

RELATIVE ENTROPY SOUP

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Oliver E. Richardson

August 2024

© 2024 Oliver E. Richardson
ALL RIGHTS RESERVED

RELATIVE ENTROPY SOUP

Oliver E. Richardson, Ph.D.

Cornell University 2024

Your abstract goes here. Make sure it sits inside the brackets. If not, your biosketch page may not be roman numeral iii, as required by the graduate school.

BIOGRAPHICAL SKETCH

Your biosketch goes here. Make sure it sits inside the brackets.

This document is dedicated to all Cornell graduate students.

ACKNOWLEDGEMENTS

Your acknowledgements go here. Make sure it sits inside the brackets.

TABLE OF CONTENTS

Biographical Sketch	iii
Dedication	iv
Acknowledgements	v
Table of Contents	vi
List of Figures	xi
1 Introduction	1
1.1	1
1.2 Overview of Results	1
1.3 Themes	1
1.3.1 Epistemic Humility	1
2 Preliminaries	2
2.0.1 Basic Notation	2
2.0.2 Algebra	2
2.0.3 Relations	3
2.0.4 Graph Theory	4
2.0.5 Categories	8
2.0.6 Measures and Probabilities	11
2.0.7 Independencies	14
2.0.8 Information Theory	14
2.0.9 Graphical Models	14
I A Universal Modeling Language	15
3 Probabilistic Dependency Graphs (PDGs)	16
3.1 Introduction	17
3.2 Syntax	24
3.3 Semantics	26
3.3.1 PDGs As Sets Of Distributions	27
3.3.2 PDGs As Distribution Scoring Functions	27
3.3.3 PDGs As Unique Distributions	32
3.4 Relationships to Other Graphical Models	33
3.4.1 Bayesian Networks	33
3.4.2 Factor Graphs	34
3.4.3 Factored Exponential Families	36
3.5 Discussion	40
Chapter 3 Appendices	
3.A Proofs	43
3.A.1 Properties of Scoring Semantics	43
3.A.2 PDGs as Bayesian Networks	49
3.A.3 Factor Graph Proofs	54

4 Representing Things with PDGs	57
4.1 Probabilities and Random Variables	57
4.2 Widgets	58
4.2.1 Incomplete CPDs and Individual (Conditional) Probabilities	59
4.2.2 Relations and Constraints	59
4.2.3 Couplings	59
4.3 Other Representations of Knowledge and Uncertainty	60
4.3.1 Belief and Plausibility Functions	60
4.3.2 Causal Models	61
4.3.3 Pseudomarginals and Clique Trees	61
4.3.4 Implicit Neural Representations	61
5 Qualitative Mecahnism Independence	62
5.1 Introduction	63
5.2 Qualitative Independent-Mechanism (QIM) Compatibility	66
5.3 QIM-Compatibility and Causality	71
5.3.1 The Equivalence Between QIM-Compatibility and Ran- domized PSEM s	74
5.3.2 Interventions and the Correspondence Between Witnesses and Causal Models	76
5.4 QIM-Compatibility and Information Theory	79
5.4.1 A Necessary Condition for QIM-Compatibility	81
5.4.2 A Scoring Function for QIM-Compatibility	85
5.5 Discussion	86
<hr/>	
Chapter 5 Appendices	
5.A Proofs	88
5.A.1 From CPDs to Distributions over Functions	88
5.A.2 Results on (In)dependence	90
5.A.3 Causality Results of Section 5.3	99
5.A.4 Information Theoretic Results of Section 5.4	106
5.B Monotonicity and Undirected Graphical Models	111
5.C Information Theory, PDGs, and QIM-Compatibility	116
5.C.1 More Detailed Primer on Information Theory	116
5.C.2 Structural Deficiency: More Motivation, and Examples .	117
5.C.3 Weights for SIM-Inc	119
5.C.4 Counter-Examples to the Converse of Theorem 5.4.1 .	120
5.D QIM-Compatibility Constructions and Counterexamples	120
5.E From Causal Models to Witnesses	123
5.F An Algorithm for Finding Witnesses: The Null Value Construction	124
5.G Even Structural Compatibility	125
5.G.1 Even QIM-Compatibility	125
5.G.2 ESIM Compatibility Scoring Rules	127
5.G.3 Complete Derandomization for Cyclic Models	129

II A Universal Objective

131

6 Loss as the Inconsistency of a PDG: Choose your Model, not your Loss	132
6.1 Introduction	133
6.2 Preliminaries	135
6.3 Standard Metrics as Inconsistencies	139
6.4 Regularizers and Priors as Inconsistencies	143
6.5 Statistical Distances as Inconsistencies	145
6.6 Variational Objectives and Bounds	149
6.6.1 PDGs and Variational Approximations	150
6.6.2 Variational Auto-Encoders and PDGs	151
6.6.3 The β -VAE Objective	153
6.7 Free Energy as Factor Graph Inconsistency	153
6.8 Beyond Standard Losses: A Concrete Example	154
6.9 Reverse-Engineering a Loss Function?	156
6.10 Conclusions	158
<hr/>	
Chapter 6 Appendices	
6.A The Fine Print for Probability Densities	160
6.B Further Results and Generalizations	161
6.B.1 Full Characterization of Gaussian Predictors	161
6.B.2 Full-Dataset ELBO and Bounds	164
6.B.3 More Variants of Cross Entropy Results	165
6.C PROOFS	167
6.C.1 Additional Proofs for Unnumbered Claims	184
6.D More Notes	193
6.D.1 Maximum A Posteriori and Priors	193
7 The Local Inconsistency Resolution (LIR) Algorithm	195
7.1 Introduction	196
7.2 Mathematical Preliminaries	197
7.3 Local Inconsistency Resolution (LIR)	201
7.4 LIR in the Classification Setting	203
7.5 The EM Algorithm as LIR	205
7.6 Generative Adversarial Training as LIR	206
7.7 Message Passing Algorithms as LIR	207
7.8 Discussion and Future Work	209
<hr/>	
Chapter 7 Appendices	
7.A Details on Belief Propagation	210
7.B Proofs	212

III Algorithms, Logic, and Complexity	215
8 Inference for PDGs, via Exponential Conic Programming	216
8.1 Introduction	216
8.2 Preliminaries & Related Work	219
8.3 Inference as a Convex Program	226
8.3.1 Minimizing Incompatiblty ($\gamma = 0$)	227
8.3.2 γ -Inference for small $\gamma > 0$	229
8.3.3 Calculating the 0^+ -semantics ($\gamma \rightarrow 0$)	231
8.4 Polynomial-Time Inference Under Bounded Treewidth	232
8.5 Experiments	238
8.6 Discussion and Conclusion	241
<hr/>	
Chapter 8 Appendices	
8.A Proofs	243
8.A.1 Novel Results about PDGs	243
8.A.2 Correctness and Efficiency of Inference via Exponential Conic Programming	251
8.B The Convex-Concave Procedure, and Implementation Details	291
8.C Details on the Empirical Evaluation	293
8.C.1 Synthetic Experiment: Comparison with Black-Box Optimizers, on Joint Distributions	293
8.C.2 Synthetic Experiment: Comparing with Black-Box Optimizers, on Tree Marginals	298
8.C.3 Comparing to Belief Propagation, on Tree Marginals	303
9 Lower Bounds, and The Deep Connection between Inconsistency and Inference	304
9.1 A Semantic Connection	304
9.2 Approximation	304
9.3 An Algorithmic Connection	307
9.4 The Reductions	309
10 Reasoning with PDGs	345
10.1 Quantitative Monotonicity and Equivalence	345
10.2 Qualitative Monotonicity and Equivalence	345
10.2.1 QIM Equivalence	345
IV Foundations	348
11 Confidence	349
12 Relative Entropy Soup	350
13 The Category Theory of PDGs	351

13.0.1	Limits	353
13.0.2	Colimits	355
13.0.3	Natural Transformations	355
13.0.4	Additional Structure Preserved by PDGs	357
13.0.5	The Category of PDGs	359
13.1	Dependency Graphs for Other Monads	359
13.1.1	Relational Dependency Graphs	360
V	Conclusions	366

LIST OF FIGURES

2.1 Examples of directed hypergraphs and their duals	7
3.1 A BN without edges, its corresponding PDG, and an illustration of how the latter can be augmented losslessly with additional cpds.	18
3.2 (a) The Bayesian Network \mathcal{B} in Example 2 (left), and (b) $p\mathcal{dg}(\mathcal{B})$, its corresponding PDG (right). The shaded box indicates a restriction of $p\mathcal{dg}(\mathcal{B})$ to only the nodes and edges it contains, and the dashed node T and its arrow to C can be added in the PDG, without taking into account S and SH	20
3.3 Grok's prior (left) and combined (right) knowledge.	23
3.4 Conversion of the PDG in Example 2 to a factor graph according to Definition 3.4.2 (left), and from that factor graph back to a PDG by Definition 3.4.3 (right). In the latter, for each J we introduce a new variable X_J (displayed as a smaller darker rectangle), whose values are joint settings of the variables connected to it, and also an edge $1 \rightarrow X_J$ (shown in blue), to which we associate the unconditional distribution given by normalizing ϕ_J	36
1 I_μ	80
5.C.1 Illustrations of the information deficiency ($IDef_{\mathcal{A}}$) for various hypergraphs \mathcal{A}	118
1 A map of the inconsistency of the PDG comprising $p(X)$ and $q(X)$, as we vary their respective confidences β_p and β_q . Solid circles indicate well-known named measures, semicircles indicate limiting values, and the heavily dashed lines are well-established classes.	145
2 A visual proof of the data-processing inequality for all PDG divergences, with monotonicity	149
1 Two Illustrations for adversarial training: with a PPDG and a PDG	204
1 Empirical results: accuracy and resource costs for the inference algorithm and baselines	239
2 Comparison of convex solver and black-box optimization baselines. Memory footprints, and accuracy/time costs for the cluster setting.	240
8.C.1 More details: resource costs for joint-distribution optimization setting	294
8.C.2 differences in performance between the Gibbs and simplex parameterizations of probabilities.	297
8.C.3 A disaggregated version of Figure 1	298

8.C.4 A graph of the gap (the difference between the attained objective value, and the best objective value obtained across all methods for that value of γ), as γ varies. The x-axis is $\log_{10}(\gamma + 10^{-15})$. As before, colors indicate the optimization method, and the size of the circle illustrates the number of optimization variables (i.e., the number of possible worlds). <code>cvx-idef</code> corresponds to just solving (8.5), and <code>cvx+idef</code> corresponds to then solving problem (8.10) afterwards. The CCCP runs are split into regimes where the entire problem is convex ($\gamma \leq 1$, labeled <code>cccp-VEX</code>), and the entire problem is concave ($\gamma > 1$, labeled <code>cccp-CAVE</code>). The optimization approaches <code>opt_dist</code> are split into three different optimizers: LBFGS, Adam, and also a third one that performs relatively poorly: accelerated gradient descent. Note that for small γ , the exponential-cone based methods significantly outperform the gradient-based ones.	299
8.C.5 An analogue of Figure 1, for the cluster setting. Note that there is even more separation between the exponential-cone based approaches, and the black-box optimization based ones. The new grey points on the bottom correspond to belief propagation, which is both faster and typically the most accurate.	300
8.C.6 Resource costs for the cluster setting. Once again, the <i>OInc</i> -optimizing exponential cone methods are in gold, the small-gamma and CCCP is in violet, and the baselines are in green. The bottom line is belief propagation, which is significantly faster and requires very little memory, but also only gives the correct answer under very specific circumstances.	300
8.C.7 Gap vs inference time for the small PDGs in the <code>bnlearn</code> repository	302
8.C.8 A variant of Figure 8.C.1, with with gap (accuracy) information on the left, and slightly different parameter settings.	303

CHAPTER 1

INTRODUCTION

1.1

1.2 Overview of Results

1.3 Themes

1.3.1 Epistemic Humility

I distinguish between

CHAPTER 2

PRELIMINARIES

2.0.1 Basic Notation

If A is a finite set, we write $\#A$ or $|A|$ for its cardinality. We will often be concerned with *variables*, which intuitively correspond to aspects of the world or properties of some object. Mathematically, a variable has two aspects. Qualitatively, a variable is just some unique identifier (the variable name), such as “height”. Quantitatively, a variable X is also associated with a set $\mathcal{V}(X)$, or simply $\mathcal{V}X$, of possible values. For example, $\mathcal{V}(\text{height})$ might be the set of positive real numbers, or the set $\{\text{short}, \text{tall}\}$.

2.0.2 Algebra

Definition 2.0.1 (Monoid). A *monoid* is a tuple $(S, *, e)$, where S is a set, $* : S \times S \rightarrow S$ is a binary operation, and $e \in S$ is a distinguished identity element, such that:

- (associativity) $\forall a, b, c \in S. (a * b) * c = a * (b * c);$
- (identity) $\forall a \in S. a * e = a = e * a.$

A monoid is called *commutative* if it also satisfies

- (commutativity): $\forall a, b \in S. a * b = b * a,$

and *idempotent* if it satisfies

- (idempotence): $\forall a \in S. a + a = a.$

An idempotent semiring defines partial order by $a \leq b \iff a + b = b$. \square

2.0.3 Relations

Let $\mathcal{X} = \{X_1, \dots, X_n\}$ be variables, traditionally called attributes. A *relation* $R(\mathcal{A}) = R(X_1, \dots, X_n) \subseteq \mathcal{V}(X_1) \times \dots \times \mathcal{V}(X_n)$, or equivalently, $R : \prod_{i=1}^n \mathcal{V}X_i \rightarrow \{0, 1\}$, is a subset of joint values of attributes. The natural number n is called the *arity* of R .

The *natural join* of two relations $R(A, B)$ and $S(B, C)$ combines them in a particularly obvious way: $(a, b, c) \in R \bowtie S$ iff $(a, b) \in R$ and $(b, c) \in S$. More generally, we have

Definition 2.0.2 (natural join). $R(\mathcal{X}) \bowtie S(\mathcal{Y}) := \left\{ \omega \in \mathcal{V}(\mathcal{X} \cup \mathcal{Y}) \mid \mathcal{X}(\omega) \in R \wedge \mathcal{Y}(\omega) \in S \right\}$. \square

At one extreme, if \mathcal{X} and \mathcal{Y} are disjoint sets of attributes, then $R(\mathcal{X}) \bowtie S(\mathcal{Y})$ coincides with the cartesian product of $R \subseteq \mathcal{V}\mathcal{X}$ and $S \subseteq \mathcal{V}\mathcal{Y}$. At the opposite extreme, if $\mathcal{X} = \mathcal{Y}$ are the same set of variables, then $R(\mathcal{X}) \bowtie S(\mathcal{X})$ coincides with the intersection of the subsets R and S .

Even when $A_1, \dots, A_n \subseteq \mathcal{X}$ are not disjoint, we give a convenient extended syntax by defining the quantity $R(a_1, \dots, a_n)$, where $a_i \in \mathcal{V}(A_i)$. Concretely, define $R(a_1, \dots, a_n) := 0$ if when $\{a_1, \dots, a_n\}$ do not agree on the value of some shared attribute (i.e., if $\exists X \in \mathcal{X}, \exists i, j \in [n]. X \in A_i \cap A_j \wedge X(a_i) \neq X(a_j)$). When $\{a, b, c\}$ do agree on all values of shared attributes, let \mathbf{x} denote the joint value of $A \cup B \cup C$ obtained from (a, b, c) by removing redundant copies of variable values. In this case, define $R(a, b, c) := R(\mathbf{x})$.

2.0.4 Graph Theory

Definition 2.0.3. A (*directed*) (*multi*) graph $G = (N, A)$, or simply a *graph*, is a set N of nodes, and a collection A of arcs, such that each $a \in A$ has a source node $S_a \in N$ and a target node $T_a \in N$. So, formally, the definition is $G = (N, A, S, T)$, with $S, T : A \rightarrow N$ often left implicit. \square

Definition 2.0.4 (Undirected (Multi) Graph). An undirected (multi)graph $G = (N, E)$ is a set N of vertices (or nodes) and a set E of edges, each element $e \in E$ of which corresponds to an unordered pair of vertices $\{u, v\}$. More formally, there is a map

$$\iota : E \rightarrow V \times V \setminus \{(v, v) : v \in N\} / \{(u, v) \sim (v, u) \mid (u, v) \in N \times N\}.$$

implicit in the definition of G , which we will write $G = (N, E, \iota)$ only when being extra careful. \square

It is common to identify a graph $H = (N, A)$ (or an undirected graph $G = (N, E)$) with its (symmetric) adjacency matrix

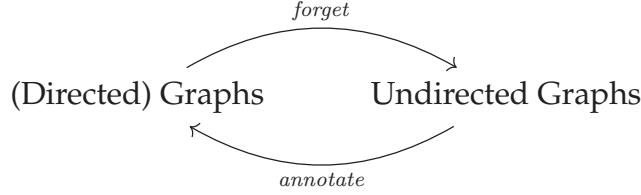
$$\mathbb{A}_H = \left[\# \left\{ a \in A : \begin{array}{l} S_a = u, \\ T_a = v \end{array} \right\} \right]_{(u,v) \in N \times N} \quad \mathbb{A}_G = \left[\# \{ e \in E : \iota(e) = \{u, v\} \} \right]_{(u,v) \in N \times N},$$

in part because there is a natural bijection between (undirected) multigraphs and (symmetric) square matrices over the natural numbers. For example:

$$\begin{bmatrix} 0 & 1 & 2 \\ 1 & 0 & 0 \\ 2 & 0 & 1 \end{bmatrix} \quad \longleftrightarrow \quad \begin{array}{c} \cap \\ c \curvearrowleft a - b \end{array}$$

A (*directed*) graph has more information than an undirected one. There are natural ways to convert between the two: one forgets the direction of arcs to

turn a directed graph into an undirected one, and annotates each arc with arrows in both directions to make an undirected graph directed. These choices are essentially locked in if we want the correspondence with square matrices to hold properly.



in which $\text{forget} \circ \text{annotate} = \text{id}_{\text{Undirected Graphs}}$ but $\text{annotate} \circ \text{forget}$ is not the identity on (directed) graphs. Technically, this makes forget a *retraction*, and annotate a *section*.

Definition 2.0.5. A bipartite graph $G = (L, R, E)$ is a graph $(L \sqcup R, E)$ whose vertices are partitioned into two components $V = L \sqcup R$, and whose edges $E \subset L \times R$ are only between L and R . □

Definition 2.0.6. A directed bipartite graph $G = (L, R, E)$ is a bipartite graph (L, R, E) whose edges $E \subset (L \times R) \cup (R \times L)$ are directed. □

Definition 2.0.7. A *hypergraph* $G = (V, \mathcal{E})$ is a set V of vertices, and a collection \mathcal{E} of edges, which correspond to finite subsets of vertices. □

Thus, a hypergraph is the generalization of an undirected graph in which the codomain of $\iota : \mathcal{E} \rightarrow 2^V$ is arbitrary subsets of V , not just those of cardinality 2.

Proposition 2.0.1. *There is a natural bijection between hypergraphs and bipartite graphs:*

$$\text{bipart}(V, \mathcal{E}) := (V, \mathcal{E}, \{(v, E) \in V \times \mathcal{E} : v \in E\})$$

$$\text{hyper}(L, R, E) := (L, \{\{v \in L : (v, r) \in E\} : r \in R\}),$$

$$bipart \circ hyper = \text{id}_{BG} \quad \text{and} \quad hyper \circ bipart = \text{id}_{HG}.$$

The consequences of this can be unintuitive. It is common to think of bipartite graphs as a strict (particularly nice) special case of ordinary undirected graphs, which themselves are a strict (particularly easy to draw) strict special case of hypergraphs. By transitivity, one might expect bipartite graphs to naturally be an extremely strict special case of hypergraphs—yet in fact they are naturally isomorphic.

Definition 2.0.8. A *directed hypergraph* (N, \mathcal{A}) is a set N of nodes, and a collection \mathcal{A} of hyperarcs, each of which has a set $S_a \subset N$ of source variables and a set $T_a \subset N$ of target variables. \square

A directed hypergraph (N, \mathcal{A}) can be equivalently defined as an (ordinary) directed graph $(2^N, \mathcal{A})$ whose set of nodes is the powerset of some set N .

Definition 2.0.9. The *dual* of the hypergraph $G = (V, \mathcal{E})$ is $\check{G} := (\mathcal{E}, \{\{e \in \mathcal{E} : v \in e\} : v \in V\})$. \square

Definition 2.0.10. The *dual* of a directed hypergraph $\mathcal{H} = (N, \mathcal{A})$ is $\check{\mathcal{H}} := (\mathcal{A}, N)$, where

$$\check{S}_n = \{a \in \mathcal{A} : n \in T_a\} \quad \text{and} \quad \check{T}_n = \{a \in \mathcal{A} : n \in S_a\}. \quad \square$$

We now verify that $\check{\check{\mathcal{H}}} = \mathcal{H}$. Observe that

$$\begin{aligned} \check{\check{S}}_a &= \{n \in N : a \in \check{T}_n\} \\ &= \{n \in N : a \in \{a' \in \mathcal{A} : n \in S_{a'}\}\} \\ &= \{n \in N : n \in S_a\} \\ &= S_a; \end{aligned}$$

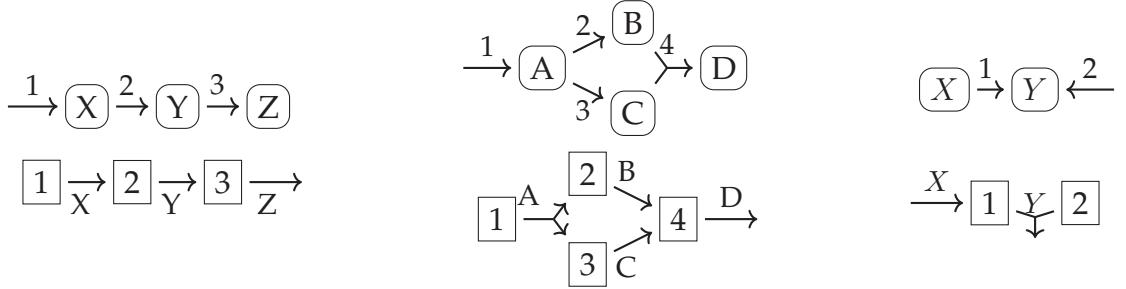


Figure 2.1: Examples of directed hypergraphs (first row) and their duals (second row).

symetrically, $T_a = \check{T}_a$.

See Figure 2.1 for some visual illustrations. We remark that the left and center diagrams on the top can be viewed as (the hypergraphs corresponding to) qualitative Bayesian Networks, by regarding X, Y, Z and A, B, C, D as variables, and imagining that there is a (randomized) causal determination occurring along each arc. One can also imagine an analogue with cycles—resuling in perhaps a (randomized) causal model of the given shape. But a causal model has one equation corresponding to each variable, and the corresponding hypergraphs thus has exactly one hyperarc leading to it. In the dual hypergraphs, one should view the nodes as processes and the arcs as wires. Such a hypergrah has precisely one hyperarc leading out of every node. When wires branch, one imagines a copy; when two arcs point to the same process (as in process 4, in the middle center), that process takes both of the wires as inputs. In the duals of hypergraphs corresponding to causal models, there are no two-tailed arrows, which might be thought of as a “merge”. Yet it is not clear how to merge the values of two variables, when they are not the same, in general—especially if we expect associativity and commutativity, as we do with *copy*.

((What can be done with these objects?))

2.0.5 Categories

Category theory is a mathematical underlingua that captures the essential form of many arguments across mathematics. Sometimes (lovingly) called “abstract nonsense”, category theory is often seen as extremely abstract meta-mathematics. Nevertheless, the basics more concrete and simpler than one might imagine. At its core, it’s essentially just the mathematic underpinnings of typed composition.

Definition 2.0.11 (category). A *category* \mathcal{C} consists of four pieces of data:

- a collection of *objects* $\text{ob}_{\mathcal{C}}$;
- a collection of *morphisms* $\text{Hom}_{\mathcal{C}}(X, Y)$, also written $\mathcal{C}(X, Y)$, for each pair of objects $(X, Y) \in \text{ob}_{\mathcal{C}}^2$;
- a *composition* operator $\circ_{X,Y,Z} : \mathcal{C}(Y, Z) \times \mathcal{C}(X, Y) \rightarrow \mathcal{C}(X, Z)$ for each triple $(X, Y, Z) \in \text{ob}_{\mathcal{C}}^3$, that is written inline (i.e., $f \circ g$ instead of $\circ(f, g)$), and is associative, i.e., $(f \circ g) \circ h = f \circ (g \circ h)$;
- a special *identity element* $\text{id}_X \in \mathcal{C}(X, X)$ for each object $X \in \mathcal{C}$, satisfying $\text{id}_X \circ f = f$ and $g \circ \text{id}_X = g$ for any morphism $f \in \mathcal{C}(Y, X)$ or $g \in \mathcal{C}(X, Y)$ for some $Y \in \text{ob}_{\mathcal{C}}$. □

Common examples of categories include:

- **Set**, the category whose objects are sets and whose morphisms are functions between them,
- **Top**, the category whose objects are topological spaces and whose morphisms are continuous maps,
- **Rel**, the category whose objects are sets, and whose morphisms are relations, and

- $\mathbb{D}\text{iff}$, the category of smooth manifolds (possibly with boundary or corners) and differentiable maps.

All of these are also known as “concrete categories”, because they all build on Set : their objects can be interpreted as sets, and their morphisms interpreted as functions. But categories can also be much more combinatorial in nature. We will be much more interested in dinkier categories. Here are some more extreme kinds of categories:

- A category with one object is just a monoid—observe that \circ is associative and has an identity.
- At the opposite extreme, a category with only identity morphisms is just a collection of objects.
- A category with at most one morphism between any two objects is a preorder—in this case we write $a \leq b$ iff there is a morphism from object a to object b ; the relation is reflexive because of the identity, and transitive because of composition.

What will be most relevant for our purposes is a construction

Definition 2.0.12 (free category generated by a graph). If $G = (N, A)$ is a directed (multi) graph with nodes N and arrows A , the *free category generated by G* is the category G^* , whose objects are the elements of N , and whose set of morphisms from x to y , for $x, y \in N$, is the collection of paths from x to y . That is,

$$\text{ob}_{G^*} = N, \quad G^*(x, y) = \left\{ \text{sequences } \langle a_1, \dots, a_n \rangle \mid \begin{array}{l} n \in \mathbb{N}, \quad n > 0 \Rightarrow (S_{a_1} = x \wedge T_{a_n} = y), \\ \forall i \in \{1, \dots, n - 1\}. \quad T_{a_i} = S_{a_{i+1}} \end{array} \right\},$$

with composition given by sequence concatenation, and the identity being the empty sequence. \square

The superscript-star notation has some standard meanings throughout mathematics, and this construction in [Definition 2.0.12](#) reduces to several of them in the appropriate contexts.

- A (multi) graph $G = (\{*\}, A)$ with one vertex can be identified with its arc set A . Every arrow has the same type $(* \rightarrow *)$, and so a path is a sequence $\langle a_1, a_2, \dots, a_n \rangle$ where each $a_i \in A$. So in this case, G^* (as given by [Definition 2.0.12](#)) coincides with the familiar set of strings A^* over the alphabet A .
- Let $R \subseteq V \times V$ be a binary relation on V . Then the transitive closure of R , often denoted R^* , is the reachability relation generated by R . That is, $(u, v) \in R^*$ if and only if there is a path $\langle u=u_1, \dots, u_n=v \rangle$ with each $(u_i, u_{i+1}) \in R$.

Equivalently, we can view R as a graph $G = (V, R)$ by regarding each $(i, j) \in R$ as an arrow $i \rightarrow j$. The free category G^* generated by these arrows (per [Definition 2.0.12](#)) has an arrow from u to v (i.e., $G^*(u, v) \neq \emptyset$) iff $(u, v) \in R^*$.

- A (directed) (multi) graph G on n vertices also has an adjacency matrix $A := \mathbb{A}_G \in \mathbb{N}^{n \times n}$. Square matrices over a semiring also have notion of a star, given by:

$$A^* = \sum_{n=0}^{\infty} A^n \in \overline{\mathbb{N}}^{n \times n}, \quad \text{where } \overline{\mathbb{N}} := \mathbb{N} \cup \{\infty\}.$$

And, yet again, we have $\#G^*(i, j) = (\mathbb{A}_G^*)_{i,j}$

What about hypergraphs? The free category generated by a directed hypergraph (N, \mathcal{A}) is the free category generated by

Definition 2.0.13.

□

2.0.6 Measures and Probabilities

Definition 2.0.14 (Measurable Space). A measurable space is a pair (X, \mathcal{F}_X) , where X is a set, and \mathcal{F}_X is a sigma-algebra over X , which is to say a set of subsets of X , containing the empty set, and closed under countable union, intersection, and complement with respect to X . The elements of \mathcal{F}_X are referred to as measurable sets. \square

Definition 2.0.15 (Measure). A *measure* λ over a measurable space (X, \mathcal{F}) is a function $\lambda : \mathcal{F} \rightarrow \mathbb{R} \cup \{\infty\}$, with the following properties.

- **Null Empty Set.** $\lambda(\emptyset) = 0$.
- **Non-negativity.** $\lambda(U) \geq 0$ for all $U \in \mathcal{F}$.
- **Countable additivity.** For every countable collection $\{U_i\}_{i=1,2,\dots}$ of pairwise disjoint measurable sets ($U_i \in \mathcal{F}$), we have $\sum_i \lambda(U_i) = \lambda(\sqcup_i U_i)$.

\square

Definition 2.0.16. Consider a measure λ on a measurable space $\mathcal{X} = (X, \mathcal{F})$.

1. If $\lambda(X) = 1$, then λ is a *probability* measure.
2. If \mathcal{T} is a topology on X , and $\lambda(U) > 0$ for every non-empty open set $U \in \mathcal{F} \cap \mathcal{T}$, then λ is said to be *strictly positive* (wrt \mathcal{T}).
3. The measure λ is called σ -*finite* if X can be covered with a countable set of sets with finite measures — that is, if there exist countable sequences $(A_1, A_2, \dots) \subset \mathcal{F}$ such that each $\lambda(A_i) < \infty$ is finite, and $X = \bigcup_{i=1}^{\infty} A_i$. \square

Definition 2.0.17 (Measurable Functions). If $\mathcal{X} = (X, \mathcal{F})$, and $\mathcal{Y} = (Y, \mathcal{G})$ are two measurable spaces, a measurable function $f : \mathcal{X} \rightarrow \mathcal{Y}$ is a function $f : X \rightarrow Y$ on the underlying spaces, such that $f^{-1}(U) \in \mathcal{F}$ for every $U \in \mathcal{G}$. \square

It is easy to verify that identity maps are measurable, and that, when f and g are measurable, so is $f \circ g$. It follows that there is a category Meas whose objects are measurable spaces, and whose maps are measurable functions.

