D Helix Manifold dataset and the point cloud data sets to evaluate the effect of GC. Table 6.2 shows the mean test log-likelihoods of the PCs trained on the 3D Helix Manifold data with and without domain knowledge encoded as GC. The baselines, EinsumNet and RatSPN, simply append the GC data points to the training data. In contrast, our proposed approach (EinsumNet+GC and RatSPN+GC) treats the GC as a constraint during training.

We observed significant performance improvements for both models when incorporating the GC. This suggests that the GC enables the models to exploit the inherent symmetries within the data, allowing them to generalize to unseen regions with similar structure. Notably, achieving this improved performance requires only a small number of samples

Table 6.2: **Quantitative Evaluation**: Mean test log-likelihood of *RatSPN* and *EinsumNet* trained with and without the Generalization Constraint (GC) on the Helix and Set-MNIST datasets, $\pm$ standard deviation across 3 independent trials.

	Helix	Set-MNIST-Even	Set-MNIST-Odd	Set-MNIST-Full	Set-Fashion-MNIST
RatSPN	-7.1 ± 1.8	-657.7 ± 15.9	-682.3 ± 16.7	-599.3 ± 3.2	-1495.0 ± 3.5
RatSPN+GC	$\mathbf{-3.6 \pm 0.1}$	$\mathbf{-562.2 \pm 0.4}$	$\mathbf{-555.1 \pm 0.4}$	$\mathbf{-566.1 \pm 0.1}$	$\mathbf{-1236.9 \pm 0.4}$
EinsumNet	-5.1 ± 0.2	-666.2 ± 13.7	-683.7 ± 19.9	-600.8 ± 4.8	-1413.9 ± 20.8
EinsumNet+GC	$\mathbf{-2.7 \pm 0.1}$	$\mathbf{-561.9 \pm 0.2}$	$\mathbf{-555.1 \pm 0.1}$	$\mathbf{-566.7 \pm 0.3}$	$\mathbf{-1235.8 \pm 0.6}$

Table 6.3: The test log-likelihoods for the RatSPN learned from data, with one form of knowledge and with two forms of knowledge averaged over 3 independent trials

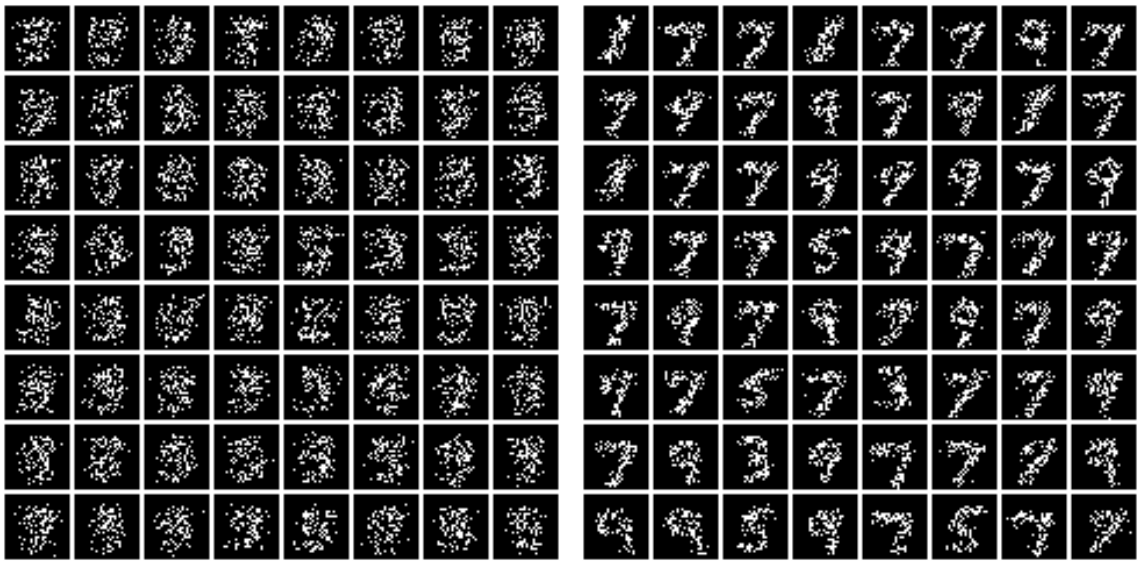
	RatSPN	+CSI	+CSI+MIS
earthquake	-272.0 ± 2.4	-137.7 ± 4.7	$\mathbf{-106.1 \pm 1.1}$
survey	-611.7 ± 7.2	-523.5 ± 4.3	$\mathbf{-470.9 \pm 6.6}$
asia	-483.3 ± 4.1	-320.5 ± 9.9	$\mathbf{-284.7 \pm 6.4}$
numom2b-b	-18281.2 ± 218.8	-15122.9 ± 201.7	$\mathbf{-14758.1 \pm 60.3}$

from the constraint's domain set. Additionally, including these data points directly in the training data without structured GC integration did not lead to better generalization (Table 6.2). Figure 6.1 visually confirms this, as samples generated by EinsumNet+GC more closely resemble the test data distribution compared to those from EinsumNet.

Similar results were observed for set-based image datasets (MNIST and fashion-MNIST) (Table 6.2). Incorporating GC significantly improved performance for both models. This is further supported by the quality of samples generated with and without GC (Figure 6.3). The impact of GC on model training is also evident in the learning curves for Set-MNIST datasets (Figure 6.2). In the absence of GC (particularly for Set-MNIST-Even and Set-MNIST-Odd with fewer data points), the model overfits as shown by the increasing training log-likelihood and decreasing validation log-likelihood. Conversely, incorporating GC enables the model to leverage symmetries and achieve better generalization. Similar visualizations for other datasets are provided in the supplementary material.

To evaluate the effect of monotonic influence statements, we used four UCI benchmark datasets. Table 6.1 shows the improvement in the test log-likelihood of RatSPN trained on UCI data sets when incorporating domain constraints.

(**Q3**) **Incorporation of different types of knowledge**. We studied the effectiveness of our unified framework in incorporating domain knowledge presented in heterogeneous forms.



(a) *EinsumNet* (b) *EinsumNet+GC*

Figure 6.3: **Qualitative Evaluation**: Visualization of randomly sampled datapoints generated by *EinsumNet* trained with and without the Generalization Constraint (*GC*) on Set-MNIST-Odd.

We considered knowledge in the form of CSI and MIS in 3 BN-based domains and 1 clinical domain. Table 6.3 presents the test log-likelihood scores of RatSPN learned from each of the 4 domains without knowledge (RatSPN), with a single CI encoded as CSIs (+CSI), and with the CSIs and a single MIS (+CSI+MIS). The models using both forms of knowledge perform better than models using CSIs.

(**Q4**) **Sensitivity and Robustness.** Our framework relies on two key hyperparameters: the size of the domain sets used for constraint evaluation (denoted by γ_{size}), and the penalty weight (λ) that balances constraint satisfaction with data-driven learning. In real-world applications, domain knowledge might be imperfect or noisy. We conducted ablation studies to assess the framework's sensitivity and robustness to these factors. γ_{size} is defined as the ratio of the domain set size for a constraint to the overall dataset size. To simulate noisy domain knowledge, we replaced a fraction of the domain set (γ_{noise}) with randomly sampled data points.

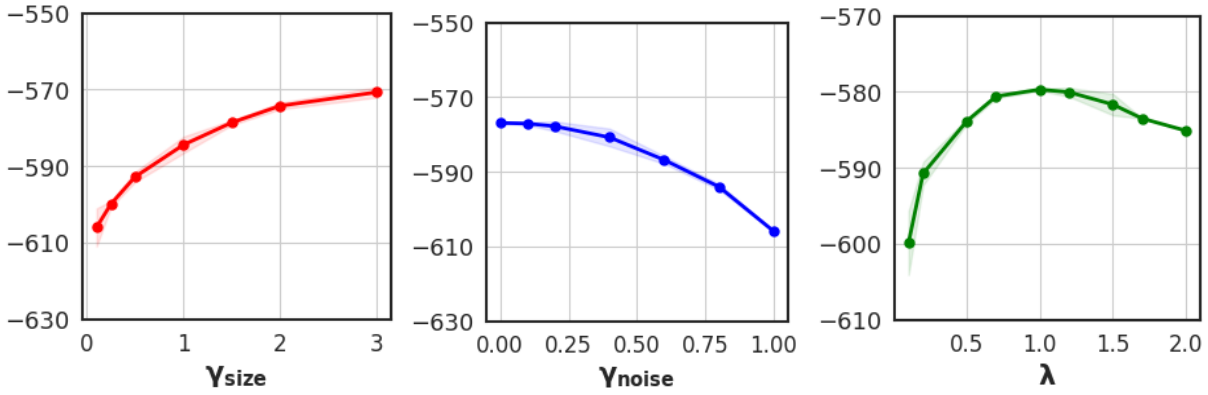
We investigated the impact of these hyperparameters on an EinsumNet model trained on the Set-MNIST-Even dataset with GC for 100 epochs. Varying γ_{size} (Figure 6.4a) showed that increasing the domain set size generally improves generalization performance, but this effect plateaus after a certain point (around $\gamma_{\text{size}} \approx 2$). Interestingly, varying γ_{noise} (Figure 6.4b) revealed that the model's performance remained relatively stable even with up to 40% noise in the constraints. This suggests the framework's robustness, as data compensates for some level of noise in the knowledge.

Finally, varying the penalty weight λ (Figure 6.4c) demonstrated that a very small λ leads to underutilization of the constraint, resulting in poor performance. Conversely, a very large λ overpowers the maximum likelihood training, disrupting the balance. The optimal λ appears to be around 1, striking a balance between data and constraints.

(**Q5**) **Performance on Real-World Data.** We evaluated the effectiveness of our framework on real-world data using the numom2b data sets. Tables 6.1 and 6.3 compare the test log-likelihood of *RatSPN* learned from the numom2b-a and numom2b-b data set with and without the knowledge in the form of MISs and CSIs. RatSPNs learned with knowledge achieved significantly higher test log-likelihood scores. This demonstrates the efficacy of our framework in exploiting domain knowledge to learn PCs from real-world datasets.

6.5 Discussion

We presented a general approach for incorporating domain knowledge as differentiable constraints into learning probabilistic models. It subsumes and extends several prior works on learning generative models (e.g., (Altendorf et al., 2005; Papantonis and Belle, 2021; Mathur et al., 2023)) and (probabilistic) discriminative models (e.g., (Kokel et al., 2020a)). Prior work by Papantonis and Belle (2021) focused specifically on knowledge encoded as equality



(a) Varying γ_{size} (b) Varying γ_{noise} (c) Varying λ

Figure 6.4: **Ablation Study:** Mean test log-likelihood of a *RatSPN* trained on the Set-MNIST-Even dataset, across varying (a) size of domain sets and (b) degrees of noise. Shaded regions represent the SD across 3 trials.

constraints, which is a special case within our broader framework. We generalize Mathur et al.(2023)'s approach of using monotonicity constraints to learn cutset networks (CNs, (Rahman et al., 2014a)) both in the forms of knowledge that can be captured and the model class, as CNs can be represented as deterministic PCs.

Altendorf et al. (2005) used *ceteris paribus*[2] monotonicity to learn the parameters of Bayesian Networks (BNs). Similar to CNs, BNs can be compiled into selective PCs (Peharz et al., 2014), making our framework applicable here as well. The KiGB algorithm (Kokel et al., 2020a) utilized soft monotonicity constraints for learning decision trees and decision trees can be converted into conditional SPNs (Shao et al., 2020). These methods can be seen as special cases of our framework.

Our framework is versatile enough to incorporate more complex forms of knowledge (e.g., (Odom et al., 2015; Xu et al., 2018)), enhancing its applicability compared to previously discussed methods. Odom et al. (2015) used relational preference information to learn gradient-boosted relational trees. While our framework is limited to learning PCs with

[2]Other parents of the influenced variable X_i are kept constant

propositional logical constraints, it can be extended to use relational preference advice over variables. For example, the relational advice that "*Hypertensive disorders of pregnancy lead to pre-term birth*" could be compiled to the propositional *preference constraint* that subjects with the presence of any of chronic hypertension, gestational hypertension, and preeclampsia should have a higher conditional probability of pre-term birth. Xu et al. (2018) use knowledge in the form of propositional logical formulas to learn deep neural networks using a semantic loss that computes the log probability of satisfying the formulas. Since this probability can be computed tractably in a PC, the semantic loss can be used to learn PCs using our framework.

As mentioned earlier, the simple yet effective objective function we propose in eq (6.6) has, in form, been successfully employed for tasks such as domain adaptation, domain generalization (Sun and Saenko, 2016; Shi et al., 2021), and multiple class novelty detection. Perera and Patel (2019) use a linear combination of cross entropy loss and membership loss as their objective function for out-of-distribution generalization. We observe that augmenting the loss function with differentiable linear constraints is effective, as shown in our experiments, easy to be specified by domain experts, and allows the incorporation of a wide variety of constraints. While more complex objective functions are conceivable, we propose this simple and effective formulation to allow easy adoption by domain experts. Integration of more complex forms of domain knowledge, handling structured data (relational data), as well as learning the structure of a PC in a knowledge-intensive manner are promising future directions.

CHAPTER 7

LEARNING INTERVENTIONAL DISTRIBUTIONS USING PROBABILISTIC CIRCUITS

The probabilistic queries introduced in Chapter 2 – evidential, marginal, and conditional – are all defined with respect to an observational distribution $P(\mathbf{X})$ learned from data. Every primary contribution in this dissertation operates at this level. Yet the background chapter also noted a fundamental limitation of observational queries: they cannot answer causal questions. In the nuMoM2b context, knowing that higher BMI is associated with increased gestational diabetes risk is an observational finding. Knowing what would happen to that risk if a patient's BMI were reduced through a clinical intervention is a categorically different question – one that sits at the second level of the Pearl Causal Hierarchy and cannot be answered from observational data alone, regardless of how faithfully the distribution has been estimated. The gap between these two questions is not merely philosophical; in clinical decision-making, where the goal is often to reason about the effects of interventions rather than to describe patterns in historical data, it has direct practical consequences. This chapter addresses that gap by extending the probabilistic circuit framework into the interventional setting.

The contribution is interventional sum-product networks (iSPNs), a model class that learns to answer $\mathcal{L}_2$ queries by conditioning on the structure of a mutilated causal graph. Where a standard SPN models the observational distribution $P(\mathbf{X})$, an iSPN models the family of interventional distributions $P(\mathbf{X} \mid do(\mathbf{X}_k = \mathbf{x}_k))$ for arbitrary interventions, doing so while retaining the tractable inference properties that make probabilistic circuits appealing. The result is a causal model that inherits the computational advantages of the SPN framework without requiring compilation to a Bayesian network, which prior work had shown to be problematic for causal reasoning.

This chapter presents collaborative work in which the author of this dissertation contributed to the framing, the experimental design, and the evaluation, with the core technical development and implementation led by co-authors. It is presented here as a complementary exploration rather than a primary contribution, extending the dissertation's central themes into a setting where the queries of interest require causal rather than associational reasoning.

7.1 Introduction

Identifying causal relationships between variables in observational data is one of the fundamental and well-studied problem in machine learning. There have been several great strides in causality (Granger, 1969; Pearl, 2009; Bareinboim and Pearl, 2016) over the years specifically characterized by efforts that focused on reasoning about interventions (Hagmayer et al., 2007; Dasgupta et al., 2019) and counterfactuals (Morgan and Winship, 2015; Oberst and Sontag, 2019).

The notion of causality has long been explored in the realm of probabilistic models (Oaksford and Chater, 2017; Beckers and Halpern, 2019) with a special focus on graphical models, called causal Bayesian networks (CBNs) (Heckerman et al., 1995; Neapolitan, 2004; Pearl, 1995; Acharya et al., 2018). CBNs have widely been applied to infer causal relationships in high-impact diverse applications such as disease progression (Koch et al., 2017), ecological risk assessment (Carriger and Barron, 2020) and more recently Covid-19 (Fenton et al., 2020; Feroze, 2020) to name a few. Although successful, classical CBN models are difficult to scale and also suffer from the problem of intractable inference. Recently, tractable probabilistic models such as probabilistic sentential decision diagrams (Kisa et al., 2014) and sum-product networks (Poon and Domingos, 2011) have emerged, which guarantee that conditional marginals can be computed in time linear in the size of the model. While weaving in the notion of interpretability, the computational view on probabilistic models allows one to exploit ideas from deep learning and can thus be very useful in modeling complex problems.