Definition 2.0.18 (absolute continuity). If μ and ν are measures over a space (X, \mathcal{F}) , we say that μ is absolutely continuous with respect to ν , denoted $\nu \ll \mu$, if, for every $U \in \mathcal{F}$ such that $\nu(U) = 0$, we also have $\mu(U) = 0$. \square

Definition 2.0.19 (Radon-Nikodym Derivative). Suppose μ and ν are both measures over a measurable space (X, \mathcal{F}) , and $\mu \ll \nu$. The Radon-Nikodym theorem states that there is then a unique \mathcal{F} -measurable function f such that for all $A \in \mathcal{F}$,

$$\mu(A) = \int_A f d\nu.$$

The function f is called the Radon-Nikodym derivative, and denoted $\frac{d\mu}{d\nu} := f$. \square

Definition 2.0.20 (Markov Kernels). If $\mathcal{X} = (X, \mathcal{F})$, and $\mathcal{Y} = (Y, \mathcal{G})$ are two measurable spaces, a Markov Kernel $\kappa : \mathcal{X} \rightarrow \mathcal{Y}$, which we sometimes write as “ $\kappa(Y|X)$ ”, is a function $\kappa : X \times \mathcal{G} \rightarrow \mathbb{R}$, such that

1. For every $x \in X$, the map $\kappa(x, -) : \mathcal{U} \rightarrow [0, 1]$ is a probability measure. (So κ is also a cpd.)
2. For every $U \in \mathcal{G}$, the map $\kappa(-, U) : X \rightarrow [0, 1]$ is a measurable function from \mathcal{X} to the Borell space $[0, 1]$. Or more explicitly: for every open set $S \subseteq [0, 1]$, and $U \in \mathcal{G}$, we have that $\{x : \kappa(x, U) \in S\} \in \mathcal{F}$.

□

Definition 2.0.21 (category of measurable spaces and Markov kernels). Let \mathbb{Stoch} be the category whose objects are measurable topological spaces with a base measure, and whose morphisms are Marov kernels that are absolutely continuous with respect to the base measure. Concretely, the objects of \mathbb{Stoch} are pairs (\mathcal{X}, λ) , where \mathcal{X} is a measurable topological space, and λ is a strictly positive and σ -finite measure on \mathcal{X} . The collection of morphisms from $(X, \mathcal{F}_X, \lambda_X)$ to $(Y, \mathcal{F}_Y, \lambda_Y)$ is the set of Markov Kernels $\kappa : \mathcal{X} \rightarrow \mathcal{Y}$ such that $\kappa(x, -) \ll \lambda_Y$ for all x . The reason we require this is so that the Radon-Nikodym derivative $\frac{d\kappa(x)}{d\lambda}$, i.e., the unique \mathcal{F}_Y -measurable function satisfying

$$\forall x. \forall A \in \mathcal{F}_Y. \quad \kappa(x, A) = \int_A \frac{d\kappa(x)}{d\lambda} d\lambda, \quad \text{exists.}$$

Composition in \mathbb{Stoch} is given by Lebesgue Integration: for Markov Kernels $p(Y|X) : \mathcal{X} \rightarrow \mathcal{Y}$ and $q(Z|Y) : \mathcal{Y} \rightarrow \mathcal{Z}$, define $(p \circ q) : \mathcal{X} \rightarrow \mathcal{Z}$ (i.e., $p \circ q : X \times \mathcal{F}_Z \rightarrow [0, 1]$) by:

$$(p \circ q)(x, U) := \int_{\mathcal{Y}} q(-, U) dp(x, -).$$

This typechecks because $q(-, U)$ is a \mathcal{Y} -measurable function, and $p(x, -)$ is a measure on \mathcal{Y} . We must also prove that the result is a Markov Kernel, which we do below. The identities are given by

$$\text{id}_{\mathcal{X}}(x, U) = \begin{cases} 1 & \text{if } x \in U \\ 0 & \text{otherwise} \end{cases}.$$

These functions are clearly identities, but are they Markov kernels, and can they be made absolutely continuous with respect to our base measure?

More explicitly:

$$\text{id}_{\mathcal{X}}(x, -) \ll \lambda_X \iff (\lambda_X(A) = 0 \Rightarrow x \notin A),$$

and so there is a problem if we can find $A \subset \mathcal{V}X$ with $\lambda_X(A) = 0$ but $x \in A$. Or equivalently, a non-empty subset $A \subseteq \mathcal{V}X$ that has measure zero. By strict positivity, this means A cannot be an open set.

In the discrete case, in which every variable can take a finite set of values, and every subset is measurable and clopen, this is not a problem so long as the base measure gives every element positive probability.

□

2.0.7 Independencies

2.0.8 Information Theory

2.0.9 Graphical Models

There are two aspects any graphical model: a “qualitative/structural” aspect, which describes influences between variables, and a “quantitative/observational” aspect, that annotates those influences with data.

A qualitative BN, for example, is a directed graph whose semantics are given in terms of independencies: any variable X is independent of its non-descendents, given the values of its parents, $\text{Pa } X$. A quantitative BN, then, includes both that directed graph, and also each variable X to a conditional probability distribution $\text{Pr}_X(X|\text{Pa } X)$.

Part I

A Universal Modeling Language

CHAPTER 3

PROBABILISTIC DEPENDENCY GRAPHS (PDGS)

RELATIVE ENTROPY SOUP

Oliver E. Richardson, Ph.D.

Cornell University 2024

We introduce Probabilistic Dependency Graphs (PDGs), a new class of directed graphical models. PDGs can capture inconsistent beliefs in a natural way and are more modular than Bayesian Networks (BNs), in that they make it easier to incorporate new information and restructure the representation. We show by example how PDGs are an especially natural modeling tool. We provide three semantics for PDGs, each of which can be derived from a scoring function (on joint distributions over the variables in the network) that can be viewed as representing a distribution's incompatibility with the PDG. For the PDG corresponding to a BN, this function is uniquely minimized by the distribution the BN represents, showing that PDG semantics extend BN semantics. We show further that factor graphs and their exponential families can also be faithfully represented as PDGs, while there are significant barriers to modeling a PDG with a factor graph.

3.1 Introduction

In this paper we introduce yet another graphical tool for modeling beliefs, *Probabilistic Dependency Graphs* (PDGs). There are already many such models in the literature, including Bayesian networks (BNs) and factor graphs. (For an overview, see Koller and Friedman [55].) Why does the world need one more?

Our original motivation for introducing PDGs was to be able capture inconsistency. We want to be able to model the process of resolving inconsistency; to do so, we have to model the inconsistency itself. But our approach to modeling inconsistency has many other advantages. In particular, PDGs are significantly more modular than other directed graphical models: operations like restriction and union that are easily done with PDGs are difficult or impossible to do with other representations. The following examples motivate PDGs and illustrate some of their advantages.

Example 1. Grok is visiting a neighboring district. From prior reading, she thinks it likely (probability .95) that guns are illegal here. Some brief conversations with locals lead her to believe that, with probability .1, the law prohibits floomps.

The obvious way to represent this as a BN is to use two random variables F and G (respectively taking values $\{f, \neg f\}$ and $\{g, \neg g\}$), indicating whether floomps and guns are prohibited. The semantics of a BN offer her two choices: either assume that F and G to be independent and give (unconditional) probabilities of F and G , or choose a direction of dependency, and give one of the two unconditional probabilities and a conditional probability distribution. As there is no reason to choose either direction of dependence, the natural choice is to assume independence, giving her the BN on the left of [Figure 3.1](#).

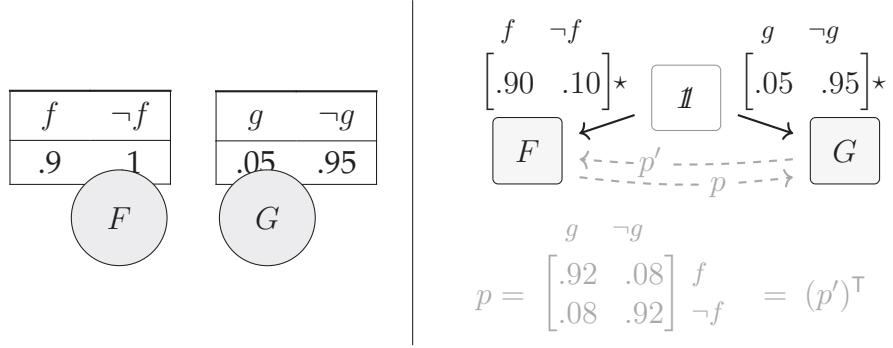


Figure 3.1: A BN (left) and corresponding PDG (right), which can be augmented with additional cpds. The cpds p and/or p' make it inconsistent.

A traumatic experience a few hours later leaves Grok believing that “floomp” is likely (probability .92) to be another word for gun. Let $p(G \mid F)$ be the conditional probability distribution (cpd) that describes the belief that if floomps are legal (resp., illegal), then with probability .92, guns are as well, and $p'(F \mid G)$ be the reverse. Starting with p , Grok’s first instinct is to simply incorporate the conditional information by adding F as a parent of G , and then associating the cpd p with G . But then what should she do with the original probability she had for G ? Should she just discard it? It is easy to check that there is no joint distribution that is consistent with both the two original priors on F and G and also p . So if she is to represent the information with a BN, which always represents a consistent distribution, she must resolve the inconsistency.

However, sorting this out immediately may not be ideal. For instance, if the inconsistency arises from a conflation between two definitions of “gun”, a resolution will have destroyed the original cpds. A better use of computation may be to notice the inconsistency and look up the actual law.

By way of contrast, consider the corresponding PDG. In a PDG, the cpds are attached to edges, rather than nodes of the graph. In order to represent unconditional probabilities, we introduce a *unit variable* \mathbb{I} which takes only one

value, denoted \star . This leads Grok to the PDG depicted in Figure 3.1, where the edges from $\mathbb{1}$ to F and G are associated with the unconditional probabilities of F and G , and the edges between F and G are associated with p and p' .

The original state of knowledge consists of all three nodes and the two solid edges from $\mathbb{1}$. This is like Bayes Net that we considered above, except that we no longer explicitly take F and G to be independent; we merely record the constraints imposed by the given probabilities.

The key point is that we can incorporate the new information into our original representation (the graph in Figure 3.1 without the edge from F to G) simply by adding the edge from F to G and the associated cpd p (the new information is shown in blue). Doing so does not change the meaning of the original edges. Unlike a Bayesian update, the operation is even reversible: all we need to do recover our original belief state is delete the new edge, making it possible to mull over and then reject an observation. \triangle

The ability of PDGs to model inconsistency, as illustrated in Example 1, appears to have come at a significant cost. We seem to have lost a key benefit of BNs: the ease with which they can capture (conditional) independencies, which, as Pearl (1988) has argued forcefully, are omnipresent.

Example 2 (emulating a BN). We now consider the classic (quantitative) Bayesian network \mathcal{B} , which has four binary variables indicating whether a person (C) develops cancer, (S) smokes, (SH) is exposed to second-hand smoke, and (PS) has parents who smoke, presented graphically in Figure 3.2a. We now walk through what is required to represent \mathcal{B} as a PDG, which we call $pdg(\mathcal{B})$, shown as the solid nodes and edges in Figure 3.2b.

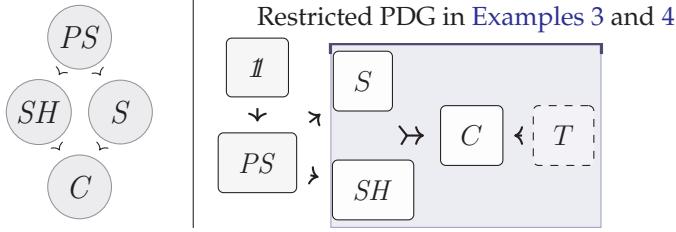


Figure 3.2: (a) The Bayesian Network \mathcal{B} in Example 2 (left), and (b) $pdg(\mathcal{B})$, its corresponding PDG (right). The shaded box indicates a restriction of $pdg(\mathcal{B})$ to only the nodes and edges it contains, and the dashed node T and its arrow to C can be added in the PDG, without taking into account S and SH .

We start with the nodes corresponding to the variables in \mathcal{B} , together with the special node \mathbb{I} from Example 1; we add an edge from \mathbb{I} to PS , to which we associate the unconditional probability given by the cpd for PS in \mathcal{B} . We can also re-use the cpds for S and SH , assigning them, respectively, to the edges $PS \rightarrow S$ and $PS \rightarrow SH$ in $pdg(\mathcal{B})$. There are two remaining problems: (1) modeling the remaining table in \mathcal{B} , which corresponds to the conditional probability of C given S and SH ; and (2) recovering the additional conditional independence assumptions in the BN.

For (1), we cannot just add the edges $S \rightarrow C$ and $SH \rightarrow C$ that are present in \mathcal{B} . As we saw in Example 1, this would mean supplying two *separate* tables, one indicating the probability of C given S , and the other indicating the probability of C given SH . We would lose significant information that is present in \mathcal{B} about how C depends jointly on S and SH . To distinguish the joint dependence on S and SH , for now, we draw an edge with two tails—a (directed) *hyperedge*—that completes the diagram in Figure 3.2b. With regard to (2), there are many distributions consistent with the conditional marginal probabilities in the cpds, and the independences presumed by \mathcal{B} need not hold for them. Rather than trying to distinguish between them with additional constraints, we develop a scoring-function semantics for PDGs which is in this case uniquely minimized

by the distribution specified by \mathcal{B} ([Theorem 3.4.1](#)). This allows us to recover the semantics of Bayesian networks without requiring the independencies that they assume.

Next suppose that we get information beyond that captured by the original BN. Specifically, we read a thorough empirical study demonstrating that people who use tanning beds have a 10% incidence of cancer, compared with 1% in the control group (call the cpd for this p); we would like to add this information to \mathcal{B} . The first step is clearly to add a new node labeled T , for “tanning bed use”. But simply making T a parent of C (as clearly seems appropriate, given that the incidence of cancer depends on tanning bed use) requires a substantial expansion of the cpd; in particular, it requires us to make assumptions about the interactions between tanning beds and smoking. The corresponding PDG, $p\text{dg}(\mathcal{B})$, on the other hand, has no trouble: We can simply add the node T with an edge to C that is associated with \mathbb{P} . But note that doing this makes it possible for our knowledge to be inconsistent. To take a simple example, if the distribution on C given S and H encoded in the original cpd was always deterministically “has cancer” for every possible value of S and H , but the distribution according to the new cpd from T was deterministically “no cancer”, the resulting PDG would be inconsistent. \triangle

We have seen that we can easily add information to PDGs; removing information is equally painless.

Example 3 (restriction). After the Communist party came to power, children were raised communally, and so parents’ smoking habits no longer had any impact on them. Grok is reading her favorite book on graphical models, and she realizes that while the node PS in [Figure 3.2a](#) has lost its usefulness, and nodes S and SH

no longer ought to have PS as a parent, the other half of the diagram—that is, the node C and its dependence on S and SH —should apply as before. Grok has identified two obstacles to modeling deletion of information from a BN by simply deleting nodes and their associated cpds. First, this restricted model is technically no longer a BN (which in this case would require unconditional distributions on S and SH), but rather a *conditional* BN [55], which allows for these nodes to be marked as observations; observation nodes do not have associated beliefs. Second, even regarded as a conditional BN, the result of deleting a node may introduce *new* independence information, incompatible with the original BN. For instance, by deleting the node B in a chain $A \rightarrow B \rightarrow C$, one concludes that A and C are independent, a conclusion incompatible with the original BN containing all three nodes. PDGs do not suffer from either problem. We can easily delete the nodes labeled 1 and PS in [Figure 3.2b](#) to get the restricted PDG shown in the figure, which captures Grok’s updated information. The resulting PDG has no edges leading to S or SH , and hence no distributions specified on them; no special modeling distinction between observation nodes and other nodes are required. Because PDGs do not directly make independence assumptions, the information in this fragment is truly a subset of the information in the whole PDG.

△

Being able to form a well-behaved local picture and restrict knowledge is useful, but an even more compelling reason to use PDGs is their ability to aggregate information.

Example 4. Grok dreams of becoming Supreme Leader (SL), and has come up with a plan. She has noticed that people who use tanning beds have significantly more power than those who don’t. Unfortunately, her mom has always told her

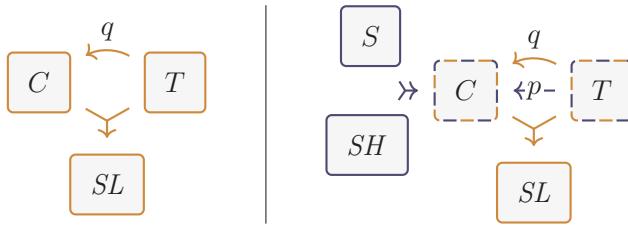


Figure 3.3: Grok’s prior (left) and combined (right) knowledge.

that tanning beds cause cancer; specifically, that 15% of people who use tanning beds get it, compared to the baseline of 2%. Call this cpd q . Grok thinks people will make fun of her if she uses a tanning bed and gets cancer, making becoming Supreme Leader impossible. This mental state is depicted as a PDG on the left of Figure 3.3.

Grok is reading about graphical models because she vaguely remembers that the variables in Example 2 match the ones she already knows about. When she finishes reading the statistics on smoking and the original study on tanning beds (associated to a cpd \mathbb{P} in Example 2), but before she has time to reflect, we can represent her (conflicted) knowledge state as the union of the two graphs, depicted graphically on the right of Figure 3.3.

The union of the two PDGs, even with overlapping nodes, is still a PDG. This is not the case in general for BNs. Note that the PDG that Grok used to represent her two different sources of information (the mother’s wisdom and the study) regarding the distribution of C is a *multigraph*: there are two edges from T to C , with inconsistent information. Had we not allowed multigraphs, we would have needed to choose between the two edges, or represent the information some other (arguably less natural) way. As we are already allowing inconsistency, merely recording both is much more in keeping with the way we have handled other types of uncertainty. \triangle

Not all inconsistencies are equally egregious. For example, even though the cpds p and q are different, they are numerically close, so, intuitively, the PDG on the right in Figure 3.3 is not very inconsistent. Making this precise is the focus of Section 3.3.2.

These examples give a taste of the power of PDGs. In the coming sections, we formalize PDGs and relate them to other approaches.

3.2 Syntax

We now provide formal definitions for PDGs. Although it is possible to formalize PDGs with hyperedges directly, we opt for a different approach here, in which PDGs have only regular edges, and hyperedges are captured using a simple construction that involves adding an extra node.¹

Definition 3.2.1. A *Probabilistic Dependency Graph* is a tuple $\mathcal{m} = (\mathcal{N}, \mathcal{A}, \mathcal{V}, \mathbb{P}, \alpha, \beta)$, where

$\mathcal{N} : \text{Set}$ is a finite set of nodes, corresponding to variables;

$\mathcal{A} \subseteq \mathcal{N} \times \mathcal{N} \times \text{Label}$ is a set of labeled edges $\{X \xrightarrow{a} Y\}$, each with a source S and target T in \mathcal{N} ;

$\mathcal{V} : \mathcal{N} \rightarrow \text{Set}$ associates each variable $N \in \mathcal{N}$ with a set $\mathcal{V}(N)$ of values that the variable N can take;

¹In the factor graph literature, especially with regard to loopy belief propagation [?], it is common to call a collection of marginals that are not necessarily all compatible with a distribution *pseudomarginals*, making a PDG in some sense a collection of ‘conditional’ pseudomarginals. This gives an alternate expansion of “PDG” as “Pseudomarginal Dependency Graph”, with nomenclature rooted in the literature.

$\mathbb{P} : ((A, B, \ell) : \mathcal{A}) \rightarrow \mathcal{V}(A) \rightarrow \Delta \mathcal{V}(B)$ associates to each edge $X \xrightarrow{a} Y \in \mathcal{A}$ a distribution $\mathbb{P}_a(x)$ on Y for each $x \in \mathcal{V}(X)$;

$\alpha : \mathcal{A} \rightarrow [0, 1]$ associates to each edge $X \xrightarrow{a} Y$ a non-negative number α_L which, roughly speaking, is the modeler's confidence in the functional dependence of Y on X implicit in L ;

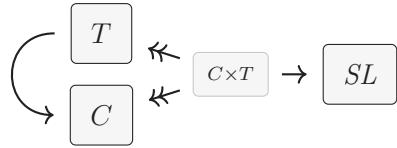
$\beta : \mathcal{A} \rightarrow \mathbb{R}^+$ associates to each edge L a positive real number β_L , the modeler's subjective confidence in the reliability of \mathbb{P} .

Note that we allow multiple edges in \mathcal{A} with the same source and target; thus $(\mathcal{N}, \mathcal{A})$ is a multigraph. We occasionally write a PDG as $\mathbf{m} = (\mathcal{G}, \mathbb{P}, \alpha, \beta)$, where $\mathcal{G} = (\mathcal{N}, \mathcal{A}, \mathcal{V})$, and abuse terminology by referring to \mathcal{G} as a multigraph. We refer to $\mathbf{n} = (\mathcal{G}, \mathbb{P})$ as an *unweighted* PDG, and give it semantics as though it were the (weighted) PDG $(\mathcal{G}, \mathbb{P}, 1, 1)$, where 1 is the constant function (i.e., so that $\alpha_L = \beta_L = 1$ for all L). In this paper, with the exception of [Section 3.4.3](#), we implicitly take $\alpha = 1$ and omit α , writing $\mathbf{m} = (\mathcal{G}, \mathbb{P}, \beta)$.² □

If \mathbf{m} is a PDG, we reserve the names $\mathcal{N}^m, \mathcal{A}^m, \dots$, for the components of \mathbf{m} , so that we may reference one without naming them all explicitly. We write $\mathcal{V}(S)$ for the set of possible joint settings of a set S of variables, and write $\mathcal{V}(\mathbf{m})$ for all settings of the variables in \mathcal{N}^m ; we refer to these settings as "worlds". While the definition above is sufficient to represent the class of all legal PDGs, we often use two additional bits of syntax to indicate common constraints: the special variable $\mathbb{1}$ such that $\mathcal{V}(\mathbb{1}) = \{\star\}$ from [Examples 1](#) and [2](#), and double-headed arrows, $A \twoheadrightarrow B$, which visually indicate that the corresponding cpd is degenerate, effectively representing a deterministic function $f : \mathcal{V}(A) \rightarrow \mathcal{V}(B)$.

²The appendix gives results for arbitrary α .

Construction 3.2.2. We can now explain how we capture the multi-tailed edges that were used in Examples 2 to 4. That notation can be viewed as shorthand for the graph that results by adding a new node at the junction representing the joint value of the nodes at the tails, with projections going back. For instance, the diagram displaying Grok’s prior knowledge in Example 4, on the left of Figure 3.3 is really shorthand for the following PDG, where we insert a node labeled $C \times T$ at the junction:



As the notation suggests, $\mathcal{V}(C \times T) = \mathcal{V}(C) \times \mathcal{V}(T)$. For any joint setting $(c, t) \in \mathcal{V}(C \times T)$ of both variables, the cpd for the edge from $C \times T$ to C gives probability 1 to c ; similarly, the cpd for the edge from $C \times T$ to T gives probability 1 to t . \square

3.3 Semantics

Although the meaning of an individual cpd is clear, we have not yet given PDGs a “global” semantics. We discuss three related approaches to doing so. The first is the simplest: we associate with a PDG the set of distributions that are consistent with it. This set will be empty if the PDG is inconsistent. The second approach associates a PDG with a scoring function, indicating the fit of an arbitrary distribution μ , and can be thought of as a *weighted* set of distributions [36]. This approach allows us to distinguish inconsistent PDGs, while the first approach does not. The third approach chooses the distributions with the best

score, typically associating with a PDG a unique distribution.

3.3.1 PDGs As Sets Of Distributions

We have been thinking of a PDG as a collection of constraints on distributions, specified by matching cpds. From this perspective, it is natural to consider the set of all distributions that are consistent with the constraints.

Definition 3.3.1. If \mathcal{m} is a PDG (weighted or unweighted) with edges \mathcal{A} and cpds \mathbb{P} , let $\{\mathcal{m}\}$ be the set of *distributions* over the variables in \mathcal{m} whose conditional marginals are exactly those given by \mathbb{P} . That is, $\mu \in \{\mathcal{m}\}$ iff, for all edges $a \in \mathcal{A}$ from X to Y , $x \in \mathcal{V}(X)$, and $y \in \mathcal{V}(Y)$, we have that $\mu(Y=y \mid X=x) = \mathbb{P}_a(T|s)$.

Formally,

$$\{\mathcal{m}\} = \left\{ \mu \in \Delta^{\mathcal{V}(\mathcal{m})} \middle| \begin{array}{l} \mu(B=b \mid A=a) \geq \mathbb{P}_a(b \mid a) \\ \forall (A, B, \ell) \in \mathcal{A}, a \in \mathcal{V}(A), b \in \mathcal{V}(B) \end{array} \right\}$$

\mathcal{m} is *inconsistent* if $\{\mathcal{m}\} = \emptyset$, and *consistent* otherwise. □

Note that $\{\mathcal{m}\}$ is independent of the weights α and β .

3.3.2 PDGs As Distribution Scoring Functions

We now generalize the previous semantics by viewing a PDG \mathcal{m} as a *scoring function* that, given an arbitrary distribution μ on $\mathcal{V}(\mathcal{m})$, returns a real-valued score indicating how well μ fits \mathcal{m} . Distributions with the lowest (best) scores are those that most closely match the cpds in \mathcal{m} , and contain the fewest unspecified correlations.

We start with the first component of the score, which assigns higher scores to distributions that require a larger perturbation in order to be consistent with \mathbf{m} . We measure the magnitude of this perturbation with relative entropy. In particular, for an edge $X \xrightarrow{a} Y$ and $x \in \mathcal{V}(X)$, we measure the relative entropy from $\mathbb{P}_a(x)$ to $\mu(Y = \cdot | X = x)$, and take the expectation over μ_X (that is, the marginal of μ on X). We then sum over all the edges L in the PDG, weighted by their reliability.

Definition 3.3.2. For a PDG \mathbf{m} , the *incompatibility* of a joint distribution μ over $\mathcal{V}(\mathbf{m})$, is given by

$$Inc_{\mathbf{m}}(\mu) := \sum_{ALX \atop Y \in \mathcal{A}^{\mathbf{m}}} \beta_L^{\mathbf{m}} \mathbb{E}_{x \sim \mu_X} \left[D\left(\mu(Y | X=x) \parallel \mathbb{P}_a^{\mathbf{m}}(x)\right) \right],$$

where $D(\mu \parallel \nu) = \sum_{w \in \text{Supp}(\mu)} \mu(w) \log \frac{\mu(w)}{\nu(w)}$ is the relative entropy from ν to μ . The *inconsistency* of PDG $\mathbf{m} = (\mathcal{N}, \mathcal{A}, \mathbb{P}, \alpha, \beta)$, denoted $Inc(\mathbf{m})$, is the minimum possible incompatibility of \mathbf{m} with any distribution μ ,

$$Inc(\mathbf{m}) = \inf_{\mu \in \Delta[\mathcal{W}_{\mathcal{V}}]} Inc_{\mathbf{m}}(\mu).$$

□

The idea behind this definition of inconsistency is that we want to choose a distribution μ that minimizes the total number of bits required to encode all of the relevant conditional marginals. More precisely, fix a distribution μ . For each edge $L = (X, Y, \ell) \in \mathcal{A}$ and $x \in \mathcal{V}(X)$, we are given a code for Y optimized for the distribution $\mathbb{P}_a(x)$, and asked to transmit data from $\mu(Y | x)$; we incur a cost for each bit required beyond what we would have used had we used a code optimized for the actual distribution $\mu(Y | X = x)$. To obtain the cost for L , we take a weighted average of these costs, where the weight for the value x is the probability $\mu_X(x)$. We do this for every edge $L \in \mathcal{A}$, summing the cost.

For even more intuition, imagine two agents (A and B) with identical beliefs described by a PDG \mathcal{M} about a set of variables that are in fact distributed according to μ . For each edge $L = (X, Y, \ell) \in \mathcal{A}^{\mathcal{M}}$, values $x, y \in \mathcal{V}(X)$ are chosen according to μ_{XY} and x is given to both agents.

At this point, the agents, having the same conditional beliefs, and the same information about Y , agree on the optimal encoding of the possible values of Y as sequences of bits, so that if y were drawn from $\mathbb{P}_a(x)$, the fewest number of bits would be needed to communicate it in expectation. The value of y —which is distributed not according to $\mathbb{P}_a(x)$, but $\mu(Y | X = x)$ —is now given to agent A. The agents pay a cost equal the number of bits needed to encode y according to the agreed-upon optimal code, but reimbursed the (smaller) cost that would have been paid, had the agents beliefs lined up with the true distribution μ .

Repeating for each edge and summing the expectations of these costs, we can view $Inc_{\mathcal{M}}(\mu)$ as the total number of *additional* expected bits required to communicate y with a code optimized for $\mathbb{P}_a(x)$ instead of the true conditional distribution $\mu(Y | X = x)$.

If \mathcal{M} is inconsistent, then there will be a cost no matter what distribution μ is the true distribution. Conversely, if \mathcal{M} is consistent, then any distribution $\mu \in \{\mathcal{M}\}$ will have $Inc_{\mathcal{M}}(\mu) = 0$.

Example ?? (continuing from p. ??). Recall our simplest example, which directly encodes an entire distribution p over the set W . In this case, there is only one edge, the expectation is over a single element, and the marginal on W is the entire distribution. Therefore, $Inc(\mathcal{M}; \mu) = D(\mu \| \mu)$, so the inconsistency is just the information μ and p , so is minimized uniquely when μ is p △

$\{\mathcal{M}\}$ and $Inc_{\mathcal{M}}$ distinguish between distributions based on their compatibility with \mathcal{M} , but even among distributions that match the marginals, some more closely match the qualitative structure of the graph than others. We think of each edge $X \xrightarrow{\alpha} Y$ as representing a qualitative claim (with confidence α_L) that the value of Y can be computed from X alone. To formalize this, we require only the multigraph $\mathcal{G}^{\mathcal{M}}$.

Given a multigraph G and distribution μ on its variables, contrast the amount of information required to

- (a) directly describe a joint outcome w drawn from μ , and
- (b) separately specify, for each edge $X \xrightarrow{\alpha} Y$, the value w_Y (of Y in world w) given the value w_X , in expectation.

If (a) = (b), a specification of (b) has exactly the same length as a full description of the world. If (b) > (a), then there are correlations in μ that allow for a more compact representation than G provides. The larger the difference, the more information is needed to determine targets Y beyond the conditional probabilities associated with the edges $X \rightarrow Y$ leading to Y (which according to G should be sufficient to compute them), and the poorer the qualitative fit of μ to G . Finally, if (a) > (b), then μ requires additional information to specify, beyond what is necessary to determine outcomes of the marginals selected by G .

Definition 3.3.3. For a multigraph $G = (\mathcal{N}, \mathcal{A}, \mathcal{V})$ over a set \mathcal{N} of variables, define the G -information deficiency of distribution μ , denoted $IDef_G(\mu)$, by considering the difference between (a) and (b), where we measure the amount of information

needed for a description using entropy:

$$IDef_G(\mu) := \sum_{(X,Y) \in \mathcal{A}} H_\mu(Y | X) - H(\mu). \quad (3.1)$$

(Recall that $H_\mu(Y | X)$, the (μ) -conditional entropy of Y given X , is defined as $-\sum_{x,y \in \mathcal{V}(X,Y)} \mu(x,y) \log \mu(y | x)$.) For a PDG \mathbf{m} , we take $IDef_{\mathbf{m}} = IDef_{\mathcal{G}\mathbf{m}}$. \square

We illustrate $IDef_{\mathbf{m}}$ with some simple examples. Suppose that \mathbf{m} has two nodes, X and Y . If \mathbf{m} has no edges, the $IDef_{\mathbf{m}}(\mu) = -H(\mu)$. There is no information required to specify, for each edge in \mathbf{m} from X to Y , the value w_Y given w_X , since there are no edges. Since we view smaller numbers as representing a better fit, $IDef_{\mathbf{m}}$ in this case will prefer the distribution that maximizes entropy. If \mathbf{m} has one edge from X to Y , then since $H(\mu) = H_\mu(Y | X) + H_\mu(X)$ by the well known *entropy chain rule* [63], $IDef_{\mathbf{m}}(\mu) = -H_\mu(X)$. Intuitively, while knowing the conditional probability $\mu(Y | X)$ is helpful, to completely specify μ we also need $\mu(X)$. Thus, in this case, $IDef_{\mathbf{m}}$ prefers distributions that maximize the entropy of the marginal on X . If \mathbf{m} has sufficiently many parallel edges from X to Y and $H_\mu(Y | X) > 0$ (so that Y is not totally determined by X) then we have $IDef_{\mathbf{m}}(\mu) > 0$, because the redundant edges add no information, but there is still a cost to specifying them. In this case, $IDef_{\mathbf{m}}$ prefers distributions that make Y a deterministic function of X will maximizing the entropy of the marginal on X . Finally, if \mathbf{m} has an edge from X to Y and another from Y to X , then a distribution μ minimizes $IDef_{\mathbf{m}}$ when X and Y vary together (so that $H_\mu(Y | X) = H_\mu(X | Y) = 0$) while maximizing $H(\mu)$, for example, by taking $\mu(0,0) = \mu(1,1) = 1/2$.

$Inc_{\mathbf{m}}(\mu)$ and $IDef_{\mathbf{m}}(\mu)$ give us two measures of compatibility between \mathbf{m} and a distribution μ . We take the score of interest to be their sum, with the tradeoff controlled by a parameter $\gamma \geq 0$:

$$[\![m]\!]_\gamma(\mu) := \text{Inc}_m(\mu) + \gamma \text{IDef}_m(\mu) \quad (3.2)$$

The following just makes precise that the scoring semantics generalizes the first semantics.

Proposition 3.3.1. $\{m\} = \{\mu : [\![m]\!]_0(\mu) = 0\}$ for all m .

link to
proof

While we focus on this particular scoring function in the paper, in part because it has deep connections to the free energy of a factor graph [55], other scoring functions may well end up being of interest.

3.3.3 PDGs As Unique Distributions

Finally, we provide an interpretation of a PDG as a probability distribution. Before we provide this semantics, we stress that this distribution does *not* capture all of the important information in the PDG—for example, a PDG can represent inconsistent knowledge states. Still, by giving a distribution, we enable comparisons with other graphical models, and show that PDGs are a surprisingly flexible tool for specifying distributions. The idea is to select the distributions with the best score. We thus define

$$[\![m]\!]^*_\gamma = \arg \min_{\mu \in \Delta \mathcal{V}(m)} [\![m]\!]_\gamma(\mu). \quad (3.3)$$

In general, $[\![m]\!]^*_\gamma$ does not give a unique distribution. But if γ is sufficiently small, then it does:

Proposition 3.3.2. If m is a PDG and $0 < \gamma \leq \min_L \beta_L^m$, then $[\![m]\!]^*_\gamma$ is a singleton.

link to
proof

In this paper, we are interested in the case where γ is small; this amounts to emphasizing the accuracy of the probability distribution as a description of probabilistic information, rather than the graphical structure of the PDG. This motivates us to consider what happens as γ goes to 0. If S_γ is a set of probability distributions for all $\gamma \in [0, 1]$, we define $\lim_{\gamma \rightarrow 0} S_\gamma$ to consist of all distributions μ such that there is a sequence $(\gamma_i, \mu_i)_{i \in \mathbb{N}}$ with $\gamma_i \rightarrow 0$ and $\mu_i \rightarrow \mu$ such that $\mu_i \in S_{\gamma_i}$ for all i . It can be further shown that

Proposition 3.3.3. *For all m , $\lim_{\gamma \rightarrow 0} \llbracket m \rrbracket_\gamma^*$ is a singleton.*

[link to
proof]

Let $\llbracket m \rrbracket^*$ be the unique element of $\lim_{\gamma \rightarrow 0} \llbracket m \rrbracket_\gamma^*$. The semantics has an important property:

Proposition 3.3.4. *$\llbracket m \rrbracket^* \in \llbracket m \rrbracket_0^*$, so if m is consistent, then $\llbracket m \rrbracket^* \in \{m\}$.*

[link to
proof]

3.4 Relationships to Other Graphical Models

We start by relating PDGs to two of the most popular graphical models: BNs and factor graphs. PDGs are strictly more general than BNs, and can emulate factor graphs for a particular value of γ .

3.4.1 Bayesian Networks

Construction 3.2.2 can be generalized to convert arbitrary Bayesian Networks into PDGs. Given a BN \mathcal{B} and a positive confidence β_X for the cpd of each variable X of \mathcal{B} , let $p\mathcal{dg}(\mathcal{B}, \beta)$ be the PDG comprising the cpds of \mathcal{B} in this way; we defer the straightforward formal details to the appendix.

Theorem 3.4.1. *If \mathcal{B} is a Bayesian network and $\text{Pr}_{\mathcal{B}}$ is the distribution it specifies, then for all $\gamma > 0$ and all vectors β such that $\beta_L > 0$ for all edges L , $\llbracket \mathbf{pdg}(\mathcal{B}, \beta) \rrbracket_{\gamma}^* = \{\text{Pr}_{\mathcal{B}}\}$, and thus $\llbracket \mathbf{pdg}(\mathcal{B}, \beta) \rrbracket^* = \text{Pr}_{\mathcal{B}}$.*

Theorem 3.4.1 is quite robust to parameter choices: it holds for every weight vector β and all $\gamma > 0$. However, it does lean heavily on our assumption that $\alpha = \mathbf{1}$, making it our only result that does not have a natural analog for general α .

3.4.2 Factor Graphs

Factor graphs [57], like PDGs, generalize BNs. In this section, we consider the relationship between factor graphs (FGs) and PDGs.

Definition 3.4.1. A *factor graph* Φ is a set of random variables $\mathcal{X} = \{X_i\}$ and factors $\{\phi_J: \mathcal{V}(X_J) \rightarrow \mathbb{R}_{\geq 0}\}_{J \in \mathcal{J}}$, where $X_J \subseteq \mathcal{X}$. More precisely, each factor ϕ_J is associated with a subset $X_J \subseteq \mathcal{X}$ of variables, and maps joint settings of X_J to non-negative real numbers. Φ specifies a distribution

$$\text{Pr}_{\Phi}(\vec{x}) = \frac{1}{Z_{\Phi}} \prod_{J \in \mathcal{J}} \phi_J(\vec{x}_J),$$

where \vec{x} is a joint setting of all of the variables, \vec{x}_J is the restriction of \vec{x} to only the variables X_J , and Z_{Φ} is the constant required to normalize the distribution. \square

The cpds of a PDG naturally constitute a collection of factors, so it is natural to wonder how the semantics of a PDG compares to simply treating the cpds as factors in a factor graph. To answer this, we start by making the translation precise.

Definition 3.4.2 (unweighted PDG to factor graph). If $\mathcal{N} = (\mathcal{G}, \mathbb{P})$ is an unweighted PDG, define the associated FG $\Phi_{\mathcal{N}}$ on the variables $(\mathcal{N}, \mathcal{V})$ by taking \mathcal{J} to be the set of edges, and for an edge L from Z to Y , taking $X_L = \{Z, Y\}$, and $\phi_L(z, y)$ to be $\mathbb{P}_a^m(y | z)$ (i.e., $(\mathbb{P}_a^m(z))(y)$). \square

It turns out we can also do the reverse. Using essentially the same idea as in [Construction 3.2.2](#), we can encode a factor graph as an assertion about the unconditional probability distribution over the variables associated to each factor.

Definition 3.4.3 (factor graph to unweighted PDG). For a FG Φ , let $updg(\Phi)$ be the unweighted PDG consisting of

- the variables in Φ together with $\mathbb{1}$ and a variable $X_J := \prod_{j \in J} X_j$ for every factor $J \in \mathcal{J}$, and
- edges $\mathbb{1} \rightarrow X_J$ for each J and $X_J \twoheadrightarrow X_j$ for each $X_j \in \mathbf{X}_J$,

where the edges $X_J \twoheadrightarrow X_j$ are associated with the appropriate projections, and each $\mathbb{1} \rightarrow X_J$ is associated with the unconditional joint distribution on X_J obtained by normalizing ϕ_J . The process is illustrated in [Figure 3.4](#). \square

PDGs are directed graphs, while factors graphs are undirected. The map from PDGs to factor graphs thus loses some important structure. As shown in [Figure 3.4](#), this mapping can change the graphical structure significantly. Nevertheless,

Theorem 3.4.2. $\Pr_{\Phi} = [updg(\Phi)]_1^*$ for all factor graphs Φ .³

(unproven!)

³Recall that we identify the unweighted PDG $(\mathcal{G}, \mathbb{P})$ with the weighted PDG $(\mathcal{G}, \mathbb{P}, \mathbf{1}, \mathbf{1})$.

(unproven!)

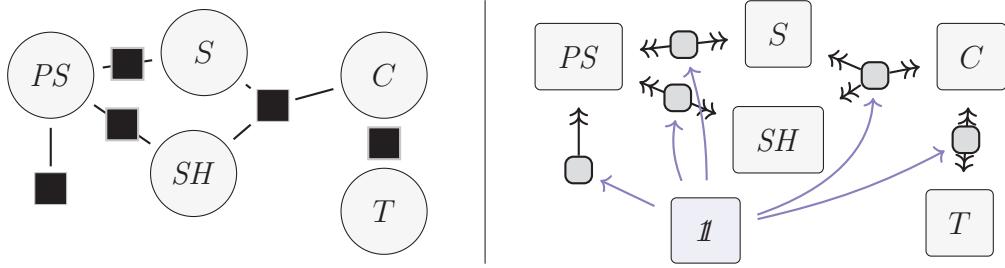


Figure 3.4: Conversion of the PDG in Example 2 to a factor graph according to Definition 3.4.2 (left), and from that factor graph back to a PDG by Definition 3.4.3 (right). In the latter, for each J we introduce a new variable X_J (displayed as a smaller darker rectangle), whose values are joint settings of the variables connected it, and also an edge $1 \rightarrow X_J$ (shown in blue), to which we associate the unconditional distribution given by normalizing ϕ_J .

Theorem 3.4.3. $[\mathcal{N}]_1^* = \Pr_{\Phi_n}$ for all unweighted PDGs \mathcal{N} .

The correspondence hinges on the fact that we take $\gamma = 1$, so that Inc and $IDef$ are weighted equally. Because the user of a PDG gets to choose γ , the fact that the translation from factor graphs to PDGs preserves semantics only for $\gamma = 1$ poses no problem. Conversely, the fact that the reverse correspondence requires $\gamma = 1$ suggests that factor graphs are less flexible than PDGs.

What about weighted PDGs $(\mathcal{G}, \mathbb{P}, \beta)$ where $\beta \neq 1$? There is also a standard notion of weighted factor graph, but as long as we stick with our convention of taking $\alpha = 1$, we cannot relate them to weighted PDGs. As we are about to see, once we drop this convention, we can do much more.

3.4.3 Factored Exponential Families

A *weighted factor graph* (WFG) Ψ is a pair (Φ, θ) consisting of a factor graph Φ together with a vector of non-negative weights $\{\theta_J\}_{J \in \mathcal{J}}$. Ψ specifies a canonical

scoring function

$$GFE_{\Psi}(\mu) := \mathbb{E}_{\vec{x} \sim \mu} \left[\sum_{J \in \mathcal{J}} \theta_J \log \frac{1}{\phi_J(\vec{x}_J)} \right] - H(\mu), \quad (3.4)$$

called the *variational Gibbs free energy* [64]. GFE_{Ψ} is uniquely minimized by the distribution $\Pr_{\Psi}(\vec{x}) = \frac{1}{Z_{\Psi}} \prod_{J \in \mathcal{J}} \phi_J(\vec{x}_J)^{\theta_J}$, which matches the unweighted case when every $\theta_J = 1$. The mapping $\theta \mapsto \Pr_{(\Phi, \theta)}$ is known as Φ 's *exponential family* and is a central tool in the analysis and development of many algorithms for graphical models [93].

PDGs can in fact capture the full exponential family of a factor graph, but only by allowing values of α other than 1. In this case, the only definition that requires alteration is *IDef*, which now depends on the *weighted multigraph* $(\mathcal{G}^m, \alpha^m)$, and is given by

$$IDef_m(\mu) := \sum_{X \xrightarrow{\alpha} Y \in \mathcal{A}} \alpha_L H_{\mu}(Y | X) - H(\mu). \quad (3.5)$$

Thus, the conditional entropy $H_{\mu}(Y | X)$ associated with the edge $X \xrightarrow{\alpha} Y$ is multiplied by the weight α_L of that edge.

One key benefit of using α is that we can capture arbitrary WFGs, not just ones with a constant weight vector. All we have to do is to ensure that in our translation from factor graphs to PDGs, the ratio α_L / β_L is a constant. (Of course, if we allow arbitrary weights, we cannot hope to do this if $\alpha_L = 1$ for all edges L .) We therefore define a family of translations, parameterized by the ratio of α_L to β_L .

Definition 3.4.4 (WFG to PDG). Given a WFG $\Psi = (\Phi, \theta)$, and positive number k , we define the corresponding PDG $p\mathcal{d}\mathcal{g}(\Psi, k) = (up\mathcal{d}\mathcal{g}(\Phi), \alpha_{\theta}, \beta_{\theta})$ by taking $\beta_J = k\theta_J$ and $\alpha_J = \theta_J$ for the edge $\mathbb{1} \rightarrow X_J$, and taking $\beta_L = k$ and $\alpha_L = 1$ for the projections $X_J \twoheadrightarrow X_j$. \square

We now extend Definitions 3.4.2 and 3.4.3 to (weighted) PDGs and WFGs. In translating a PDG to a WFG, there will necessarily be some loss of information: PDGs have two sets, while WFGs have only one. Here we throw out α and keep β , though in its role here as a left inverse of Definition 3.4.4, either choice would suffice.

Definition 3.4.5 (PDG to WFG). Given a (weighted) PDG $\mathbf{m} = (\mathbf{n}, \beta)$, we take its corresponding WFG to be $\Psi_{\mathbf{m}} := (\Phi_{\mathbf{n}}, \beta)$; that is, $\theta_L := \beta_L$ for all edges L . \square

We now show that we can capture the entire exponential family of a factor graph, and even its associated free energy, but only for γ equal to the constant k used in the translation.

Theorem 3.4.4. For all WFGs $\Psi = (\Phi, \theta)$ and all $\gamma > 0$, we have that $GFE_{\Psi} = \frac{1}{\gamma} [\mathbf{m}_{\Psi, \gamma}]_{\gamma} + C$ for some constant C , so \Pr_{Ψ} is the unique element of $[\mathbf{m}_{\Psi, \gamma}]_{\gamma}^*$.

In particular, for $k=1$, so that θ is used for both the functions α and β of the resulting PDG, Theorem 3.4.4 strictly generalizes Theorem 3.4.2.

Corollary 3.4.4.1. For all weighted factor graphs (Φ, θ) , we have that $\Pr_{(\Phi, \theta)} = [[\mathbf{updg}(\Phi), \theta, \theta]]_1^*$

Conversely, as long as the ratio of α_L to β_L is constant, the reverse translation also preserves semantics.

Theorem 3.4.5. For all unweighted PDGs \mathbf{n} and non-negative vectors \mathbf{v} over \mathcal{A}^n , and all $\gamma > 0$, we have that $[(\mathbf{n}, \mathbf{v}, \gamma \mathbf{v})]_{\gamma} = \gamma GFE_{(\Phi_{\mathbf{n}}, \mathbf{v})}$; consequently, $[(\mathbf{n}, \mathbf{v}, \gamma \mathbf{v})]_{\gamma}^* = \{\Pr_{(\Phi_{\mathbf{n}}, \mathbf{v})}\}$.

The key step in proving [Theorems 3.4.4](#) and [3.4.5](#) (and in the proofs of a number of other results) involves rewriting $\llbracket \mathbf{m} \rrbracket_\gamma$ as follows:

Proposition 3.4.6. *Letting x^w and y^w denote the values of X and Y , respectively, in $w \in \mathcal{V}(\mathbf{m})$, we have*

$$\begin{aligned} \llbracket \mathbf{m} \rrbracket(\mu) = \mathbb{E}_{w \sim \mu} \left\{ \sum_{X \xrightarrow{a} Y} \underbrace{\left[\beta_L \log \frac{1}{\mathbb{P}_a(y^w | x^w)} + (\alpha_L \gamma - \beta_L) \log \frac{1}{\mu(y^w | x^w)} \right]}_{\text{local regularization (if } \beta_L > \alpha_L \gamma \text{)}} - \underbrace{\gamma \log \frac{1}{\mu(w)}}_{\text{global regularization}} \right\}. \end{aligned} \quad (3.6)$$

For a fixed γ , the first and last terms of (3.6) are equal to a scaled version of the free energy, γGFE_Φ , if we set $\phi_J := \mathbb{P}_a$ and $\theta_J := \beta_L/\gamma$. If, in addition, $\beta_L = \alpha_L \gamma$ for all edges L , then the local regularization term disappears, giving us the desired correspondence.

[Equation \(3.6\)](#) also makes it clear that taking $\beta_L = \alpha_L \gamma$ for all edges L is essentially necessary to get [Theorems 3.4.2](#) and [3.4.3](#). Of course, fixed γ precludes taking the limit as γ goes to 0, so [Proposition 3.3.4](#) does apply. This is reflected in some strange behavior in factor graphs trying to capture the same phenomena as PDGs, as the following example shows.

Example 5. Consider the PDG \mathbf{m} containing just X and 1, and two edges $p, q : 1 \rightarrow X$. (Recall that such a PDG can arise if we get different information about the probability of X from two different sources; this is a situation we certainly want to be able to capture!) Consider the simplest situation, where p and q are both associated with the same distribution on X ; further suppose that the agent is certain about the distribution, so $\beta_p = \beta_q = 1$. For definiteness, suppose that $\mathcal{V}(X) = \{x_1, x_2\}$, and that the distribution associated with both edges is

$\mu_{.7}$, which ascribes probability .7 to x_1 . Then, as we would hope $[\![m]\!]^* = \{\mu_{.7}\}$; after all, both sources agree on the information. However, it can be shown that $\Pr_{\Psi_m} = \mu_{.85}$, so $[\![m]\!]_1^* = \{\mu_{.85}\}$. \triangle

Although both θ and β are measures of confidence, the way that the Gibbs free energy varies with θ is quite different from the way that the score of a PDG varies with β . The scoring function that we use for PDGs can be viewed as extending $GFE_{\Phi,\theta}$ by including the local regularization term. As γ approaches zero, the importance of the global regularization terms decreases relative to that of the local regularization term, so the PDG scoring function becomes quite different from Gibbs free energy.

3.5 Discussion

We have introduced PDGs, a powerful tool for representing probabilistic information. They have a number of advantages over other probabilistic graphical models.

- They allow us to capture inconsistency, including conflicting information from multiple sources with varying degrees of reliability.
- They are much more modular than other representations; for example, we can combine information from two sources by simply taking the union of two PDGs, and it is easy to add new information (edges) and features (nodes) without affecting previously-received information.
- They allow for a clean separation between quantitative information (the cpds and weights β) and more qualitative information contained by the

graph structure (and the weights α); this is captured by the terms Inc and $IDef$ in our scoring function.

- PDGs have (several) natural semantics; one of them allows us to pick out a unique distribution. Using this distribution, PDGs can capture BNs and factor graphs. In the latter case, a simple parameter shift in the corresponding PDG eliminates arguably problematic behavior of a factor graph.

We have only scratched the surface of what can be done with PDGs here. Two major issues that need to be tackled are inference and dynamics. How should we query a PDG for probabilistic information? How should we modify a PDG in light of new information or to make it more consistent? These issues turn out to be closely related. Due to space limitations, we just briefly give some intuitions and examples here.

Suppose that we want to compute the probability of Y given X in a PDG m . For a cpd $p(Y|X)$, let m^{+p} be the PDG obtained by associating p with a new edge in m from X to Y , with $\alpha_p = 0$. We judge the quality of a candidate answer p by the best possible score that m^{+p} gives to any distribution (which we call the *degree of inconsistency* of m^{+p}). It can be shown that the degree of inconsistency is minimized by $\llbracket m \rrbracket^*(Y | X)$. Since the degree of inconsistency of m^{+p} is smooth and strongly convex as a function of p , we can compute its optimum values by standard gradient methods. This approach is inefficient as written (since it involves computing the full joint distribution $\llbracket m^{+p} \rrbracket^*$), but we believe that standard approximation techniques will allow us to draw inferences efficiently.

To take another example, conditioning can be understood in terms of resolving inconsistencies in a PDG. To condition on an observation $Y = y$, given a situation

described by a PDG m , we can add an edge from $\mathbb{1}$ to Y in m , annotated with the cpd that gives probability 1 to y , to get the (possibly inconsistent) PDG $m^{(Y=y)}$. The distribution $\llbracket m^{(Y=y)} \rrbracket^*$ turns out to be the result of conditioning $\llbracket m \rrbracket^*$ on $Y = y$. This account of conditioning generalizes without modification to give Jeffrey’s Rule [49], a more general approach to belief updating.

Issues of updating and inconsistency also arise in variational inference. A variational autoencoder [52], for instance, is essentially three cpds: a prior $p(Z)$, a decoder $p(X|Z)$, and an encoder $q(Z|X)$. Because two cpds target Z (and the cpds are inconsistent until fully trained), this situation can be represented by PDGs but not by other graphical models. We hope to report further on the deep connection between inference, updating, and the resolution of inconsistency in PDGs in future work.

APPENDICES FOR CHAPTER 3

3.A Proofs

For brevity, we use the standard notation and write $\mu(x, y)$ instead of $\mu(X = x, Y = y)$, $\mu(x | y)$ instead of $\mu(X = x | Y = y)$, and so forth.

3.A.1 Properties of Scoring Semantics

In this section, we prove the properties of scoring functions that we mentioned in the main text, Propositions 3.3.1, 3.3.2, and 3.3.4. We repeat the statements for the reader's convenience.

Proposition 3.3.1. $\{\mathcal{m}\} = \{\mu : \llbracket \mathcal{m} \rrbracket_0(\mu) = 0\}$ for all \mathcal{m} .

Proof. By taking $\gamma = 0$, the score is just Inc . By definition, a distribution $\mu \in \{\mathcal{m}\}$ satisfies all the constraints, so $\mu(Y = \cdot | X = x) = \mathbb{P}_a(x)$ for all edges $X \rightarrow Y \in \mathcal{A}^m$ and x with $\mu(X = x) > 0$. By Gibbs inequality [63], $D(\mu(Y|x) \parallel \mathbb{P}_a(x)) = 0$. Since this is true for all edges, we must have $Inc_m(\mu) = 0$. Conversely, if $\mu \notin \{\mathcal{m}\}$, then it fails to marginalize to the cpd \mathbb{P}_a on some edge L , and so again by Gibbs inequality, $D(\mu(Y|x) \parallel \mathbb{P}_a(x)) > 0$. As relative entropy is non-negative, the sum of these terms over all edges must be positive as well, and so $Inc_m(\mu) \neq 0$. \square

Before proving the remaining results, we prove a lemma that will be useful in other contexts as well.

Lemma 3.A.1. $\text{Inc}_m(\mu)$ is a convex function of μ .

Proof. It is well known that D is convex [20, Theorem 2.7.2], in the sense that

$$D(\lambda q_1 + (1 - \lambda)q_2 \| \lambda p_1 + (1 - \lambda)p_2) \leq \lambda D(q_1 \| p_1) + (1 - \lambda)D(q_2 \| p_2).$$

Given an edge $\ell \in \mathcal{A}$ from A to B and $a \in \mathcal{V}(A)$, and setting $q_1 = q_2 = \mathbb{P}_\ell(a)$, we get that

$$D(\mathbb{P}_\ell(a) \| \lambda p_1 + (1 - \lambda)p_2) \leq \lambda D(\mathbb{P}_\ell(a) \| p_1) + (1 - \lambda)D(\mathbb{P}_\ell(a) \| p_2).$$

Since this is true for every a and edge, we can take a weighted sum of these inequalities for each a weighted by $p(A = a)$; thus,

$$\mathbb{E}_{a \sim p_A} D(\mathbb{P}_\ell(a) \| \lambda p_1 + (1 - \lambda)p_2) \leq \mathbb{E}_{a \sim p_A} \lambda D(\mathbb{P}_\ell(a) \| p_1) + (1 - \lambda)D(\mathbb{P}_\ell(a) \| p_2).$$

Taking a sum over all edges, we get that

$$\sum_{(A,B) \in \mathcal{A}} \mathbb{E}_{a \sim p_A} D(\mathbb{P}_\ell(a) \| \lambda p_1 + (1 - \lambda)p_2) \leq \sum_{(A,B) \in \mathcal{A}} \mathbb{E}_{a \sim p_A} \lambda D(\mathbb{P}_\ell(a) \| p_1) + (1 - \lambda)D(\mathbb{P}_\ell(a) \| p_2).$$

It follows that

$$\text{Inc}_m(\lambda p_1 + (1 - \lambda)p_2) \leq \lambda \text{Inc}_m(p_1) + (1 - \lambda) \text{Inc}_m(p_2).$$

Therefore, $\text{Inc}_m(\mu)$ is a convex function of μ . \square

The next proposition gives us a useful representation of $\llbracket M \rrbracket_\gamma$.

Proposition 3.4.6. Letting x^w and y^w denote the values of X and Y , respectively, in $w \in \mathcal{V}(M)$, we have

$$\begin{aligned} \llbracket M \rrbracket(\mu) &= \mathbb{E}_{w \sim \mu} \left\{ \underbrace{\sum_{X \xrightarrow{a} Y} \left[\overbrace{\beta_L \log \frac{1}{\mathbb{P}_a(y^w | x^w)}}^{\text{log likelihood / cross entropy}} + \right.} \\ &\quad \left. \underbrace{(\alpha_L \gamma - \beta_L) \log \frac{1}{\mu(y^w | x^w)}}_{\text{local regularization (if } \beta_L > \alpha_L \gamma \text{)}} \right] - \underbrace{\gamma \log \frac{1}{\mu(w)}}_{\text{global regularization}} \right\}. \end{aligned} \tag{3.6}$$

Proof. We use the more general formulation of $IDef$ given in [Section 3.4.3](#), in which each arc a 's conditional information is weighted by α_a .

$$\begin{aligned}
\llbracket \mathbf{m} \rrbracket_\gamma(\mu) &:= Inc_{\mathbf{m}}(\mu) + \gamma IDef_{\mathbf{m}}(\mu) \\
&= \left[\sum_{X \xrightarrow{a} Y} \beta_a \mathbb{E}_{x \sim \mu_X} D\left(\mu(Y|X=x) \parallel \mathbb{P}_a(x)\right) \right] + \gamma \left[\sum_{X \xrightarrow{a} Y} \alpha_a H_\mu(Y|X) - H(\mu) \right] \\
&= \sum_{X \xrightarrow{a} Y} \mathbb{E}_{x \sim \mu_X} \left[\beta_a D\left(\mu(Y|x) \parallel \mathbb{P}_a(Y|x)\right) + \gamma \alpha_a H(\mu|X=x) \right] - \gamma H(\mu) \\
&= \sum_{X \xrightarrow{a} Y} \mathbb{E}_{x \sim \mu_X} \left[\beta_a \left(\sum_{y \in \mathcal{V}(Y)} \mu(y|x) \log \frac{\mu(y|x)}{\mathbb{P}_a(y|x)} \right) \right. \\
&\quad \left. + \alpha_a \gamma \left(\sum_{y \in \mathcal{V}(Y)} \mu(y|x) \log \frac{1}{\mu(y|x)} \right) \right] - \gamma H(\mu) \\
&= \sum_{X \xrightarrow{a} Y} \mathbb{E}_{x \sim \mu_X} \left[\sum_{y \in \mathcal{V}(Y)} \mu(y|x) \left(\beta_a \log \frac{\mu(y|x)}{\mathbb{P}_a(y|x)} + \alpha_a \gamma \log \frac{1}{\mu(y|x)} \right) \right] - \gamma H(\mu) \\
&= \sum_{X \xrightarrow{a} Y} \mathbb{E}_{x \sim \mu_X} \left[\mathbb{E}_{y \sim \mu(Y|X=x)} \left(\beta_a \log \frac{\mu(y|x)}{\mathbb{P}_a(y|x)} + \alpha_a \gamma \log \frac{1}{\mu(y|x)} \right) \right] - \gamma \sum_{\mathbf{w} \in \mathcal{V}\mathcal{X}} \mu(\mathbf{w}) \log \frac{1}{\mu(\mathbf{w})} \\
&= \sum_{X \xrightarrow{a} Y} \mathbb{E}_{x,y \sim \mu_{XY}} \left[\beta_a \log \frac{\mu(y|x)}{\mathbb{P}_a(y|x)} + \alpha_a \gamma \log \frac{1}{\mu(y|x)} \right] - \gamma \mathbb{E}_{\mathbf{w} \sim \mu} \left[\log \frac{1}{\mu(\mathbf{w})} \right] \\
&= \mathbb{E}_{\mathbf{w} \sim \mu} \left\{ \sum_{X \xrightarrow{a} Y} \left[\beta_a \log \frac{1}{\mathbb{P}_a(y|x)} - \beta_a \log \frac{1}{\mu(y|x)} + \alpha_a \gamma \log \frac{1}{\mu(y|x)} \right] \right\} - \gamma \mathbb{E}_{\mathbf{w} \sim \mu} \left[\log \frac{1}{\mu(\mathbf{w})} \right] \\
&= \mathbb{E}_{\mathbf{w} \sim \mu} \left\{ \sum_{X \xrightarrow{a} Y} \left[\beta_a \log \frac{1}{\mathbb{P}_a(y|x)} + (\alpha_a \gamma - \beta_a) \log \frac{1}{\mu(y|x)} \right] - \gamma \log \frac{1}{\mu(\mathbf{w})} \right\}. \quad \square
\end{aligned}$$

We can now prove [Proposition 3.3.2](#).