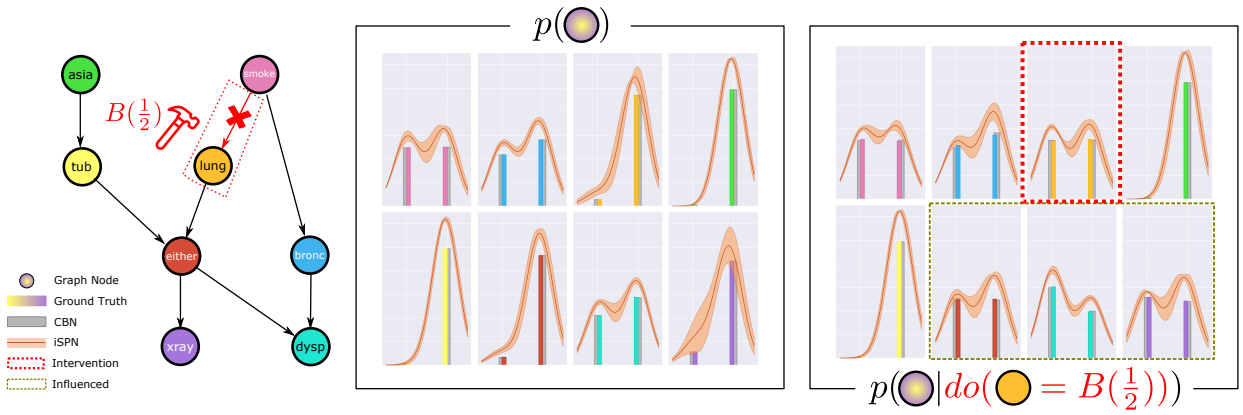


Figure 7.1: **Capturing interventional distributions using iSPN**. The interventional distributions for the ASIA data set using a causal Bayesian network (CBN, small-scale gold standard, gray bars) as well as an interventional SPN (iSPN) by intervening on *lung*. The iSPN is sensible to all the influences of the given intervention onto the system i.e., subsequent effects in the causal hierachy. (Best viewed in color.)

Recently, there has been an effort to take advantage of this tractability to reason for causality. Zhao et al. (2015b) showed how to compile back and forth between sum-product networks (SPNs) and Bayesian networks (BNs). Although this opened up a whole range of possibilities for tractable causal models, (Papantonis and Belle, 2020) argued that such conversion leads to degenerated BNs thereby rendering it ineffective for causal reasoning. For the considered compilation of SPNs to BNs, this is indeed the case since a bipartite graph between the hidden and observed variables loses the relationships between the actual variables. Thus, either a new compilation method for transforming between tractable and causal model or alternatively a method to condition the probabilistic models directly on the *do*-operator to obtain interventional distributions, $P(y|\, do(x))$, is being required.

Here, we consider the latter strategy and extend the idea of conditionally parameterizing SPNs (Shao et al., 2019) by conditioning on the *do*-operator while predicting the complete set of observed variables therefore capturing the effect of intervention(s). The resulting *interventional sum-product networks (iSPNs)* take advantage of both the expressivity, due to the neural network, and the tractability, due to the SPN in order to capture the interventional distributions faithfully. This shows that the dream of tractable causal models

is not insurmountable, since iSPNs are causal. (Pearl, 2019) defined a three-level causal hierarchy that separates association (purely statistical level 1) from intervention (level 2) and counterfactuals (level 3), and argued that the latter two levels involve causal inference. iSPNs are a modeling scheme for arbitrary interventional distributions. They thus belong to level 2 and, in turn, are causal. So, while SPNs are not universal function approximators and the use of gate functions turns them into universal approximators, we go one step ahead and make the first effort towards introducing causality to SPNs without the need for compilation to Bayesian networks, as the functional approximator subsumes the *do*-operator. Fig. 7.1 shows an example of the effectiveness of iSPNs to capture interventional distributions on the ASIA data set. Our extensive experiments against strong baselines demonstrate that our method is able to capture ideal interventional distributions.

To summarize, we make the following contributions:

1. We introduce iSPNs, the first method that applies the idea of tractable probabilistic models to causality without the need for compilation and that accordingly generate interventional distributions w.r.t. the provided causal structure.

2. We formulate the inductive bias necessary for turning conditional SPN into iSPN while taking advantage of the neural network modeling capacities within the gating nodes for for modeling interventional distributions while capturing all influences within the given intervention (i.e., consequences propagated through the structural hierarchy).

3. We show that, by construction, iSPNs can identify any interventional distribution permitted by the underlying structural causal model due to the inducted bias on the interface modalities in junction with the universal function approximation of the underlying gated SPN.

We proceed as follows. We start by reviewing the basic concepts required and related work, namely the tractable model class of sum-product networks and key concepts from

causality. Then we motivate using a newly curated causal model themed around personal health. Subsequently, we introduce iSPNs formally and prove that by construction they are capable of approximating any *do*-query (given corresponding data). Before concluding, we present our experimental evaluation in which we challenge iSPN in its density estimation and causal inference capabilities against various baselines. We make our code publicly available at: `https://github.com/zecevic-matej/iSPN`.

7.2 Background and Related Work

Let us briefly review the background on tractable probabilistic models and causal models used in subsequent sections for developing our new model class based on CSPNs that allow for identifying causal quantities i.e., interventional distributions.

Notation. We denote indices by lower-case letters, functions by the general form $g(\cdot)$, scalars or random variables interchangeably by upper-case letters, vectors, matrices and tensors with different boldface font $\mathbf{v}, \mathbf{V}, \mathsf{V}$ respectively, and probabilities of a set of random variables $\mathbf{X}$ as $p(\mathbf{X})$.

Causal Models. A Structural Causal Model (SCM) as defined by (Peters et al., 2017) is specified as $\mathfrak{C} := (\mathbf{S}, P_{\mathbf{N}})$ where $P_{\mathbf{N}}$ is a product distribution over noise variables and $\mathbf{S}$ is defined to be a set of d structural equations

$$X_i := f_i(\mathrm{pa}(X_i), N_i), \quad \text{where } i = 1, \ldots, d \tag{7.1}$$

with $\mathrm{pa}(X_i)$ representing the parents of X_i in graph $G(\mathfrak{C})$. An intervention on a SCM $\mathfrak{C}$ as defined in (7.1) occurs when (multiple) structural equations are being replaced through new non-parametric functions $\hat{f}(\widehat{\mathrm{pa}(X_i)}, \hat{N}_i)$ thus effectively creating an alternate SCM $\hat{\mathfrak{C}}$. Interventions are referred to as *imperfect* if $\widehat{\mathrm{pa}(X_i)} = \mathrm{pa}(X_i)$ and as *atomic* if $\hat{f} = a$ for $a \in \mathbb{R}$. An important property of interventions often referred to as "modularity" or "autonomy"[1]

[1]See Section 6.6 in (Peters et al., 2017).

states that interventions are fundamentally of local nature, formally

$$p^{\mathfrak{C}}(X_i \mid \mathrm{pa}(X_i)) = p^{\hat{\mathfrak{C}}}(X_i \mid \mathrm{pa}(X_i)) \ , \tag{7.2}$$

where the intervention of $\hat{\mathfrak{C}}$ occured on variable X_k opposed to X_i. This suggests that mechanisms remain invariant to changes in other mechanisms which implies that only information about the effective changes induced by the intervention need to be compensated for. An important consequence of autonomy is the truncated factorization

$$p(V) = \prod_{i \notin S} p(X_i \mid \mathrm{pa}(X_i)) \tag{7.3}$$

derived by (Pearl, 2009), which suggests that an intervention S introduces an independence of an intervened node X_i to its causal parents. Another important assumption in causality is that causal mechanisms do not change through intervention suggesting a notion of invariance to the cause-effect relations of variables which further implies an invariance to the origin of the mechanism i.e., whether it occurs naturally or through means of intervention (Pearl et al., 2016).

A SCM $\mathfrak{C}$ is capable of emitting various mathematical objects such as graph structure, statistical and causal quantities placing it at the heart of causal inference, rendering it applicable to machine learning applications in marketing (Hair Jr and Sarstedt, 2021)), healthcare (Bica et al., 2020)) and education (Hoiles and Schaar, 2016). A SCM induces a causal graph G, an observational/associational distribution $p^{\mathfrak{C}}$, can be intervened upon using the *do*-operator and thus generate interventional distributions $p^{\mathfrak{C};do(\dots)}$ and given some observations $\mathbf{v}$ can also be queried for interventions within a system with fixed noise terms amounting to counterfactual distributions $p^{\mathfrak{C}|\mathbf{V}=\mathbf{v};do(\dots)}$. To query for samples of a given SCM, the structural equations are being simulated sequentially following the underlying causal structure starting from independent, exogenous variables and then moving along the causal hierarchy of endogenous variables (i.e., following the causal descendants).

The work closest to our work is by (Brouillard et al., 2020) although it solves the different problem of causal discovery. Causality for machine learning has recently gained a lot of traction (Schölkopf, 2019) with the study of both interventions (Shanmugam et al., 2015) and counterfactuals (Kusner et al., 2017) gaining speed. For more details the readers are referred to (Zhang et al., 2018).

7.3 Interventional SPNs

Now we are ready to develop interventional SPNs (iSPNs). To this end, we re-introduce the importance of adaptability of models to interventional queries and present a newly curated synthetic data set to both motivate and validate the formalism of iSPNs that, through over-parametric extension of SPNs, allows them to adhere to causal quantities.

7.3.1 Adaptation to Causal Change

(Peters et al., 2017) motivated the necessity of causality via the adequate generalizability of predictive models. Specifically, consider a simple regression problem $f(a) = b$ with data vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^k$ that are strongly positively correlated in some given region, e.g. $(1 < a < \infty, 1 < b < \infty)$. Now a query is posed outside the data support, e.g. $(0, f(0))$. As argued by Peters et al., the underlying data generating processes can be an ambiguous causal process i.e., the data at hand can be explained by two different causal structures being either $A \rightarrow B$ or a common confounder with $A \leftarrow C \rightarrow B$.

Assuming the wrong causal structure or ignoring it altogether could be fatal, therefore, any form of generalization out of data support requires assumptions to be made about the underlying causal structure. We adopt this point of view and further argue that *ignoring causal change(s) in a system, i.e., the change of structural equation(s) underlying the system,*

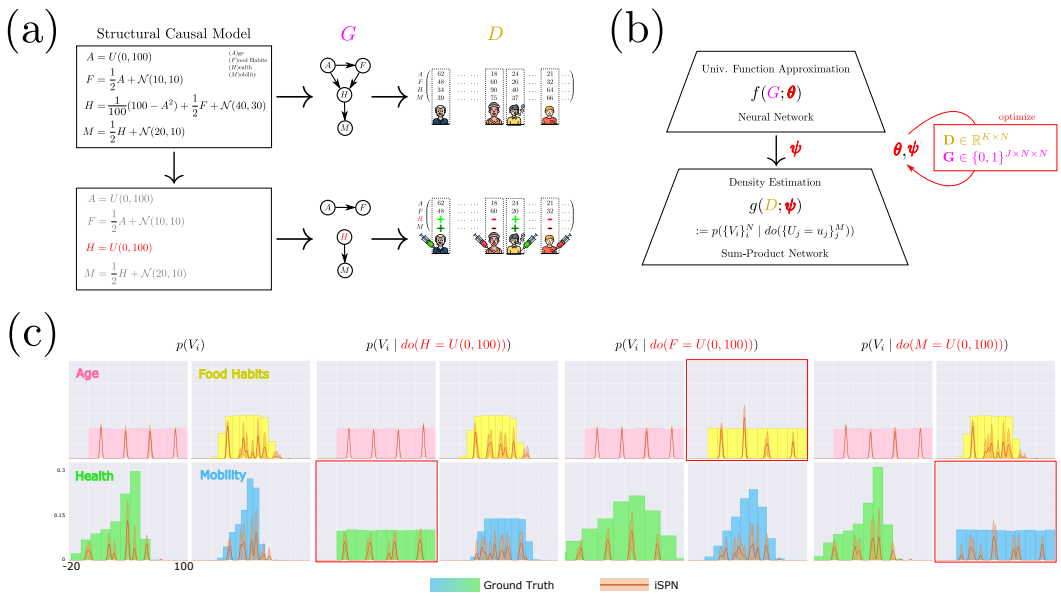


Figure 7.2: **An Overview of iSPN.** (a) The hidden process underlying the observable reality is modeled via a SCM that can be modified through interventions. The given SCM induces a causal graph and can generate data. An intervention can significantly alter the resulting data. (b) An over-parameterized density estimation framework using SPN is presented. The universal function approximator (here neural network) conditions on the mutilated causal graph and provides parameters to the SPN such that the given data's density can be modeled accordingly. The FA subsumes the *do*-operator. (c) Different causal queries are being presented. Furthermore, iSPN adapts to intervention-consequences.

can lead to a significant performance decrease and safety hazards[2]. Therefore, it is important to account for distributional changes present in the data due to experimental (thus interventional) settings.

Consequently, we consider the learning problem, where the given data samples have been generated by different interventions $do(\mathbf{U}_j = \mathbf{u}_j)$ in a common SCM $\mathfrak{C}$ while the induced mutilated causal graphs $G(\mathfrak{C}, do(\mathbf{U}_j = \mathbf{u}_j))$ are assumed to be known, such that the trained model is capable of at least inferring all involved causal distributions $p(V(\mathfrak{C}) \mid do(\mathbf{U}_j = \mathbf{u}_j))$ with V being the variables.

[2]This extended notion of performance degeneration through ignorance to the underlying causality is prioritized in this paper.

7.3.2 Data Generating Process

To both validate and demonstrate the expressivity of interventional SPNs in modeling arbitrary interventional distributions, we curate a new causal data set based on the SCM $\mathfrak{C}$ presented in Fig. 7.2(a), which we subsequently refer to as *Causal Health* data set. The SCM encompasses four different structural equations of the form $V_i = f(\text{pa}(V_i), N_i)$, where $\text{pa}(V_i)$ are the parents of variable V_i and N_i are the respective noise terms that form a factor distribution $P^{N_{1:N}}$ i.e., the N_i are jointly independent. Now, the SCM $\mathfrak{C}$ describes the causal relations of an individual's health and mobility attributes with respect to their age and nutrition.

Note that the Causal Health data set does not impose assumptions over the type of random variables or functional domains of the structural equations[3], which additionally constraints a learned model to adapt flexibly. While we generally do not restrict our method to any particular type of intervention, the following mainly considers perfect interventions as introduced in Sec. 7.2.

Perfect interventions fully remove the causal mechanism of the parents of a given node, which is consistent with the idea behind randomized controlled trials (RCTs) where the given intervention randomizes the given specific node, often referred to as gold standard in causality related literature. We consider the special case of uniform randomization, i.e., uniform across the domain of the given variable[4]. An intervention performed on one node immediately changes the population and, thus, has a major effect on the generating processes of subsequent causal mechanisms in the respective causal sequence of events.

To provide the reader with a concrete example of interventions within the causal health data set, consider the following: In concern of a virus infection the individuals of the Causal

[3]Assumptions on the func. form of structural eq. are crucial for identification (Tab. 7.1 (Peters et al., 2017))

[4]Note that for binary variables this amounts to Bernoulli $B(\frac{1}{2})$.

Health study should be vaccinated. The vaccine is expected to have side-effect(s), however, it has been poorly designed and has reached the population with the capability of completely changing the individual's health state. The observed changes do not show any form of pattern and are therefore assumed to be random. A young fit person could thus become sick, while an old person might feel better health wise. This sudden change will have an effect on the individual's mobility and also be independent of their age and nutrition. Such a scenario is mathematically being captured through $do(H = U(H))$ where $U(\cdot)$ is the uniform distribution over a given domain.

7.3.3 Introducing Interventional SPNs

After motivating both the importance and the occurrences of interventions within relevant systems, we now start introducing interventional SPNs (iSPNs).

Definition of iSPN. As motivated in Sec. 7.1, the usage of the compilation method from (Zhao et al., 2015b) for causal inference within SPN is arguably of degenerate[5] nature given the properties of the compilation method (Papantonis and Belle, 2020). While the results of (Papantonis and Belle, 2020) are arguably negative, there exists *yet no proof of non-existence of such a compilation method* and as the authors point out in their argument for future lines of research in this direction, a model class extension poses a viable candidate for overcoming the problems of using SPN for causal inference.

While agreeing on the latter aspect, we do not go the "compilation road" but extend the idea of conditional parameterization for SPN (Shao et al., 2019) by conditioning on a modified form of the *do*-operator introduced by (Pearl, 2009) while predicting the complete set of observed variables.

Mathematically, we estimate $p(V_i \mid do(\mathbf{U}_j = \mathbf{u}_j))$ by learning a non-parametric function approximator $f(\mathbf{G}; \boldsymbol{\theta})$ (e.g. neural network), which takes as input the (mutilated) causal

[5]A bipartite graph in which the actual variables of interest are not connected is called degenerate.

graph $\mathbf{G} \in \{0,1\}^{N \times N}$ encoded as an adjacency matrix, to predict the parameters $\boldsymbol{\psi}$ of a SPN $g(\mathbf{D}; \boldsymbol{\psi})$ that estimates the density of the given data matrix $\{\mathbf{V}_k\}_k^K = \mathbf{D} \in \mathbb{R}^{K \times N}$. With this, iSPNs are defined as follows:

Definition 7.3.1 (Interventional Sum-Product Network)**.** An interventional sum-product network (iSPN) is the joint model $m(\mathbf{G}, \mathbf{D}) = g(\mathbf{D}; \boldsymbol{\psi} = f(\mathbf{G}; \boldsymbol{\theta}))$, where $g(\cdot)$ is a SPN, $f(\cdot)$ a non-parametric function approximator and $\boldsymbol{\psi} = f(\mathbf{G})$ are shared parameters.