Proposition 3.3.2. *If \mathbf{m} is a PDG and $0 < \gamma \leq \min_L \beta_L^m$, then $\llbracket \mathbf{m} \rrbracket_\gamma^*$ is a singleton.*

Proof. It suffices to show that $\llbracket \mathbf{m} \rrbracket_\gamma$ is a strictly convex function of μ , since every

strictly convex function has a unique minimum. Note that

$$\begin{aligned}
\llbracket M \rrbracket_\gamma(\mu) &= \mathbb{E}_{\mathbf{w} \sim \mu} \left\{ \sum_{\substack{L \\ X \xrightarrow{L} Y}} \left[\beta_L \log \frac{1}{\mathbb{P}_a(y \mid x)} + (\alpha_L \gamma - \beta_L) \log \frac{1}{\mu(y \mid x)} \right] - \gamma \log \frac{1}{\mu(\mathbf{w})} \right\} \\
&= \mathbb{E}_{\mathbf{w} \sim \mu} \left\{ \sum_{\substack{L \\ X \xrightarrow{L} Y}} \left[\gamma \alpha_L \log \frac{1}{\mathbb{P}_a(y \mid x)} + (\beta_L - \alpha_L \gamma) \log \frac{1}{\mathbb{P}_a(y \mid x)} - (\beta_L - \alpha_L \gamma) \log \frac{1}{\mu(y \mid x)} \right] - \right. \\
&\quad \left. = \mathbb{E}_{\mathbf{w} \sim \mu} \left\{ \sum_{\substack{L \\ X \xrightarrow{L} Y}} \left[\gamma \alpha_L \log \frac{1}{\mathbb{P}_a(y \mid x)} + (\beta_L - \alpha_L \gamma) \log \frac{\mu(y \mid x)}{\mathbb{P}_a(y \mid x)} \right] - \gamma \log \frac{1}{\mu(\mathbf{w})} \right\} \right. \\
&\quad \left. = \sum_{\substack{L \\ X \xrightarrow{L} Y}} \left[\gamma \alpha_L \mathbb{E}_{x,y \sim \mu_{XY}} \left[\log \frac{1}{\mathbb{P}_a(y \mid x)} \right] + (\beta_L - \alpha_L \gamma) \mathbb{E}_{x \sim \mu_X} D(\mu(Y \mid x) \parallel \mathbb{P}_a(x)) \right] - \gamma H(\mu).
\right.
\end{aligned}$$

The first term, $\mathbb{E}_{x,y \sim \mu_{XY}} [-\log \mathbb{P}_a(y \mid x)]$ is linear in μ , as $\mathbb{P}_a(y \mid x)$ does not depend on μ . As for the second term, it is well-known that KL divergence is convex, in the sense that

$$D(\lambda q_1 + (1-\lambda)q_2 \parallel \lambda p_1 + (1-\lambda)p_2) \leq \lambda D(q_1 \parallel p_1) + (1-\lambda)D(q_2 \parallel p_2).$$

Therefore, for a distribution on Y , setting $p_1 = p_2 = \mathbb{P}_a(x)$, for all conditional marginals $\mu_1(Y \mid X = x)$ and $\mu_2(Y \mid X = x)$,

$$D(\lambda \mu_1(Y \mid x) + (1-\lambda) \mu_2(Y \mid x) \parallel \mathbb{P}_a(x)) \leq \lambda D(\mu_1(Y \mid x) \parallel \mathbb{P}_a(x)) + (1-\lambda) D(\mu_2(Y \mid x) \parallel \mathbb{P}_a(x)).$$

So $D(\mu(Y \mid x) \parallel \mathbb{P}_a(Y \mid x))$ is convex. As convex combinations of convex functions are convex, the second term, $\mathbb{E}_{x \sim \mu_X} D(\mu(Y \mid x) \parallel \mathbb{P}_a(x))$, is convex. Finally, negative entropy is well known to be strictly convex.

Any non-negative linear combinations of the three terms is convex, and if this combination applies a positive coefficient to the (strictly convex) negative entropy, it must be strictly convex. Therefore, as long as $\beta_L \geq \gamma$ for all edges $L \in \mathcal{A}^m$, $\llbracket M \rrbracket_\gamma$ is strictly convex. The result follows. \square

We next prove Proposition 3.3.3. The first step is provided by the following lemma.

Lemma 3.A.2. $\lim_{\gamma \rightarrow 0} \llbracket \mathbf{m} \rrbracket_{\gamma}^* \subseteq \llbracket \mathbf{m} \rrbracket_0^*$.

Proof. Since $IDef_{\mathbf{m}}$ is a finite weighted sum of entropies and conditional entropies over the variables $\mathcal{N}^{\mathbf{m}}$, which have finite support, it is bounded. Thus, there exist bounds k and K depending only on $\mathcal{N}^{\mathbf{m}}$ and $\mathcal{V}^{\mathbf{m}}$, such that $k \leq IDef_{\mathbf{m}}(\mu) \leq K$ for all μ . Since $\llbracket \mathbf{m} \rrbracket_{\gamma} = Inc_{\mathbf{m}} + \gamma IDef_{\mathbf{m}}$, it follows that, for all $\mu \in \mathcal{V}(\mathbf{m})$, we have

$$Inc_{\mathbf{m}}(\mu) + \gamma k \leq \llbracket \mathbf{m} \rrbracket_{\gamma}(\mu) \leq Inc_{\mathbf{m}}(\mu) + \gamma K.$$

For a fixed γ , since this inequality holds for all μ , and both Inc and $IDef$ are bounded below, it must be the case that

$$\min_{\mu \in \Delta \mathcal{V}(\mathbf{m})} [Inc_{\mathbf{m}}(\mu) + \gamma k] \leq \min_{\mu \in \Delta \mathcal{V}(\mathbf{m})} \llbracket \mathbf{m} \rrbracket_{\gamma}(\mu) \leq \min_{\mu \in \Delta \mathcal{V}(\mathbf{m})} [Inc_{\mathbf{m}}(\mu) + \gamma K],$$

even though the distributions that minimize each expression will in general be different. Let $Inc(\mathbf{m}) = \min_{\mu} Inc_{\mathbf{m}}(\mu)$. Since $\Delta \mathcal{V}(\mathbf{m})$ is compact, the minimum of the middle term is achieved. Therefore, for $\mu_{\gamma} \in \llbracket \mathbf{m} \rrbracket_{\gamma}^*(\mu)$ that minimizes it, we have

$$Inc(\mathbf{m}) + \gamma k \leq \llbracket \mathbf{m} \rrbracket_{\gamma}(\mu_{\gamma}) \leq Inc(\mathbf{m}) + \gamma K$$

for all $\gamma \geq 0$. Now taking the limit as $\gamma \rightarrow 0$ from above, we get that $Inc(\mathbf{m}) = \llbracket \mathbf{m} \rrbracket_0(\mu^*)$. Thus, $\mu^* \in \llbracket \mathbf{m} \rrbracket_0^*$, as desired. \square

We now apply Lemma 3.A.2 to show that the limit as $\gamma \rightarrow 0$ is unique, as stated in Proposition 3.3.3.

Proposition 3.3.3. *For all \mathbf{m} , $\lim_{\gamma \rightarrow 0} \llbracket \mathbf{m} \rrbracket_{\gamma}^*$ is a singleton.*

Proof. First we show that $\lim_{\gamma \rightarrow 0} \llbracket \mathbf{m} \rrbracket_{\gamma}^*$ cannot be empty. Let $(\gamma_n) = \gamma_1, \gamma_2, \dots$ be a sequence of positive reals converging to zero. For all n , choose some

$\mu_n \in [\![\mathcal{M}]\!]_{\gamma_n}^*$. Because $\Delta\mathcal{V}(\mathcal{M})$ is a compact metric space, it is sequentially compact, and so, by the Bolzano–Weierstrass Theorem, the sequence (μ_n) has at least one accumulation point, say ν . By our definition of the limit, $\nu \in \lim_{\gamma \rightarrow 0} [\![\mathcal{M}]\!]_\gamma^*$, as witnessed by the sequence $(\gamma_n, \mu_n)_n$. It follows that $\lim_{\gamma \rightarrow 0} [\![\mathcal{M}]\!]_\gamma^* \neq \emptyset$.

Now, choose $\nu_1, \nu_2 \in \lim_{\gamma \rightarrow 0} [\![\mathcal{M}]\!]_\gamma^*$. Thus, there are subsequences (μ_i) and (μ_j) of (μ_n) converging to ν_1 and ν_2 , respectively. By Lemma 3.A.2, $\nu_1, \nu_2 \in [\![\mathcal{M}]\!]_0^*$, so $Inc_m(\nu_1) = Inc_m(\nu_2)$. Because $(\mu_{j_n}) \rightarrow \nu_1$, $(\mu_{k_n}) \rightarrow \nu_2$, and $IDef_m$ is continuous on $\Delta\mathcal{V}(\mathcal{M})$, we conclude that $(IDef_m(\mu_i)) \rightarrow IDef_m(\nu_1)$ and $(IDef_m(\mu_j)) \rightarrow IDef_m(\nu_2)$.

Suppose that $IDef_m(\nu_1) \neq IDef_m(\nu_2)$. Without loss of generality, suppose that $IDef_m(\nu_1) > IDef_m(\nu_2)$. Since $(IDef_m(\mu_i)) \rightarrow IDef_m(\nu_1)$, there exists some $i^* \in \mathbb{N}$ such that for all $i > i^*$, $IDef_m(\mu_i) > IDef_m(\nu_2)$. But then for all γ and $i > i^*$, we have

$$[\![\mathcal{M}]\!]_\gamma(\mu_i) = Inc(\mu_i) + \gamma IDef_m(\mu_i) > Inc(\nu_2) + \gamma IDef_m(\nu_2) = [\![\mathcal{M}]\!]_\gamma(\nu_2),$$

contradicting the assumption that μ_i minimizes $[\![\mathcal{M}]\!]_{\gamma_i}$. We thus conclude that we cannot have $IDef_m(\nu_1) > IDef_m(\nu_2)$. By the same argument, we also cannot have $IDef_m(\nu_1) < IDef_m(\nu_2)$, so $IDef_m(\nu_1) = IDef_m(\nu_2)$.

Now, suppose that ν_1 and ν_2 distinct. Since $[\![\mathcal{M}]\!]_\gamma$ is strictly convex for $\gamma > 0$, among the possible convex combinations of ν_1 and ν_2 , the distribution $\nu_3 = \lambda\nu_1 + (1 - \lambda)\nu_2$ that minimizes $[\![\mathcal{M}]\!]_\gamma$ must lie strictly between ν_1 and ν_2 . Because Inc itself is convex and $Inc_m(\nu_1) = Inc_m(\nu_2) =: v$, we must have $Inc_m(\nu_3) \leq v$. But since $\nu_1, \nu_2 \in [\![\mathcal{M}]\!]_0^*$ minimize Inc , we must have $Inc_m(\nu_3) \geq v$. Thus, $Inc_m(\nu_3) = v$. Now, because, for all $\gamma > 0$,

$$[\![\mathcal{M}]\!]_\gamma(\nu_3) = v + \gamma IDef_m(\nu_3) < v + \gamma IDef_m(\nu_1) = [\![\mathcal{M}]\!]_\gamma(\nu_1),$$

it must be the case that $IDef_m(\nu_3) < IDef_m(\nu_1)$.

We can now get a contradiction by applying the same argument as that used to show that $IDef_m(\nu_1) = IDef_m(\nu_2)$. Because $(\mu_i) \rightarrow \nu_1$, there exists some i^* such that for all $i > i^*$, we have $IDef_m(\mu_i) > IDef_m(\nu_3)$. Thus, for all $i > i^*$ and all $\gamma > 0$,

$$[\![m]\!]_\gamma(\mu_i) = Inc(\mu_i) + \gamma IDef_m(\mu_i) > Inc(\nu_3) + \gamma IDef_m(\nu_3) = [\![m]\!]_\gamma(\nu_3),$$

again contradicting the assumption that μ_i minimizes $[\![m]\!]_{\gamma_i}$. Thus, our supposition that ν_1 was distinct from ν_2 cannot hold, and so $\lim_{\gamma \rightarrow 0} [\![m]\!]_\gamma^*$ must be a singleton, as desired. \square

Finally, Proposition 3.3.4 is a simple corollary of Lemma 3.A.2 and Proposition 3.3.3, as we now show.

Proposition 3.3.4. $[\![m]\!]^* \in [\![m]\!]_0^*$, so if m is consistent, then $[\![m]\!]^* \in \{m\}$.

Proof. By Proposition 3.3.3, $\lim_{\gamma \rightarrow 0} [\![m]\!]_\gamma^*$ is a singleton. As in the body of the paper, we refer to its unique element by $[\![m]\!]^*$. Lemma 3.A.2 therefore immediately gives us $[\![m]\!]^* \in [\![m]\!]_0^*$. If m is consistent, then by Proposition 3.3.1, $Inc(m) = 0$, so $[\![m]\!]_0([\![m]\!]^*) = 0$, and thus $[\![m]\!]^* \in \{m\}$. \square

3.A.2 PDGs as Bayesian Networks

In this section, we prove Theorem 3.4.1. We start by recounting some standard results and notation, all of which can be found in a standard introduction to information theory (e.g., [63, Chapter 1]).

First, note that just as we introduced new variables to model joint dependence in PDGs, we can view a finite collection $\mathcal{X} = X_1, \dots, X_n$ of random variables, where each X_i has the same sample space, as itself a random variable, taking the value (x_1, \dots, x_n) iff each X_i takes the value x_i . Doing so allows us to avoid cumbersome and ultimately irrelevant notation which treats sets of random variables differently, and requires lots of unnecessary braces, bold face, and uniqueness issues. Note the notational convention that the joint variable X, Y may be indicated by a comma.

Definition 3.A.1 (Conditional Independence). If X, Y , and Z are random variables, and μ is a distribution over them, then X is *conditionally independent of Z given Y* , (according to μ), denoted ' $X \perp\!\!\!\perp Z | Y$ ', iff for all $x, y, z \in \mathcal{V}(X, Y, Z)$, we have $\mu(x | y)\mu(z | y) = \mu(x, z | y)$. □

Fact 3.A.3 (Entropy Chain Rule). *If X and Y are random variables, then the entropy of the joint variable (X, Y) can be written as $H_\mu(X, Y) = H_\mu(Y | X) + H_\mu(X)$. It follows that if μ is a distribution over the n variables X_1, \dots, X_n , then*

$$H(\mu) = \sum_{i=1}^n H_\mu(X_i | X_1, \dots, X_{i-1}).$$

Definition 3.A.2 (Conditional Mutual Information). The *conditional mutual information* between two (sets of) random variables is defined as

$$I_\mu(X; Y | Z) := \sum_{x,y,z \in \mathcal{V}(X,Y,Z)} \mu(x, y, z) \log \frac{\mu(z)\mu(x, y, z)}{\mu(x, z)\mu(y, z)}.$$

□

Fact 3.A.4 (Properties of Conditional Mutual Information). *For random variables X, Y , and Z over a common set of outcomes, distributed according to a distribution μ , the following properties hold:*

1. (*difference identity*) $I_\mu(X; Y | Z) = H_\mu(X | Y) - H_\mu(X | Y, Z);$
2. (*non-negativity*) $I_\mu(X; Y | Z) \geq 0;$
3. (*relation to independence*) $I_\mu(X; Y | Z) = 0 \text{ iff } X \perp\!\!\!\perp Z | Y.$

We now provide the formal details of the transformation of a BN into a PDG.

Definition 3.A.3 (Transformation of a BN to a PDG). Recall that a (quantitative) Bayesian Network (G, f) consists of two parts: its qualitative graphical structure G , described by a dag, and its quantitative data f , an assignment of a cpd $p_i(X_i | \text{Pa}(X_i))$ to each variable X_i . If \mathcal{B} is a Bayesian network on random variables X_1, \dots, X_n , we construct the corresponding PDG $p\&g(\mathcal{B})$ as follows: we take $\mathcal{N} := \{X_1, \dots, X_n\} \cup \{\text{Pa}(X_1), \dots, \text{Pa}(X_n)\}$. That is, the variables of $p\&g(\mathcal{B})$ consist of all the variables in \mathcal{B} together with a variable corresponding to the parents of X_i . (This will be used to deal with the hyperedges.) The values $\mathcal{V}(X_i)$ for a random variable X_i are unchanged, (i.e., $\mathcal{V}^{p\&g(\mathcal{B})}(\{X_i\}) := \mathcal{V}(X_i)$) and $\mathcal{V}^{p\&g(\mathcal{B})}(\text{Pa}(X_i)) := \prod_{Y \in \text{Pa}(X_i)} \mathcal{V}(Y)$ (if $\text{Pa}(X_i) = \emptyset$, so that X_i has no parents, then we then we identify $\text{Pa}(X_i)$ with $\mathbb{1}$ and take $\mathcal{V}(\text{Pa}(X_i)) = \{\star\}$). We take the set of edges $\mathcal{A}^{p\&g(\mathcal{B})} := \{(\text{Pa}(X_i), X_i) : i = 1, \dots, n\} \cup \{(\text{Pa}_i, Y) : Y \in \text{Pa}(X_i)\}$ to be the set of edges to a variable X_i from its parents, together with an edge from $\text{Pa}(X_i)$ to each of the elements of $\text{Pa}(X_i)$, for $i = 1, \dots, n$. Finally, we set $\mathbb{P}_{(\text{Pa}(X_i), X_i)}^{p\&g(\mathcal{B})}$ to be the cpd associated with X_i in \mathcal{B} , and for each node $X_j \in \text{Pa}(X_i)$, we define

$$\mathbb{P}_{(\text{Pa}(X_i), X_j)}^{p\&g(\mathcal{B})}(\dots, x_j, \dots) = \delta_{x_j};$$

that is, $\mathbb{P}_{(\text{Pa}(X_i), X_j)}^{p\&g(\mathcal{B}, \beta)}$ is the the cpd on X_j that, given a setting (\dots, x_j, \dots) of $\text{Pa}(X_i)$, yields the distribution that puts all mass on x_j . \square

Let \mathcal{X} be the variables of some BN \mathcal{B} , and $\mathcal{M} = (\mathcal{N}, \mathcal{A}, \mathbb{P}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ be the PDG

$\text{pdg}(\mathcal{B})$. Because the set \mathcal{N} of variables in $\text{pdg}(\mathcal{B}, \beta)$ includes variables of the form $\text{Pa}(X_i)$, it is a strict superset of $\mathcal{X} = \{X_1, \dots, X_n\}$, the set of variables of \mathcal{B} . For the purposes of this theorem, we identify a distribution $\mu_{\mathcal{X}}$ over \mathcal{X} with the unique distribution $\text{Pr}_{\mathcal{B}}$ whose marginal on the variables in \mathcal{X} is $\mu_{\mathcal{X}}$ such that if $X_j \in \text{Pa}(X_i)$, then $\mu_{\mathcal{N}}(X_j = x'_j \mid \text{Pa}(X_i) = (\dots, x_j, \dots)) = 1$ iff $x_j = x'_j$. In the argument below, we abuse notation, dropping the the subscripts \mathcal{X} and \mathcal{N} on a distribution μ .

Theorem 3.4.1. *If \mathcal{B} is a Bayesian network and $\text{Pr}_{\mathcal{B}}$ is the distribution it specifies, then for all $\gamma > 0$ and all vectors β such that $\beta_L > 0$ for all edges L , $[\![\text{pdg}(\mathcal{B}, \beta)]\!]^*_{\gamma} = \{\text{Pr}_{\mathcal{B}}\}$, and thus $[\![\text{pdg}(\mathcal{B}, \beta)]\!]^* = \text{Pr}_{\mathcal{B}}$.*

Proof. For the cpd $p(X_i \mid \text{Pa}(X_i))$ associated to a node X_i in \mathcal{B} , we have that $\text{Pr}_{\mathcal{B}}(X_i \mid \text{Pa}(X_i)) = p(X_i \mid \text{Pa}(X_i))$. For all nodes X_i in \mathcal{B} and $X_j \in \text{Pa}(X_i)$, by construction, $\text{Pr}_{\mathcal{B}}$, when viewed as a distribution on \mathcal{N} , is also with the cpd on the edge from $\text{Pa}(X_i)$ to X_j . Thus, $\text{Pr}_{\mathcal{B}}$ is consistent with all the cpds in $\text{pdg}(\mathcal{B}, \beta)$; so $\text{Inc}_{\text{pdg}(\mathcal{B}, \beta)}(\text{Pr}_{\mathcal{B}}) = 0$.

We next want to show that $\text{IDef}_{\text{pdg}(\mathcal{B}, \beta)}(\mu) \geq 0$ for all distributions μ . To do this, we first need some definitions. Let ρ be a permutation of $1, \dots, n$. Define an order \prec_{ρ} by taking $j \prec_{\rho} i$ if j precedes i in the permutation; that is, if $\rho^{-1}(j) < \rho^{-1}(i)$. Say that a permutation is *compatible with \mathcal{B}* if $X_j \in \text{Pa}(X_i)$ implies $j \prec_{\rho} i$. There is at least one permutation compatible with \mathcal{B} , since the graph underlying \mathcal{B} is acyclic.

Consider an arbitrary distribution μ over the variables in \mathcal{X} (which we also view as a distribution over the variables in \mathcal{N} , as discussed above). Recall from Definition 3.A.3 that the cpd on the edge in $\text{pdg}(\mathcal{B}, \beta)$ from $\text{Pa}(X_i)$ to X_i is just

the cpd associated with X_i in \mathcal{B} , while the cpd on the edge in $p\mathbf{dg}(\mathcal{B}, \beta)$ from $\mathbf{Pa}(X_i)$ to $X_j \in \mathbf{Pa}(X_i)$ consists only of deterministic distributions (i.e., ones that put probability 1 on one element), which all have entropy 0. Thus,

$$\sum_{X \xrightarrow{a} Y \in \mathbf{A}^{\mathbf{dg}}(\mathcal{B})} H_\mu(Y | X) = \sum_{i=1}^n H_\mu(X_i | \mathbf{Pa}(X_i)). \quad (3.7)$$

Given a permutation ρ , let $\mathbf{X}_{\prec_{\rho} i} = \{X_j : j \prec_{\rho} i\}$. Observe that

$$\begin{aligned} IDef_{p\mathbf{dg}(\mathcal{B}, \beta)}(\mu) &= \left[\sum_{X \xrightarrow{a} Y \in \mathbf{A}^{\mathbf{dg}}(\mathcal{B})} H_\mu(Y | X) \right] - H(\mu) \\ &= \sum_{i=1}^n H_\mu(X_i | \mathbf{Pa}(X_i)) - \sum_{i=1}^n H_\mu(X_i | \mathbf{X}_{\prec_{\rho} i}) \quad [\text{by Fact 3.A.3 and (3.7)}] \\ &= \sum_{i=1}^n \left[H_\mu(X_i | \mathbf{Pa}(X_i)) - H_\mu(X_i | \mathbf{X}_{\prec_{\rho} i}) \right] \\ &= \sum_{i=1}^n I_\mu \left(X_i ; \mathbf{X}_{\prec_{\rho} i} \setminus \mathbf{Pa}(X_i) \mid \mathbf{Pa}(X_i) \right). \quad [\text{by Fact 3.A.4}] \end{aligned}$$

Using Fact 3.A.4, it now follows that, for all distributions μ , $IDef_{p\mathbf{dg}(\mathcal{B})}(\mu) \geq 0$.

Furthermore, for all μ and permutations ρ ,

$$IDef_{p\mathbf{dg}(\mathcal{B})}(\mu) = 0 \quad \text{iff} \quad \forall i. X_i \perp\!\!\!\perp_{\mu} \mathbf{X}_{\prec_{\rho} i}. \quad (3.8)$$

Since the left-hand side of (3.8) is independent of ρ , it follows that X_i is independent of $\mathbf{X}_{\prec_{\rho} i}$ for some permutation ρ iff X_i is independent of $\mathbf{X}_{\prec_{\rho} i}$ for every permutation ρ . Since there is a permutation compatible with \mathcal{B} , we get that $IDef_{p\mathbf{dg}(\mathcal{B}, \beta)}(\Pr_{\mathcal{B}}) = 0$. We have now shown that that $IDef_{p\mathbf{dg}(\mathcal{B}, \beta)}$ and Inc are non-negative functions of μ , and both are zero at $\Pr_{0\mathcal{B}}$. Thus, for all $\gamma \geq 0$ and all vectors β , we have that $\|\mathbf{pdg}(\mathcal{B}, \beta)\|_{\gamma}(\Pr_{\mathcal{B}}) \leq \|\mathbf{pdg}(\mathcal{B}, \beta)\|_{\gamma}(\mu)$ for all distributions μ . We complete the proof by showing that if $\mu \neq \Pr_{\mathcal{B}}$, then $\|\mathbf{pdg}(\mathcal{B}, \beta)\|_{\gamma}(\mu) > 0$ for $\gamma > 0$.

So suppose that $\mu \neq \text{Pr}_{\mathcal{B}}$. Then μ must also match each cpd of \mathcal{B} , for otherwise $\text{Inc}_{\text{pdg}(\mathcal{B}, \beta)}(\mu) > 0$, and we are done. Because $\text{Pr}_{\mathcal{B}}$ is the *unique* distribution that matches the both the cpds and independencies of \mathcal{B} , μ must not have all of the independencies of \mathcal{B} . Thus, some variable X_i , X_i is not independent of some nondescendant X_j in \mathcal{B} with respect to μ . There must be some permutation ρ of the variables in \mathcal{X} compatible with \mathcal{B} such that $X_j \prec_{\rho} X_i$ (e.g., we can start with X_j and its ancestors, and then add the remaining variables appropriately). Thus, it is not the case that X_i is independent of $X_{\prec_{\rho} i}$, so by (3.8), $\text{IDef}_{\text{pdg}(\mathcal{B})}(\mu) > 0$. This completes the proof. \square

3.A.3 Factor Graph Proofs

Theorems 3.4.2 and 3.4.3 are immediate corollaries of their more general counterparts, Theorems 3.4.4 and 3.4.5, which we now prove.

Theorem 3.4.5. *For all unweighted PDGs \mathcal{N} and non-negative vectors \mathbf{v} over \mathcal{A}^n , and all $\gamma > 0$, we have that $\llbracket (\mathcal{N}, \mathbf{v}, \gamma \mathbf{v}) \rrbracket_{\gamma} = \gamma \text{GFE}_{(\Phi_{\mathcal{N}}, \mathbf{v})}$; consequently, $\llbracket (\mathcal{N}, \mathbf{v}, \gamma \mathbf{v}) \rrbracket_{\gamma}^* = \{\text{Pr}_{(\Phi_{\mathcal{N}}, \mathbf{v})}\}$.*

Proof. Let $\mathcal{M} := (\mathcal{N}, \mathbf{v}, \gamma \mathbf{v})$ be the PDG in question. Explicitly, $\alpha_L^m = v_L$ and $\beta_L^m = \gamma v_L$. By Proposition 3.4.6,

$$\llbracket \mathcal{M} \rrbracket_{\gamma}(\mu) = \mathbb{E}_{\mathbf{w} \sim \mu} \left\{ \sum_{X \xrightarrow{L} Y} \left[\beta_L \log \frac{1}{\mathbb{P}_a(y \mid x)} + (\alpha_L \gamma - \beta_L) \log \frac{1}{\mu(y \mid x)} \right] - \gamma \log \frac{1}{\mu(\mathbf{w})} \right\}.$$

Let $\{\phi_L\}_{L \in \mathcal{A}} := \Phi_{\mathcal{N}}$ denote the factors of the factor graph associated with \mathcal{M} .

Because we have $\alpha_L \gamma = \beta_L$, the middle term cancels, leaving us with

$$\begin{aligned}
\llbracket \mathbf{m} \rrbracket_\gamma(\mu) &= \mathbb{E}_{\mathbf{w} \sim \mu} \left\{ \sum_{\substack{X \xrightarrow{L} Y}} \left[\beta_L \log \frac{1}{\mathbb{P}_a(y \mid x)} \right] - \gamma \log \frac{1}{\mu(\mathbf{w})} \right\} \\
&= \mathbb{E}_{\mathbf{w} \sim \mu} \left\{ \sum_{\substack{X \xrightarrow{L} Y}} \left[\gamma v_L \log \frac{1}{\phi(x, y)} \right] - \gamma \log \frac{1}{\mu(\mathbf{w})} \right\} \quad [\text{as } \beta_L = v_L \gamma] \\
&= \gamma \mathbb{E}_{\mathbf{w} \sim \mu} \left\{ \sum_{\substack{X \xrightarrow{L} Y}} \left[v_L \log \frac{1}{\phi(x, y)} \right] - \log \frac{1}{\mu(\mathbf{w})} \right\} \\
&= \gamma GFE_{(\Phi_n, \mathbf{v})}.
\end{aligned}$$

It immediately follows that the associated factor graph has $\llbracket \mathbf{m} \rrbracket_\gamma^* = \{\Pr_{\Phi(m)}\}$, because the free energy is clearly a constant plus the KL divergence from its associated probability distribution. \square

Theorem 3.4.4. *For all WFGs $\Psi = (\Phi, \theta)$ and all $\gamma > 0$, we have that $GFE_\Psi = 1/\gamma \llbracket \mathbf{m}_{\Psi, \gamma} \rrbracket_\gamma + C$ for some constant C , so \Pr_Ψ is the unique element of $\llbracket \mathbf{m}_{\Psi, \gamma} \rrbracket_\gamma^*$.*

Proof. In $p\mathbf{dg}(\Psi, \gamma)$, there is an edge $1 \rightarrow X_J$ for every $J \in \mathcal{J}$, and also edges $X_J \twoheadrightarrow X_j$ for each $X_j \in X_J$. Because the latter edges are deterministic, a distribution μ that is not consistent with one of the edges, say $X_J \twoheadrightarrow X_j$, has $Inc_m(\mu) = \infty$. This is a property of relative entropy: if there exist $j^* \in \mathcal{V}(X_j)$ and $\mathbf{z}^* \in \mathcal{V}(J)$ such that $\mathbf{z}_j^* \neq j^*$ and μ places positive probability on their co-occurrence (i.e., $\mu(j^*, \mathbf{z}^*) > 0$), then we would have

$$\begin{aligned}
\mathbb{E}_{\mathbf{z} \sim \mu_J} D\left(\mu(X_j \mid X_J = \mathbf{z}) \parallel \mathbb{1}[X_j = \mathbf{z}_j]\right) &= \sum_{\substack{\mathbf{z} \in \mathcal{V}(X_J), \\ \iota \in \mathcal{V}(X_j)}} \mu(\mathbf{z}, \iota) \log \frac{\mu(\iota \mid \mathbf{z})}{\mathbb{1}[\mathbf{z}_j = \iota]} \\
&\geq \mu(\mathbf{z}^*, j^*) \log \frac{\mu(j^* \mid \mathbf{z})}{\mathbb{1}[\mathbf{z}_j^* = j_*]} = \infty.
\end{aligned}$$

Consequently, a distribution μ that does not satisfy the the projections has $\llbracket \mathbf{m}_{\Psi, \gamma} \rrbracket_\gamma(\mu) = \infty$ for every γ . Thus, a distribution that has a finite score must

match the constraints, so we can identify such a distribution with its restriction to the original variables of Φ . Moreover, for all distributions μ with finite score and projections $X_J \rightarrow X_j$, the conditional entropy $H(X_j | X_J) = -\mathbb{E}_\mu \log(\mu(x_j | x_J))$ and divergence from the constraints are both zero. Therefore the per-edge terms for both $IDef_m$ and Inc_m can be safely ignored for the projections. Let \mathbb{P}_J be the normalized distribution $\frac{1}{Z_J} \phi_J$ over X_J , where $Z_J = \sum_{x_J} \phi_J(x_J)$ is the appropriate normalization constant. By Definition 3.4.4, we have $p\mathcal{dg}(\Psi, \gamma) = (up\mathcal{dg}(\Phi), \theta, \gamma\theta)$, so by Proposition 3.4.6,

$$\begin{aligned}
\llbracket p\mathcal{dg}(\Psi, \gamma) \rrbracket_\gamma(\mu) &= \mathbb{E}_{\mathbf{x} \sim \mu} \left\{ \sum_{J \in \mathcal{J}} \left[\beta_J \log \frac{1}{\mathbb{P}_J(x_J)} + (\alpha_J \gamma - \beta_J) \log \frac{1}{\mu(x_J)} \right] - \gamma \log \frac{1}{\mu(\mathbf{x})} \right\} \\
&= \mathbb{E}_{\mathbf{x} \sim \mu} \left\{ \sum_{J \in \mathcal{J}} \left[(\gamma \theta_J) \log \frac{1}{\mathbb{P}_J(x_J)} + (\theta_J \gamma - \gamma \theta_J) \log \frac{1}{\mu(x_J)} \right] - \gamma \log \frac{1}{\mu(\mathbf{x})} \right\} \\
&= \mathbb{E}_{\mathbf{x} \sim \mu} \left\{ \sum_{J \in \mathcal{J}} \left[\gamma \theta_J \log \frac{1}{\mathbb{P}_J(x_J)} \right] - \gamma \log \frac{1}{\mu(\mathbf{x})} \right\} \\
&= \gamma \cdot \mathbb{E}_{\mathbf{x} \sim \mu} \left\{ \sum_{J \in \mathcal{J}} \theta_J \log \frac{Z_J}{\phi_J(x_J)} - \log \frac{1}{\mu(\mathbf{x})} \right\} \\
&= \gamma \cdot \mathbb{E}_{\mathbf{x} \sim \mu} \left\{ \sum_{J \in \mathcal{J}} \theta_J \left[\log \frac{1}{\phi_J(x_J)} + \log Z_J \right] - \log \frac{1}{\mu(\mathbf{x})} \right\} \\
&= \gamma \cdot \mathbb{E}_{\mathbf{x} \sim \mu} \left\{ \sum_{J \in \mathcal{J}} \theta_J \log \frac{1}{\phi_J(x_J)} - \log \frac{1}{\mu(\mathbf{x})} \right\} + \sum_{J \in \mathcal{J}} \theta_J \log Z_J \\
&= \gamma GFE_\Psi + k \log \prod_J Z_J,
\end{aligned}$$

which differs from GFE_Ψ by the value $\sum_J \theta_J \log Z_J$, which is constant in μ . \square

CHAPTER 4

REPRESENTING THINGS WITH PDGS

PDGs are extremely expressive. We saw in Sections 3.4.1 to 3.4.3 that PDGs can capture graphical models such as Bayesian Networks and Factor graphs—but this is only the beginning. In this chapter, we will see how a wide variety of other mathematical models and fragments of epistemic information can be implicitly viewed as PDGs.

4.1 Probabilities and Random Variables

Probability is the dominant way that computer scientists and microeconomists think about an epistemic state. This is due to standard betting arguments suggesting that any sufficiently rational agent (e.g., one resistant to dutch books) must act as if it had probabilistic beliefs [? ?]. Still, probability has real drawbacks [], and there have been many other representations of uncertainty [39]. Many approaches are built on binary truth and falsehood, and epistemic logics dealing with possibility and necessity are built on top of them. Possibility is one important one; others include belief/plausibility measures, which have been touted as generalizations of probability. Perhaps surprisingly, many of these objects can be represented with PDGs as well.

We start with an obvious construction, that is nonetheless important to keep in mind: a joint distribution can be viewed as a very special case of a PDG. Let \mathcal{X} be a set of variables, and recall that $\mathcal{V}\mathcal{X}$ is the set of all joint settings of the variables \mathcal{X} . A joint distribution $\mu \in \Delta\mathcal{V}\mathcal{X}$ can be implicitly regarded as that has a single hyperarc $\emptyset \rightarrow \mathcal{X}$. We attach μ as the (c)pd, and give it weights $\alpha = 0$

(if, as is usual, the distribution represents purely observational and not causal information) and $\beta = 1$ (default confidence).

To a probability theorist, these joint distributions μ may seem to be of a very special form, because they are over product spaces. In probability theory, the setup is typically instead that one has a (measurable) set Ω of outcomes, and then random variables are in fact (*measurable*) functions $X : \Omega \rightarrow \mathcal{VX}$. Observe that this too is an immediate special case of a PDG:

(In fact, this PDG happens to also be a BN, if one isn't worried about calling Ω , often itself a product of variables, a variable.) It is easy to verify that these PDGs have inconsistency zero, and represent precisely the distribution μ .

The semantics of PDGs make heavy use of the usual definition of a joint distribution $\mu \in \Delta\mathcal{VX}$, and thus would be circular if one were to implicitly convert joint distributions to PDGs before developing the results of the previous chapter. But now that the theory of PDGs is on solid ground, we may freely regard joint distribution as a PDG. This will be useful to keep in mind later, but it is neither surprising, nor is this property unique to PDGs. Indeed, μ can also be viewed as a factor graph with only one factor.

4.2 Widgets

PDGs may be expressive, but they are structured objects with clear and specific specific syntax. In fact, upon careful examination, one might find the syntax

unnecessarily restrictive. In specifying the data for an arc $X \rightarrow Y$, you must specify a probability distribution $p(Y|x)$ over $\mathcal{V}Y$ for *every* value $x \in \mathcal{V}X$.

At least—to a first approximation. It turns out that all of these can be captured by PDGs.

4.2.1 Incomplete CPDs and Individual (Conditional) Probabilities

4.2.2 Relations and Constraints

4.2.3 Couplings

Let $\Pi(p, q)$ be the set of couplings of p and q , i.e.,

$$\Pi(p(X), q(Y)) := \left\{ \mu \in \Delta \mathcal{V}(X, Y) : \quad \mu(X) = p, \mu(Y) = q \right\}.$$

Observe that this is exactly the set of distributions consistent with a PDG containing p and q .

$$\Pi(p, q) = \left\{ \begin{array}{cc} p \downarrow & q \downarrow \\ \boxed{X} & \boxed{Y} \end{array} \right\}.$$

Now, suppose we have a distance metric d on a spce X . For $k \in [1, \infty)$, the k -Wasserstein distance between $p, q \in \Delta X$ is given by

$$W_k(p, q) := \inf_{\mu \in \Pi(p, q)} \mathbb{E}_{\mu} \left[d(X, Y)^k \right]^{\frac{1}{k}}.$$

This definition effectively takes $p(X)$ and $q(X)$ with high confidence, by constraining to $\mu \in \Pi(p, q)$. But, in order to represent this as a PDG, we need to represent the d in probabilistic terms. A distance is not a probability. But we can encode the belief that the values of X and Y are close, according to d .

Let T be a variable that in principle can be either t or f , but happens to always be t . To place more probability in f as $d(X, Y)$ increases.

$$\hat{d}(T = t | X, Y) \propto \exp(-d(X, Y)).$$

We then have:

Proposition 4.2.1.

$$\left\langle \begin{array}{c} p! \downarrow \\ X \\ \hat{d} \\ q! \downarrow \\ Y \\ \xrightarrow[t]{} T \end{array} \right\rangle = \inf_{\mu \in \Pi(p,q)} \mathbb{E}_{\mu} [d(X, Y)] = W_1(p, q),$$

We also have:

Proposition 4.2.2.

$$\left\langle \begin{array}{c} p! \downarrow \\ X \\ \xrightarrow[Gd]{} Y \\ q! \downarrow \end{array} \right\rangle = \inf_{\mu \in \Pi(p,q)} \mathbb{E}_{\mu} [d(X, Y) + \log \sum_y \exp(-d(X, y)) - H_{\mu}(Y|X)]$$

Proposition 4.2.3.

$$\lim_{t \rightarrow \infty} \frac{1}{t} \left\langle \begin{array}{c} p! \downarrow \\ X \\ \xrightarrow[(t)]{Gd} Y \\ q! \downarrow \end{array} \right\rangle = \inf_{\mu \in \Pi(p,q)} \mathbb{E}_{\mu} [d(X, Y)]$$

4.3 Other Representations of Knowledge and Uncertainty

4.3.1 Belief and Plausibility Functions

We now move on to another representation of uncertainty, which generalizes the notion of a probability distribution over a (for simplicity, finite) set W , called

a *belief function* [86]. Like a probability measure, a belief function Bel assigns a degree of belief in $[0, 1]$ to subsets $U \subseteq W$. Belief functions must satisfy certain axioms ensuring that $Bel(U) + Bel(\bar{U}) \leq 1$, and thus $Plaus(U) := 1 - Bel(\bar{U}) \geq Bel(U)$. It can be shown that a probability distribution is the special case when these two relationships hold with equality, so that $Bel = Plaus$.

Belief functions admit an alternate representation in terms of a *mass function* $m : 2^W \rightarrow [0, 1]$, which yields a belief function according to $Bel(U) := \sum_{V \subseteq U} m(V)$. In fact, there is a bijection between belief functions Bel and mass functions m . The only requirement on m is that $\sum_{V \subseteq W} m(V) = 1$. So, in other words, m is a probability over subsets V of W . There is also a natural relation between values of V (i.e., subsets of W) and values of W (i.e., elements of W): containment (\ni). Both m and \ni can be modeled with a PDG. What happens if we put them together?

Theorem 4.3.1. (a) If m is the mass function corresponding to $Plaus$, then for all

$$U \subseteq W, \left\langle \xrightarrow{m} V \rightsquigarrow \ni W \xrightarrow{\in U} \right\rangle = -\log Plaus(U).$$

4.3.2 Causal Models

(TODO: Move Causality Intro Here)

4.3.3 Pseudomarginals and Clique Trees

4.3.4 Implicit Neural Representations

CHAPTER 5

QUALITATIVE MECAHNISM INDEPENDENCE

In Section 3.3.1, we defined what it meant for a joint distribution μ to be *quantitatively* compatible with the information in a PDG—it must match all of the cpds. But conspicuously absent is a qualitative analogue. What should it mean for a distribution to be compatible with the qualitative data of the hypergraph?

In this chapter, we define what it means for a joint probability distribution to be compatible with a set of independent causal mechanisms, at a qualitative level—or, more precisely, with a directed hypergraph \mathcal{A} , i.e., the qualitative structure of a PDG. When \mathcal{A} represents a qualitative Bayesian network (BN), this notion of *QIM-compatibility* with \mathcal{A} reduces to satisfying the appropriate conditional independencies. But giving semantics to hypergraphs using QIM-compatibility lets us do much more. For one thing, we can capture functional *dependencies*. For another, we can capture important aspects of causality using compatibility: we can use compatibility to understand cyclic causal graphs, and to demonstrate structural compatibility, we must essentially produce a causal model. Finally, compatibility has deep connections to Shannon information. Applying compatibility to cyclic structures helps to clarify a longstanding conceptual issue in information theory. Compatibility also has a close, but far from obvious, relationship with the original scoring-function semantics for qualitative PDGs, which underlies many of our results.

5.1 Introduction

The structure of a probabilistic graphical model encodes a set of conditional independencies among variables. This is useful because it enables a compact description of probability distributions that have those independencies; it also lets us use graphs as a visual language for describing important qualitative properties of a probabilistic world. Yet these kinds of independencies are not the only important qualitative aspects of a probability measure. In this paper, we study a natural generalization of standard graphical model structures that can describe far more than conditional independence.

For example, another qualitative aspect of a probability distribution is that of functional *dependence*, which is also exploited across computer science to enable compact representations and simplify probabilistic analysis. Acyclic causal models, for instance, specify a distribution via a probability over *contexts* (the values of variables whose causes are viewed as outside the model), and a collection of equations (i.e., functional dependencies) [75]. And in deep learning, a popular class of models called *normalizing flows* [89, 54] specify a distribution by composing a fixed distribution over some latent space, say a standard normal distribution, with a function (i.e., a functional dependence) fit to observational data. **Similarly, complexity theorists often regard a probabilistic Turing machine as a deterministic function that takes as input a uniformly random string [87].** Functional dependence and independence are deeply related and interacting notions. For instance, if B is a function of A (written $A \twoheadrightarrow B$) and A is independent of C (written $A \perp\!\!\!\perp C$), then B and C are also independent ($B \perp\!\!\!\perp C$).¹ Moreover, dependence can be written in terms of independence: Y is a function

¹This well-known fact (Lemma 5.A.1) is formalized and proved in Section 7.B, where all proofs can be found.

of X if and only if Y is conditionally independent of itself given X (i.e., $X \rightarrowtail Y$ iff $Y \perp\!\!\!\perp Y | X$). Traditional graph-based languages such as Bayesian Networks (BNs) and Markov Random Fields (MRFs) cannot capture these relationships. Indeed, the graphoid axioms (which describe BNs and MRFs) [76] and axioms for conditional independence [67], do not even consider statements like $A \perp\!\!\!\perp A$ to be syntactically valid. Yet such statements are perfectly meaningful, and reflect a deep relationship between independence, dependence, and generalizations of both notions (grounded in information theory, a point we will soon revisit).

So the paper describes a simple yet expressive graphical language for describing qualitative structure such as dependence and independence in probability distributions. The idea behind our approach is to specify the inputs and outputs of a set of *independent mechanisms*. In slightly more detail, by “independent mechanism”, we mean a process by which some (set of) the target variables T are determined as a (possibly randomized) function of a (set of) source variables S . So, at a qualitative level, the modeler specifies not a graph, but rather a *directed hypergraph*—which is the structure of another type of probabilistic graphical model: a *probabilistic dependency graph* (PDG) [82, 83, 81].

Although the qualitative aspects of PDGs were characterized by Richardson and Halpern [82] using a scoring function, that scoring function does not seem to get at the qualitative aspects that we are most interested in here. In this work, we develop from first principles an alternate qualitative semantics for directed hypergraphs. More precisely, we define what it means for a distribution to be *QIM-compatible* (qualitatively independent-mechanism compatible, or just *compatible* when unambiguous) with a directed hypergraph \mathcal{A} . This definition allows us to use directed hypergraphs as a language for specifying structure in probability distributions, of which the semantics of qualitative BNs are a special

case ([Theorem 5.2.1](#)).

But QIM-compatibility can do much more than represent conditional independencies in acyclic networks. For one thing, it can encode arbitrary functional dependencies ([Theorem 5.2.2](#)); for another, it gives meaningful semantics to cyclic models. Indeed, compatibility lets us go well beyond capturing dependence and independence. The fact that Pearl [75] also views causal models as representing independent mechanisms suggests that there might be a connection between causality and compatibility. In fact, there is. A *witness* that a distribution μ is compatible with a hypergraph \mathcal{A} is an extended distribution $\bar{\mu}$ that is nearly equivalent to (and guarantees the existence of) a causal model that explains μ with dependency structure \mathcal{A} . As we shall see, thinking in terms of witnesses and compatibility allows us to tie together causality, dependence, and independence.

Perhaps surprisingly, compatibility also has deep connections with information theory ([Section 5.4](#)). The conditional independencies of a BN can be viewed as a very specific kind of information-theoretic constraint. Our notion of compatibility with a hypergraph \mathcal{A} turns out to imply a generalization of this constraint (closely related to the qualitative PDG scoring function) that is meaningful for all hypergraphs. Applied to cyclic models, it yields a causally inspired notion of pairwise interaction that clarifies some important misunderstandings in information theory ([Examples 10 and 11](#)). **It also gracefully handles incomplete fragments of a causal picture, as well as “over-determined” ones.**

Saying that one approach to qualitative graphical modeling has connections to so many different notions is a rather bold claim. We spend the rest of the paper justifying it.

5.2 Qualitative Independent-Mechanism (QIM) Compatibility

In this section, we present the central definition of our paper: a way of making precise Pearl’s notion of “independent mechanisms”, used to motivate Bayesian Networks from a causal perspective. Pearl [77, p.22] states that “*each parent-child relationship in a causal Bayesian network represents a stable and autonomous physical mechanism.*” But, technically speaking, a parent-child relationship only partially describes the mechanism. Instead, the autonomous mechanism that determines the child is really represented by that child’s joint relationship with all its parents. So, the qualitative aspect of a mechanism is best represented as a directed *hyperarc* [30], that can have multiple sources.

Definition 5.2.1. A *directed hypergraph* (or simply a hypergraph, since all our hypergraphs will be directed) consists of a set \mathcal{N} of nodes and a set \mathcal{A} of directed hyperedges, or *hyperarcs*; each hyperarc $a \in \mathcal{A}$ is associated with a set $S_a \subseteq \mathcal{N}$ of source nodes and a set $T_a \subseteq \mathcal{N}$ of target nodes. We write $S \xrightarrow{a} T \in \mathcal{A}$ to specify a hyperarc $a \in \mathcal{A}$ together with its sources $S = S_a$ and targets $T = T_a$. Nodes that are neither a source nor a target of any hyperarc will seldom have any effect on our constructions; the other nodes can be recovered from the hyperarcs (by selecting $\mathcal{N} := \bigcup_{a \in \mathcal{A}} S_a \cup T_a$). Thus, we often leave \mathcal{N} implicit, referring to the hypergraph simply as \mathcal{A} . □

Following the graphical models literature, we are interested in hypergraphs whose nodes represent variables, so that each $X \in \mathcal{N}$ will ultimately be associated with a (for simplicity, finite) set $\mathcal{V}(X)$ of possible values. However, one should not think of \mathcal{V} as part of the information carried by the hypergraph. It makes perfect sense to say that X and Y are independent without specifying

the possible values of X and Y . Of course, when we talk concretely about a distribution μ on a set of variables $\mathcal{X} \cong (\mathcal{N}, \mathcal{V})$, those variables must have possible values—but the *qualitative* properties of μ , such as independence, can be expressed purely in terms of \mathcal{N} , without reference to \mathcal{V} .

Intuitively, we expect a joint distribution $\mu(\mathcal{X})$ to be qualitatively compatible with a set of independent mechanisms (whose structure is given by a hypergraph \mathcal{A}) if there is a mechanistic explanation of how each target arises as a function of the variable(s) on which it depends and independent random noise. This is made precise by the following definition.

Definition 5.2.2 (QIM-compatibility). Let \mathcal{X} and \mathcal{Y} be (possibly identical) sets of variables, and $\mathcal{A} = \{S_a \xrightarrow{a} T_a\}_{a \in \mathcal{A}}$ be a hypergraph with nodes \mathcal{X} . We say a distribution $\mu(\mathcal{Y})$ is *qualitatively independent-mechanism compatible*, or (QIM-)compatible, with \mathcal{A} (symbolically: $\mu \models \Diamond \mathcal{A}$) iff there exists an extended distribution $\bar{\mu}(\mathcal{Y} \cup \mathcal{X} \cup \mathcal{U}_{\mathcal{A}})$ of $\mu(\mathcal{Y})$ to \mathcal{X} and to $\mathcal{U}_{\mathcal{A}} = \{U_a\}_{a \in \mathcal{A}}$, an additional set of “noise” variables (one variable per hyperarc) according to which:

- (a) the variables \mathcal{Y} are distributed according to μ (i.e., $\bar{\mu}(\mathcal{Y}) = \mu(\mathcal{Y})$),
- (b) the variables $\mathcal{U}_{\mathcal{A}}$ are mutually independent (i.e., $\bar{\mu}(\mathcal{U}_{\mathcal{A}}) = \prod_{a \in \mathcal{A}} \bar{\mu}(U_a)$), and
- (c) the target variable(s) T_a of each hyperarc $a \in \mathcal{A}$ are determined by U_a and the source variable(s) S_a (i.e.,
 $\forall a \in \mathcal{A}. \bar{\mu} \models (S_a, U_a) \rightarrow\!\!\! \rightarrow T_a$).

We call such a distribution $\bar{\mu}(\mathcal{X} \cup \mathcal{Y} \cup \mathcal{U}_{\mathcal{A}})$ a *witness* that μ is QIM-compatible with \mathcal{A} . □

While Definition 5.2.2 requires the noise variables $\{U_a\}_{a \in \mathcal{A}}$ to be independent of one another, note that they need not be independent of any variables in \mathcal{X} . In

particular, U_a may not be independent of S_a , and so the situation can diverge from what one would expect from a randomized algorithm, whose randomness U is assumed to be independent of its input S . Furthermore, the variables in \mathcal{U} may not be independent of one another conditional on the value of some $X \in \mathcal{X}$.

Example 6. $\mu(X, Y)$ is compatible with $\mathcal{A} = \{\emptyset \xrightarrow{1} \{X\}, \emptyset \xrightarrow{2} \{Y\}\}$ (depicted in PDG notation as $\rightarrow[X] [Y] \leftarrow$) iff X and Y are independent, i.e., $\mu(X, Y) = \mu(X)\mu(Y)$. For if U_1 and U_2 are independent and respectively determine X and Y , then X and Y must also be independent. \triangle

This is a simple illustration of a more general phenomenon: when \mathcal{A} describes the structure of a Bayesian Network (BN), then QIM-compatibility with \mathcal{A} coincides with satisfying the independencies of that BN (which are given, equivalently, by the *ordered Markov properties* [59], factoring as a product of probability tables, or *d-separation* [31]). To state the general result ([Theorem 5.2.1](#)), we must first clarify how the graphs of standard graphical and causal models give rise to directed hypergraphs.

Suppose that $G = (V, E)$ is a graph, whose edges may be directed or undirected. Given a vertex $u \in V$, write $\text{Pa}_G(u) := \{v : (v, u) \in E\}$ for the set of vertices that can “influence” u . There is a natural way to interpret the graph G as giving rise to a set of mechanisms: one for each variable u , which determines the value of u based the values of the variables on which u can depend. Formally, let $\mathcal{A}_G := \{ \text{Pa}_G(u) \xrightarrow{u} \{u\} \}_{u \in V}$ be the hypergraph *corresponding* to the graph G .

Theorem 5.2.1. *If G is a directed acyclic graph and $\mathcal{I}(G)$ consists of the independencies of its corresponding Bayesian network, then $\mu \models \Diamond \mathcal{A}_G$ if and only if μ satisfies $\mathcal{I}(G)$.*

[link to
proof]

Theorem 5.2.1 shows, for hypergraphs that correspond to directed acyclic graphs (dags), our definition of compatibility reduces exactly to the well-understood independencies of BNs. This means that QIM-compatibility, a notion based on the independence of causal mechanisms, and seemingly unrelated to other notions of independence in BNs, gives us a completely different way of characterizing these independencies—one that can be generalized to much larger classes of graphical models, that includes, for example, cyclic variants [6]. Moreover, QIM-compatibility can capture properties other than independence. As the following example shows, it can capture determinism.

Example 7. If $\mathcal{A} = \{\overset{1}{\rightarrow} X, \overset{2}{\rightarrow} X\}$ consists of just two hyperarcs pointing to a single variable X , then a distribution $\mu(X)$ is QIM-compatible with \mathcal{A} iff μ places all mass on a single value $x \in \mathcal{V}(X)$. \triangle

Intuitively, if two independent coins always give the same answer (the value of X), then neither coin can be random. This simple example shows that we can capture determinism with multiple hyperarcs pointing to the same variable. Such hypergraphs do not correspond to graphs; recall that in a BN, two arrows pointing to X (e.g., $Y \rightarrow X$ and $Z \rightarrow X$) represent a single mechanism by which X is jointly determined (by Y and Z), rather than two distinct mechanisms. A central thrust of Richardson and Halpern's original argument for PDGs over BNs is their ability to describe two different probabilities describing a single variable, such as $\Pr(X|Y)$ and $\Pr(X|Z)$. The qualitative analogue of that expressiveness is precisely what allows us to capture functional dependence.

Given a hypergraph $\mathcal{A} = (\mathcal{N}, \mathcal{A})$, $X, Y \subseteq \mathcal{N}$, and a natural number $n \geq 0$, let $\mathcal{A} \sqcup_{X \rightarrow Y}^{(+n)}$ denote the hypergraph that results from augmenting \mathcal{A} with n additional (distinct) hyperarcs from X to Y .

- Theorem 5.2.2.** (a) $\mu \models X \rightarrow\!\!\!\rightarrow Y \wedge \diamond \mathcal{A}$ if and only if $\forall n \geq 0. \mu \models \diamond \mathcal{A} \sqcup_{X \rightarrow Y}^{(+n)}$.
- (b) if $\mathcal{A} = \mathcal{A}_G$ for a dag G , then $\mu \models X \rightarrow\!\!\!\rightarrow Y \wedge \diamond \mathcal{A}$ if and only if $\mu \models \diamond \mathcal{A} \sqcup_{X \rightarrow Y}^{(+1)}$.
- (c) if $\exists a \in \mathcal{A}$ such that $S_a = \emptyset$ and $X \in T_a$, then $\mu \models X \rightarrow\!\!\!\rightarrow Y \wedge \diamond \mathcal{A}$ iff $\mu \models \diamond \mathcal{A} \sqcup_{X \rightarrow Y}^{(+2)}$.

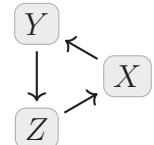
Based on the intuition given after [Example 7](#), it may seem unnecessary to ever add more than two parallel hyperarcs to ensure functional dependence in part (a). However, this intuition implicitly assumes that the randomness U_1 and U_2 of the two mechanisms is independent conditional on X , which may not be the case. See [Section 5.D](#) for counterexamples.

Finally, as alluded to above, QIM-compatibility gives meaning to cyclic structures, a topic that we will revisit often in [Sections 5.3](#) and [5.4](#). We start with some simple examples.

Example 8. Every $\mu(X, Y)$ is compatible with $\boxed{X} \not\rightarrow \boxed{Y}$, because every distribution is compatible with $\rightarrow \boxed{X} \rightarrow \boxed{Y}$, and a mechanism with no inputs is a special case of one that can depend on Y . \triangle

The logic above is an instance of an important reasoning principle, which we develop in [Section 5.B](#). Although the 2-cycle in [Example 8](#) is straightforward, generalizing it even slightly to a 3-cycle raises a not-so-straightforward question, whose answer will turn out to have surprisingly broad implications.

Example 9. What $\mu(X, Y, Z)$ are compatible with the 3-cycle, shown on the right? By the reasoning above, among them must be all distributions consistent with a linear chain $\rightarrow X \rightarrow Y \rightarrow Z$. Thus, any distribution in which two variables are conditionally independent given the third is compatible with the 3-cycle. Are there distributions that



are *not* compatible with this hypergraph? It is not obvious. We return to this in [Section 5.4](#). \triangle

Because QIM-compatibility applies to cyclic structures, one might wonder if it also captures the independencies of undirected models. Our definition of \mathcal{A}_G , as is common, implicitly identifies a undirected edge $A-B$ with the pair $\{A \rightarrow B, B \rightarrow A\}$ of directed edges; in this way, it naturally converts even an *undirected* graph G to a (directed) hypergraph. Compatibility with \mathcal{A}_G , however, does not coincide with any of the standard Markov properties corresponding to G [56]. This may appear to be a flaw in [Definition 5.2.2](#), but it is unavoidable (see [Section 5.B](#)) if we wish to also capture causality, as we do in the next section.

5.3 QIM-Compatibility and Causality

Recall that in the definition of QIM-compatibility, each hyperarc represents an independent mechanism. Equations in a causal model are also viewed as representing independent mechanisms. This suggests a possible connection between the two formalisms, which we now explore. We will show that QIM-compatibility with \mathcal{A} means exactly that a distribution can be generated by a causal model with the corresponding dependency structure ([Section 5.3.1](#)). Moreover, such causal models and QIM-compatibility witnesses are themselves closely related ([Section 5.3.2](#)). In this section, we establish a causal grounding for QIM-compatibility. To do so, we must first review some standard definitions.

Definition 5.3.1 (Pearl [77]). A *structural equations model* (SEM) is a tuple $M = (\mathcal{U}, \mathcal{V}, \mathcal{F})$, where

- \mathcal{U} is a set of exogenous variables;
- \mathcal{V} is a set of endogenous variables (disjoint from \mathcal{U});
- $\mathcal{F} = \{f_Y\}_{Y \in \mathcal{V}}$ associates to each endogenous variable Y an equation $f_Y : \mathcal{V}(\mathcal{U} \cup \mathcal{V} - Y) \rightarrow \mathcal{V}(Y)$ that determines its value as a function of the other variables.

□

In a SEM M , a variable $X \in \mathcal{V}$ does not depend on $Y \in \mathcal{V} \cup \mathcal{U}$ if $f_X(\dots, y, \dots) = f_X(\dots, y', \dots)$ for all $y, y' \in \mathcal{V}(Y)$. Let the parents $\text{Pa}_M(X)$ of X be the set of variables on which X depends. M is acyclic iff $\text{Pa}_M(X) \cap \mathcal{V} = \text{Pa}_G(X)$ for some dag G with vertices \mathcal{V} . In an acyclic SEM, it is easy to see that a setting of the exogenous variables determines the values of the endogenous variables (symbolically: $M \models \mathcal{U} \twoheadrightarrow \mathcal{V}$). A probabilistic SEM (PSEM) $\mathcal{M} = (M, P)$ is a SEM, together with a probability P over the exogenous variables. When $\mathcal{M} \models \mathcal{U} \twoheadrightarrow \mathcal{V}$ (such as when M is acyclic), the distribution $P(\mathcal{U})$ extends uniquely to a distribution over $\mathcal{V}(\mathcal{V} \cup \mathcal{U})$. A cyclic PSEM, however, may induce more than one such distribution, or none at all. In general, a PSEM \mathcal{M} induces a (possibly empty) convex set of distributions over $\mathcal{V}(\mathcal{U} \cup \mathcal{V})$. This set is defined by two (linear) constraints: the equations \mathcal{F} must hold with probability 1, and , in the case of a PSEM, the marginal probability over \mathcal{U} must equal P . Formally, for a PSEM $\mathcal{M} = (M, P)$, define $\{\mathcal{M}\} :=$

$$\left\{ \nu \in \Delta \mathcal{V}(\mathcal{V} \cup \mathcal{U}) \mid \begin{array}{l} \forall Y \in \mathcal{V}. \nu(f_Y(\mathcal{U}, \mathcal{V} - Y) = Y) = 1, \\ \nu(\mathcal{U}) = P(\mathcal{U}) \end{array} \right\}$$

and define $\{M\}$ for an “ordinary” SEM M in the same way, except without the constraint involving P . To unpack the other constraint, $f_Y(\mathcal{U}, \mathcal{V} - Y)$ is a random variable on the outcome space $\mathcal{V}(\mathcal{V}, \mathcal{U})$, and that it has the same value as Y is an event which, according to the equation f_Y , must always occur. Given a PSEM \mathcal{M} , let $\{\mathcal{M}\}$ consist of all joint distributions $\nu(\mathcal{U}, \mathcal{V})$ that satisfy the two constraints

above (or just the first of them, in the case of a non-probabilistic SEM). $\{\mathcal{M}\}$ can be thought of as the set of distributions compatible wth \mathcal{M} . It ; this set captures the behavior of \mathcal{M} in the absence of interventions. A joint distribution $\mu(\mathbf{X})$ over $\mathbf{X} \subseteq \mathcal{V} \cup \mathcal{U}$ can arise from a (P)SEM \mathcal{M} iff there is some $\nu \in \{\mathcal{M}\}$ whose marginal on \mathbf{X} is μ .