They are called interventional because we consider it to be a causal model given its capability of answering queries from the second level of the causal hierarchy (Pearl, 2019), namely, that of interventions[6]. The shared parameters $\boldsymbol{\psi}$ allow for information flow during learning between the conditions and the estimated densities. Setting the conditions such that they contain information about the interventions, in the form of the mutilated graphs $\mathbf{G}$, effectively renders f to subsume a sort of *do*-calculus in the spirit of truncated factorization shown in Eq.(7.3) i.e., the gate model acts as an estimand selector. Generally, we note that our formulation allows for different function and density estimators f, g. We choose f to be a neural network for two reasons (1) their empirically established capability to act as causal sub-modules (e.g. (Ke et al., 2019) use a cohort of neural nets to mimic a set of structural equations, thus, a SCM) and (2) their model capacity being a universal function approximator, while we choose g to be a SPN for its tractability properties for inference.

We have argued the importance of adaptability to interventional changes within the causal system and intend now to prove that iSPN are capable of approximating these different causal quantities.

Proposition 7.3.2 (Expressivity)**.** *Assuming autonomy and invariance, an iSPN* $m(\mathbf{G}, \mathbf{D})$ *is able to identify any interventional distribution* $p_G(\mathbf{V}_i = \mathbf{v}_i \mid do(\mathbf{U}_j = \mathbf{u}_j))$*, permitted by a SCM*

[6]The first level of the causal hierarchy - association - is considered to be purely statistical.

$\mathfrak{C}$ *through interventions, with knowledge of the mutilated causal graph* $\hat{G}$ *and data* $\mathbf{D}$ *generated from the intervened SCMs by modeling the conditional distribution* $p_{\hat{G}}(\mathbf{V}_i = \mathbf{v}_i \mid \mathbf{U}_j = \mathbf{u}_j)$.

Proof. It follows directly from the definition of the *do*-calculus (Pearl, 2009) that $p_G(\mathbf{V}_i = \mathbf{v}_i \mid do(\mathbf{U}_j = \mathbf{u}_j)) = p_{\hat{G}}(\mathbf{V}_i = \mathbf{v}_i \mid \mathbf{U}_j = \mathbf{u}_j)$ where $\hat{G}$ is the mutilated causal graph according to the intervention $do(\mathbf{U}_j = \mathbf{u}_j)$ i.e., observations in the intervened system are akin to observations made when intervening on the system. Given the mutilated causal graph $\hat{G}$ (as adjacency matrix), the only remaining aspect to show is that the density estimating SPN can approximate a joint probability $p(\mathbf{X})$ using $\mathbf{D}$. This naturally follows from (Poon and Domingos, 2011). □

The expressivity of iSPN stems from both the capacities of gate function and the knowledge of intervention as well as availability of respective data. As an important remark, causal inference is often interested in estimating interventional distributions, i.e, causal quantities from purely observational models. Therefore, an alternative formulation to Prop. 7.3.2 would be to replace the knowledge of the intervened causal structure $\mathbf{G}$ with knowledge on a valid adjustment set. In the following, we only consider the direct setting where actual interventional data from the system is assumed to be captured[7] thereby freeing the investigation of iSPN from the independent research around hidden confounding.

Universal Function Approximation (UFA). The gating nodes of CSPNs extend SPNs in a way that allows them to also induce functions which are universal approximators. For instance, using threshold gates $x_i \leq c \in \mathbb{R}$, one can realize testing arithmetic circuits (Choi and Darwiche, 2018) which have been proven to be universal approximators - rendering iSPN to be UFA by construction.

Learning of iSPN. An interventional sum-product network is being learned using a set of mixed-distribution samples generated from simulating the Causal Health SCM for different

[7]For details on the remarked alternative formulation consider the Supplement.

interventions, where the observational case is considered to be equivalent to an intervention on the empty set. The parameters $\boldsymbol{\theta}, \boldsymbol{\psi}$ of the iSPN describe the weights of the gate nodes and the distributions at the leaf nodes. The full model m is differentiable if the provided gate function f and each of the leaf models of g are differentiable. Therefore, to train an iSPN, as depicted in Fig. 7.2(b), it is sufficient to optimize the conditional log-likelihood end-to-end using gradient based optimization techniques. We assess the performance of our learned model through inspection of the adaptation of the model to the different interventions manifesting in the resulting marginals $p(V_i \mid do(U_j))$.

As can be observed in Fig. 7.2(c) or alternatively in Fig. 7.4 (top row), the learned iSPN successfully adapts to both the interventions as well as its consequences. Considering for instance the intervention $p(V_i \mid do(F = B(\frac{1}{2})))$ which removes the edge $A \to F$ and thus renders Age (A) and Food Habits (F) independent. Given the drastic population change in F and the fact that the Health (H) of an individual is causally dependent on both A and F a significant change in H is being expected. Indeed, both H and Mobility (M), being a causal child of H, broaden distribution wise and also these subsequent changes are captured correctly.

Causal Estimation. The algebraic-graphical *do*-calculus (Pearl, 2009) is complete in that it can find estimands for any identifiable query in a finite amount of transformation. An SPN, and thereby also iSPN, is capable of providing estimates to said demands when given observational data ($\mathcal{L}_1$ on the PCH). Consider for instance the well-known non-Markovian "Napkin" graph $G = \{W \to Z \to X \to Y, W \overset{*}{\leftrightarrow} X, W \overset{*}{\leftrightarrow} Y\}$ where $\overset{*}{\leftrightarrow}$ denotes a confounded relation. The causal effect is identified as $p(y|\, do(x)) = (\sum_w p(x|w,z)p(w))^{-1}(\sum_w p(y,x|w,z)p(w))$ using the *do*-calculus where each of the r.h.s. components can be modeled by (i)SPN respectively. However, iSPN are also capable of directly expressing the causal quantity $p(y|\, do(x))$ as proven in Prop.7.3.2, similar to other neural-causal models like CausalGAN (Kocaoglu et al., 2019) or MLP-based NCM (Xia et al., 2021).

Tractability. Inference in BNs and Markov networks is at least an NP-problem and existing exact algorithms have worst-case exponential complexity (Cooper, 1990b; Roth, 1996), while SPNs can do inference in time proportional to the number of links in the graph. I.e., SPNs in general are able to compute any marginalization and conditioning query in time linear of the model's representation size r, that is $O(r)$. We deploy simple neural network architectures for which the runtime for the forward-pass is that of matrix-multiplication i.e., the time complexity scales cubically in the size of the input n, that is $O(n^3)$. The gradient descent procedure that involves forward and backward passes, assuming m gradient descent iterations, scales to $O(mn^3)$ which is the overall complexity we achieve during training phase. For SPN we deploy a random SPN structure which circumvents structure learning as such. Therefore, overall, training will generally be in the $O(mn^3r)$ regime while any causal query (that is, both L_1 observational and L_2 interventional in our case) will be answered within $O(n^3r)$. However, assuming that the weights of the random SPN were already initialized by the neural parameter-provider in a previous step, any causal query becomes answerable in $O(r)$. Since asking for inferences/queries q within a single distribution ($q \to d \in L_i$) are more common than changes between distributions ($d, \hat{d} \subset L_i$), this linear complexity by our tractable model is being leveraged fully for performing causal inference.

Discussion. To reconsider and answer the general question of why the modeling of a conditional distribution via an over-parameterized architecture is a sensible idea consider the following. One can represent a conditional distribution $p(\mathbf{Y} \mid \mathbf{X})$ by applying Bayes Rule to a joint distribution density model (e.g. a regular SPN) $p(\mathbf{Y} \mid \mathbf{X}) = \frac{p(\mathbf{Y},\mathbf{X})}{p(\mathbf{X})}$. However, this assumes non-empty support i.e., $p(\mathbf{X}) > 0$. Furthermore, the joint distribution $p(\mathbf{Y}, \mathbf{X})$ optimizes *all* possibly derivable distributions, diminishing single distr. expressivity. Therefore, our considered formulation of a gate model allows for effectively subsuming the *do*-operator i.e., the gate model orchestrates the *do* queries such that the density estimator can easily switch between different interventional distributions. While not introducing specific limitations, general CSPN limitation regarding OOD generalization are inherited.

Table 7.1: **Jensen-Shannon-Divergence Evaluation of Estimated Interventional Distributions**. Numerical counterpart to Fig. 7.4, mean and standard deviation per $p(V_{j\setminus i} \mid do(V_i = U(V_i)))$ where U is the uniform distribution across all data sets. Lower = better.

Method \ Query	V_1	V_2	V_3	V_4
iSPN	$.001 \pm .00$	$.007 \pm .01$	$.003 \pm .00$	$.013 \pm .01$
MADE	$.588 \pm .59$	$.108 \pm .16$	$.015 \pm .02$	$.105 \pm .12$
MDN	$.178 \pm .14$	$.263 \pm .14$	$.184 \pm .12$	$.079 \pm .01$

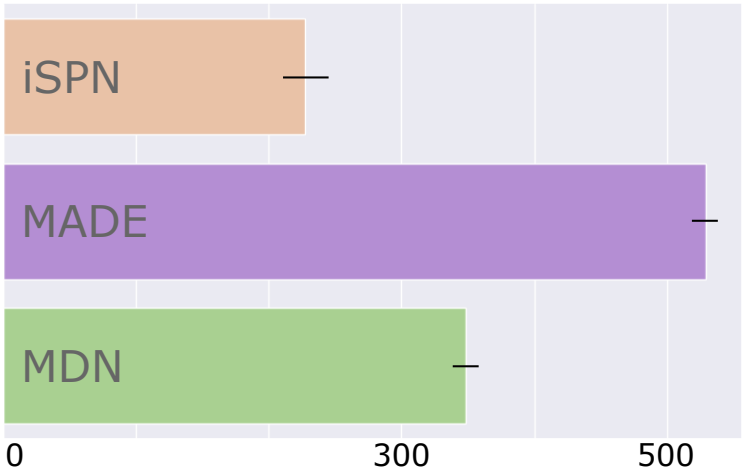


Figure 7.3: **Mean running times in seconds until convergence (Causal Health)** for 50 full passes. More data set results are in the supplementary material.

7.4 Experimental Results

The assumptions made in causality usually require control over the data generating process which is almost never readily available in the real world. This amounts to scarcity of the available public data sets and also their implications for transfers to the real world and even when available, they are usually artificially generated as some causal extension of a known, pre-existing data set (e.g. MorphoMNIST data set introduced by Castro et al. [2019]). While it is difficult to consider real-world-esque experimental settings for causal models, we do not restrict our investigations of iSPN to rather specific problem settings like certain noise or decision variable instances (which is common in causal inference). The evaluation is

performed on data sets with varying number of variables and in both continuous and discrete domains. For the introduced causal health data set, we even consider arbitrary underlying noise distributions. We have made our code repository publicly available[8].

Data Sets. We evaluate iSPNs on four data sets. Three benchmarks: ASIA (A) (Lauritzen and Spiegelhalter, 1988) with 8 variables, Earthquake (E) with 5 variables and Cancer (C) (Korb and Nicholson, 2010b) with 5 variables. One newly curated synthetic causal health (H) data set with 4 variables. More information about the data sets is presented in the Supplement.

Baselines. For *generative* capacities, we compare our method against Mixture Density Networks (MDN) (Bishop, 1994) and Masked Autoencoder for Density Estimation (MADE) (Germain et al., 2015). Both methods are expressive, parametric neural network based approaches for density estimation. Generally, the causality for machine learning literature suggests a strong favor for neural based function approximators for modeling causal mechanisms (Ke et al., 2019). For *causal* capacities, we compare our method against the renown causal baselines from CausalML (Chen et al., 2020) and DoWhy (Sharma and Kiciman, 2020) for the modeling of average treatment effects (ATE) (Pearl, 2009; Peters et al., 2017) considered to be a gold standard task within causal inference.

Protocol and Parameters. To account for reproducibility and stability of the presented results, we used learned models for five different random seeds per configuration. For means of visual clarity, the competing baselines MDN and MADE only present the best performing seed while iSPN is being presented with a mean plot and the standard deviation. The CBN, which performs exact inference according to the *do*-calculus (while best performing) has to be considered as gold standard and is therefore not part of the visualization. Furthermore, as it cannot compare in terms of feasibility. The considered interventions were of uniform nature. Each block of four columns represents a variable being randomized uniformly. We deployed

[8] `https://github.com/zecevic-matej/iSPN`

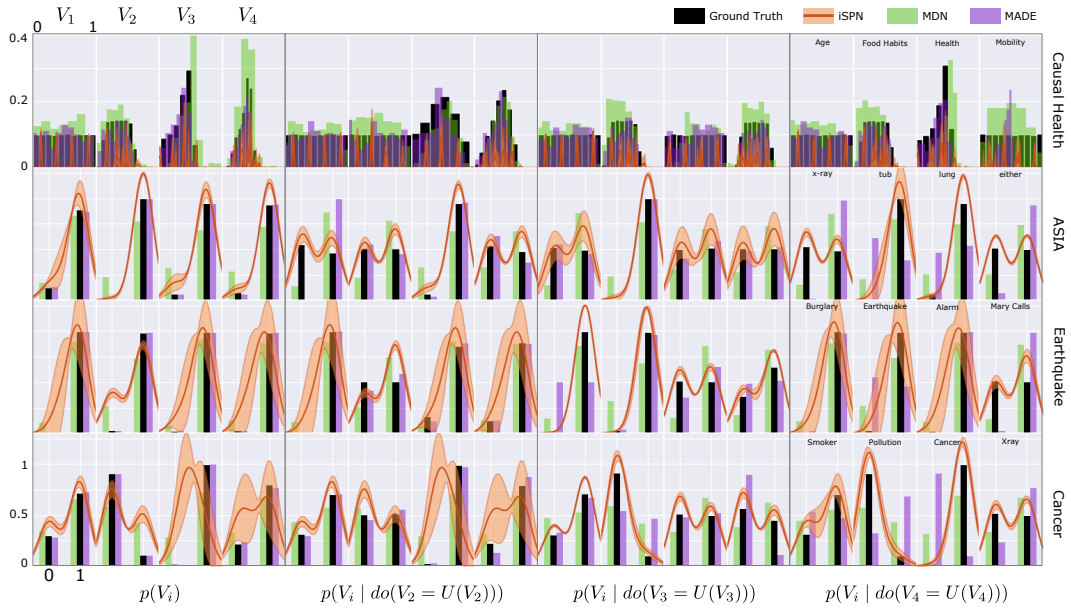


Figure 7.4: **Generative Baselines.** A comparison to the ground-truth (via underlying SCM) and competing estimated distributions. Each row represents a data set and each column represents a variable for a given causal query. (Best in color.)

a RAT-SPN (Peharz et al., 2020b) selecting the leaf node distributions to be Gaussian distributions. For further experimental details consider the Supplement.

Our empirical analysis investigates iSPN for the following questions: **Q1**: How is the estimation quality for interv. distributions? **Q2**: How important is the model's capacity? **Q3**: How is the runtime performance? **Q4/5**: How is the performance relative to state-of-the-art generative and causal models? **Q6**: How does the model adapt to different types of interventions?

(Q1. Precise Estimation of *do*-influenced variables) The density functions learned by iSPN (see Fig. 7.4) fit with a high degree of precision as visualized by the difference in the peak of the modes of the learned distribution and that of the ground truth. While our visual analysis is arguably the superior method of evaluation as it conveys information about how the distributions of interest compare (which is feasible due to marginal inference and the locality of interventions within a SCM), we have also considered numerical evaluation criteria like the Jenson-Shannon-Divergence on which (as visually confirmed) iSPN outperforms

the baselines (see Tab.7.1). For more detailled observation and interpretation consider the Supplement.

(Q2. Capacity Ablation Study) We test the robustness of iSPN as the size of the associated SPN $g(\mathbf{D};\boldsymbol{\psi})$ is varied. We obtain 5 different iSPNs for each of the 4 data sets by using 5 different numbers of sum node weights, 600, 1200, 1800, 2400, 3200, effectively changing the capacity of the parameter-sharing neural network $f(\mathbf{G};\boldsymbol{\theta})$. We observe iSPNs to be robust to varying hyper-parameters that control the size of the SPN $g(\cdot)$ and effectively the complexity of the associated function approximator $f(\cdot)$. More details and figures are in the Supplement.

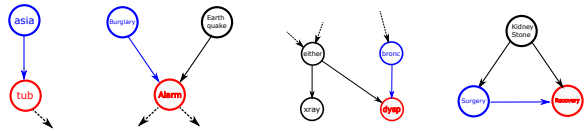
(Q3. Comparison of running times) SPNs are tractable by design as long the networks size is polynomial in the input size. The superiority in running time also becomes apparent during training on the same (Causal Health) data set where we observe the mean run times over 50 passes of the whole data set to be significantly faster than competing methods (see Fig.7.3).

(Q4. Comparison to Generative models) We compare the performances in terms of precision of fit of the learned distributions as well as the flexibility of the models. iSPN outperforms the baselines across all 4 data sets both in precision of the fit of the learned density functions and flexibility of adaptation to the different interventions as seen in figure 7.4. In our experiments, MDN had the worst performance with estimated densities being consistently and significantly different from the ground truth, as the model settled for an average distribution across all interventions. MADE is able to estimate to high precisions but showed to be generally inconsistent across the experimental settings.

(Q5. Comparison to Causal models) We compare the numerical results as seen in figure 7.5 and observe that iSPN matches the performance of the causal baselines. The simple regressor employed by CausalML fails in the confounding case as it estimates wrongly the conditional, both DoWhy and iSPN can handle even the more difficult Simpson's paradox (Simpson, 1951) scenario. An analytical derivation for the ATE is exampled in the Supplement.