We now review the syntax of a language for describing causality. A *basic causal formula* is one of the form $[\mathbf{Y} \leftarrow \mathbf{y}] \varphi$, where φ is a Boolean expression over the endogenous variables \mathcal{V} , $\mathbf{Y} \subseteq \mathcal{V}$ is a subset of them, and $\mathbf{y} \in \mathcal{V}(\mathbf{Y})$. The language then consists of all Boolean combinations of basic formulas. In a causal model M and context $\mathbf{u} \in \mathcal{V}(\mathcal{U})$, a Boolean expression φ over \mathcal{V} is true iff it holds for all $(\mathbf{u}, \mathbf{x}) \in \mathcal{V}(\mathcal{U}, \mathcal{V})$ consistent with the equations of M . Basic causal formulas are then given semantics by $(M, \mathbf{u}) \models [\mathbf{Y} \leftarrow \mathbf{y}] \varphi$ iff $(M_{\mathbf{Y} \leftarrow \mathbf{y}}, \mathbf{u}) \models \varphi$, where $M_{\mathbf{Y} \leftarrow \mathbf{y}}$ is the result of changing each f_Y , for $Y \in \mathbf{Y}$, to the constant function $s \mapsto \mathbf{y}[Y]$, which returns (on all inputs s) the value of Y in the joint setting \mathbf{y} . From here, the truth relation can be extended to arbitrary causal formulas by structural induction in the usual way.² The dual formula $\langle \mathbf{Y} \leftarrow \mathbf{y} \rangle \varphi := \neg[\mathbf{Y} \leftarrow \mathbf{y}] \neg \varphi$ is equivalent to $[\mathbf{Y} \leftarrow \mathbf{y}] \varphi$ in SEMs where each context \mathbf{u} induces a unique setting of the endogenous variables [37]. A PSEM $\mathcal{M} = (M, P)$ assigns probabilities to causal formulas according to $\Pr_{\mathcal{M}}(\varphi) := P(\{\mathbf{u} \in \mathcal{V}(\mathcal{U}) : (M, \mathbf{u}) \models \varphi\})$.

Some authors assume that for each variable X , there is a special “independent noise” exogenous variable U_X (often written ϵ_X in the literature) on which only the equation f_X can depend; we call a PSEM (M, P) *randomized* if it contains such exogenous variables that are mutually independent according to P , and *fully randomized* if all its exogenous variables are of this form. Randomized PSEMs are clearly a special class of PSEMs but any PSEM can be converted to an equivalent

² $M \models \varphi_1 \wedge \varphi_2$ iff $M \models \varphi_1$ and $M \models \varphi_2$; $M \models \neg \varphi$ iff $M \not\models \varphi$.

randomized PSEM by extending it with additional dummy variables $\{U_X\}_{X \in \mathcal{V}}$ that can take only a single value. Thus, we do not lose expressive power by using randomized PSEMs. In fact, *qualitatively*, randomized PSEMs are more expressive: they can encode independence. **It should come as no surprise that randomized PSEMs and QIM-compatibility are related.**

5.3.1 The Equivalence Between QIM-Compatibility and Randomized PSEMs

We are now equipped to formally describe the connection between QIM-compatibility and causality. At a high level, this connection should be unsurprising: witnesses and causal models both relate dependency structures to distributions, but in “opposite directions”. QIM-compatibility starts with distributions and asks what dependency structures they are compatible with. Causal models, on the other hand, are explicit (quantitative) representations of dependency structures that give rise to sets of distributions. We now show that the existence of a causal model coincides with the existence of a witness. We start by showing this for the hypergraphs generated by graphs (like Bayesian networks, except possibly cyclic), which we show correspond to fully randomized causal models ([Proposition 5.3.1](#)). We then give a natural generalization of a causal model that exactly captures QIM-compatibility with an arbitrary hypergraph ([Proposition 5.3.2](#)). In both cases, the high-level result is the same: $\mu \models \mathcal{A}$ iff there is a causal model that “has dependency structure \mathcal{A} ” that gives rise to μ .

More precisely, we say that a randomized causal model \mathcal{M} *has dependency structure \mathcal{A}* iff there is a 1-1 correspondence between $a \in \mathcal{A}$ and the equations of \mathcal{M} , such that the equation f_a produces a value of T_a and depends only on

S_a and U_a . This definition emphasizes the hypergraph; here is a more concrete alternative emphasizing the randomized PSEM: \mathcal{M} is of dependency structure \mathcal{A} iff the targets of \mathcal{A} are disjoint singletons (the elements of \mathcal{V}), and $\text{Pa}_{\mathcal{M}}(Y) \subseteq S_Y \cup \{U_Y\}$ for all $Y \in \mathcal{V}$. We start by presenting the result in the case where \mathcal{A} corresponds to a directed graph.

Proposition 5.3.1. *Given a graph G and a distribution μ , $\mu \models \Diamond \mathcal{A}_G$ iff there exists a fully randomized PSEM of dependency structure \mathcal{A}_G from which μ can arise.*

[link to proof]

Proposition 5.3.1 shows that, for those hypergraphs induced by graphs, QIM-compatibility means arising from a fully randomized PSEM of the appropriate dependency structure. Theorem 5.2.1 makes precise a phenomenon that seems to be almost universally implicitly understood but, to the best of our knowledge, has not been formalized before: every acyclic fully randomized SEM induces a distribution with the independencies of the corresponding Bayesian Network—and, conversely, every distribution with those independencies arises from such a causal model.

It is easy to extend this result to the dependency structures of all randomized PSEMs. But what happens if \mathcal{A} contains hyperarcs with overlapping targets? Here the correspondence starts to break down for a simple reason: by definition, there is at most one equation per variable in a (P)SEM; thus, no PSEM can have dependency structure \mathcal{A} . Nevertheless, the correspondence between witnesses and causal models persists if we simply drop the (traditional) requirement that \mathcal{F} is indexed by \mathcal{V} . This leads us to consider a natural generalization of a (randomized) PSEM that has an arbitrary set of equations—not just one per variable.

Definition 5.3.2. Let $(\mathcal{N}, \mathcal{A})$ be a hypergraph. A *generalized randomized PSEM* $\mathcal{M} = (\mathcal{X}, \mathcal{U}, \mathcal{F}, P)$ with structure \mathcal{A} consists of sets of variables \mathcal{X} and $\mathcal{U} = \{U_a\}_{a \in \mathcal{A}}$, together with a set of functions $\mathcal{F} = \{f_a : \mathcal{V}(S_a) \times \mathcal{V}(U_a) \rightarrow \mathcal{V}(T_a)\}_{a \in \mathcal{A}}$, and a probability P_a over each independent noise variable U_a . The meanings of $\{\mathcal{M}\}$ and *can arise* are the same as for a PSEM. □

Proposition 5.3.2. $\mu \models \Diamond \mathcal{A}$ iff there exists a generalized randomized PSEM with structure \mathcal{A} from which μ can arise. link to proof

Generalized randomized PSEMs can capture functional dependencies, and constraints. For instance, an equality (say $X = Y$) can be encoded in a generalized randomized PSEM with a second equation for X . Indeed, we believe that generalized randomized PSEMs can capture a wide class of constraints, and are closely related to *causal models with constraints* [8], a discussion we defer to future work.

5.3.2 Interventions and the Correspondence Between Witnesses and Causal Models

We have seen that QIM-compatibility with \mathcal{A} (i.e., the existence of a witness $\bar{\mu}$) coincides exactly with the existence of a causal model \mathcal{M} from which a distribution can arise. But which witnesses correspond to which causal models? The answer to this question will be critical to extend the correspondence we have given so that it can deal with interventions. Different causal models may give rise to the same distribution, yet handle interventions differently.

There are two directions of the correspondence. Given a randomized PSEM

\mathcal{M} , distributions arising from it are compatible with its dependency structure, and the corresponding witnesses are exactly the distributions in $\{\mathcal{M}\}$ (see Section 5.E). In particular, if \mathcal{M} is acyclic, there is a unique witness. The converse is more interesting: how can we turn a witness into a causal model?

Construction 5.3.3. Given a witness $\bar{\mu}(\mathcal{X})$ to compatibility with a hypergraph \mathcal{A} with disjoint targets, construct a PSEM according to the following (non-deterministic) procedure. Take $\mathcal{V} := \cup_{a \in \mathcal{A}} T_a$, $\mathcal{U} := \mathcal{U}_{\mathcal{A}} \cup (\mathcal{X} - \mathcal{V})$, and $P(\mathcal{U}) := \bar{\mu}(\mathcal{U})$. For each $X \in \mathcal{V}$, there is a unique $a_X \in \mathcal{A}$ whose targets T_{a_X} contain X . Since $\bar{\mu} \models (U_{a_X}, S_{a_X}) \rightarrow T_{a_X}$ (this is just property (c) in Definition 5.2.2), $X \in T_{a_X}$ must also be a function of S_{a_X} and U_{a_X} ; take f_X to be such a function. More precisely, for each $u \in \mathcal{V}(U_{a_X})$ and $s \in \mathcal{V}(S_{a_X})$ for which $\bar{\mu}(U_{a_X} = u, S_{a_X} = s) > 0$, there is a unique $t \in \mathcal{V}(T_{a_X})$ such that $\bar{\mu}(u, s, t) > 0$. In this case, set $f_X(u, s, \dots) := t[X]$. If $\bar{\mu}(U_{a_X} = u, S_{a_X} = s) = 0$, $f_X(u, s, \dots)$ can be an arbitrary function of u and s . Let $\text{PSEMs}_{\mathcal{A}}(\bar{\mu})$ denote the set of PSEMs that can result. \square

It's clear from Construction 5.3.3 that $\text{PSEMs}_{\mathcal{A}}(\bar{\mu})$ is always nonempty, and is a singleton iff $\bar{\mu}(u, s) > 0$ for all $(a, u, s) \in \sqcup_{a \in \mathcal{A}} \mathcal{V}(U_a, S_a)$. A witness with this property exists when μ is positive (i.e., $\mu(\mathcal{X} = \mathbf{x}) > 0$ for all $\mathbf{x} \in \mathcal{V}(\mathcal{X})$), in which case the construction gives a unique causal model. Conversely, we have seen that an acyclic model \mathcal{M} gives rise to a unique witness. So, in the simplest cases, models \mathcal{M} with structure \mathcal{A} and witnesses $\bar{\mu}$ to compatibility with \mathcal{A} are equivalent. But there are two important caveats.

1. A causal model \mathcal{M} can contain more information than a witness $\bar{\mu}$ if some events have probability zero. For instance, $\bar{\mu}$ could be a point mass on a single joint outcome ω of all variables that satisfies the equations of \mathcal{M} . But \mathcal{M} cannot be reconstructed uniquely from $\bar{\mu}$ because there may be many

causal models for which ω is a solution.

2. A witness $\bar{\mu}$ can contain more information than a causal model \mathcal{M} if \mathcal{M} is cyclic. For example, suppose that \mathcal{M} consists of two variables, X and X' , and equations $f_X(X') = X'$ and $f_{X'}(X) = X$. In this case, $\bar{\mu}$ cannot be reconstructed from \mathcal{M} , because \mathcal{M} does not contain information about the distribution of X .

These two caveats appear to be very different, but they fit together in a surprisingly elegant way.

Proposition 5.3.3. *If $\bar{\mu}(\mathcal{X}, \mathcal{U}_{\mathcal{A}})$ is a witness for QIM-compatibility with \mathcal{A} and \mathcal{M} is a PSEM with dependency structure \mathcal{A} , then $\bar{\mu} \in \{\mathcal{M}\}$ if and only if $\mathcal{M} \in \text{PSEMs}_{\mathcal{A}}(\bar{\mu})$.*

link to
proof

Equivalently, this means that $\text{PSEMs}_{\mathcal{A}}(\bar{\mu})$, the possible outputs of [Construction 5.3.3](#), are precisely the randomized PSEMs of dependency structure \mathcal{A} that can give rise to $\bar{\mu}$. This is already substantial evidence that causal models $\mathcal{M} \in \text{PSEMs}_{\mathcal{A}}(\bar{\mu})$ are closely related to the QIM-compatibility witness $\bar{\mu}$. But everything we have seen so far describes only the correspondence in the absence of intervention, a setting in which many causal models are indistinguishable. We now show that the correspondence goes deeper, by extending it to interventions.

In any randomized PSEM M , we can define an event

$$\text{do}_M(\mathbf{X}=\mathbf{x}) := \bigcap_{X \in \mathbf{X}} \bigcap_{\mathbf{s} \in \mathcal{V}(\mathbf{Pa}(X))} f_X(U_X, \mathbf{s}) = \mathbf{x}[X], \quad \begin{array}{l} \text{where } \mathbf{x}[X] \text{ is} \\ \text{the value of } X \\ \text{in } \mathbf{x}. \end{array} \quad (5.1)$$

This is intuitively the event in which the randomness is such that $\mathbf{X} = \mathbf{x}$ regardless of the values of the parent variables.³ As we now show, conditioning on

³This is essentially the event in which, for each $X \in \mathbf{X}$, the response variable $\hat{U}_X := \lambda s. f_X(s, U_X)$, whose possible values $\mathcal{V}(\hat{U}_X)$ are functions from $\mathcal{V}(\mathbf{Pa}_M(X))$ to $\mathcal{V}(X)$ [84, 7], takes on the constant function $\lambda p. x$.

$\text{do}_M(\mathbf{X}=\mathbf{x})$ has the effect of intervention.

Theorem 5.3.4. Suppose that $\bar{\mu}$ is a witness to $\mu \models \Diamond \mathcal{A}$, $\mathcal{M} \in \text{PSEM}_{\mathcal{A}}(\bar{\mu})$, $\mathbf{X} \subseteq \mathcal{X}$ and $\mathbf{x} \in \mathcal{V}(\mathbf{X})$. If $\bar{\mu}(\text{do}_{\mathcal{M}}(\mathbf{X}=\mathbf{x})) > 0$, then:

[link to proof]

(a) $\bar{\mu}(\mathcal{X} \mid \text{do}_{\mathcal{M}}(\mathbf{X}=\mathbf{x}))$ can arise from $\mathcal{M}_{\mathbf{X} \leftarrow \mathbf{x}}$;

(b) for all events $\varphi \subseteq \mathcal{V}(\mathcal{X})$, $\Pr_{\mathcal{M}}([\mathbf{X} \leftarrow \mathbf{x}] \varphi) \leq \bar{\mu}(\varphi \mid \text{do}_{\mathcal{M}}(\mathbf{X}=\mathbf{x})) \leq \Pr_{\mathcal{M}}([\langle \mathbf{X} \leftarrow \mathbf{x} \rangle] \varphi)$

and all three are equal when $\mathcal{M} \models \mathcal{U} \twoheadrightarrow \mathcal{X}$ (such as when \mathcal{M} is acyclic).

Theorem 5.3.4 shows that the relationship between witnesses and causal models extends to interventions. Even when $\text{do}_M(\mathbf{X}=\mathbf{x})$ has probability zero, it is always possible to find a nearly equivalent setting where the bounds of the theorem apply.⁴ Intervention and conditioning are conceptually very different, so it may seem surprising that conditioning can have the effect of intervention (and also that the Pearl's $\text{do}(\cdot)$ notation actually corresponds to an event [43]). We emphasize that the conditioning (on $\text{do}_M(\mathbf{X}=\mathbf{x})$) is on the randomness U_X and not X itself; intervening on $\mathbf{X}=\mathbf{x}$ is indeed fundamentally different from conditioning on $\mathbf{X}=\mathbf{x}$.

5.4 QIM-Compatibility and Information Theory

The fact that the dependency structure of a (causal) Bayesian network describes the independencies of the distribution it induces is fundamental to both causal-

⁴More precisely, for all $\epsilon > 0$, there exists some \mathcal{M}' that differs from \mathcal{M} on the probabilities all causal formulas by at most ϵ , and a distribution $\bar{\mu}'$ that is ϵ -close to $\bar{\mu}$, such that $\bar{\mu}'(\text{do}_{\mathcal{M}'}(\mathbf{X}=\mathbf{x})) > 0$. As a result, Theorem 5.3.4 places bounds on the conditional probabilities that are possible limits of sequences of distributions $(\nu_k)_{k \geq 0}$ where $\nu_k(\text{do}_M(\mathbf{X}=\mathbf{x})) > 0$, i.e., the possible outcomes of conditioning a *non-standard* probability measure [38] on this probability-zero event.

ity and probability. It makes explicit the distributional consequences of BN structure. Yet, despite substantial interest [6], generalizing the BN case to more complex (e.g., cyclic) dependency structures remains largely an open problem. In Section 5.4.1, we generalize the BN case by providing an information-theoretic constraint, capable of capturing conditional independence, functional dependence, and more, on the distributions that can arise from an *arbitrary* dependency structure. This connection between causality and information theory has implications for both fields. It grounds the cyclic dependency structures found in causality in concrete constraints on the distributions they represent. At the same time, it allows us to resolve longstanding confusion about structure in information theory, clarifying the meaning of the so-called “interaction information”, and recasting a standard counterexample to substantiate the claim it was intended to oppose. In Section 5.4.2, we strengthen this connection. Using entropy to measure distance to (in)dependence, we develop a scoring function to measure how far a distribution is from being QIM-compatible with a given dependency structure. This function turns out to have an intimate relationship with the qualitative PDG scoring function $IDef$, which we use to show that our information-theoretic constraints degrade gracefully on “near-compatible” distributions.

We now review the critical information theoretic concepts and their relationships to (in)dependence (see Section 5.C.1 for a full primer). Conditional entropy $H_\mu(Y|X)$ measures how far μ is from satisfying the functional dependency $X \twoheadrightarrow Y$. Conditional mutual information $I_\mu(Y; Z|X)$ measures how far μ is from satisfying the conditional independence $Y \perp\!\!\!\perp Z | X$. Linear combinations of these quantities

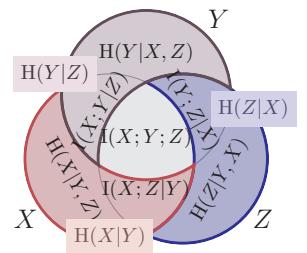


Figure 1: I_μ .

(for $X, Y, Z \subseteq \mathcal{X}$) can be viewed as the inner product between a coefficient vector \mathbf{v} and a $2^{|\mathcal{X}|} - 1$ dimensional vector \mathbf{I}_μ that we will call the *information profile* of μ . For three variables, the components of this vector are illustrated in Figure 1 (right). It is not hard to see that an arbitrary conjunction of (conditional) (in)dependencies can be expressed as a constraint $\mathbf{I}_\mu \cdot \mathbf{v} \geq 0$, for some appropriate vector \mathbf{v} .

We now formally introduce the qualitative PDG scoring function $IDef$, which interprets a hypergraph structure \mathcal{A} as a function of the form $\mathbf{I}_\mu \cdot \mathbf{v}_{\mathcal{A}}$. This *information deficiency*, given by

$$IDef_{\mathcal{A}}(\mu) = \mathbf{I}_\mu \cdot \mathbf{v}_{\mathcal{A}} := -H_\mu(\mathcal{X}) + \sum_{a \in \mathcal{A}} H_\mu(T_a \mid S_a), \quad (5.2)$$

is the difference between the number of bits needed to (independently) specify the randomness in μ along the hyperarcs of \mathcal{A} , and the number of bits needed to specify a sample of μ according to its own structure ($\emptyset \rightarrow \mathcal{X}$). While $IDef$ has some nice properties⁵, it can also behave unintuitively in some cases; for instance, it can be negative. Clearly, it does not measure how close μ is to being structurally compatible with \mathcal{A} , in general. Nevertheless, there is still a fundamental relationship between $IDef$ and QIM-compatibility, as we now show.

5.4.1 A Necessary Condition for QIM-Compatibility

What constraints does QIM-compatibility with \mathcal{A} place on a distribution μ ? When G is a dag, we have seen that if $\mu \models \Diamond \mathcal{A}_G$, then μ must satisfy the independencies

⁵It captures BN independencies and the dependencies of Theorem 5.2.2, reduces to maximum entropy for the empty hypergraph, and combines with the quantitative PDG scoring function [82] to capture factor graphs.

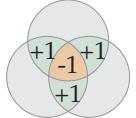
of the corresponding Bayesian network ([Theorem 5.2.1](#)); we have also seen that additional hyperarcs impose functional dependencies ([Theorem 5.2.2](#)). But these results apply only when \mathcal{A} is of a very special form. More generally, $\mu \models \Diamond \mathcal{A}$ implies that μ can arise from some randomized causal model whose equations have dependency structure \mathcal{A} ([Propositions 5.3.1](#) and [5.3.2](#)). Still, unless \mathcal{A} has a particularly special form, it is not obvious whether or not this says something about μ . The primary result of this section is an information-theoretic bound ([Theorem 5.4.1](#)) that generalizes most of the concrete consequences of QIM-compatibility we have seen so far ([Theorems 5.2.1](#) and [5.2.2](#)). The result is a connection between information theory and causality; it yields an information-theoretic test for complex causal dependency structures, and enables causal notions of structure to dispel misconceptions in information theory.

Theorem 5.4.1. *If $\mu \models \Diamond \mathcal{A}$, then $IDef_{\mathcal{A}}(\mu) \leq 0$.*

link to proof

[Theorem 5.4.1](#) applies to all hypergraphs, and subsumes every general-purpose technique we know of for proving that $\mu \not\models \mathcal{A}$. Indeed, the negative directions of [Theorems 5.2.1](#) and [5.2.2](#) are immediate consequences of it. To illustrate some of its subtler implications, let's return to the 3-cycle in [Example 9](#).

Example 10. It is easy to see (e.g., by inspecting [Figure 1](#)) that $IDef_{3\text{-cycle}}(\mu) = H_{\mu}(Y|X) + H_{\mu}(Z|Y) + H_{\mu}(X|Z) - H_{\mu}(XYZ) = -I_{\mu}(X; Y; Z)$. [Theorem 5.4.1](#) therefore tells us that a distribution μ that is QIM-compatible with the 3-cycle cannot have negative interaction information $I_{\mu}(X; Y; Z)$. What does this mean? **Overall, conditioning on the value of one variable can only reduce the amount of remaining information in other variables (in expectation).** When $I(X; Y; Z) < 0$, conditioning on one variable causes the other two to share more information than they did

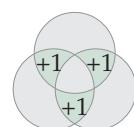


before. The most extreme instance is μ_{xor} , the distribution in which two variables are independent and the third is their parity (illustrated on the right). It seems intuitively clear that μ_{xor} cannot arise from the 3-cycle, a causal model with only pairwise dependencies. This is difficult to prove directly, but is an immediate consequence of [Theorem 5.4.1](#). \triangle

For many, there is an intuition that $I(X; Y; Z) < 0$ should require a fundamentally “3-way” interaction between the variables, and should not arise through pairwise interactions alone [46]. This has been a source of conflict [95, 63, 61, 20], because traditional ways of making precise “pairwise interactions” (e.g., maximum entropy subject to pairwise marginal constraints and pairwise factorization) do not ensure that $I(X; Y; Z) \geq 0$. But QIM-compatibility does. One can verify by enumeration that the 3-cycle is the most expressive causal structure with no joint dependencies, and we have already proven that QIM-compatibility with that hypergraph implies non-negative interaction information. QIM-compatibility has another even more noteworthy clarifying effect on information theory.

There is a school of thought that contends that *all* structural information in $\mu(\mathcal{X})$ is captured by its information profile I_μ . This position has fallen out of favor in some communities due to standard counterexamples: distributions that have intuitively different structures yet share an information profile [47]. However, with “structure” explicated by compatibility, the prototypical counterexample of this kind suddenly supports the very notion it was meant to challenge, suggesting in an unexpected way that the information profile may yet capture the essence of probabilistic structure.

Example 11. Let A, B , and C be variables with $\mathcal{V}(A), \mathcal{V}(B), \mathcal{V}(C) = \{0, 1\}^2$. Using independent fair coin flips X_1, X_2 , and X_3 , define



two joint distributions, P and Q , over A, B, C as follows. Define $P(A, B, C)$ by letting $A := (X_1, X_2)$, $B := (X_2, X_3)$, and $C := (X_3, X_1)$. Define Q by letting $A := (X_1, X_2)$, $B := (X_1, X_3)$, and $C := (X_1, X_2 \oplus X_3)$. Structurally, P and Q appear to be very different. According to P , the first components of the three variables (A, B, C) are independent, yet they are identical according to Q . Moreover, P has only simple pairwise interactions between the variables, while P has μ_{xor} (a clear 3-way interaction) embedded within it. Yet P and Q have identical information profiles (see right): in both cases, each of $\{A, B, C\}$ is determined by the values of the other two, each pair share one bit of information given the third, and $I(A; B; C) = 0$.

This example has been used to argue that multivariate Shannon information does not take into account important structural differences between distributions [47]. We are now in a position to give a novel and particularly persuasive response, by appealing to QIM-compatibility.⁶ Unsurprisingly, P is compatible with the 3-cycle; it clearly consists of “2-way” interactions, as each pair of variables shares a bit. But, counterintuitively, the distribution Q is *also* compatible with the 3-cycle! (The reader is encouraged to verify that $U_1 = X_3 \oplus X_1$, $U_2 = X_2$, and $U_3 = X_3$ serves as a witness.) To emphasize: this is despite the fact that Q is just μ_{xor} (which is certainly not compatible with the 3-cycle) together with a seemingly irrelevant random bit X_1 . By the results of Section 5.3, this means there is a causal model without joint dependence giving rise to Q —so, despite appearances, Q does not require a 3-way interaction. Indeed, P and Q are QIM-compatible with precisely the same hypergraphs over $\{A, B, C\}$, suggesting that they don’t have a structural difference after all. \triangle

⁶Note that P and Q no longer have the same profile if we split each variable into its two components. Since the notion of “component” is based on the assignment \mathcal{V} of variables to possible values, our view that \mathcal{V} is not structural information diffuses this counterexample by assumption—but the present argument is much stronger.

In light of [Example 11](#), one might reasonably conjecture that the converse of [Theorem 5.4.1](#) holds. Unfortunately, it does not (see [Section 5.C.4](#)); the quantity $IDef_{\mathcal{A}}(\mu)$ does not completely determine whether or not $\mu \models \Diamond \mathcal{A}$. We now pursue a new (entropy-based) scoring function that does. This will allow us to generalize [Theorem 5.4.1](#) to distributions that are only “near-compatible” with \mathcal{A} .

5.4.2 A Scoring Function for QIM-Compatibility

Here is a function that measures how far a distribution μ is from being QIM-compatible with \mathcal{A} .

$$QIMInc_{\mathcal{A}}(\mu) := \inf_{\substack{\nu(\mathcal{U}, \mathcal{X}) \\ \nu(\mathcal{X}) = \mu(\mathcal{X})}} -H_{\nu}(\mathcal{U}) + \sum_{a \in \mathcal{A}} H_{\nu}(U_a) + \sum_{a \in \mathcal{A}} H_{\nu}(T_a | S_a, U_a). \quad (5.3)$$

$QIMInc$ is a direct translation of [Definition 5.2.2](#) (a-c); it measures the (optimal) quality of an extended distribution ν as a witness. The infimum restricts the search to ν satisfying (a), the first two terms measure ν 's discrepancy of with (b), and the last term measures ν 's discrepancy with (c). Therefore:

Proposition 5.4.2. $QIMInc_{\mathcal{A}}(\mu) \geq 0$, with equality iff $\mu \models \mathcal{A}$.

[[link to proof](#)]

Although they seem to be very different, $QIMInc$ and $IDef$ turn out to be closely related. In fact, modulo the infimum, $QIMInc_{\mathcal{A}}$ is a special case of $IDef$ —not for the hypergraph \mathcal{A} , but rather for a transformed one \mathcal{A}^{\dagger} that models the noise variables explicitly. To construct \mathcal{A}^{\dagger} from \mathcal{A} , add new nodes $\mathcal{U} = \{U_a\}_{a \in \mathcal{A}}$, and replace each hyperarc



Finally, add one additional hyperarc $\mathcal{U} \rightarrow \mathcal{X}$. (Intuitively, this hyperarc creates functional dependencies in the spirit of [Theorem 5.2.2](#).) With these definitions in place, we can state a theorem that bounds QIMInc above and below with information deficiencies. The lower bound generalizes [Theorem 5.4.1](#) by giving an upper limit on $IDef_{\mathcal{A}}(\mu)$ even for distributions μ that are not QIM-compatible with \mathcal{A} . The upper bound is tight in general, and shows that $\text{QIMInc}_{\mathcal{A}}$ can be equivalently defined as a minimization over $IDef_{\mathcal{A}^\dagger}$.

Theorem 5.4.3. (a) If $(\mathcal{X}, \mathcal{A})$ is a hypergraph, $\mu(\mathcal{X})$ is a distribution, and $\nu(\mathcal{X}, \mathcal{U})$ is an extension of ν to additional variables $\mathcal{U} = \{U_a\}_{a \in \mathcal{A}}$ indexed by \mathcal{A} , then:

$$IDef_{\mathcal{A}}(\mu) \leq \text{QIMInc}_{\mathcal{A}}(\mu) \leq IDef_{\mathcal{A}^\dagger}(\nu).$$

(b) For all μ and \mathcal{A} , there is a choice of ν that achieves the upper bound. That is,

$$\text{QIMInc}_{\mathcal{A}}(\mu) = \min \left\{ IDef_{\mathcal{A}^\dagger}(\nu) : \begin{array}{l} \nu \in \Delta\mathcal{V}(\mathcal{X}, \mathcal{U}) \\ \nu(\mathcal{X}) = \mu(\mathcal{X}) \end{array} \right\}.$$

where the minimization is over all possible ways of assigning values to the variables in \mathcal{U} . The minimum is achieved when $|\mathcal{V}(U_a)| \leq |\mathcal{V}(T_a)|^{|\mathcal{V}(S_a)|}$.

5.5 Discussion

We have shown how directed hypergraphs can be used to represent structural aspects of distributions. Moreover, they can do so in a way that generalizes conditional independencies and functional dependencies and has deep connections to causality and information theory. Many open questions remain. A major one is that of more precisely understanding QIM-compatibility in cyclic models. We do not yet know, for example, whether the same set of distributions are

QIM-compatible with the clockwise and counter-clockwise 3-cycles. A related problem is to find an efficient procedure that can determine whether a given distribution is QIM-compatible with a hypergraph. We hope to explore all these questions in future work.

APPENDICES FOR CHAPTER 5

5.A Proofs

We begin with a de-randomization construction, that will be useful for the proofs.

5.A.1 From CPDs to Distributions over Functions

Compare two objects:

- a cpd $p(Y|X)$, and
- a distribution $q(Y^X)$ over functions $g : \mathcal{V}X \rightarrow \mathcal{V}Y$.

The latter is significantly larger — if both $|\mathcal{V}X| = |\mathcal{V}Y| = N$, then q is a N^N dimensional object, while p is only dimension N^2 . A choice of distribution $q(Y^X)$ corresponds to a unique choice cpd $p(Y|X)$, according to

$$p(Y=y | X=x) := q(Y^X(x) = y).$$

Claim 1. 1. *The definition above in fact yields a cpd, i.e., $\sum_y p(Y=y|X=x) = 1$ for*

all $x \in \mathcal{V}X$.

2. *This definition of $p(Y|X)$ is the conditional marginal of any joint distribution $\mu(X, Y, Y^X)$ satisfying $\mu(Y^X) = q$ and $\mu(Y = Y^X(X)) = 1$.*

Both p and q give probabilistic information about Y conditioned on X . But $q(Y^X)$ contains strictly more information. Not only does it specify the distribution over Y given $X=x$, but it also contains counter-factual information about

the distribution of Y if X were equal to x' , conditioned on the fact that, in reality, $X=x$.

Is there a natural construction that goes in the opposite direction, intuitively making as many independence assumptions as possible? It turns out there is:

$$q(Y^X = g) = \prod_{x \in \mathcal{V}X} p(Y=g(x) \mid X=x).$$

Think of Y^X as a collection of variables $\{Y^x : x \in \mathcal{V}X\}$ describing the value of the function for each input, so that q is a joint distribution over them. This construction simply asks that these variables be independent. Specifying a distribution with these independences amounts to a choice of “marginal” distribution $q(Y^x)$ for each $x \in \mathcal{V}X$, and hence is essentially a function of type $\mathcal{V}X \rightarrow \Delta \mathcal{V}Y$, the same as p . In addition, if we apply the previous construction, we recover p , since:

$$\begin{aligned} q(Y^X(x) = y) &= \sum_{g: \mathcal{V}X \rightarrow \mathcal{V}Y} \mathbb{1}[g(x) = y] \prod_{x' \in \mathcal{V}X} p(Y=g(x') \mid X=x') \\ &= \sum_{g: \mathcal{V}X \rightarrow \mathcal{V}Y} \mathbb{1}[g(x) = y] p(Y=g(x) \mid X=x) \prod_{x' \neq x} p(Y=g(x') \mid X=x') \\ &= p(Y=y \mid X=x) \sum_{g: \mathcal{V}X \rightarrow \mathcal{V}Y} \mathbb{1}[g(x) = y] \prod_{x' \neq x} p(Y=g(x') \mid X=x') \\ &= p(Y=y \mid X=x) \sum_{g: \mathcal{V}X \setminus \{x\} \rightarrow \mathcal{V}Y} \prod_{x' \in \mathcal{V}X \setminus \{x\}} p(Y=g(x') \mid X=x') \\ &= p(Y=y \mid X=x). \end{aligned}$$

The final equality holds because the remainder of the terms can be viewed as the probability of selecting any function from $X \setminus \{x\}$ to Y , under an analogous measure; thus, it equals 1. This will be a useful construction for us in general.

5.A.2 Results on (In)dependence

Lemma 5.A.1. Suppose X_1, \dots, X_n are variables, Y_1, \dots, Y_n are sets, and for each $i \in \{1, \dots, n\}$, we have a function $f_i : \mathcal{V}(X_i) \rightarrow Y_i$. Then if X_1, \dots, X_n are mutually independent (according to a joint distribution μ), then so are $f_1(X_1), \dots, f_n(X_n)$.

Proof. This is an intuitive fact, but we provide a proof for completeness. Explicitly, mutual independence of X_1, \dots, X_n means that, for all joint settings $\mathbf{x} = (x_1, \dots, x_n)$, we have $\mu(X_1=x_1, \dots, X_n=x_n) = \prod_{i=1}^n \mu(X_i=x_i)$. So, for any joint setting $\mathbf{y} = (y_1, \dots, y_n) \in Y_1 \times \dots \times Y_n$, we have

$$\begin{aligned}
\mu(f_1(X_1)=y_1, \dots, f_n(X_n)=y_n) &= \mu(\{\mathbf{x} : \mathbf{f}(\mathbf{x}) = \mathbf{y}\}) \\
&= \sum_{\substack{(x_1, \dots, x_n) \in \mathcal{V}(X_1, \dots, X_n) \\ f_1(x_1)=y_1, \dots, f_n(x_n)=y_n}} \mu(X_1=x_1, \dots, X_n=x_n) \\
&= \sum_{\substack{x_1 \in \mathcal{V}X_1 \\ f_1(x_1)=y_1}} \dots \sum_{\substack{x_n \in \mathcal{V}X_n \\ f_n(x_n)=y_n}} \mu(X_1=x_1, \dots, X_n=x_n) \\
&= \sum_{\substack{x_1 \in \mathcal{V}X_1 \\ f_1(x_1)=y_1}} \dots \sum_{\substack{x_n \in \mathcal{V}X_n \\ f_n(x_n)=y_n}} \prod_{i=1}^n \mu(X_i=x_i) \\
&= \left(\sum_{\substack{x_1 \in \mathcal{V}X_1 \\ f_1(x_1)=y_1}} \mu(X_1=x_1) \right) \dots \left(\sum_{\substack{x_n \in \mathcal{V}X_n \\ f_n(x_n)=y_n}} \mu(Y_1=y_1) \right) \\
&= \prod_{i=1}^n \mu(f_i(X_i)=y_i). \quad \square
\end{aligned}$$

Lemma 5.A.2 (properties of determination).

1. If $\nu \models A \twoheadrightarrow B$ and $\nu \models A \twoheadrightarrow C$, then $\nu \models A \twoheadrightarrow (B, C)$.
2. If $\nu \models A \twoheadrightarrow B$ and $\nu \models B \twoheadrightarrow C$, then $\nu \models A \twoheadrightarrow C$.

Proof. $\nu \models X \twoheadrightarrow Y$, means there exists a function $f : V(A) \rightarrow V(B)$ such that

$\nu(f(Y) = X) = 1$, i.e., the event $f(A) = B$ occurs with probability 1.

1. Let $f : \mathcal{V}(A) \rightarrow \mathcal{V}(B)$ and $g : \mathcal{V}(A) \rightarrow \mathcal{V}(C)$ be such that $\nu(f(A) = B) = 1 = \nu(g(A) = C)$. Since both events happen with probability 1, so must the event $f(A) = B \cap g(A) = C$. Thus the event $(f(A), g(A)) = (B, C)$ occurs with probability 1. Therefore, $\nu \models A \twoheadrightarrow (B, C)$.
2. The same ideas, but faster: we have $f : \mathcal{V}(A) \rightarrow \mathcal{V}(B)$ as before, and $g : \mathcal{V}(B) \rightarrow \mathcal{V}(C)$, such that the events $f(A) = B$ and $g(B) = C$ occur with probability 1. By the same logic, it follows that their conjunction holds with probability 1, and hence $C = f(g(A))$ occurs with probability 1. So $\nu \models A \twoheadrightarrow C$.

□

Theorem 5.2.1. *If G is a directed acyclic graph and $\mathcal{I}(G)$ consists of the independencies of its corresponding Bayesian network, then $\mu \models \Diamond \mathcal{A}_G$ if and only if μ satisfies $\mathcal{I}(G)$.*

Proof. Label the vertices of $G = (\mathcal{N}, E)$ by natural numbers so that they are a topological sort of G —that is, without loss of generality, suppose $\mathcal{N} = [n] := \{1, 2, \dots, n\}$, and $i < j$ whenever $i \rightarrow j \in E$. By the definition of \mathcal{A}_G , the arcs $\mathcal{A}_G = \{S_i \xrightarrow{i} i\}_{i=1}^n$ are also indexed by integers. Finally, write $\mathcal{X} = (X_1, \dots, X_n)$ for the variables \mathcal{X} corresponding to \mathcal{N} over which μ is defined.

(\implies). Suppose $\mu \models \mathcal{A}_G$. This means there is an extension of $\bar{\mu}(\mathcal{X}, \mathcal{U})$ of $\mu(\mathcal{X})$ to additional independent variables $\mathcal{U} = (U_1, \dots, U_n)$, such that $\bar{\mu} \models (S_i, U_i) \twoheadrightarrow i$ for all $i \in [n]$.

First, we claim that if $\bar{\mu}$ is such a witness, then $\bar{\mu} \models (U_1, \dots, U_k) \rightarrow\!\!\!\rightarrow (X_1, \dots, X_k)$ for all $k \in [n]$, and so in particular, $\bar{\mu} \models \mathcal{U} \rightarrow\!\!\!\rightarrow \mathcal{X}$. This follows from QIM-compatibility's condition (c) and the fact that G is acyclic, by induction. In more detail: The base case of $k = 0$ holds vacuously. Suppose that $\bar{\mu} \models (X_1, \dots, X_k)$ for some $k < n$. Now, condition (c) of [Definition 5.2.2](#) says $\bar{\mu} \models (S_{k+1}, U_{k+1}) \rightarrow\!\!\!\rightarrow X_{k+1}$. Because the variables are sorted in topological order, the parent variables S_{k+1} are a subset of $\{X_1, \dots, X_n\}$, which are determined by \mathcal{U} by the induction hypothesis; at the same time clearly $\bar{\mu} \models (U_1, \dots, U_{k+1}) \rightarrow\!\!\!\rightarrow U_{k+1}$ as well. So, by two instances of [Lemma 5.A.2](#), $\bar{\mu} \models (U_1, \dots, U_{k+1}) \rightarrow\!\!\!\rightarrow X_{k+1}$. Combining with our inductive hypothesis, we find that $\bar{\mu} \models (U_1, \dots, U_{k+1}) \rightarrow\!\!\!\rightarrow (X_1, \dots, X_{k+1})$. So, by induction, $\bar{\mu} \models (U_1, \dots, U_k) \rightarrow\!\!\!\rightarrow (X_1, \dots, X_k)$ for $k \in [n]$, and in particular, $\bar{\mu} \models \mathcal{U} \rightarrow\!\!\!\rightarrow \mathcal{X}$.

With this in mind, we now return to proving that μ has the required independencies. It suffices to show that $\mu(\mathcal{X}) = \prod_{i=1}^n \mu(X_i \mid S_i)$. We do so by showing that, for all $k \in [n]$, $\mu(X_1, \dots, X_k) = \mu(X_1, \dots, X_{k-1})\mu(X_k \mid S_k)$. By QIM-compatibility witness condition (c), we know that $\bar{\mu} \models (S_k, U_k) \rightarrow\!\!\!\rightarrow X_k$, and so there exists a function $f_k : \mathcal{V}(S_k) \times \mathcal{V}(U_k) \rightarrow \mathcal{V}(X_k)$ for which the event $f_k(S_k, U_k) = X_k$ occurs with probability 1. Since $\bar{\mu} \models (U_1, \dots, U_{k-1}) \rightarrow\!\!\!\rightarrow (X_1, \dots, X_{k-1})$, and U_k is independent of (U_1, \dots, U_{k-1}) , it follows from [Lemma 5.A.1](#) that $\bar{\mu} \models (X_1, \dots, X_{k-1}) \perp\!\!\!\perp U_k$. Thus

$$\begin{aligned}\mu(X_1, \dots, X_{k-1}, X_k) &= \sum_{u \in \mathcal{V}(U_k)} \mu(X_1, \dots, X_{k-1})\bar{\mu}(U_k = u) \cdot \mathbb{1}[X_k = f_k(S_k, u)] \\ &= \mu(X_1, \dots, X_{k-1}) \sum_{u \in \mathcal{V}(U_k)} \bar{\mu}(U_k = u) \cdot \mathbb{1}[X_k = f_k(S_k, u)]\end{aligned}$$

Observe that the quantity on the right, including the sum, is a function of X_k and S_k , but no other variables; let $\varphi(X_k, S_k)$ denote this quantity. Because μ is a probability distribution, know that $\varphi(X_k, S_k)$ must be the conditional probability of X_k given X_1, \dots, X_{k-1} , and it depends only on the variables S_k . Thus

$$\mu(X_1, \dots, X_k) = \mu(X_1, \dots, X_{k-1})\mu(X_k \mid S_k).$$

Therefore $\nu(\mathcal{X}) = \mu(\mathcal{X})$ factors as required by the BN G , meaning that μ has the independencies specified by G . (See Koller & Friedman Thm 3.2, for instance.)

(\Leftarrow). Suppose μ satisfies the independencies of G , meaning that each node is conditionally independent of its non-descendants given its parents. We now repeatedly apply the construction [Section 5.A.1](#) to construct a QIM-compatibility witness. Specifically, for $k \in \{1, \dots, n\}$, let U_k be a variable whose values $\mathcal{V}(U_k) := \mathcal{V}(X_k)^{\mathcal{V}(S_k)}$ are functions from values of X_k 's parents, to values of X_k . Let \mathcal{U} denote the joint variable (U_1, \dots, U_n) , and observe that a setting $\mathbf{g} = (g_1, \dots, g_n)$ of \mathcal{U} uniquely picks out a value of \mathcal{X} , by evaluating each function in order. Let's call this function $f : \mathcal{V}(\mathcal{U}) \rightarrow \mathcal{V}(\mathcal{X})$.

To be more precise, we now construct $f(\mathbf{g})$ inductively. The first component we must produce is X_1 , but since X_1 has no parents, g_1 effectively describes a single value of X_1 , so we define the first component $f(\mathbf{g})[X_1]$ to be that value. More generally, assuming that we have already defined the components X_1, \dots, X_{i-1} , among which are the variables S_k on which X_i depends, we can determine the value of X_i ; formally, this means defining

$$f(\mathbf{g})[X_i] := g_i(f(\mathbf{g})[S_i]),$$

which, by our inductive assumption, is well-defined. Note that, for all $\mathbf{g} \in \mathcal{V}(\mathcal{U})$ and $\mathbf{x} \in \mathcal{V}(\mathcal{X})$, the function f is characterized by the property

$$f(\mathbf{g}) = \mathbf{x} \iff \bigwedge_{i=1}^n g_i(\mathbf{x}[S_i]) = \mathbf{x}[X_i]. \quad (5.4)$$

To quickly verify this: if $f(\mathbf{g}) = \mathbf{x}$, then in particular, for $i \in [n]$, then $\mathbf{x}[X_i] = f(\mathbf{g})[X_i] = g_i(\mathbf{x}[S_i])$ by the definition above. Conversely, if the right hand side of

(5.4) holds, then we can prove $f(\mathbf{g}) = \mathbf{x}$ by induction over our construction of f : if $f(\mathbf{g})[X_j] = \mathbf{x}[X_j]$ for all $j < i$, then $f(\mathbf{g})[X_i] = g_i(f(\mathbf{g})[S_i]) = g_i(\mathbf{x}[S_i]) = \mathbf{x}[X_i]$.

Next, we define an unconditional probability over each U_k according to

$$\bar{\mu}_i(U_i = g) := \prod_{\mathbf{s} \in \mathcal{V}(S_k)} \mu(X_i = g(s) \mid S_i = \mathbf{s}),$$

which, as verified in [Section 5.A.1](#), is indeed a conditional probability, and has the property that $\bar{\mu}_i(U_i(\mathbf{s}) = x) = \mu(X_i = x \mid S_i = \mathbf{s})$ for all $x \in \mathcal{V}(X_i)$ and $\mathbf{s} \in \mathcal{V}(S_i)$. By taking an independent combination (tensor product) of each of these unconditional distributions, we obtain a joint distribution $\bar{\mu}(\mathcal{U}) = \prod_{i=1}^n \bar{\mu}_i(U_i)$. Finally, we extend this distribution to a full joint distribution $\bar{\mu}(\mathcal{U}, \mathcal{X})$ via the pushforward of $\bar{\mu}(\mathcal{U})$ through the function f defined by induction above. In this distribution, each X_i is determined by U_i and S_i .

By construction, the variables \mathcal{U} are mutually independent (for [Definition 5.2.2\(b\)](#)), and satisfy $(S_k, U_k) \rightarrowtail X_k$ for all $k \in [n]$ ([Definition 5.2.2\(c\)](#)). It remains only to verify that the marginal of $\bar{\mu}$ on the variables \mathcal{X} is the original distribution μ ([Definition 5.2.2\(a\)](#)). Here is where we rely on the fact that μ satisfies the independencies of G , which means that we can factor $\mu(\mathcal{X})$ as

$$\mu(\mathcal{X}) = \prod_{i=1}^n \mu(X_i \mid S_i).$$

$$\begin{aligned}
\bar{\mu}(\mathcal{X}=\mathbf{x}) &= \sum_{\mathbf{g} \in \mathcal{V}(\mathcal{U})} \bar{\mu}(\mathcal{U}=\mathbf{g}) \cdot \delta f(\mathbf{x} \mid \mathbf{g}) \\
&= \sum_{(g_1, \dots, g_n) \in \mathcal{V}(\mathcal{U})} \mathbb{1}[\mathbf{x} = f(\mathbf{g})] \prod_{i=1}^n \bar{\mu}(U_i=g_i) \\
&= \sum_{(g_1, \dots, g_n) \in \mathcal{V}(\mathcal{U})} \mathbb{1}\left[\bigwedge_{i=1}^n g_i(\mathbf{x}[S_i]) = \mathbf{x}[X_i]\right] \prod_{i=1}^n \bar{\mu}(U_i=g_i) \quad [\text{by (5.4)}] \\
&= \prod_{i=1}^n \sum_{g \in \mathcal{V}(U_i)} \mathbb{1}[g(\mathbf{x}[S_i]) = \mathbf{x}[X_i]] \cdot \bar{\mu}(U_i=g) \\
&= \prod_{i=1}^n \bar{\mu}\left(\left\{g \in \mathcal{V}(U_i) \mid g(\mathbf{s}_i) = x_i\right\}\right) \quad \text{where } \begin{array}{l} x_i := \mathbf{x}[X_i], \\ \mathbf{s}_i := \mathbf{x}[S_i] \end{array} \\
&= \prod_{i=1}^n \bar{\mu}(U_i(\mathbf{s}_i) = x_i) \\
&= \prod_{i=1}^n \mu(X_i = x_i \mid S_i = \mathbf{s}_i) \\
&= \mu(\mathcal{X} = \mathbf{x}).
\end{aligned}$$

Therefore, when μ satisfies the independencies of a BN G , it is QIM-compatible with \mathcal{A}_G . \square

Before we move on to proving the other results in the paper, we first illustrate how this relatively substantial first half of the proof of [Theorem 5.2.1](#) can be dramatically simplified by relying on two information theoretic arguments.

Alternate, information-based proof. (\implies). Let G be a dag. If $\mu \models \mathcal{A}_G$, then by [Theorem 5.4.1](#), $IDef_{\mathcal{A}_G}(\mu) \leq 0$. In the appendix of [82], it is shown that $IDef_{\mathcal{A}_G}(\mu) \geq 0$ with equality iff μ satisfies the BN's independencies. Thus μ must satisfy the appropriate independencies. \square

Theorem 5.2.2.

- (a) $\mu \models X \rightarrow\!\!\!\rightarrow Y \wedge \Diamond \mathcal{A}$ if and only if $\forall n \geq 0. \mu \models \Diamond \mathcal{A} \sqcup_{X \rightarrow Y}^{(+n)}$.
- (b) if $\mathcal{A} = \mathcal{A}_G$ for a dag G , then $\mu \models X \rightarrow\!\!\!\rightarrow Y \wedge \Diamond \mathcal{A}$ if and only if $\mu \models \Diamond \mathcal{A} \sqcup_{X \rightarrow Y}^{(+1)}$.
- (c) if $\exists a \in \mathcal{A}$ such that $S_a = \emptyset$ and $X \in T_a$, then $\mu \models X \rightarrow\!\!\!\rightarrow Y \wedge \Diamond \mathcal{A}$ iff $\mu \models \Diamond \mathcal{A} \sqcup_{X \rightarrow Y}^{(+2)}$.

Proof. (a). The forward direction is straightforward. Suppose that $\mu \models \mathcal{A}$ and $\mu \models X \rightarrow\!\!\!\rightarrow Y$. The former condition gives us a witness $\nu(\mathcal{X}, \mathcal{U})$ in which $\mathcal{U} = \{U_a\}_{a \in \mathcal{A}}$ are mutually independent variables indexed by \mathcal{A} , that determine their respective edges. “Extend” ν in the unique way to n additional constant variables U_1, \dots, U_n , each of which can only take on one value. We claim that this “extended” distribution ν' , which we conflate with ν because it is not meaningfully different, is a witness to $\mu \models \Diamond \mathcal{A} \sqcup_{X \rightarrow Y}^{(+n)}$. Since $\mu \models X \rightarrow\!\!\!\rightarrow Y$ it must also be that $\nu \models X \rightarrow\!\!\!\rightarrow Y$, and it follows that $\nu \models (X, U_i) \rightarrow\!\!\!\rightarrow Y$ for all $i \in \{1, \dots, n\}$, demonstrating that the new requirements of ν' imposed by Definition 5.2.2(c) hold. (The remainder of the requirements for condition (c), namely that $\nu' \models (S_a, U_a) \rightarrow\!\!\!\rightarrow T_a$ for $a \in \mathcal{A}$, still hold because ν' is an extension of ν , which we know has this property.) Finally, since \mathcal{U} are mutually independent and each U_i is a constant (and hence independent of everything), the variables $\mathcal{U}' := \mathcal{U} \sqcup \{U_i\}_{i=1}^n$ are also mutually independent. Thus ν (or, more precisely, an isomorphic “extension” of it to additional trivial variables) is a witness of $\mu \models \Diamond \mathcal{A} \sqcup_{X \rightarrow Y}^{(+n)}$.

The reverse direction is difficult to prove directly, yet it is a straightforward application of Theorem 5.4.1. Suppose that $\mu \models \mathcal{A} \sqcup_{X \rightarrow Y}^{(+n)}$ for all $n \geq 0$. By

Theorem 5.4.1, we know that

$$0 \geq SDef_{\mathcal{A} \sqcup \overset{(+n)}{X \rightarrow Y}}(\mu) = SDef_{\mathcal{A}}(\mu) + n H_\mu(Y|X).$$

Because $SDef_{\mathcal{A}r}(\mu)$ is bounded below (by $-\log |\mathcal{V}(\mathcal{X})|$), it cannot be the case that $H_\mu(Y|X) > 0$; otherwise, the inequality above would not hold for large n (specifically, for $n > \log |\mathcal{V}(\mathcal{X})|/H_\mu(Y|X)$). By Gibbs inequality, $H_\mu(Y|X)$ is non-negative, and thus it must be the case that $H_\mu(Y|X) = 0$. Thus $\mu \models X \twoheadrightarrow Y$. It is also true that $\mu \models \diamond \mathcal{A}$ by monotonicity (Theorem 5.B.2), which is itself a direct application of Theorem 5.4.1

(b). Now $\mathcal{A} = \mathcal{A}_G$ for some graph G . The forward direction of the equivalence is strictly weaker than the one we already proved in part (a); we have shown $\mu \models \diamond \mathcal{A} \sqcup \overset{(+n)}{X \rightarrow Y}$ for all $n \geq 0$, and needed only to show it for $n = 1$. The reverse direction is what's interesting. As before, we will take a significant shortcut by using Theorem 5.4.1. Suppose $\mu \models \diamond \mathcal{A} \sqcup \overset{(+1)}{X \rightarrow Y}$. In this case where $\mathcal{A} = \mathcal{A}_G$, it was shown by Richardson and Halpern [82] that $SDef_{\mathcal{A}}(\mu) \geq 0$. It follows that

$$0 \stackrel{(\text{Theorem 5.4.1})}{\geq} SDef_{\mathcal{A} \sqcup \overset{(+n)}{X \rightarrow Y}}(\mu) = SDef_{\mathcal{A}}(\mu) + H_\mu(Y|X) \geq 0,$$

and thus $H_\mu(Y|X) = 0$, meaning that $\mu \models X \twoheadrightarrow Y$ as promised. As before, we also have $\mu \models \diamond \mathcal{A}$ by monotonicity.

(c). As in part (b), the forward direction is a special case of the forward direction of part (a), and it remains only to prove the reverse direction. Equipped with the additional information that $\mathcal{A} \rightsquigarrow \{\rightarrow \{X\}\}$, suppose that $\mu \models \diamond \mathcal{A} \sqcup \overset{(+2)}{X \rightarrow Y}$. By monotonicity, this means $\mu \models \mathcal{A}$ and also that $\mu \models \rightarrow \boxed{X} \not\supset \boxed{Y}$. Let \mathcal{A}' denote this hypergraph. Once again by appeal to Theorem 5.4.1, we have that

$$0 \geq SDef_{\mathcal{A}'} = -H_\mu(X, Y) + H(X) + 2H_\mu(Y|X) = H_\mu(Y|X) \geq 0.$$

It follows that $H_\mu(Y|X) = 0$, and thus $\mu \models X \twoheadrightarrow Y$. As mentioned above, we also know that $\mu \models \mathcal{A}$, and thus $\mu \models \Diamond\mathcal{A} \wedge X \twoheadrightarrow Y$ as promised. \square

5.A.3 Causality Results of Section 5.3

Proposition 5.3.1. *Given a graph G and a distribution μ , $\mu \models \Diamond \mathcal{A}_G$ iff there exists a fully randomized PSEM of dependency structure \mathcal{A}_G from which μ can arise.*

Proof. (\implies). Suppose $\mu \models \mathcal{A}_G$. Thus there exists some witness $\bar{\mu}(\mathcal{X}, \mathcal{U})$ to this fact, satisfying conditions (a-c) of Definition 5.2.2. Because \mathcal{A}_G is partitional, the elements of $\text{PSEMs}_{\mathcal{A}_G}(\bar{\mu})$ are ordinary (i.e., not generalized) randomized PSEMs. We claim that every $\mathcal{M} = (M, P) \in \text{PSEMs}_{\mathcal{A}_G}(\bar{\mu})$ that is a randomized PSEM from which μ can arise, and also has the property that $\text{Pa}_M(Y) \subseteq \text{Pa}_G(Y) \cup \{U_Y\}$ for all $Y \in \mathcal{X}$.

- The hyperarcs of \mathcal{A}_G correspond to the vertices of G , which in turn correspond to the variables in \mathcal{X} ; thus $\mathcal{U} = \{U_X\}_{X \in \mathcal{X}}$. By property (b) of QIM-compatibility witnesses (Definition 5.2.2), these variables $\{U_X\}_{X \in \mathcal{X}}$ are mutually independent according to $\bar{\mu}$. Furthermore, because $\mathcal{M} = (M, P) \in \text{PSEMs}_{\mathcal{A}_G}(\bar{\mu})$, we know that $\bar{\mu}(\mathcal{U}) = P$, and thus the variables in \mathcal{U} must be mutually independent according to P . By construction, in causal models $\mathcal{M} \in \text{PSEMs}_{\mathcal{A}_G}(\bar{\mu})$ the equation f_Y can depend only on $S_Y = \text{Pa}_G(Y) \subseteq \mathcal{X}$ and U_Y . So, in particular, f_Y does not depend on U_X for $X \neq Y$.

Altogether, we have shown that \mathcal{M} contains exogenous variables $\{U_X\}_{X \in \mathcal{X}}$ that are mutually independent according to P , and that f_Y does not depend on U_X when $X \neq Y$. Thus, \mathcal{M} is a randomized PSEM.

- By condition (a) on QIM-compatibility witnesses (Definition 5.2.2), we know that $\bar{\mu}(\mathcal{X}) = \mu$. By Proposition 5.3.3(a), we know that $\mu \in \{\mathcal{M}\}$. Together, the previous two sentences mean that μ can arise from \mathcal{M} .

- Finally, as mentioned in the first bullet item, the equation f_Y in M can depend only on $S_Y = \text{Pa}_G(Y)$ and on U_Y . Thus $\text{Pa}_M(Y) \subseteq \text{Pa}_G(Y) \cup \{U_Y\}$ for all $Y \in \mathcal{X}$.

Under the assumption that $\mu \models \mathcal{A}_G$, we have now shown that there exists a randomized causal model \mathcal{M} from which μ can arise, with the property that $\text{Pa}_{\mathcal{M}}(Y) \subseteq \text{Pa}_G(Y) \cup \{U_Y\}$ for all $Y \in \mathcal{X}$.

(\Leftarrow). Conversely, suppose there is a randomized PSEM $\mathcal{M} = (M = (\mathcal{Y}, \mathcal{U}, \mathcal{F}), P)$ with the property that $\text{Pa}_M(Y) \subseteq \text{Pa}_G(Y) \cup \{U_Y\}$ for all Y , from which μ can arise. The last clause means there exists some $\nu \in \{\mathcal{M}\}$ such that $\nu(\mathcal{X}) = \mu$. We claim that this ν is a witness to $\mu \models \mathcal{A}_G$. We already know that condition (a) of being a QIM-compatibility witness is satisfied, since $\nu(\mathcal{X}) = \mu$. Condition (b) holds because of the assumption that $\{U_X\}_{X \in \mathcal{X}}$ are mutually independent in the distribution P for a randomized PSEM (and the fact that $\nu(\mathcal{U}) = P$, since $\nu \in \{\mathcal{M}\}$). Finally, we must show that (c) for each $Y \in \mathcal{X}$, $\nu \models \text{Pa}_G(Y) \cup \{U_Y\} \rightarrowtail Y$. Since $\nu \in \{\mathcal{M}\}$, we know that M 's equation holds with probability 1 in ν , and so it must be the case that $\nu \models \text{Pa}_M(Y) \rightarrowtail Y$. Note that, in general, if $A \subseteq B$ and $A \rightarrowtail C$, then $B \rightarrowtail C$. By assumption, $\text{Pa}_M(Y) \subseteq \text{Pa}_G(Y) \cup \{U_Y\}$, and thus $\nu \models \text{Pa}_G(Y) \cup \{U_Y\} \rightarrowtail Y$.

Thus ν satisfies all conditions (a-c) for a QIM-compatibility witness, and hence $\mu \models \mathcal{A}_G$. □

Proposition 5.3.2. $\mu \models \Diamond \mathcal{A}$ iff there exists a generalized randomized PSEM with structure \mathcal{A} from which μ can arise.

Proof. (\implies). Suppose $\mu \models \mathcal{A}$, meaning there exists a witness $\nu(\mathcal{X}, \mathcal{U})$ with property [Definition 5.2.2\(c\)](#), meaning that, for all $a \in \mathcal{A}$, there is a functional dependence $(S_a, U_a) \twoheadrightarrow T_a$. Thus, there is some set of functions \mathcal{F} with these types that holds with probability 1 according to ν . Meanwhile, by [Definition 5.2.2\(b\)](#), $\nu(\mathcal{U})$ are mutually independent, so defining $P_a(U_a) := \nu(U_a)$, we have $\nu(\mathcal{U}) = \prod_{a \in \mathcal{A}} P_a(U_a)$. Together, the previous two conditions (non-deterministically) define a generalized randomized PSEM \mathcal{M} of shape \mathcal{A} for which $\nu \in \{\mathcal{M}\}$. Finally, by [Definition 5.2.2\(a\)](#), we know that μ can arise from \mathcal{M} .

(\Leftarrow). Conversely, suppose there is a generalized randomized SEM \mathcal{M} of shape \mathcal{A} from which $\mu(\mathcal{X})$ can arise. Thus, there is some $\nu \in \{\mathcal{M}\}$ whose marginal on \mathcal{X} is μ . We claim that this ν is also a witness that $\mu \models \mathcal{A}$. The marginal constraint from [Definition 5.2.2\(a\)](#) is clearly satisfied. Condition (b) is immediate as well, because $\nu(\mathcal{U}) = \prod_a P_a(U_a)$. Finally, condition (c) is satisfied, because the equations of \mathcal{M} hold with probability 1, ensuring the appropriate functional dependencies. \square

Proposition 5.3.3. *If $\bar{\mu}(\mathcal{X}, \mathcal{U}_{\mathcal{A}})$ is a witness for QIM-compatibility with \mathcal{A} and \mathcal{M} is a PSEM with dependency structure \mathcal{A} , then $\bar{\mu} \in \{\mathcal{M}\}$ if and only if $\mathcal{M} \in \text{PSEMs}_{\mathcal{A}}(\bar{\mu})$.*

Proof. (a) is straightforward. Suppose $\mathcal{M} \in \text{PSEMs}(\nu)$. By construction, the equations of \mathcal{M} reflect functional dependencies in ν , and hence hold with probability 1.⁷ Furthermore, the distribution $P(\mathcal{U})$ in all $\mathcal{M} \in \text{PSEMs}(\nu)$ is equal to $\nu(\mathcal{U})$. These two facts, demonstrate that ν satisfies the two constraints required for

⁷When the probability of some combination of source variables is zero, there is typically more than one choice of functions that holds with probability 1; the choice of functions is essentially the choice of $\mathcal{M} \in \text{PSEMs}(\nu)$.

membership in $\{\mathcal{M}\}$.