Confounding	Conditioning	Ground Truth[1]	CausalML	DoWhy	iSPN	
No	0.0374	0.0397 (0.04)	0.0397	0.0397	**0.0347** ✓	ATE(asia, tub)
No	0.9271	0.9337 (0.93342)*	0.9337	0.9337	‡ **0.9139** ✓	ATE(Burglary, Alarm)
Yes	0.6766	0.6703 (0.667586)	0.6703	0.6697	**0.6551** ✓	ATE(bronc, dysp)
Yes	−0.0457	0.0537 (0.05)	−0.0454	0.0538	**0.0545** ✓	ATE(Surgery, Recovery)

$$\text{ATE}(T, E) := \mathbb{E}[E \mid do(T = 1)] - \mathbb{E}[E \mid do(T = 0)]$$



‡ Visual

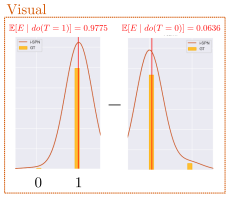


Figure 7.5: **Causal Baselines.** Different causal structures and corresponding causal effect estimation methods (CausalML, DoWhy) are being compared against iSPN. When confounding is present, then conditioning becomes different from intervening $p(Y \mid X) \neq p(Y \mid do(X))$ and iSPN correctly captures all evaluated cases. (* are analytical solutions, [1] differences of means for actual interventional distributions, Best viewed in color.)

(Q6. Different Types of Intervention) By construction, iSPNs are capable to handle arbitrary interventions and our empirical results corroborate this impression. Fig.7.6 shows the training of multiple models on different interventions alongside two example interventions and their appearance in the marginal distributions. Depending on the training setup, e.g. how many different interventional distributions need be learned, any single intervention might become more easily optimizable since the model can exploit similarities between distributions. For a more detailled elaboration and also visualizations of other interventional distributions (including atomic and non-continuous interventions), we direct the reader to our extensive supplementary material.

7.5 Conclusions

We presented a way to connect causality with tractable probabilistic models by using sum-product networks parameterized by universal function approximators in the form of neural networks. We show that our proposed method can adapt to the underlying causal changes in a given domain and generate near perfect interventional distributions irrespective of the data

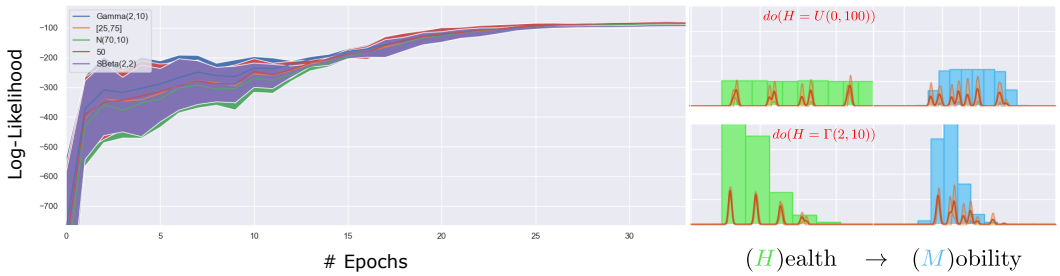


Figure 7.6: **Adaptation to Different Interventions.** Training results for different kinds of interventions on the continuous CH data set. Left, the respective mean objective curves (log-likelihood), indicating consistent training and convergence for all three random seeds per configuration. Right, the (mean) density functions for two different interventions on H: Uniform $U(a, b)$ and Gamma $\Gamma(p, q)$ (other interventions shown in the supplementary). (Best viewed in color.)

distribution and the intervention type thereby exhibiting flexibility. Our empirical evaluation shows that our method is able to precisely estimate the conditioned variables and outperform generative baselines.

Finding a different compilation method for SPNs such as by making use of tree CBNs is important for learning pure causal probabilistic models. Testing our method on larger real world causal data sets is an interesting direction. Finally, using rich expert domain knowledge in addition to observational data is essential for causality and extending our method to incorporate such knowledge is essential.

CHAPTER 8

EXPLOITING DOMAIN KNOWLEDGE AS CAUSAL INDEPENDENCIES

The work presented in this chapter is a collaborative exploration. It is the second and final collaborative exploration of the dissertation, and in several respects the most grounded: it returns to the nuMoM2b gestational diabetes study that has served as a running example throughout, applies the qualitative influence framework that Chapter 4 developed for knowledge extraction, and does so within a model class whose interpretability and feasibility under scarce data are explicitly motivated by the clinical setting.

Where Chapter 7 extended tractable probabilistic models to answer interventional queries at the second level of the causal hierarchy, this chapter takes a different and complementary approach to causal structure. Rather than modeling interventional distributions directly, it exploits causal independence assumptions to reduce the parameter complexity of a Noisy-Or probabilistic model, making learning feasible from the small labeled samples characteristic of clinical datasets. The two chapters thus reflect distinct uses of causal knowledge: **one for expanding the class of queries a model can answer, the other for constraining the parameterization of a model to make learning tractable and the result interpretable.**

This chapter also serves as a fitting closing argument for the dissertation as a whole. The three tenets of effectiveness, efficiency, and explainability are each operative here, in the modest and honest sense that Chapter 6 established: a model that incorporates structured domain knowledge to improve predictive performance, reduces the parameter burden through independence assumptions, and produces outputs that a clinician can interrogate and validate. That this is achieved within a well-understood probabilistic model class, without deep learning machinery, *is not a limitation but a deliberate choice – one that reflects the core premise of this dissertation, that principled, expert-aware, and transparent learning systems are not only possible in data-scarce and safety-critical domains, but necessary.*

8.1 Introduction

We consider the problem of predicting the onset of gestational diabetes mellitus (GDM) from a combination of risk factors and a polygenic risk score. To this effect, we consider data from the **Nulliparous Pregnancy Outcomes Study: Monitoring Mothers-to-Be** (nuMoM2b (Haas et al., 2015)) study and develop a probabilistic model for modeling GDM. While the success of deep learning methods (Goodfellow et al., 2016) in medical tasks (Zhou et al., 2020) has significantly increased the interest in machine-learning-based methods, these models suffer from the twin problems of being data-hungry and uninterpretable. While quite powerful in their classification abilities, these models are not easy to use in decision-making systems that require human interaction.

Consequently, we propose a probabilistic learning method that can effectively and efficiently incorporate domain knowledge. Inspired by previous work in probabilistic learning with expert knowledge (Altendorf et al., 2005; Yang and Natarajan, 2013b), we develop a framework for modeling GDM from a few risk factors including age, BMI, metabolism, family history, blood pressure, etc., and combine the results with a polygenic risk score.

Specifically, our work considers two types of knowledge: causal independencies and qualitative influences. Causal independencies (Heckerman and Breese, 2013; Srinivas, 1993; Vomlel, 2013; Pearl, 1989) specify sets of risk factors (called random variables in probabilistic learning terminology) that are independent of each other when affecting the target. The idea here is that each of these variables has an independent effect on the target – for instance, BMI and age affect GDM independently – and their effects can be combined by a probabilistic combination function. One such example is Noisy-Or. The advantage of such independencies lies in the fact that they lead to a drastic reduction in the number of parameters needed to learn the model.

While powerful, specifying only causal independencies could be insufficient. As an example, consider age and BMI as risk factors for GDM. While both these risk factors could

be independent, when they both are higher, the risk of GDM could be increased. This information is not captured by simple causal independencies. To model such knowledge, earlier methods employ the use of qualitative constraints (Wellman, 1990c; Altendorf et al., 2005; Feelders and van der Gaag, 2005). A qualitative constraint could be a monotonic statement of the form *as X increases Y increases*. For instance, in our task, it is easy to specify that as age increases the risk of GDM increases.

Inspired by our prior work (Yang and Natarajan, 2013b), we combine these two types of domain knowledge to learn a probabilistic model for predicting GDM from the nuMoM2b data and employ the use of polygenic risk score to provide a prior over the incidence of GDM. Specifically, we take the view of a temporal model due to Heckerman and Breese (Heckerman and Breese, 2013) and combine the influence due to the different risk factors using Noisy-Or. For each of these risk factors, we also employ monotonicity constraints whenever applicable. Our empirical evaluations demonstrate that the proposed method with the knowledge from domain experts outperforms probabilistic learning only from data and is comparable with the best machine learning methods that are not interpretable or interactive.

To summarize, we make the following key contributions: (1) We view the problem of modeling GDM using a probabilistic lens and in the presence of domain expert knowledge in the form of qualitative constraints and causal independencies. (2) We take the temporal view and derive the gradients for learning the probabilistic model. (3) We evaluate the algorithm on a real GDM study and establish its effectiveness.

8.2 Data Description

The **Nulliparous Pregnancy Outcomes Study: Monitoring Mothers-to-Be** (nuMoM2b (Haas et al., 2015)) study was established to study individuals without previous pregnancy lasting 20 weeks or more (nulliparous) and to elucidate factors associated with adverse pregnancy outcomes. The study enrolled a racially/ethnically/geographically diverse

population of 10,038 nulliparous women with singleton gestations. The enrolled participants were followed for the duration of their pregnancy and visits were scheduled four times during the pregnancy: 6 weeks 0 days through 13 weeks 6 days estimated gestational age (EGA), 16 weeks 0 days through 21 weeks 6 days EGA, 22 weeks 0 days through 29 weeks 6 days EGA, and at the time of delivery. Our subset has 7 variables - *BMI, PRS, METs, Age, Hist, PCOS, HiBP*.

For our work, we excluded 193 cases where women were diagnosed with pregestational diabetes. Additionally, 3,368 cases with missing features in the dataset were excluded. In our experiments, we use two cohorts. Figure 8.1 illustrates the mechanism for choosing these cohorts. A sub-cohort of 3,533 non-Hispanic white participants with European ancestry was used for experiments involving PRS and a cohort of 6,164 participants was used for experiments not involving PRS. Of the 7 variables, $Hist$, *PCOS, HiBP* are binary, *Age* is discrete while *BMI, PRS and METs* are continuous. *Age* was categorized into 4 values based on quantiles to limit the number of possible values. The continuous variables *BMI, PRS, and METs* were discretized into 5 categories based on quantiles.

8.3 Background: Knowledge-Guided Learning

We now present the necessary background on the two types of expert knowledge that we consider in this work – qualitative influences and causal independencies.

8.3.1 Causal Independence

Causal independence, in simple terms, states that (1) the effect is independent of the order in which causes are introduced, and (2) the impact of a single cause on the effect does not depend on what other causes have previously been applied. This definition facilitates a (probabilistic) belief network representation that is consistent with a set of causal independence statements (Heckerman and Breese, 2013; Srinivas, 1993). The Noisy-Or model, illustrated

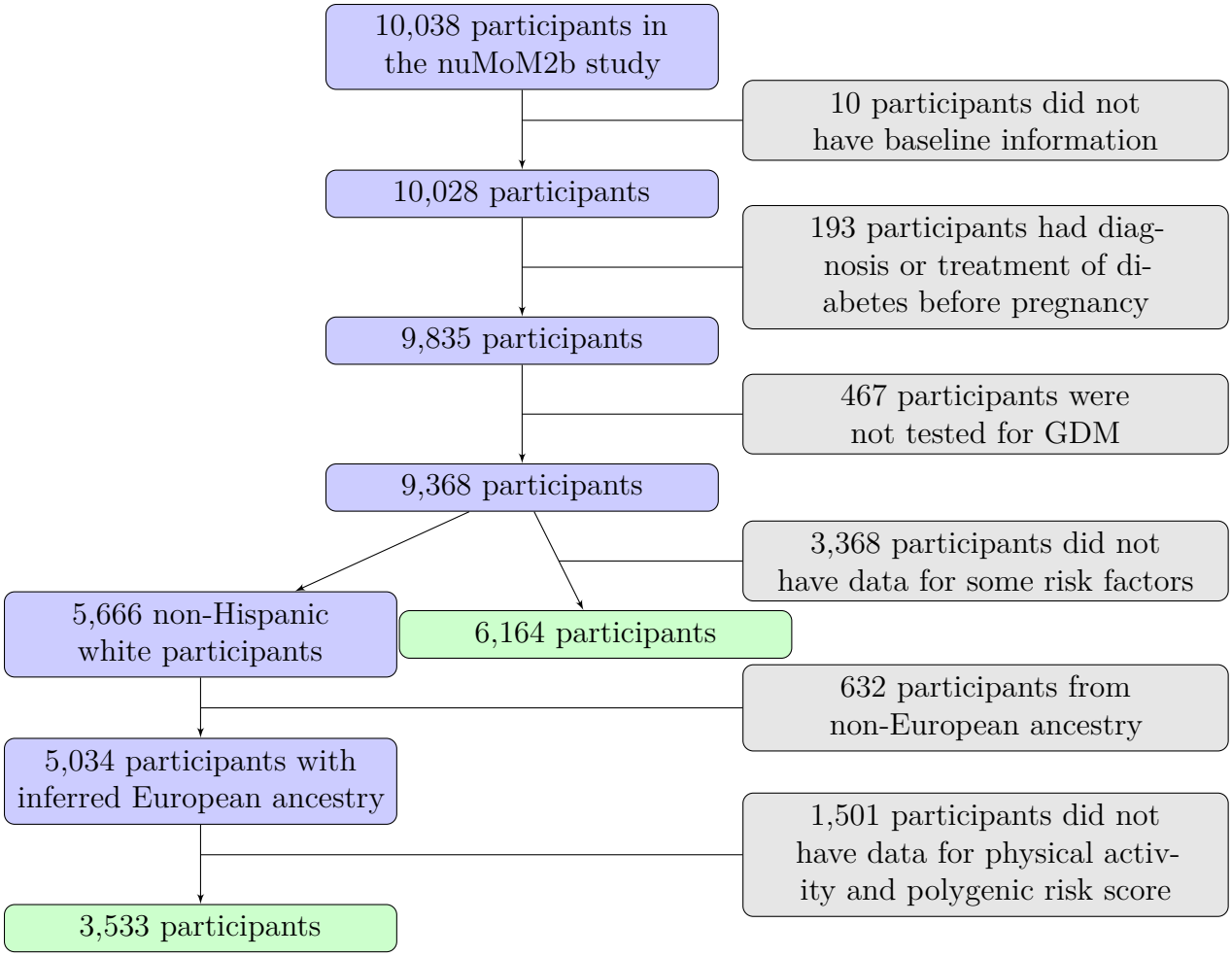


Figure 8.1: Flowchart illustrating the process of selecting the cohorts for our experiments. The two sub-cohorts used in our experiments are indicated in green.

in Figure 8.3, belongs to a class of causal interactions which are characterized by the independence of causal inputs. The belief network in Figure 8.2 represents a general multiple-cause interaction wherein n causes influence a single effect (target variable) y. While this representation provides an intuitive way to capture the causal interaction between the risk factors x_i and the target variable, it requires 2^n parameter assessments for binary variables - one parameter for each instantiation of the causes. This leads to an exponentially large number of examples required to learn a robust conditional distribution.

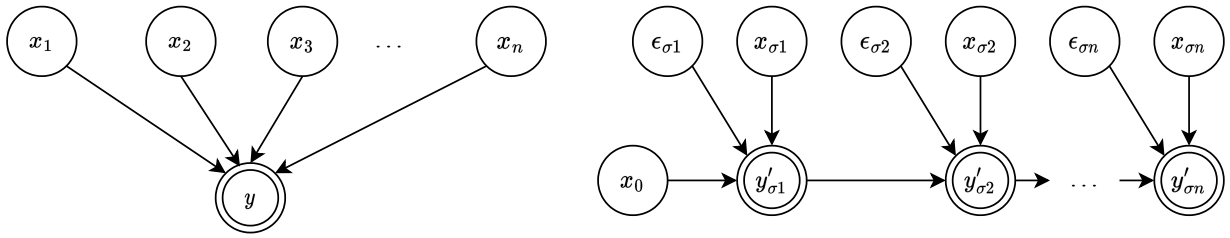


Figure 8.2: A belief network for multiple causes and a single effect (left) and Temporal interpretation of Independence of causal influence (right).

Akin to conditional independence assumptions in Bayesian networks, causal independence assumptions allow efficient parameter learning by causing an exponential reduction in the total number of model parameters as compared to the case of general multiple-cause interaction. Concretely, the presence of independence of causal influences allows us to represent the belief network in Figure 8.2 on the left as the temporal network on the right, for any ordering of causes $\sigma = \{\sigma 1, \ldots, \sigma n\}$. Here, the unobserved effect variable at a timestep $y'_{\sigma i}$ is defined as a deterministic function of the cause $x_{\sigma i}$, the previous state of the effect $y'_{\sigma i-1}$ and $\epsilon_{\sigma i}$, a dummy variable representing the uncertainty. Finally, x_0 represents all causes not considered in the model and $y'_{\sigma n}$ is the observed effect variable. This relation can be expressed as

$$y'_{\sigma 1} = h_\sigma(x_0, x_{\sigma 1}, \epsilon_{\sigma 1}) \tag{8.1}$$

$$y'_{\sigma i} = h_\sigma(y'_{\sigma i-1}, x_{\sigma i}, \epsilon_{\sigma i}),\ \forall i \in \{2, \ldots, n\} \tag{8.2}$$

For the case where h_σ is the Noisy-Or function, the temporal belief network is equivalent to the Noisy-Or model shown in Figure 8.3. The number of parameters in the Noisy-Or model is linear in the number of causes, n, while it is exponential in the original model.

Causal independence statements, in conjunction with qualitative influence statements, allow the injection of rich domain knowledge into an interpretable model while ensuring feasible parameter learning from data. We build upon prior work (Yang and Natarajan, 2013b) in employing this knowledge in the context of GDM modeling.

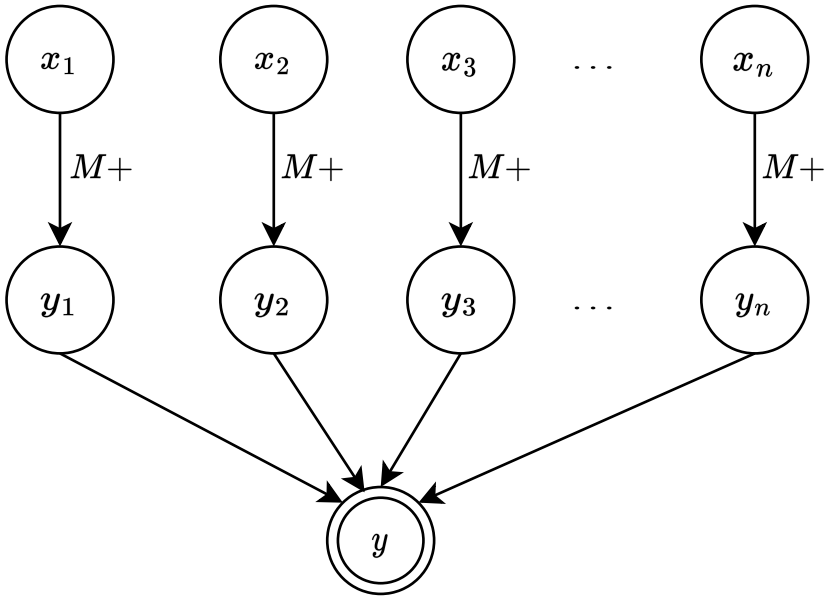


Figure 8.3: The Noisy-Or model

8.4 Causal Independencies with Qualitative Constraints for Modeling GDM

Given: A set of causally independent risk factors $\mathbf{X}$ for the target GDM Y and a set of qualitative influences C.
To Do: Learn an interpretable model $\mathfrak{m}$ that models the conditional probability of the target variable given the risk factors.

As mentioned earlier, $\mathbf{X}$ is the set of risk factors $\langle BMI, PRS, METs, Age, Hist, PCOS, HiBP \rangle$ while Y denotes GDM. Thus, the goal of our work is to learn $P(GDM \mid \mathbf{X})$ given the constraints C. In the rest of this section, we use $\mathbf{X}$ and Y instead of specific risk factors and GDM to demonstrate the generality of the approach.

In the Noisy-Or model, the target variable is activated if any of the causes is active, unless the active causes are inhibited. Formally, the probability of a cause being active is called the *link probability*, and we parameterize it using the sigmoid function σ, i.e.,

$$P(Y_i = 1 \mid X_i = x_i) = \sigma(w_i x_i + b_i), \qquad \forall i \in \{1, \ldots, n\}.$$

The key assumption of the Noisy-Or model is that the inhibitory effect for each cause is independent. Consequently, we parameterize these *inhibition probabilities* as

$$P(Y = 0 \mid Y_i = 1) = \sigma(q_i), \qquad \forall i \in \{1, \ldots, n\}.$$

Finally, the target variable may still be activated even if none of the causes are active. This is called *leakage* and represents all other possible causes that are not included as risk factors. We parameterize the leak probability as

$$P(Y = 1 \mid Y_1 = 0, \ldots, Y_n = 0) = \sigma(q_l).$$

Thus, the target distribution under Noisy-Or is

$$P(Y = 1 \mid \mathbf{X} = x) = 1 - (1 - q_l) \prod_{i=1}^{n} \Big(P(Y_i = 1 \mid X_i = x_i) q_i + P(Y_i = 0 \mid X_i = x_i) \Big). \quad (8.3)$$

Following previous work (Altendorf et al., 2005; Yang and Natarajan, 2013b), we define positive (or negative) monotonic influence $X_i \prec^{M+} Y$ (or $X_i \prec^{M-} Y$) as

$$P(Y_i = 0 \mid X_i = a) \leq P(Y_i = 0 \mid X_i = b), \qquad \forall a, b \in \text{domain}(X_i),\ a > b$$

(or $a < b$ for the negative case). The Noisy-Or model with monotonic influences is shown in Figure 8.3.

8.4.1 Parameter Learning under Monotonicity Constraints

The log-likelihood under the Noisy-Or model can be written as

$$\begin{aligned} \mathcal{L}(\mathbf{w}, \mathbf{b}, \mathbf{q}, q_l; \mathcal{D}) &= \sum_{j=1}^{N} \log P(Y = y^{(j)} \mid \mathbf{X} = x^{(j)}) \\ &= \sum_{j=1}^{N} y^{(j)} \log\big(1 - P(Y = 0 \mid \mathbf{X} = x^{(j)})\big) + (1 - y^{(j)}) \log P(Y = 0 \mid \mathbf{X} = x^{(j)}). \end{aligned} \quad (8.4)$$

We encode the monotonic influences as the margin constraints $\delta_i^{a,b} \leq 0$, where

$$\delta_i^{a,b} = \begin{cases} P(Y_i = 0 \mid X_i = a) - P(Y_i = 0 \mid X_i = b) + \epsilon, & X_{i\prec}^{M+} Y \in C, \\ -P(Y_i = 0 \mid X_i = a) + P(Y_i = 0 \mid X_i = b) + \epsilon, & X_{i\prec}^{M-} Y \in C, \\ 0, & \text{otherwise.} \end{cases} \tag{8.5}$$

Intuitively, if the monotonicity constraint is satisfied, then $\delta_i^{a,b} \leq 0$; otherwise, $\delta_i^{a,b} > 0$. Here ϵ is a small margin. Using these constraints, we define the penalty function

$$\zeta_i^{a,b} = I_{\delta_i^{a,b} > 0} \, (\delta_i^{a,b})^2.$$

Intuitively, the penalty is applied only if the constraint is violated, and it is equal to the square of the magnitude of the violation. Thus, the model does not penalize cases in which the constraints are already satisfied.

Including the penalty function, the final objective to be maximized is

$$J(\mathbf{w}, \mathbf{b}, \mathbf{q}, q_l; \mathcal{D}) = \mathcal{L}(\mathbf{w}, \mathbf{b}, \mathbf{q}, q_l; \mathcal{D}) - \lambda \sum_{i=1}^{n} \sum_{a>b} \zeta_i^{a,b}, \tag{8.6}$$

where λ is the penalty weight. The first term is the classic log-likelihood computed using the conditional distributions, and the second term is the weighted sum of the non-zero penalties. Recall that $\mathbf{w}$ and $\mathbf{b}$ are the link probability parameters, while $\mathbf{q}$ and q_l are the inhibition and leak parameters, respectively.

Intuitively, the penalty function serves as a regularizer that forces the model to satisfy the constraints as much as possible given the data. An advantage of this formalism is that, because the objective is a weighted combination, **the data may be noisy or the constraints may be incorrect**. The model can trade off between fitting the data and satisfying the constraints accordingly. Exploring the case in which both the data and the domain expert are noisy is outside the scope of this work. Thus, the model is robust to both data noise and expert-advice noise. Although λ could be selected by cross-validation, in our experiments and in prior work (Yang and Natarajan, 2013b), the model is robust to the choice of λ as long as it is not close to 0 or 1.

8.4.2 Derivation of the gradients of the log-likelihood term

First, we define the following intermediate gradient terms:

$$
\begin{aligned}
U_j &= \frac{\partial \log P(Y = y^{(j)} \mid \mathbf{X} = x^{(j)})}{\partial P(Y = 0 \mid \mathbf{X} = x^{(j)})} = \frac{-y^{(j)}}{P(Y = 1 \mid \mathbf{X} = x^{(j)})} + \frac{1 - y^{(j)}}{P(Y = 0 \mid \mathbf{X} = x^{(j)})}, \\
Q_{lj} &= \frac{\partial P(Y = 0 \mid \mathbf{X} = x^{(j)})}{\partial q_l} = -\frac{P(Y = 0 \mid \mathbf{X} = x^{(j)})\, \sigma'(q_l)}{1 - q_l}, \\
Q_{ij} &= \frac{\partial P(Y = 0 \mid \mathbf{X} = x^{(j)})}{\partial q_i} = \frac{P(Y = 0 \mid \mathbf{X} = x^{(j)}) P(Y_i = 1 \mid X_i = x_i^{(j)})\, \sigma'(q_i)}{P(Y_i = 1 \mid X_i = x_i^{(j)}) q_i + P(Y_i = 0 \mid X_i = x_i^{(j)})}, \\
V_{ij} &= \frac{\partial P(Y = 0 \mid \mathbf{X} = x^{(j)})}{\partial P(Y_i = 1 \mid X_i = x_i^{(j)})} = \frac{P(Y = 0 \mid \mathbf{X} = x^{(j)})(q_j - 1)}{P(Y_i = 1 \mid X_i = x_i^{(j)}) q_i + P(Y_i = 0 \mid X_i = x_i^{(j)})}, \\
W_{ij} &= \frac{\partial P(Y_i = 1 \mid X_i = x_i^{(j)})}{\partial w_i} = \sigma'(w_i x_i + b_i)\, x_i, \\
B_{ij} &= \frac{\partial P(Y_i = 1 \mid X_i = x_i^{(j)})}{\partial b_i} = \sigma'(w_i x_i + b_i).
\end{aligned}
$$

Here, U_j is the gradient of the log-likelihood of the jth data example with respect to the probability that the target Y is 0. Q_{lj}, Q_{ij}, and V_{ij} are the gradients of the probability that the target is 0 for the jth data example with respect to the leak parameter q_l, the inhibition parameter q_i, and the link probability $P(Y_i = 1 \mid X_i = x_i^{(j)})$, respectively. W_{ij} and B_{ij} are the gradients of the link probability with respect to its parameters w_i and b_i. Finally, σ' denotes the derivative of the sigmoid function, i.e.,

$$\sigma'(x) = \sigma(x)\big(1 - \sigma(x)\big).$$

The gradients of the log-likelihood function with respect to the link parameters w_i and b_i can be computed in terms of U_j, V_{ij}, W_{ij}, and B_{ij} as

$$
\begin{aligned}
\frac{\partial \mathcal{L}(\mathbf{w}, \mathbf{b}, \mathbf{q}, q_l; \mathcal{D})}{\partial w_i} &= \sum_{j=1}^{N} \frac{\partial \log P(Y = y^{(j)} \mid \mathbf{X} = x^{(j)})}{\partial w_i} \\
&= \sum_{j=1}^{N} U_j V_{ij} W_{ij}, \qquad (8.7)
\end{aligned}
$$

and

$$\frac{\partial \mathcal{L}(\mathbf{w}, \mathbf{b}, \mathbf{q}, q_l; \mathcal{D})}{\partial b_i} = \sum_{j=1}^{N} \frac{\partial \log P(Y = y^{(j)} \mid \mathbf{X} = x^{(j)})}{\partial b_i}$$
$$= \sum_{j=1}^{N} U_j V_{ij} B_{ij}. \tag{8.8}$$

The gradients of the log-likelihood function with respect to the inhibition and leak parameters q_i and q_l can be computed in terms of U_j, Q_{ij}, and Q_{lj} as

$$\frac{\partial \mathcal{L}(\mathbf{w}, \mathbf{b}, \mathbf{q}, q_l; \mathcal{D})}{\partial q_i} = \sum_{j=1}^{N} \frac{\partial \log P(Y = y^{(j)} \mid \mathbf{X} = x^{(j)})}{\partial q_i}$$
$$= \sum_{j=1}^{N} \frac{\partial \log P(Y = y^{(j)} \mid \mathbf{X} = x^{(j)})}{\partial P(Y = 0 \mid \mathbf{X} = x^{(j)})} \frac{\partial P(Y = 0 \mid \mathbf{X} = x^{(j)})}{\partial q_i}$$
$$= \sum_{j=1}^{N} U_j Q_{ij}, \tag{8.9}$$
$$\frac{\partial \mathcal{L}(\mathbf{w}, \mathbf{b}, \mathbf{q}, q_l; \mathcal{D})}{\partial q_l} = \sum_{j=1}^{N} \frac{\partial \log P(Y = y^{(j)} \mid \mathbf{X} = x^{(j)})}{\partial q_l}$$
$$= \sum_{j=1}^{N} \frac{\partial \log P(Y = y^{(j)} \mid \mathbf{X} = x^{(j)})}{\partial P(Y = 0 \mid \mathbf{X} = x^{(j)})} \frac{\partial P(Y = 0 \mid \mathbf{X} = x^{(j)})}{\partial q_l}$$
$$= \sum_{j=1}^{N} U_j Q_{lj}. \tag{8.10}$$

8.4.3 Derivation of the gradients of the penalty term

The gradients of the penalty function are given by

$$\frac{\partial \zeta_i^{a,b}}{\partial w_i} = \frac{\partial \zeta_i^{a,b}}{\partial \delta_i^{a,b}} \frac{\partial \delta_i^{a,b}}{\partial w_i}$$
$$= I_{\delta_i^{a,b} > 0} \, 2\delta_i^{a,b} \frac{\partial \delta_i^{a,b}}{\partial w_i}, \tag{8.11}$$

where

$$\frac{\partial \delta_i^{a,b}}{\partial w_i} = \begin{cases} \sigma'(w_i a + b_i)\, a - \sigma'(w_i b + b_i)\, b + \epsilon, & X_i \prec^{M+} Y \in C, \\ -\sigma'(w_i a + b_i)\, a + \sigma'(w_i b + b_i)\, b + \epsilon, & X_i \prec^{M-} Y \in C, \\ 0, & \text{otherwise.} \end{cases} \tag{8.12}$$

Similarly,

$$\begin{aligned} \frac{\partial \zeta_i^{a,b}}{\partial b_i} &= \frac{\partial \zeta_i^{a,b}}{\partial \delta_i^{a,b}} \frac{\partial \delta_i^{a,b}}{\partial b_i} \\ &= I_{\delta_i^{a,b} > 0}\, 2\delta_i^{a,b} \frac{\partial \delta_i^{a,b}}{\partial b_i}, \end{aligned} \tag{8.13}$$

where

$$\frac{\partial \delta_i^{a,b}}{\partial b_i} = \begin{cases} \sigma'(w_i a + b_i) - \sigma'(w_i b + b_i) + \epsilon, & X_i \prec^{M+} Y \in C, \\ -\sigma'(w_i a + b_i) + \sigma'(w_i b + b_i) + \epsilon, & X_i \prec^{M-} Y \in C, \\ 0, & \text{otherwise.} \end{cases} \tag{8.14}$$

Using these gradients, we solve the maximization problem using the L-BFGS-B algorithm, increasing the value of λ until the solution satisfies all the constraints.[1]

The high-level flowchart of our model construction is presented in Figure 8.4. Given the entire GDM data set, after preprocessing and obtaining the causal independencies, we construct the smaller data set in which we learn the model such that the qualitative constraints are satisfied. The final model is then evaluated on the test set, and the results are presented in the next section.

8.5 Experimental Evaluation

Our experiments explicitly aim at answering the following questions,

Q1: Does inclusion of QIs improve model performance over a base model that does not have background knowledge in the form of QIs?

[1]The code is available at `https://github.com/saurabhmathur96/noisy_or`.

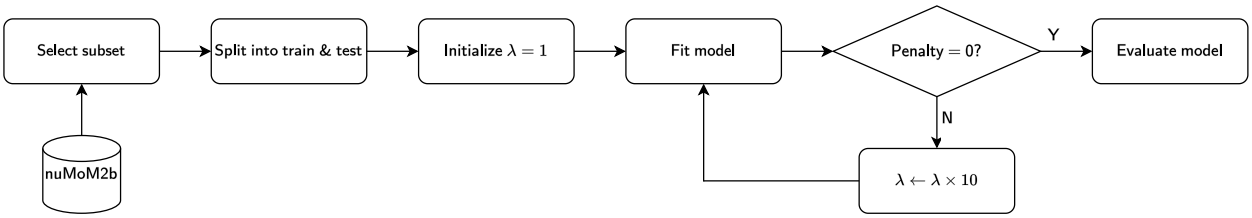


Figure 8.4: Flowchart for the Noisy-Or model construction process

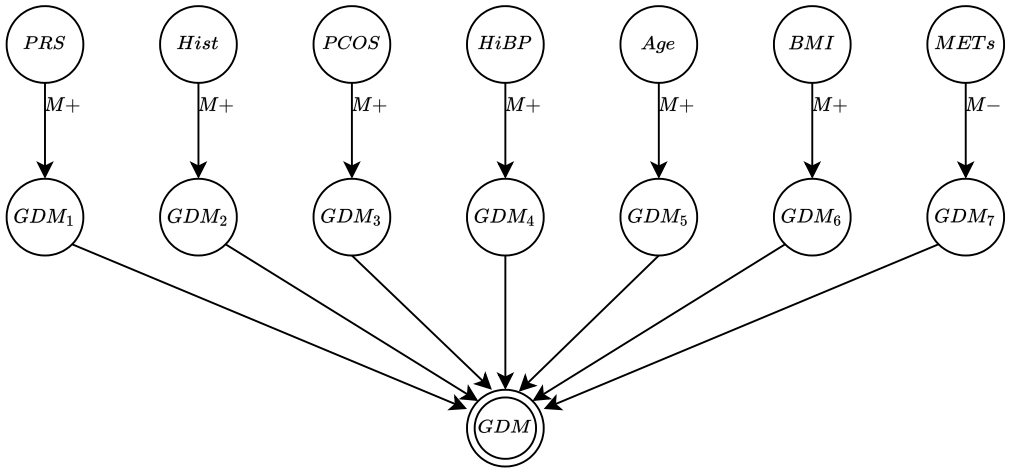


Figure 8.5: Noisy-OR model used for the GDM dataset. Both QIs and causal independence knowledge are incorporated in this model. This representation shows that *PRS, Hist, PCOS, HiBP, Age and BMI* have a positive monotonic influence on GDM whereas *METs* have a negative monotonic influence. Additionally, all the risk factors are causally independent in this model.

Q2: Can our proposed model incorporate causal independencies to efficiently estimate model parameters without significantly losing performance?

We evaluate our proposed approach on two sub-cohorts in the nuMoM2b study - one sub-cohort with PRS as a risk factor and one without it - as described in section 8.2. The domain knowledge in the form of causal independencies and QIs were provided by our domain expert Dr. Haas. Figure 8.5 presents our proposed noisy-OR model that incorporates this domain knowledge for the task of GDM prediction given the 7 risk factors.

To answer the first question, we train noisy-OR models for the two cohorts with and without the inclusion of QIs. Figure 8.6 presents the AUC-ROC (Hanley and McNeil, 1982)

for our model trained on each of the sub-cohorts. In the case of the sub-cohort using the PRS (bottom in Figure 8.6), it can be clearly noted that incorporating QIs improves AUC-ROC from 0.6409 ± 0.0408 to 0.7371 ± 0.0149. In the sub-cohort not using the PRS, incorporating QIs improves the AUC-ROC from 0.6640 ± 0.0079 to 0.6863 ± 0.0091. It is evident from these charts that the inclusion of QIs as domain knowledge improves model performance. This analysis helps us answer Q1. Our proposed approach can effectively incorporate QIs to improve model performance.

To answer the second question, we compare our proposed approach to a strong discriminative baseline: gradient boosted trees (GBT). Figure 8.6 presents a comparison of our model with the baseline for the two sub-cohorts (left and center). GBT achieves AUC-ROC scores of 0.7261 ± 0.0174 and 0.6831 ± 0.0130 for the sub-cohort with and without PRS, respectively. This is comparable to the performance of our proposed approach when QIs are incorporated. However, unlike the noisy-OR model, GBT does not make any causal independence assumptions and hence has no causal meaning and is much more difficult to interpret. This analysis helps us answer Q2. Our proposed model can incorporate causal independencies to allow feasible parameter learning without losing model performance as compared to models that do not make causal independence assumptions.

To summarize, our experiments on two sub-cohorts of the GDM dataset suggest that our proposed approach can leverage domain knowledge in the form of QIs and causal independencies to effectively and efficiently learn an interpretable model without losing model performance as compared to a strong discriminative baseline that is uninterpretable.

8.6 Discussion

We adapted the use of qualitative constraints and causal independencies to build an interpretable and explainable probabilistic model for modeling GDM given a **small number of risk factors**. We presented the learning method that learned the parameters of the model.

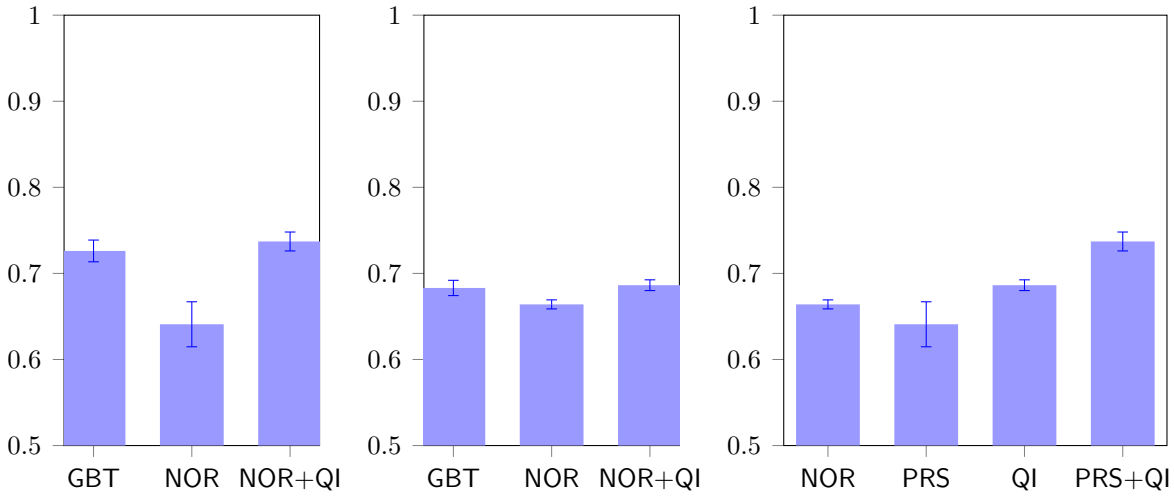


Figure 8.6: The AUC-ROC scores for the Noisy OR model (NOR) as compared to the Gradient Boosted Trees model (GBT) with PRS (left) and without PRS (center). The AUC-ROC scores for the Noisy OR model (NOR) in the presence of PRS and Qualitative Influences (right). The bars show the mean score over 10 boostrap samples and the error bars show the standard deviation.

Our empirical evaluations on nuMoM2b dataset clearly demonstrated that the use of the two types of constraints yielded better results than learning only from data and most importantly, exhibit similar performance as the state-of-the-art machine learning algorithm. Extending the model to include more risk factors is an immediate research direction. Learning a fully generative model such as Bayesian network would provide valuable insights in the interactions between risk factors. Finally, evaluating the learned models on larger and diverse data such as EHRs remains an interesting future direction.

CHAPTER 9

CONCLUSION AND FUTURE WORK

This dissertation began with a premise: that the prevailing "more data, more compute" paradigm, while successful in benchmark settings, is poorly suited to the realities of data-scarce and safety-critical domains. In such domains, with healthcare being the motivating example throughout this work, data is expensive to collect, noisy by nature, and governed by privacy constraints that preclude the kind of scale on which modern deep learning depends. The solutions developed across these chapters are not based on a single method but a perspective: that effectiveness, efficiency, and explainability are not competing priorities to be traded off against one another, but complementary properties that a well-designed learning system should pursue simultaneously, each reinforcing the others.

In the direction of explainability, $\mathcal{EXSPN}$ and QuaKE establish that tractable probabilistic models are not necessarily black boxes. $\mathcal{EXSPN}$ demonstrates that the context-specific independence structure encoded in a learned SPN can be surfaced as compact, human-readable CSI-trees, making the model's internal geometry interpretable to domain experts. QuaKE complements this by extracting qualitative influence statements from the model's behavior, revealing the monotonic and synergistic relationships between variables that a learned distribution implicitly encodes. Both methods were validated by a clinical domain expert on the nuMoM2b gestational diabetes study, grounding their utility in a real and consequential application. Together they operationalize a consistent language of interpretable knowledge structures, grounded in the established formalisms of CSI relations and qualitative influences, that runs as a connective thread through the remaining contributions.

In the direction of building effective models, the *unified framework for human-allied learning of PCs* formalizes seven types of domain knowledge as differentiable constraints on PC parameter learning and optimizes them through a penalized likelihood objective, providing a unified and practically grounded framework for knowledge-intensive learning in data-scarce

settings. The sample efficiency gains that emerge as a secondary finding, where a single well-chosen constraint can compensate for a meaningful reduction in training data, illustrate that domain knowledge is not merely a modeling convenience but a genuine substitute for data when it accurately reflects the underlying distribution. The efficiency contribution, ORIENT, addresses a closely related challenge from a different angle. In supervised domain adaptation, the target domain is by definition data-scarce, making it a natural instance of the same fundamental problem that motivates this dissertation. Existing SDA methods are effective but computationally expensive, and that expense is not merely inconvenient - in resource-constrained clinical environments where specialized hardware and infrastructure cannot be taken for granted, it is a genuine barrier to deployment. ORIENT addresses this by using submodular mutual information to select the subset of source data most relevant to the target domain, reducing training time substantially while maintaining model quality. The argument it makes is that efficiency and effectiveness in adaptation are not in tension, and that principled data selection is a practical path to both.

The two collaborative explorations that close the dissertation extend its themes into causal probabilistic learning. iSPNs push tractable probabilistic models beyond the associational tier of Pearl's causal hierarchy, enabling the modeling of interventional distributions that can be learned from data and mutilated graphs. The work presented in Chapter 8 takes a complementary approach, exploiting causal independence assumptions and qualitative influence statements to make parameter learning feasible on the nuMoM2b dataset, demonstrating that structured domain knowledge yields interpretable and competitive models even when data is severely limited.

The methods developed in this dissertation collectively **make an argument that goes beyond any individual contribution.** Domain knowledge, when expressed in forms that are natural to experts and incorporated in ways that are principled and verifiable, is not a workaround for insufficient data but a significant input to the learning process. One piece of

advice is equivalent to one fell swoop of examples. Simpler models, grounded in structured knowledge and designed with interpretability as a requirement rather than an afterthought, can match the performance of complex data-hungry alternatives while remaining auditable by the people who must ultimately act on their outputs. And the tenets of effectiveness, efficiency, and explainability, though treated separately in much of the literature, are more productively understood as mutually reinforcing dimensions of a single design goal. The cumulative case made by these contributions is for a particular kind of machine learning system: one that is expert-aware, resource-mindful, and transparent by design.

Future Work

The contributions presented in this dissertation open several directions for future investigation. We organize these along three axes: a deeper theoretical understanding of knowledge-intensive learning, an expansion of the dissertation's scope toward causal reasoning, and a set of concrete applications that demonstrate the practical value of tractable probabilistic models beyond the settings considered here.

Theoretical foundations of knowledge-intensive learning. A recurring observation across the experiments of this dissertation is that domain knowledge, when it accurately reflects the underlying distribution, functions as a genuine substitute for data. The unified framework of Chapter 6 surfaces this as an empirical finding: a single well-chosen constraint can compensate for a meaningful reduction in training data. This observation is offered there as an interpretation of results rather than a primary claim, but the signal is consistent enough to motivate a principled theoretical investigation. The framework is built around a general class of constraints expressible as linear combinations of tractable probabilistic queries, of which the seven constraint types enumerated and analyzed empirically in Chapter 6 are representative instances. A theoretical analysis must therefore address this broader class

rather than its specific instantiations, asking what properties of a constraint, understood as a functional of the model's distribution, determine how much it reduces the effective hypothesis space and what that reduction implies for learning. A sample efficiency analysis from a learning theory perspective offers one productive entry point into this question, with the choice of theoretical tools guided by the generative nature of probabilistic circuits; PAC-style bounds, while natural for discriminative settings, are less well suited here, and alternatives such as compression-based arguments may provide sharper and more interpretable guarantees. The information-theoretic perspective offers a complementary lens: domain knowledge reduces prior entropy over the model class, and understanding how this reduction propagates through parameter learning to tighten posterior concentration could yield bounds that are both analytically tractable and meaningful to practitioners.

A particularly promising direction within this investigation concerns the *expressive efficiency* of the constraint types formalized in Chapter 6. The expressive efficiency of probabilistic circuits is a well-studied, if still incompletely understood, concept: certain distributions require circuits of exponential size to represent without specific structural properties, while the same distributions admit compact representations when those properties are present. An analogous question applies to domain knowledge constraints. Some constraints may impose equivalent inductive biases despite arising from structurally different expert beliefs; others may subsume one another in terms of the distributions they rule out. Formalizing a notion of expressive efficiency for domain knowledge constraints, within the general class of linear combinations of tractable queries, would not only clarify the relationships among the specific constraint types introduced in this dissertation, but would also provide a principled vocabulary for reasoning about which constraints are most informative in a given setting and when adding further constraints yields diminishing returns. Taken together, these theoretical tools would move the framework from its current empirical grounding toward a principled design methodology, informing the development of practical algorithms that are adaptive

to the type and quality of available domain knowledge. Although the contributions of this dissertation are developed in the context of probabilistic circuits, the theoretical questions raised here are largely model-agnostic, and extending the analysis to other tractable or approximate model families would be a natural and valuable next step.

Extending toward causal reasoning. The methods developed in this dissertation operate almost entirely at the associational tier of Pearl's causal hierarchy, modeling statistical dependencies among variables without making commitments about the mechanisms that generate them. The collaborative explorations in Chapters 7 and 8 take early steps toward the interventional tier, demonstrating that tractable probabilistic models can be extended to represent interventional distributions and that causal independence assumptions can improve learning in severely data-scarce settings. An important direction for future work is to continue this ascent through the causal hierarchy, with particular attention to the counterfactual tier, which has received comparatively little treatment in the tractable models literature.

Counterfactual reasoning is not merely a theoretical extension; it is a practical prerequisite for deploying probabilistic models in settings where decisions must be justified to the people they affect. In clinical practice, a clinician who receives a risk prediction for a patient may reasonably ask what change in the patient's modifiable risk factors would have led to a different outcome. Answering this question requires counterfactual inference, and doing so with the exactness guarantees that motivated the use of probabilistic circuits throughout this dissertation demands tractable counterfactual computation. Developing PC architectures capable of such computation would unlock a class of practically important queries, including algorithmic recourse, where the goal is to identify the minimal intervention on a patient's features that would change a model's prediction, and causal fairness auditing, where the goal is to determine whether a model's behavior on a subpopulation is attributable to a protected attribute through a causal pathway.

The knowledge structures operationalized in this dissertation offer a natural conceptual bridge toward this setting. The context-specific independence relations surfaced by CSI-trees and the qualitative influence statements extracted by QuaKE are associational objects, grounded in the conditional independence structure of a learned distribution. **They do not themselves encode causal claims.** However, both types of representation use a vocabulary that maps naturally onto the language of causal models: independencies, monotonic effects, and synergistic interactions are precisely the kinds of relationships that domain experts express when constructing causal graphs and that appear in the structural equations of causal models. Extending these representations so that they can be grounded in the interventional and counterfactual tiers of the causal hierarchy would make them substantially more powerful as tools for clinical decision support, allowing practitioners to reason not only about what a model has learned from historical data but about what it implies for future actions. Causal discovery under domain knowledge constraints, which would complement the causal learning direction of Chapters 7 and 8 while connecting back to the unified framework of Chapter 6, represents a further avenue of investigation in this direction.

Applications enabled by tractable inference. Throughout this dissertation, tractability is treated primarily as a modeling property: the guarantee that a rich class of probabilistic queries can be answered exactly and in time linear in the circuit size. This guarantee is well motivated as a theoretical design requirement in safety-critical domains, and the practical consequences of violating it, in the form of unreliable probability estimates driving clinical decisions, are clear. What is less immediately visible is the degree to which tractable inference becomes not merely desirable but *necessary* for several practically important downstream tasks. We highlight two such tasks as illustrative examples; the broader point is that any application currently relying on approximate inference or surrogate generative models is a candidate for this kind of treatment, with the central research question being how much exact and efficient inference changes outcomes in that application.

The first is active feature acquisition. In many real-world settings, including the nuMoM2b study that serves as the running application throughout this dissertation, not all features are available at prediction time; obtaining them requires tests, measurements, or queries that carry a cost. The optimal strategy for deciding which feature to acquire next, given a partially observed patient profile and a budget, involves computing the expected reduction in predictive uncertainty that each candidate feature would provide. This expectation is an integral over the model's distribution, and computing it exactly for each candidate at each step requires tractable marginal inference. Approximate inference methods introduce errors that compound across acquisition steps and cannot be controlled without problem-specific analysis. The tractability of probabilistic circuits makes them uniquely well positioned for this task, and the domain knowledge framework of Chapter 6 would allow clinical constraints to shape not only the learned model but also the acquisition policy derived from it.

The second is query-driven post-hoc explanation. The explainability contributions of this dissertation, CSI-trees and QuaKE, are oriented toward extracting global structural representations from a learned model. A complementary and practically important form of explanation is local and contrastive: given a specific prediction for a specific instance, why did the model produce that output rather than a different one? Answering such questions involves evaluating conditional and marginal probabilities under the learned distribution with particular variables fixed, which is exactly the class of queries for which probabilistic circuits guarantee efficient exact computation. Developing a framework for query-driven contrastive explanation grounded in tractable inference would extend the explainability contributions of this work from the global to the local level, directly addressing the kinds of questions that arise in practice when a clinician or patient seeks to understand a model's output for a specific case.

These two applications are illustrative rather than exhaustive. The common thread is a dependency on repeated exact probabilistic inference at decision time, in settings where errors

compound and the cost of approximation is difficult to bound. Tractable inference is not, in this light, a theoretical selling point but a practical design requirement. Demonstrating this through concrete instantiations would substantially strengthen the case for tractable probabilistic models as the right tool for real-world deployment in data-scarce and safety-critical domains.

REFERENCES

Acharya, J., A. Bhattacharyya, C. Daskalakis, and S. Kandasamy (2018). Learning and testing causal models with interventions. *NeurIPS*.

Agrawal, R., R. Srikant, et al. (1994). Fast algorithms for mining association rules. In *VLDB*.

Altendorf, E., A. C. Restificar, and T. G. Dietterich (2005). Learning from sparse data by exploiting monotonicity constraints. In *UAI*, pp. 18–26. AUAI Press.

Arpit, D., S. Jastrzebski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. C. Courville, Y. Bengio, and S. Lacoste-Julien (2017). A closer look at memorization in deep networks. In *ICML*, Volume 70, pp. 233–242. PMLR.

Ash, J. T., C. Zhang, A. Krishnamurthy, J. Langford, and A. Agarwal (2020). Deep batch active learning by diverse, uncertain gradient lower bounds. In *ICLR*.

Axelrod, A., X. He, and J. Gao (2011). Domain adaptation via pseudo in-domain data selection. In *EMNLP*, pp. 355–362. ACL.

Bareinboim, E. and J. Pearl (2016). Causal inference and the data-fusion problem. *PNAS*.

Beckers, S. and J. Y. Halpern (2019). Abstracting causal models. In *AAAI*.

Ben-David, S., J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Vaughan (2010). A theory of learning from different domains. *Machine Learning 79*, 151–175.

Ben-David, S., J. Blitzer, K. Crammer, and F. Pereira (2006). Analysis of representations for domain adaptation. In *NIPS*, pp. 137–144. MIT Press.

Bica, I., A. Alaa, and M. Van Der Schaar (2020). Time series deconfounder: Estimating treatment effects over time in the presence of hidden confounders. In *ICML*.

Bilmes, J. A. (2022). Submodularity in machine learning and artificial intelligence. *CoRR abs/2202.00132*.

Bishop, C. M. (1994). Mixture density networks.

Borsos, Z., M. Mutný, M. Tagliasacchi, and A. Krause (2021). Data summarization via bilevel optimization. *CoRR abs/2109.12534*.

Boutilier, C., N. Friedman, M. Goldszmidt, and D. Koller (1996). Context-specific independence in bayesian networks. In *UAI*.

Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone (1984). *Classification and Regression Trees*. Wadsworth Publishing Company.

Brouillard, P., S. Lachapelle, A. Lacoste, S. Lacoste-Julien, and A. Drouin (2020). Differentiable causal discovery from interventional data. *NeurIPS*.

Carriger, J. F. and M. G. Barron (2020). A bayesian network approach to refining ecological risk assessments: Mercury and the florida panther (puma concolor coryi). *Ecological Modelling*.

Castro, D. C., J. Tan, B. Kainz, E. Konukoglu, and B. Glocker (2019). Morpho-mnist: quantitative assessment and diagnostics for representation learning. *Journal of Machine Learning Research 20*(178), 1–29.

Chen, H., T. Harinen, J.-Y. Lee, M. Yung, and Z. Zhao (2020). Causalml: Python package for causal machine learning. *arXiv preprint arXiv:2002.11631*.

Choi, A. and A. Darwiche (2017). On relaxing determinism in arithmetic circuits. In *ICML*.

Choi, A. and A. Darwiche (2018). On the relative expressiveness of bayesian and neural networks. In *International Conference on Probabilistic Graphical Models*, pp. 157–168. PMLR.

Choi, Y., A. Vergari, and G. Van den Broeck (2020). Probabilistic circuits: A unifying framework for tractable probabilistic models. *UCLA*.

Chu, B., V. Madhavan, O. Beijbom, J. Hoffman, and T. Darrell (2016). Best practices for fine-tuning visual classifiers to new domains. In *ECCV Workshops (3)*, Volume 9915 of *Lecture Notes in Computer Science*, pp. 435–442.

Ciresan, D. C., U. Meier, and J. Schmidhuber (2012). Multi-column deep neural networks for image classification. In *CVPR*, pp. 3642–3649. IEEE Computer Society.

Colombo, D. and M. H. Maathuis (2014). Order-independent constraint-based causal structure learning. *Journal of Machine Learning Research 15*(1), 3741–3782.

Cooper, G. F. (1990a). The computational complexity of probabilistic inference using bayesian belief networks. *Artif. Intell. 42*(2-3), 393–405.

Cooper, G. F. (1990b). The computational complexity of probabilistic inference using bayesian belief networks. *Artificial intelligence 42*(2-3), 393–405.

Darwiche, A. (2003). A differential approach to inference in bayesian networks. *J. ACM 50*(3), 280–305.

Dasgupta, I., J. Wang, S. Chiappa, J. Mitrovic, P. Ortega, D. Raposo, E. Hughes, P. Battaglia, M. Botvinick, and Z. Kurth-Nelson (2019). Causal reasoning from meta-reinforcement learning. *arXiv preprint arXiv:1901.08162*.

de Campos, C. P., Y. Tong, and Q. Ji (2008). Constrained maximum likelihood learning of bayesian networks for facial action recognition. In *ECCV (3)*, Volume 5304 of *Lecture Notes in Computer Science*, pp. 168–181. Springer.

Deng, L. (2012). The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*.

Domingos, P. (1999). The role of occam's razor in knowledge discovery. *Data mining and knowledge discovery 3*, 409–425.

Feelders, A. J. and L. C. van der Gaag (2005). Learning bayesian network parameters with prior knowledge about context-specific qualitative influences. In *UAI*, pp. 193–200. AUAI Press.

Feldman, D. (2020). Core-sets: Updated survey. In *Sampling Techniques for Supervised or Unsupervised Tasks*, pp. 23–44. Springer.

Fenton, N. E., M. Neil, M. Osman, and S. McLachlan (2020). Covid-19 infection and death rates: the need to incorporate causal explanations for the data and avoid bias in testing. *Journal of Risk Research*.

Feroze, N. (2020). Forecasting the patterns of covid-19 and causal impacts of lockdown in top five affected countries using bayesian structural time series models. *Chaos, Solitons & Fractals*.

Fujishige, S. (2005). *Submodular functions and optimization*. Elsevier.

Fung, G., O. L. Mangasarian, and J. W. Shavlik (2002). Knowledge-based support vector machine classifiers. In *NIPS*, pp. 521–528. MIT Press.

Fürnkranz, J. and T. Kliegr (2015). A brief overview of rule learning. Springer International Publishing.

Ganin, Y. and V. S. Lempitsky (2015). Unsupervised domain adaptation by backpropagation. In *ICML*, Volume 37 of *JMLR Workshop and Conference Proceedings*, pp. 1180–1189. JMLR.org.

Geirhos, R., C. R. M. Temme, J. Rauber, H. H. Schütt, M. Bethge, and F. A. Wichmann (2018). Generalisation in humans and deep neural networks. In *NeurIPS*, pp. 7549–7561.

Gens, R. and P. M. Domingos (2012). Discriminative learning of sum-product networks. In *NeurIPS*.

Gens, R. and P. M. Domingos (2013). Learning the structure of sum-product networks. In *ICML*.

Germain, M., K. Gregor, I. Murray, and H. Larochelle (2015). Made: Masked autoencoder for distribution estimation. In *International conference on machine learning*, pp. 881–889. PMLR.

Girshick, R. B., J. Donahue, T. Darrell, and J. Malik (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pp. 580–587. IEEE Computer Society.

Gong, B., Y. Shi, F. Sha, and K. Grauman (2012). Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, pp. 2066–2073. IEEE Computer Society.

Goodfellow, I., Y. Bengio, and A. Courville (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Goodfellow, I., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio (2014). Generative adversarial nets. In *Advances in Neural Information Processing Systems*.

Granger, C. W. (1969). Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: journal of the Econometric Society*.

Guo, Y. and M. Xiao (2012). Cross language text classification via subspace co-regularized multi-view learning. In *ICML*. icml.cc / Omnipress.

Gupta, A. and R. Levin (2020). The online submodular cover problem. In *SODA*, pp. 1525–1537. SIAM.

Haas, D. M., C. B. Parker, et al. (2015). A description of the methods of the nulliparous pregnancy outcomes study: monitoring mothers-to-be (nuMoM2b). *American journal of obstetrics and gynecology 212*(4), 539.e1–539.e24.

Hagmayer, Y., S. A. Sloman, D. A. Lagnado, and M. R. Waldmann (2007). Causal reasoning through intervention. *Causal learning: Psychology, philosophy, and computation*.

Hair Jr, J. F. and M. Sarstedt (2021). Data, measurement, and causal inferences in machine learning: opportunities and challenges for marketing. *Journal of Marketing Theory and Practice*.

Hanley, J. A. and B. J. McNeil (1982). The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology 143*(1), 29–36.

Hastings, W. K. (1970, 04). Monte carlo sampling methods using markov chains and their applications. *Biometrika 57*(1), 97–109.

He, K., X. Zhang, S. Ren, and J. Sun (2016). Deep residual learning for image recognition. In *CVPR*, pp. 770–778. IEEE Computer Society.

Heckerman, D. and J. S. Breese (2013). A new look at causal independence. *CoRR abs/1302.6814*.

Heckerman, D., D. Geiger, and D. M. Chickering (1995). Learning bayesian networks: The combination of knowledge and statistical data. *Machine learning*.

Hedderson, M. M., J. A. Darbinian, and A. Ferrara (2010). Disparities in the risk of gestational diabetes by race-ethnicity and country of birth. *Paediatric and Perinatal Epidemiology 24*(5), 441–448.

Hedegaard, L., O. A. Sheikh-Omar, and A. Iosifidis (2021). Supervised domain adaptation: A graph embedding perspective and a rectified experimental protocol. *IEEE Trans. Image Process. 30*, 8619–8631.

Hershey, J. R., S. J. Rennie, P. A. Olsen, and T. T. Kristjansson (2010). Super-human multi-talker speech recognition: A graphical modeling approach. *Comput. Speech Lang. 24*(1), 45–66.

Hoiles, W. and M. Schaar (2016). Bounded off-policy evaluation with missing data for course recommendation and curriculum design. In *ICML*.

Iyer, R. K., N. Khargonkar, J. A. Bilmes, and H. Asnani (2022). Generalized submodular information measures: Theoretical properties, examples, optimization algorithms, and applications. *IEEE Trans. Inf. Theory 68*(2), 752–781.

Karanam, A., A. L. Hayes, H. Kokel, D. M. Haas, P. Radivojac, and S. Natarajan (2021). A probabilistic approach to extract qualitative knowledge for early prediction of gestational diabetes. In *AIME*.

Kaushal, V., R. K. Iyer, S. Kothawade, R. Mahadev, K. Doctor, and G. Ramakrishnan (2019). Learning from less data: A unified data subset selection and active learning framework for computer vision. In *WACV*, pp. 1289–1299. IEEE.

Ke, N. R., O. Bilaniuk, A. Goyal, S. Bauer, H. Larochelle, B. Schölkopf, M. C. Mozer, C. Pal, and Y. Bengio (2019). Learning neural causal models from unknown interventions. *arXiv preprint arXiv:1910.01075*.

Killamsetty, K., G. S. Abhishek, Aakriti, A. V. Evfimievski, L. Popa, G. Ramakrishnan, and R. Iyer (2022). AUTOMATA: Gradient based data subset selection for compute-efficient hyper-parameter tuning.

Killamsetty, K., D. Sivasubramanian, G. Ramakrishnan, A. De, and R. K. Iyer (2021b). GRAD-MATCH: gradient matching based data subset selection for efficient deep model training. In *ICML*, Volume 139, pp. 5464–5474. PMLR.

Killamsetty, K., D. Sivasubramanian, G. Ramakrishnan, and R. K. Iyer (2021a). GLISTER: generalization based data subset selection for efficient and robust learning. In *AAAI*, pp. 8110–8118. AAAI Press.

Killamsetty, K., X. Zhao, F. Chen, and R. K. Iyer (2021d). RETRIEVE: coreset selection for efficient and robust semi-supervised learning. In *NeurIPS*, pp. 14488–14501.

Kim, J., J. Yoo, J. Lee, and S. Hong (2021). Setvae: Learning hierarchical composition for generative modeling of set-structured data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15059–15068.

Kindermann, R. and J. L. Snell (1980). *Markov random fields and their applications*, Volume 1 of *Contemporary Mathematics*. Providence, R.I.: American Mathematical Society.

Kingma, D. P. and M. Welling (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kingma, D. P. and M. Welling (2014). Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR*.

Kirchhoff, K. and J. A. Bilmes (2014). Submodularity for data selection in machine translation. In *EMNLP*, pp. 131–141. ACL.

Kisa, D., G. Van den Broeck, A. Choi, and A. Darwiche (2014). Probabilistic sentential decision diagrams. In *KR*.

Kocaoglu, M., C. Snyder, A. Dimakis, and S. Vishwanath (2019). Causalgan.

Koch, D., R. S. Eisinger, and A. Gebharter (2017). A causal bayesian network model of disease progression mechanisms in chronic myeloid leukemia. *Journal of theoretical biology*.

Kokel, H., P. Odom, S. Yang, and S. Natarajan (2020a). A unified framework for knowledge intensive gradient boosting: Leveraging human experts for noisy sparse domains. In *AAAI*, pp. 4460–4468. AAAI Press.

Kokel, H., P. Odom, S. Yang, and S. Natarajan (2020b). A unified framework for knowledge intensive gradient boosting: Leveraging human experts for noisy sparse domains. In *AAAI*, Volume 34, pp. 4460–4468.

Koller, D. and N. Friedman (2009). *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press.

Korb, K. B. and A. E. Nicholson (2010a). *Bayesian artificial intelligence*. CRC press.

Korb, K. B. and A. E. Nicholson (2010b). *Bayesian artificial intelligence*. CRC press.

Kothawade, S., N. Beck, K. Killamsetty, and R. K. Iyer (2021). SIMILAR: submodular information measures based active learning in realistic scenarios. In *NeurIPS*, pp. 18685–18697.

Kothawade, S., V. Kaushal, G. Ramakrishnan, J. A. Bilmes, and R. K. Iyer (2022). PRISM: a rich class of parameterized submodular information measures for guided subset selection. *AAAI 36*.

Kusner, M. J., J. R. Loftus, C. Russell, and R. Silva (2017). Counterfactual fairness. *NeurIPS*.

Lauritzen, S. L. (1988). Local computations with probabilities on graphical structures and their application to expert systems (with discussion). *J. Roy. Statis., Soc., B 50*(2), 251–263.

Lauritzen, S. L. and D. J. Spiegelhalter (1988). Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society: Series B (Methodological)*.

Li, J., T. Xu, J. Zhang, A. Hertzmann, and J. Yang (2019). LayoutGAN: Generating graphic layouts with wireframe discriminator. In *International Conference on Learning Representations*.

Liu, A., H. Zhang, and G. V. den Broeck (2023). Scaling up probabilistic circuits by latent variable distillation. In *The Eleventh International Conference on Learning Representations*.

Liu, M. and O. Tuzel (2016). Coupled generative adversarial networks. In *NeurIPS*, pp. 469–477.

Liu, X., A. Liu, G. Van den Broeck, and Y. Liang (2023). Understanding the distillation process from deep generative models to tractable probabilistic circuits. In *International Conference on Machine Learning*, pp. 21825–21838. PMLR.

Lovász, L. (1982). Submodular functions and convexity. In *ISMP*, pp. 235–257. Springer.

Lowd, D. and J. Davis (2010). Learning markov network structure with decision trees. In *ICDM*.

Luenberger, D. G. and Y. Ye (2016). *Linear and Nonlinear Programming*. Springer International Publishing.

Lüdtke, S., C. Bartelt, and H. Stuckenschmidt (2022, 7). Exchangeability-aware sum-product networks. In L. D. Raedt (Ed.), *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pp. 4864–4870. International Joint Conferences on Artificial Intelligence Organization.

Mathur, S., V. Gogate, and S. Natarajan (2023, 31 Jul–04 Aug). Knowledge intensive learning of cutset networks. In *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, Volume 216 of *Proceedings of Machine Learning Research*, pp. 1380–1389. PMLR.

McCarthy, J. (1960). *Programs with common sense*. RLE and MIT computation center.

Minoux, M. (1978). Accelerated greedy algorithms for maximizing submodular set functions. In *Optimization techniques*, pp. 234–243. Springer.

Mirzasoleiman, B., J. A. Bilmes, and J. Leskovec (2020). Coresets for data-efficient training of machine learning models. In *ICML*, Volume 119, pp. 6950–6960. PMLR.

Mirzasoleiman, B., K. Cao, and J. Leskovec (2020). Coresets for robust training of neural networks against noisy labels. *Advances in Neural Information Processing Systems 33*.

Molina, A., S. Natarajan, and K. Kersting (2017). Poisson sum-product networks: A deep architecture for tractable multivariate poisson distributions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Volume 31.

Molina, A., A. Vergari, N. D. Mauro, S. Natarajan, F. Esposito, and K. Kersting (2018). Mixed sum-product networks: A deep architecture for hybrid domains. In *AAAI*.

Molina, A., A. Vergari, K. Stelzner, R. Peharz, P. Subramani, N. D. Mauro, P. Poupart, and K. Kersting (2019). Spflow: An easy and extensible library for deep probabilistic learning using sum-product networks.

Morgan, S. L. and C. Winship (2015). *Counterfactuals and causal inference*. Cambridge University Press.

Morsing, L. H., O. A. Sheikh-Omar, and A. Iosifidis (2020). Supervised domain adaptation using graph embedding. In *ICPR*, pp. 7841–7847. IEEE.

Motiian, S., Q. Jones, S. M. Iranmanesh, and G. Doretto (2017b). Few-shot adversarial domain adaptation. In *NIPS*, pp. 6670–6680.

Motiian, S., M. Piccirilli, D. A. Adjeroh, and G. Doretto (2017a). Unified deep supervised domain adaptation and generalization. In *ICCV*, pp. 5716–5726. IEEE Computer Society.

Muandet, K., D. Balduzzi, and B. Schölkopf (2013). Domain generalization via invariant feature representation. In *ICML (1)*, Volume 28 of *JMLR Workshop and Conference Proceedings*, pp. 10–18. JMLR.org.

Murphy, K. P., Y. Weiss, and M. I. Jordan (1999). Loopy belief propagation for approximate inference: An empirical study. In *UAI*, pp. 467–475. Morgan Kaufmann.

Neapolitan, R. E. (2004). *Learning bayesian networks*. Pearson Prentice Hall Upper Saddle River, NJ.

Nyman, H., J. Pensar, T. Koski, and J. Corander (2014). Stratified graphical models-context-specific independence in graphical models. *Bayesian Analysis 9*(4), 883–908.

Nyman, H., J. Pensar, T. Koski, and J. Corander (2016). Context-specific independence in graphical log-linear models. *Computational Statistics*.

Oaksford, M. and N. Chater (2017). Causal models and conditional reasoning. *Oxford library of psychology. The Oxford handbook of causal reasoning*.

Oberst, M. and D. Sontag (2019). Counterfactual off-policy evaluation with gumbel-max structural causal models. In *ICML*.

Odom, P., T. Khot, R. Porter, and S. Natarajan (2015). Knowledge-based probabilistic logic learning. In *AAAI*, pp. 3564–3570. AAAI Press.

Papantonis, I. and V. Belle (2020). Interventions and counterfactuals in tractable probabilistic models: Limitations of contemporary transformations. *arXiv preprint arXiv:2001.10905*.

Papantonis, I. and V. Belle (2021). Closed-form results for prior constraints in sum-product networks. *Frontiers Artif. Intell. 4*, 644062.

París, I., R. Sánchez-Cauce, and F. J. Díez (2020). Sum-product networks: A survey. *arXiv preprint arXiv:2004.01167*.

Patel, V. M., R. Gopalan, R. Li, and R. Chellappa (2015). Visual domain adaptation: A survey of recent advances. *IEEE Signal Process. Mag. 32*(3), 53–69.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann.

Pearl, J. (1989). *Probabilistic reasoning in intelligent systems – networks of plausible inference*. Morgan Kaufmann.

Pearl, J. (1995). From bayesian networks to causal networks. In *Mathematical models for handling partial knowledge in artificial intelligence*.

Pearl, J. (2000). *Causality: Models, Reasoning, and Inference*. New York: Cambridge University Press.

Pearl, J. (2009). *Causality*. Cambridge university press.

Pearl, J. (2019). The seven tools of causal inference, with reflections on machine learning. *Communications of the ACM 62*(3), 54–60.

Pearl, J., M. Glymour, and N. P. Jewell (2016). *Causal inference in statistics: A primer*. John Wiley & Sons.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. (2011). Scikit-learn: Machine learning in Python. *JMLR 12*(85), 2825–2830.

Peharz, R., R. Gens, and P. Domingos (2014). Learning selective sum-product networks. In *LTPM workshop, ICML*, Volume 32.

Peharz, R., R. Gens, F. Pernkopf, and P. M. Domingos (2017). On the latent variable interpretation in sum-product networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence 39*, 2030–2044.

Peharz, R., S. Lang, A. Vergari, K. Stelzner, A. Molina, M. Trapp, G. V. den Broeck, K. Kersting, and Z. Ghahramani (2020). Einsum networks: Fast and scalable learning of tractable probabilistic circuits. In *ICML*.

Peharz, R., A. Vergari, K. Stelzner, A. Molina, X. Shao, M. Trapp, K. Kersting, and Z. Ghahramani (2020a). Random sum-product networks: A simple and effective approach to probabilistic deep learning. In *UAI*.

Peharz, R., A. Vergari, K. Stelzner, A. Molina, X. Shao, M. Trapp, K. Kersting, and Z. Ghahramani (2020b). Random sum-product networks: A simple and effective approach to probabilistic deep learning. In *UAI*.

Perera, P. and V. M. Patel (2019). Deep transfer learning for multiple class novelty detection. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pp. 11544–11552.

Peters, J., D. Janzing, and B. Schölkopf (2017). *Elements of causal inference*. The MIT Press.

Poon, H. and P. Domingos (2011). Sum-product networks: A new deep architecture. In *UAI*.

Rahman, T. and V. Gogate (2016). Learning ensembles of cutset networks. In *AAAI*, pp. 3301–3307. AAAI Press.

Rahman, T., S. Jin, and V. Gogate (2019, 09–15 Jun). Look ma, no latent variables: Accurate cutset networks via compilation. In K. Chaudhuri and R. Salakhutdinov (Eds.), *Proceedings of the 36th International Conference on Machine Learning*, Volume 97 of *Proceedings of Machine Learning Research*, pp. 5311–5320. PMLR.

Rahman, T., P. Kothalkar, and V. Gogate (2014a). Cutset networks: A simple, tractable, and scalable approach for improving the accuracy of chow-liu trees. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part II 14*, pp. 630–645. Springer.

Rahman, T., P. V. Kothalkar, and V. Gogate (2014b). Cutset networks: A simple, tractable, and scalable approach for improving the accuracy of chow-liu trees. In *ECML/PKDD (2)*, Volume 8725 of *Lecture Notes in Computer Science*, pp. 630–645. Springer.

Ramanan, N., M. Das, K. Kersting, and S. Natarajan (2020, 23–25 Sep). Discriminative non-parametric learning of arithmetic circuits. In M. Jaeger and T. D. Nielsen (Eds.), *Proceedings of the 10th International Conference on Probabilistic Graphical Models*, Volume 138 of *Proceedings of Machine Learning Research*, pp. 353–364. PMLR.

Raschka, S. (2018). Mlxtend: Providing machine learning and data science utilities and extensions to python's scientific computing stack. *The Journal of Open Source Software*.

Rooshenas, A. and D. Lowd (2013). Learning tractable graphical models using mixture of arithmetic circuits. In *AAAI (Late-Breaking Developments)*.

Rooshenas, A. and D. Lowd (2014, 22–24 Jun). Learning sum-product networks with direct and indirect variable interactions. In *Proceedings of the 31st International Conference on Machine Learning*, Volume 32 of *Proceedings of Machine Learning Research*, Bejing, China, pp. 710–718. PMLR.

Rooshenas, A. and D. Lowd (2016). Discriminative structure learning of arithmetic circuits. In *Artificial Intelligence and Statistics*, pp. 1506–1514.

Roth, D. (1996). On the hardness of approximate reasoning. *Artificial Intelligence 82*(1-2), 273–302.

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nat. Mach. Intell. 1*(5), 206–215.

Rudin, C., C. Zhong, L. Semenova, M. Seltzer, R. Parr, J. Liu, S. Katta, J. Donnelly, H. Chen, and Z. Boner (2024). Amazing things come from having many good models. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Sachs, K., O. Perez, D. Pe'er, D. A. Lauffenburger, and G. P. Nolan (2005). Causal protein-signaling networks derived from multiparameter single-cell data. *Science 308*(5721), 523–529.

Saenko, K., B. Kulis, M. Fritz, and T. Darrell (2010). Adapting visual category models to new domains. In *ECCV (4)*, Volume 6314 of *Lecture Notes in Computer Science*, pp. 213–226. Springer.

Saito, K., D. Kim, S. Sclaroff, T. Darrell, and K. Saenko (2019). Semi-supervised domain adaptation via minimax entropy. *ICCV*.

Schölkopf, B. (2019). Causality for machine learning. *arXiv preprint arXiv:1911.10500*.

Schwartz, R., J. Dodge, N. Smith, and O. Etzioni (2020). Green ai. *Communications of the ACM 63*, 54 – 63.

Schwarz, G. (1978). Estimating the dimension of a model. *The annals of statistics*, 461–464.

Scutari, M. and J.-B. Denis (2021). *Bayesian networks: with examples in R*. Chapman and Hall/CRC.

Selvaraju, R. R., M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, pp. 618–626. IEEE Computer Society.

Sener, O. and S. Savarese (2018). Active learning for convolutional neural networks: A core-set approach. In *ICLR*. OpenReview.net.

Shanmugam, K., M. Kocaoglu, A. G. Dimakis, and S. Vishwanath (2015). Learning causal graphs with small interventions. *NeurIPS*.

Shao, X., A. Molina, A. Vergari, K. Stelzner, R. Peharz, T. Liebig, and K. Kersting (2019). Conditional sum-product networks: Imposing structure on deep probabilistic architectures. *arXiv preprint arXiv:1905.08550*.

Shao, X., A. Molina, A. Vergari, K. Stelzner, R. Peharz, T. Liebig, and K. Kersting (2020). Conditional sum-product networks: Imposing structure on deep probabilistic architectures. In *PGM*, Volume 138 of *Proceedings of Machine Learning Research*, pp. 401–412. PMLR.

Sharma, A. and E. Kiciman (2020). Dowhy: An end-to-end library for causal inference. *arXiv preprint arXiv:2011.04216*.

Shen, Y., A. Choi, and A. Darwiche (2020). A new perspective on learning context-specific independence. In *PGM2020*. PMLR.

Shi, Y., J. Seely, P. H. Torr, N. Siddharth, A. Hannun, N. Usunier, and G. Synnaeve (2021). Gradient matching for domain generalization. *arXiv preprint arXiv:2104.09937*.

Shimodaira, H. (2000). Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference 90*(2), 227–244.

Sidheekh, S., C. B. Dock, T. Jain, R. Balan, and M. K. Singh (2022, 01–05 Aug). Vq-flows: Vector quantized local normalizing flows. In J. Cussens and K. Zhang (Eds.), *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, Volume 180 of *Proceedings of Machine Learning Research*, pp. 1835–1845. PMLR.

Sidheekh, S., K. Kersting, and S. Natarajan (2023, 31 Jul–04 Aug). Probabilistic flow circuits: Towards unified deep models for tractable probabilistic inference. In *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, Volume 216 of *Proceedings of Machine Learning Research*, pp. 1964–1973. PMLR.

Sidheekh, S. and S. Natarajan (2024, 8). Building expressive and tractable probabilistic generative models: A review. In *IJCAI*, pp. 8234–8243. Survey Track.

Simpson, E. H. (1951). The interpretation of interaction in contingency tables. *Journal of the Royal Statistical Society: Series B (Methodological) 13*(2), 238–241.

Singh, A. (2021). CLDA: contrastive learning for semi-supervised domain adaptation. In *NeurIPS*, pp. 5089–5101.

Spirtes, P. and C. Glymour (1991). An algorithm for fast recovery of sparse causal graphs. *Social Science Computer Review 9*(1), 62–72.

Srinivas, S. (1993). A generalization of the noisy-or model. In *UAI*, pp. 208–218. Morgan Kaufmann.

Sun, B. and K. Saenko (2016). Deep coral: Correlation alignment for deep domain adaptation. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*, pp. 443–450. Springer.

Tikka, S., A. Hyttinen, and J. Karvanen (2019). Identifying causal effects via context-specific independence relations. In *NeurIPS*.

Tiwari, R., K. Killamsetty, R. Iyer, and P. Shenoy (2021). GCR: Gradient coreset based replay buffer selection for continual learning.

Torralba, A. and A. A. Efros (2011). Unbiased look at dataset bias. In *CVPR*, pp. 1521–1528. IEEE Computer Society.

Towell, G. G. and J. W. Shavlik (1994). Knowledge-based artificial neural networks. *Artificial intelligence*.

Tzeng, E., J. Hoffman, K. Saenko, and T. Darrell (2017). Adversarial discriminative domain adaptation. In *CVPR*, pp. 2962–2971. IEEE Computer Society.

Van den Broeck, G., N. Di Mauro, and A. Vergari (2019). Tractable probabilistic models: Representations, algorithms, learning, and applications. *Tutorial at UAI*.

Venkateswara, H., J. Eusebio, S. Chakraborty, and S. Panchanathan (2017). Deep hashing network for unsupervised domain adaptation. In *CVPR*, pp. 5385–5394. IEEE Computer Society.

Vergari, A., Y. Choi, A. Liu, S. Teso, and G. V. den Broeck (2021). A compositional atlas of tractable circuit operations for probabilistic inference. In *Advances in Neural Information Processing Systems*.

Vergari, A., N. Mauro, and F. Esposito (2015). Simplifying, regularizing and strengthening sum-product network structure learning. In *ECML-PKDD*.

Vergari, A., N. D. Mauro, and F. Esposito (2019). Visualizing and understanding sum-product networks. *Mach. Learn. 108*(4), 551–573.

Vomlel, J. (2013). Exploiting functional dependence in bayesian network inference. *CoRR abs/1301.0609*.

Wang, M. and W. Deng (2018). Deep visual domain adaptation: A survey. *Neurocomputing 312*, 135–153.

Wei, K., R. K. Iyer, and J. A. Bilmes (2015). Submodularity in data subset selection and active learning. In *ICML*, Volume 37, pp. 1954–1963. JMLR.org.

Wei, K., Y. Liu, K. Kirchhoff, C. D. Bartels, and J. A. Bilmes (2014b). Submodular subset selection for large-scale speech training data. In *ICASSP*, pp. 3311–3315. IEEE.

Wei, K., Y. Liu, K. Kirchhoff, and J. A. Bilmes (2014a). Unsupervised submodular subset selection for speech data. In *ICASSP*, pp. 4107–4111. IEEE.

Wellman, M. P. (1990a). Fundamental concepts of qualitative probabilistic networks. *Artif. Intell. 44*(3), 257–303.

Wellman, M. P. (1990b). Fundamental concepts of qualitative probabilistic networks. *Artificial Intelligence 44*(3), 257–303.

Wellman, M. P. (1990c). Fundamental concepts of qualitative probabilistic networks. *Artif. Intell. 44*(3), 257–303.

Xia, K., K.-Z. Lee, Y. Bengio, and E. Bareinboim (2021). The causal-neural connection: Expressiveness, learnability, and inference.

Xu, J., Z. Zhang, T. Friedman, Y. Liang, and G. V. den Broeck (2018). A semantic loss function for deep learning with symbolic knowledge. In *ICML*, Volume 80 of *Proceedings of Machine Learning Research*, pp. 5498–5507. PMLR.

Xu, X., X. Zhou, R. Venkatesan, G. Swaminathan, and O. Majumder (2019). d-sne: Domain adaptation using stochastic neighborhood embedding. In *CVPR*, pp. 2497–2506. Computer Vision Foundation / IEEE.

Yang, S. and S. Natarajan (2013a). Knowledge intensive learning: Combining qualitative constraints with causal independence for parameter learning in probabilistic models. In *ECML/PKDD (2)*, Volume 8189 of *Lecture Notes in Computer Science*, pp. 580–595. Springer.

Yang, S. and S. Natarajan (2013b). Knowledge intensive learning: Combining qualitative constraints with causal independence for parameter learning in probabilistic models. In *ECML-PKDD*.

Yao, T., Y. Pan, C. Ngo, H. Li, and T. Mei (2015). Semi-supervised domain adaptation with subspace learning for visual recognition. In *CVPR*, pp. 2142–2150. IEEE Computer Society.

Yosinski, J., J. Clune, Y. Bengio, and H. Lipson (2014). How transferable are features in deep neural networks? In *NeurIPS*, pp. 3320–3328.

Zaheer, M., S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola (2017). Deep sets. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), *Advances in Neural Information Processing Systems*, Volume 30. Curran Associates, Inc.

Zečević, M., D. Dhami, A. Karanam, S. Natarajan, and K. Kersting (2021). Interventional sum-product networks: Causal inference with tractable probabilistic models. In *Advances in Neural Information Processing Systems*, Volume 34, pp. 15019–15031. Curran Associates, Inc.

Zhang, K., B. Schölkopf, P. Spirtes, and C. Glymour (2018). Learning causality and causality-related learning: some recent progress. *National science review*.

Zhang, Y., J. Hare, and A. Prugel-Bennett (2019). Deep set prediction networks. *Advances in Neural Information Processing Systems 32*.

Zhao, H., M. Melibari, and P. Poupart (2015a). On the relationship between sum-product networks and bayesian networks. In *ICML '15*, Volume 37, pp. 116–124.

Zhao, H., M. Melibari, and P. Poupart (2015b). On the relationship between sum-product networks and bayesian networks. In *ICML*.

Zhao, H. and P. Poupart (2016). A unified approach for learning the parameters of sum-product networks. *CoRR*.

Zhou, S. K., H. Greenspan, C. Davatzikos, J. S. Duncan, B. van Ginneken, A. Madabhushi, J. L. Prince, D. Rueckert, and R. M. Summers (2020). A review of deep learning in medical imaging: Image traits, technology trends, case studies with progress highlights, and future promises.