(b). We do the two directions separately. First, suppose $\mathcal{M} \in \text{PSEMs}(\nu)$. We have already shown (in part (a)) that $\nu \in \{\mathcal{M}\}$. The construction of $\text{PSEMs}(\nu)$ depends on the hypergraph \mathcal{A} (even if the dependence is not explicitly clear from our notation) in such a way that f_X does not depend on any variables beyond U_a and S_{a_X} . Thus, $\text{Pa}_{\mathcal{M}}(X) \subseteq S_{a_X} \cup \{U_{a_X}\}$.

Conversely, suppose $\mathcal{M} = (\mathcal{X}, \mathcal{U}, \mathcal{F})$ is a PSEM satisfying $\nu \in \{\mathcal{M}\}$ and $\text{Pa}_{\mathcal{M}}(X) \subseteq S_{a_X} \cup \{U_{a_X}\}$. We would like to show that $\mathcal{M} \in \text{PSEMs}(\nu)$. Because $\nu \in \{\mathcal{M}\}$, we know that the distribution $P(\mathcal{U})$ over the exogenous variables in the PSEM \mathcal{M} is equal to $\nu(\mathcal{U})$, matching the first part of our construction. What remains is to show that the equations \mathcal{F} are consistent with our transformation. Choose any $X \in \mathcal{X}$. Because \mathcal{A} is subpartitional, there is a unique $a_X \in \mathcal{A}$ such that $X \in T_{a_X}$. Now choose any values $s \in \mathcal{V}(S_{a_X})$ and $u \in \mathcal{V}(U_{a_X})$. If $\nu(s, u) > 0$, then we know there is a unique value of $x \in \mathcal{V}(X)$ such that $\nu(s, u, x) > 0$. Since \mathcal{M} 's equation for X , f_X , depends only on s and u , and holds with probability 1, we know that $f_X(s, u) = t$, as required. On the other hand, if $\nu(s, u) = 0$, then any choice of $f_X(s, u)$ is consistent with our procedure. Since this is true for all X , and all possible inputs to the equation f_X , we conclude that the equations \mathcal{F} can arise from the procedure described in the main text, and therefore $\mathcal{M} \in \text{PSEMs}(\nu)$. \square

Theorem 5.3.4. Suppose that $\bar{\mu}$ is a witness to $\mu \models \Diamond \mathcal{A}$, $\mathcal{M} \in \text{PSEMs}_{\mathcal{A}}(\bar{\mu})$, $\mathbf{X} \subseteq \mathcal{X}$ and $\mathbf{x} \in \mathcal{V}(\mathbf{X})$. If $\bar{\mu}(\text{do}_{\mathcal{M}}(\mathbf{X}=\mathbf{x})) > 0$, then:

(a) $\bar{\mu}(\mathcal{X} \mid \text{do}_{\mathcal{M}}(\mathbf{X}=\mathbf{x}))$ can arise from $\mathcal{M}_{\mathbf{X} \leftarrow \mathbf{x}}$;

(b) for all events $\varphi \subseteq \mathcal{V}(\mathcal{X})$, $\Pr_{\mathcal{M}}([\mathbf{X} \leftarrow \mathbf{x}] \varphi) \leq \bar{\mu}(\varphi \mid \text{do}_{\mathcal{M}}(\mathbf{X}=\mathbf{x})) \leq \Pr_{\mathcal{M}}(\langle \mathbf{X} \leftarrow \mathbf{x} \rangle \varphi)$

and all three are equal when $\mathcal{M} \models \mathcal{U} \rightarrowtail \mathcal{X}$ (such as when \mathcal{M} is acyclic).

Proof. (part a). Let $(M, P) := \mathcal{M}$ be the SEM and probability over exogenous variable in the PSEM \mathcal{M} , and $\mathcal{F} = \{f_Y\}_{Y \in \mathcal{X}}$ be its set of equations. Because we have assumed $\nu(\text{do}_M(\mathbf{X}=\mathbf{x})) > 0$, the conditional distribution

$$\nu \mid \text{do}_M(\mathbf{X}=\mathbf{x}) = \nu(\mathcal{U}, \mathcal{X}) \cdot \prod_{X \in \mathbf{X}} \mathbb{1}[\forall \mathbf{s}. f_X(U_X, \mathbf{s}) = \mathbf{x}[X]] \Big/ \nu(\text{do}_M(\mathbf{X}=\mathbf{x}))$$

is defined. By assumption, $\mathcal{M} \in \text{PSEMs}(\nu)$ and ν is a witness to $\mu \models \mathcal{A}$. Thus, by Proposition 5.3.3, we know that $\nu \in \{\mathcal{M}\}$. So in particular, all equations of \mathcal{M} hold for all joint settings $(\mathbf{u}, \omega) \in \mathcal{V}(\mathcal{X} \cup \mathcal{U})$ in the support of ν . But the support of the conditional distribution $\nu \mid \text{do}_M(\mathbf{X}=\mathbf{x})$ is a subset of the support of ν , so all equations of \mathcal{M} also hold in the conditioned distribution. Furthermore, the event $\text{do}_M(\mathbf{X}=\mathbf{x})$ is the event in which, for all $X \in \mathbf{X}$, the variable U_X takes on a value such that $f_X(\dots, U_X, \dots) = \mathbf{x}[X]$. Thus the equations corresponding to $\mathbf{X} = \mathbf{x}$ also hold with probability 1 in $\nu \mid \text{do}_M(\mathbf{X}=\mathbf{x})$.

This shows that all equations of $\mathcal{M}_{\mathbf{X} \leftarrow \mathbf{x}}$ hold with probability 1 in $\nu \mid \text{do}_M(\mathbf{X}=\mathbf{x})$. However, the marginal distribution $\nu(\mathcal{U} \mid \text{do}_M(\mathbf{X}=\mathbf{x}))$ over \mathcal{U} will typically not be the distribution $P(\mathcal{U})$ —indeed, we have altered collapsed distribution of the variables $\mathcal{U}_{\mathbf{X}} := \{U_X : X \in \mathbf{X}\}$. So, strictly speaking, $\nu \mid \text{do}_M(\mathbf{X}=\mathbf{x}) \notin \{\mathcal{M}_{\mathbf{X} \leftarrow \mathbf{x}}\}$. Our objective, therefore, is to show that there is a *different* distribution $\nu' \in \{\mathcal{M}_{\mathbf{X} \leftarrow \mathbf{x}}\}$ such that $\nu'(\mathcal{X}) = \nu(\mathcal{X} \mid \text{do}_M(\mathbf{X}=\mathbf{x}))$. Let $\mathbf{Z} := \mathcal{X} \setminus \mathbf{X}$, and $\mathcal{U}_{\mathbf{Z}} := \{U_Z : Z \in \mathbf{Z}\}$. We can define ν' according to

$$\nu'(\mathcal{X}, \mathcal{U}_{\mathbf{X}}, \mathcal{U}_{\mathbf{Z}}) := \nu(\mathcal{X}, \mathcal{U}_{\mathbf{Z}} \mid \text{do}_M(\mathbf{X}=\mathbf{x}))P(\mathcal{U}_{\mathbf{X}}).$$

This distribution satisfies three critical properties:

1. Clearly ν' has the appropriate marginal $\nu'(\mathcal{X}) = \nu(\mathcal{X} \mid \text{do}_M(\mathbf{X}=\mathbf{x}))$ on exogenous variables \mathcal{X} , by construction.

2. At the same time, the marginal on exogenous variables is

$$\begin{aligned}
\nu'(\mathcal{U}) &= \nu'(\mathcal{U}_X, \mathcal{U}_Z) \\
&= \int_{\mathcal{V}(\mathcal{X})} \nu(\omega, \mathcal{U}_Z \mid \text{do}_M(\mathbf{X}=\mathbf{x})) P(\mathcal{U}_X) d\omega \\
&= P(\mathcal{U}_X) \nu(\mathcal{U}_Z \mid \text{do}_M(\mathbf{X}=\mathbf{x})) \\
&= P(\mathcal{U}_X) P(\mathcal{U}_Z \mid \text{do}_M(\mathbf{X}=\mathbf{x})) && [\text{since } \text{do}_M(\mathbf{X}=\mathbf{x}) \text{ depends only on } \mathcal{U}] \\
&= P(\mathcal{U}_X) P(\mathcal{U}_Z) && \left[\begin{array}{l} \text{since } \text{do}_M(\mathbf{X}=\mathbf{x}) \text{ depends} \\ \text{only on } \mathcal{U}_X, \text{ while } \mathcal{U}_X \text{ and} \\ \mathcal{U}_Z \text{ are independent in } \nu \text{ (by} \\ \text{the witness condition).} \end{array} \right] \\
&= P(\mathcal{U}_X, \mathcal{U}_Z) && [\text{same reason as above}]
\end{aligned}$$

3. Finally, ν' satisfies all equations of $\mathcal{M}_{X \leftarrow x}$. It satisfies the equations for the variables \mathbf{X} because $\mathbf{X} = \mathbf{x}$ holds with probability 1. At the same time, the equations in $\mathcal{M}_{X \leftarrow x}$ corresponding to the variables \mathbf{Z} hold with probability 1, because the marginal $\nu'(\mathcal{U}_Z, \mathcal{X})$ is shared with the distribution $\nu \mid \text{do}_M(\mathbf{X}=\mathbf{x})$ —and that distribution satisfies these equations. (It suffices to show that they share this particular marginal because the equations for \mathbf{Z} do not depend on \mathcal{U}_X .)

Together, items 2 and 3 show that $\nu' \in \{\mathcal{M}_{X \leftarrow x}\}$, and item 1 shows that $\nu(\mathcal{X} \mid \text{do}_M(\mathbf{X}=\mathbf{x}))$ can arise from $\mathcal{M}_{X \leftarrow x}$.

(part b). We will again make use of the distribution ν' defined in part (a), and its three critical properties listed above. Given a setting $\mathbf{u} \in \mathcal{V}(\mathcal{U})$ of the exogenous variables, let

$$\mathcal{F}_{X \leftarrow x}(\mathbf{u}) := \left\{ \omega \in \mathcal{V}(\mathcal{X}) \mid \begin{array}{ll} \forall X \in \mathbf{X}. & \omega[X] = \mathbf{x}[X] \\ \forall Y \in \mathcal{X} \setminus \mathbf{X}. & \omega[Y] = f_X(\omega[\mathcal{X} \setminus Y], \mathbf{u}) \end{array} \right\}$$

denote the set of joint settings of endogenous variables that are consistent with the equations of $\mathcal{M}_{\mathbf{X} \leftarrow \mathbf{x}}$.

If $\mathbf{u} \in \mathcal{V}(\mathcal{U})$ is such that

$$\begin{aligned} (M, \mathbf{u}) \models [\mathbf{X} \leftarrow \mathbf{x}] \varphi &\iff (M_{\mathbf{X} \leftarrow \mathbf{x}}, \mathbf{u}) \models \varphi \\ &\iff \forall \omega \in \mathcal{F}_{\mathbf{X} \leftarrow \mathbf{x}}(\mathbf{u}). \omega \in \varphi \\ &\iff \mathcal{F}_{\mathbf{X} \leftarrow \mathbf{x}}(\mathbf{u}) \subseteq \varphi, \end{aligned}$$

then ϕ holds at all points that satisfy the equations of $M_{\mathbf{X} \leftarrow \mathbf{x}}$. So, since ν' is supported only on such points (property 3), it must be that $\nu'(\varphi) = 1$. By property 1, $\nu'(\varphi) = \nu(\varphi \mid \text{do}_M(\mathbf{X} = \mathbf{x}))$.

Furthermore, if $\nu'(\varphi) > 0$, then there must exist some $\omega \in \mathcal{F}_{\mathbf{X} \leftarrow \mathbf{x}}(\mathbf{u})$ satisfying φ , and thus $(M, \mathbf{u}) \models \langle \mathbf{X} \leftarrow \mathbf{x} \rangle \varphi$. Putting both of these observations together, and with a bit more care to the symbolic manipulation, we find that:

$$\begin{aligned} \Pr_{\mathcal{M}}([\mathbf{X} \leftarrow \mathbf{x}] \varphi) &= P(\{\mathbf{u} \in \mathcal{V}(\mathcal{U}) : (M, \mathbf{u}) \models [\mathbf{X} \leftarrow \mathbf{x}] \varphi\}) \\ &= \sum_{\mathbf{u} \in \mathcal{V}(\mathcal{U})} P(\mathbf{u}) \mathbb{1}[\mathcal{F}_{\mathbf{X} \leftarrow \mathbf{x}}(\mathbf{u}) \subseteq \varphi] \\ &\leq \sum_{\mathbf{u} \in \mathcal{V}(\mathcal{U})} P(\mathbf{u}) \nu'(\varphi \mid \mathbf{u}) = \nu'(\varphi) = \nu(\varphi \mid \text{do}_M(\mathbf{X} = \mathbf{x})) \\ &\leq \sum_{\mathbf{u} \in \mathcal{V}(\mathcal{U})} P(\mathbf{u}) \mathbb{1}[\mathcal{F}_{\mathbf{X} \leftarrow \mathbf{x}}(\mathbf{u}) \cap \varphi \neq \emptyset] \\ &= P(\{\mathbf{u} \in \mathcal{V}(\mathcal{U}) : (M, \mathbf{u}) \models \langle \mathbf{X} \leftarrow \mathbf{x} \rangle \varphi\}) \\ &= \Pr_{\mathcal{M}}(\langle \mathbf{X} \leftarrow \mathbf{x} \rangle \varphi), \quad \text{as desired.} \end{aligned}$$

Finally, if $\nu \models \mathcal{U} \twoheadrightarrow \mathcal{X}$, then $\mathcal{F}_{\mathbf{X} \leftarrow \mathbf{x}}(\mathbf{u})$ is a singleton for all \mathbf{u} , and hence φ holding for all $\omega \in \mathcal{F}_{\mathbf{X} \leftarrow \mathbf{x}}$ and for some $\omega \in \mathcal{F}_{\mathbf{X} \leftarrow \mathbf{x}}$ are equivalent. So, in this case,

$$(M, \mathbf{u}) \models [\mathbf{X} \leftarrow \mathbf{x}] \varphi \iff (M, \mathbf{u}) \models \langle \mathbf{X} \leftarrow \mathbf{x} \rangle \varphi,$$

and thus the probability of both formulas are the same—and it must also equal $\nu(\varphi \mid \text{do}_M(\mathbf{X}=\mathbf{x}))$ which we have shown lies between them. \square

Proposition 5.B.4. *

Proof. 1. Suppose $\mu \models \mathcal{A}_G$, $\bar{\mu}(\mathcal{U}, \mathcal{X})$ is a witness to this, and $\mathcal{M} \in \text{PSEM}_{\mathcal{A}_G}(\bar{\mu})$.

(\implies). Suppose $\mathbf{X} \perp\!\!\!\perp_G \mathbf{Y} \mid \mathbf{Z}$, meaning every path from \mathbf{X} to \mathbf{Y} in G goes through \mathbf{Z} . We now show that

⟨ INCOMPLETE ⟩

□

5.A.4 Information Theoretic Results of Section 5.4

To prove [Theorem 5.4.1](#) and [Theorem 5.4.3\(a\)](#), we will need the following Lemma.

Lemma 5.A.3. Consider a set of variables $\mathbf{Y} = \{Y_1, \dots, Y_n\}$, and another (set of) variable(s) X . Every joint distribution $\mu(X, \mathbf{Y})$ over the values of X and \mathbf{Y} satisfies

$$\sum_{i=1}^n I_\mu(X; Y_i) \leq I_\mu(X; \mathbf{Y}) + \sum_{i=1}^n H_\mu(Y_i) - H_\mu(\mathbf{Y}).$$

Proof. Since there is only one joint distribution in scope, we omit the subscript μ , writing $I(-)$ instead of $I_\mu(-)$ and $H(-)$ instead of $H_\mu(-)$, in the body of this proof. The following fact will also be very useful:

$$I(A; B, C) = I(A; C) + I(A; B \mid C) \quad (\text{the chain rule for mutual information}). \tag{5.5}$$

We prove this by induction on n . In the base case ($n = 1$), we must show that $I(X; Y) \leq I(X; Y) + H(Y) - H(Y)$, which is an obvious tautology. Now, suppose inductively that

$$\sum_{i=1}^k I(X; Y_i) \leq I(X; \mathbf{Y}_{1:k}) + \sum_{i=1}^k H(Y_i) - H(\mathbf{Y}_{1:k}) \quad (\text{IH}_k)$$

for some $k < n$, where $\mathbf{Y}_{1:k} = (Y_1, \dots, Y_k)$. We now prove that the analogue for $k + 1$ also holds. Some calculation reveals:

$$\begin{aligned} I(X; Y_{k+1}) &= I(X; \mathbf{Y}_{1:k+1}) - I(X; \mathbf{Y}_{1:k} | Y_{k+1}) && [\text{by MI chain rule (5.5)}] \\ &\leq I(X; \mathbf{Y}_{1:k+1}) && [\text{since } I(X; \mathbf{Y}_{1:k} | Y_{k+1}) \geq 0] \\ &= I(X; Y_{k+1} | \mathbf{Y}_{1:k}) + I(\mathbf{Y}_{1:k}; Y_{k+1}) && [\text{by MI chain rule (5.5)}] \\ &= \begin{pmatrix} I(X; \mathbf{Y}_{1:k+1}) + H(Y_{k+1}) - H(\mathbf{Y}_{1:k+1}) \\ -I(X; \mathbf{Y}_{1:k}) & +H(\mathbf{Y}_{1:k}) \end{pmatrix} && \left[\begin{array}{l} \text{left: one more MI chain rule (5.5);} \\ \text{right: defn of mutual information} \end{array} \right] \end{aligned}$$

Observe: adding this inequality to our inductive hypothesis (IH $_k$) yields (IH $_{k+1}$)! So, by induction, the lemma holds for all k . \square

Theorem 5.4.1. *If $\mu \models \diamond \mathcal{A}$, then $IDef_{\mathcal{A}}(\mu) \leq 0$.*

Proof. Suppose that $\mu \models \mathcal{A}$, meaning that there is a witness $\nu(\mathcal{X}, \mathcal{U})$ that extends μ , and has properties (a-c) of Definition 5.2.2. For each hyperarc a , since $\nu \models (S_a, U_a) \rightarrow\!\!\!\rightarrow T_a$, we have $H_{\nu}(T_a | S_a, U_a) = 0$, and so

$$H_{\mu}(T_a | S_a) = H_{\nu}(T_a | S_a, U_a) + I_{\nu}(T_a; U_a | S_a) = I_{\nu}(T_a; U_a | S_a).$$

Thus, we compute

$$\begin{aligned}
\sum_{a \in \mathcal{A}} H_\mu(T_a | S_a) &= \sum_{a \in \mathcal{A}} I_\nu(U_a; T_a | S_a) \\
&= \sum_{a \in \mathcal{A}} I_\nu(U_a; T_a, S_a) - I_\nu(U_a; S_a) && \text{by MI chain rule (5.5)} \\
&\leq \sum_{a \in \mathcal{A}} I_\nu(U_a; T_a, S_a) && \text{since } I_\nu(U_a; S_a) \geq 0 \\
&\leq \sum_{a \in \mathcal{A}} I_\nu(U_a; \mathcal{X}) && \text{since } \mathcal{X} \twoheadrightarrow (S_a, T_a) \\
&\leq I_\nu(\mathcal{X}; \mathcal{U}) + \sum_{a \in \mathcal{A}} H_\nu(U_a) - H_\nu(\mathcal{U}) && \text{by Lemma 5.A.3} \\
&= I_\nu(\mathcal{X}; \mathcal{U}) && \text{since } \mathcal{U} \text{ are independent} \\
&&& (\text{per condition (b) of Definition 5.2.2}) \\
&\leq H_\nu(\mathcal{X}) = H_\mu(\mathcal{X}). && (\text{per condition (a) of Definition 5.2.2})
\end{aligned}$$

Thus, $IDef_{\mathcal{A}}(\mu) \leq 0$. □

Proposition 5.4.2. $QIMInc_{\mathcal{A}}(\mu) \geq 0$, with equality iff $\mu \models \mathcal{A}$.

Proof. The first term in the definition of $QIMInc$ be written as

$$\left(-H_\nu(\mathcal{U}) + \sum_{a \in \mathcal{A}} H_\nu(U_a) \right) = \mathbb{E}_\nu \left[\log \frac{\nu(\mathcal{U})}{\prod_a \nu(U_a)} \right]$$

and is therefore the relative entropy between $\nu(\mathcal{U})$ and the independent product distribution $\prod_{a \in \mathcal{A}} \nu(U_a)$. Thus, it is non-negative. The remaining terms of $QIMInc_{\mathcal{A}}(\mu)$, are all conditional entropies, and hence non-negative as well. Thus $QIMInc_{\mathcal{A}}(\mu) \geq 0$.

Now, suppose μ is s2-comaptible with \mathcal{A} , i.e., there exists some $\nu(\mathcal{U}, \mathcal{X})$ such that (a) $\nu(\mathcal{X}) = \mu(\mathcal{X})$, (b) $H_\nu(T_a | S_a, U_a) = 0$, and (d) $\{U_a\}_{a \in \mathcal{A}}$ are mutually independent. Then clearly ν satisfies the condition under the infemum, every

$H_\nu(T_a|S_a, U_a)$ is zero. It is also immediate that the final term is zero as well, because it equals $D(\nu(\mathcal{U}) \parallel \prod_a \nu(U_a))$, and $\nu(\mathcal{U}) = \prod_a \nu(U_a)$, per the definition of mutual independence. Thus, ν witnesses that $\text{QIMInc}_{(\mathcal{A}, \lambda)} = 0$.

Conversely, suppose $\text{QIMInc}_{(\mathcal{A}, \lambda)} = 0$. Because the feasible set is closed and bounded, as is the function, the infimum is achieved by some joint distribution $\nu(\mathcal{X}, \mathcal{A})$ with marginal $\mu(\mathcal{X})$. In this distribution ν , we know that every $H_\nu(T_a|S_a, U_a) = 0$ and $D(\nu(\mathcal{U}) \parallel \prod_a \nu(U_a)) = 0$ —because if any of these terms were positive, then the result would be positive as well. So ν satisfies (a) and (b) by definition. And, because relative entropy is zero iff its arguments are identical we have $\nu(\mathcal{U}) = \prod_a \nu(U_a)$, so the U_a 's are mutually independent, and ν satisfies (d) as well. \square

Theorem 5.4.3.

(a) If $(\mathcal{X}, \mathcal{A})$ is a hypergraph, $\mu(\mathcal{X})$ is a distribution, and $\nu(\mathcal{X}, \mathcal{U})$ is an extension of ν to additional variables $\mathcal{U} = \{U_a\}_{a \in \mathcal{A}}$ indexed by \mathcal{A} , then:

$$IDef_{\mathcal{A}}(\mu) \leq \text{QIMInc}_{\mathcal{A}}(\mu) \leq IDef_{\mathcal{A}^\dagger}(\nu).$$

(b) For all μ and \mathcal{A} , there is a choice of ν that achieves the upper bound. That is,

$$\text{QIMInc}_{\mathcal{A}}(\mu) = \min \left\{ IDef_{\mathcal{A}^\dagger}(\nu) : \begin{array}{l} \nu \in \Delta^{\mathcal{V}}(\mathcal{X}, \mathcal{U}) \\ \nu(\mathcal{X}) = \mu(\mathcal{X}) \end{array} \right\}.$$

where the minimization is over all possible ways of assigning values to the variables in \mathcal{U} . The minimum is achieved when $|\mathcal{V}(U_a)| \leq |\mathcal{V}(T_a)|^{|\mathcal{V}(S_a)|}$.

Proof. Part (a). The left hand side of the theorem ($IDef_{\mathcal{A}}(\nu) \leq \text{QIMInc}_{\mathcal{A}}(\mu)$) is a strengthening of the argument used to prove [Theorem 5.4.1](#). Specifically, let ν^*

be a minimizer of the optimization problem defining QIMInc . We calculate

$$\begin{aligned}
& \text{QIMInc}_{\mathcal{A}}(\mu) - \text{IDef}_{\mathcal{A}}(\mu) \\
&= \left(\sum_{a \in \mathcal{A}} H_{\nu^*}(T_a | S_a, U_a) - H_{\nu^*}(\mathcal{U}) + \sum_{a \in \mathcal{A}} H_{\nu^*}(U_a) \right) - \left(\sum_{a \in \mathcal{A}} H_{\mu}(T_a | S_a) - H_{\mu}(\mathcal{X}) \right) \\
&= \sum_{a \in \mathcal{A}} \left(H_{\nu^*}(T_a | S_a, U_a) - H_{\nu^*}(T_a | S_a) \right) + H_{\mu}(\mathcal{X}) - H_{\nu^*}(\mathcal{U}) + \sum_{a \in \mathcal{A}} H_{\nu^*}(U_a) \\
&= - \sum_{a \in \mathcal{A}} I_{\nu^*}(T_a; U_a | S_a) + H_{\mu}(\mathcal{X}) - H_{\nu^*}(\mathcal{U}) + \sum_{a \in \mathcal{A}} H_{\nu^*}(U_a).
\end{aligned}$$

The argument given in the first five lines of the proof of [Theorem 5.4.1](#), gives us a particularly convenient bound for the first group of terms on the left:

$$\sum_{a \in \mathcal{A}} I_{\nu^*}(U_a; T_a | S_a) \leq I_{\nu^*}(\mathcal{X}; \mathcal{U}) + \sum_{a \in \mathcal{A}} H_{\nu^*}(U_a) - H_{\nu^*}(\mathcal{U}).$$

Substituting this into our previous expression, we have:

$$\begin{aligned}
& \text{QIMInc}_{\mathcal{A}}(\mu) - \text{IDef}_{\mathcal{A}}(\mu) \\
&\geq - \left(I_{\nu^*}(\mathcal{X}; \mathcal{U}) + \sum_{a \in \mathcal{A}} H_{\nu^*}(U_a) - H_{\nu^*}(\mathcal{U}) \right) + H_{\mu}(\mathcal{X}) - H_{\nu^*}(\mathcal{U}) + \sum_{a \in \mathcal{A}} H_{\nu^*}(U_a) \\
&= H_{\mu}(\mathcal{X}) - I_{\nu^*}(\mathcal{X}; \mathcal{U}) \\
&\geq 0.
\end{aligned}$$

The final inequality holds because of our assumption that the marginal $\nu^*(\mathcal{X})$ equals $\mu(\mathcal{X})$. Thus, $\text{QIMInc}_{\mathcal{A}}(\mu) \geq \text{IDef}_{\mathcal{A}}(\mu)$, as promised.

We now turn to the right hand inequality, and part (b) of the theorem. Recall that ν^* is defined to be a minimizer of the optimization problem defining QIMInc . For the right inequality ($\text{QIMInc}_{\mathcal{A}}(\mu) \leq \text{IDef}_{\mathcal{A}^\dagger}(\nu)$) of part (a), observe that

$$\begin{aligned}
\text{IDef}_{\mathcal{A}^\dagger}(\nu) &= -H_{\nu}(\mathcal{X}, \mathcal{U}) + \sum_{a \in \mathcal{A}} H_{\nu}(U_a) + \sum_{a \in \mathcal{A}} H_{\nu}(T_a | S_a, U_a) + H_{\nu}(\mathcal{X} | \mathcal{U}) \\
&= \left(-H_{\nu}(\mathcal{U}) + \sum_{a \in \mathcal{A}} H_{\nu}(U_a) \right) + \sum_{a \in \mathcal{A}} H_{\nu}(T_a | S_a, U_a)
\end{aligned}$$

$$\begin{aligned}
&\geq \left(-H_{\nu^*}(\mathcal{U}) + \sum_{a \in \mathcal{A}} H_{\nu^*}(U_a) \right) + \sum_{a \in \mathcal{A}} H_{\nu^*}(T_a | S_a, U_a) \\
&= QIMInc(\mu).
\end{aligned}$$

This proves the right hand side of the inequality of part (a). Moreover, because the one inequality holds with equality when $\nu = \nu^*$ is a minimizer of this quantity (subject to having marginal $\mu(\mathcal{X})$) we have shown part (b) as well.

□

5.B Monotonicity and Undirected Graphical Models

Monotonicity of PDG inconsistency [81] is a powerful reasoning principle. Many important inequalities (e.g., the data processing inequality, relationships between statistical distances, the evidence lower bound, ...) can be proved using only a simple inference rule: “more beliefs can only increase inconsistency”. In this section, we develop and apply an analogous principle for QIM-compatibility. But first, we start with something simple. The fact that (quantitative) PDG inconsistency is monotonic is a powerful reasoning principle that can be used to prove many important inequalities [81]. In this section, we develop a related principle for QIM-compatibility. One classical representation of knowledge is a list of formulas $[\phi_1, \phi_2, \dots, \phi_n]$ that one knows to be true. This representation has a nice property: learning an additional formula ϕ_{n+1} can only narrow the set of worlds one considers possible. The same is true of QIM-compatibility.

Proposition 5.B.1. *If $\mathcal{A} \subseteq \mathcal{A}'$ and $\mu \models \Diamond \mathcal{A}'$, then $\mu \models \Diamond \mathcal{A}$.*

Here is a direct but not very useful analogue: if $\mathcal{A} \subseteq \mathcal{A}'$ and $\mu \models \Diamond \mathcal{A}'$, conclude

$\mu \models \Diamond \mathcal{A}$. After all, if μ is consistent with a set of independent causal mechanisms, then surely it is consistent with a causal picture in which only a subset of those mechanisms are present and independent. There is a sense in which BNs and MRFs are also monotonic, but in the opposite direction: adding edges to a graph results in a weaker independence statement. We will soon see why.

Since we use *directed* hypergraphs, there is actually a finer notion of monotonicity at play. Inputs and outputs play opposite roles, and they are naturally monotonic in opposite directions. If there is an obvious way to regard an element of B as an element of B' (abbreviated $B \hookrightarrow B'$), and $A' \hookrightarrow A$, then a function $f : A \rightarrow B$ can be regarded as one of type $A' \rightarrow B'$. This is depicted to the right. The same principle applies in our setting. If X and Z are sets of variables and $X \subseteq Z$, then $\mathcal{V}(Z) \hookrightarrow \mathcal{V}(X)$, by restriction. It follows, for example, that any mechanism by which X determines (Y, Y') can be viewed as a mechanism by which (X, X') determines Y . The general phenomenon is captured by the following.

$$\begin{array}{ccc} A & \xrightarrow{f} & B \\ \uparrow & \swarrow & \downarrow \\ A' & \dashrightarrow & B' \end{array}$$

Definition 5.B.1. If $\mathcal{A} = \{S \xrightarrow{a} T\}_a$, $\mathcal{A}' = \{S' \xrightarrow{a'} T'\}_{a'}$, and there is an injective map $\iota : \mathcal{A}' \rightarrow \mathcal{A}$ such that $T'_a \subseteq T_{\iota(a)}$ and $S'_a \supseteq S_{\iota(a)}$ for all $a \in \mathcal{A}'$, then \mathcal{A}' is a *weakening* of \mathcal{A} (written $\mathcal{A} \rightsquigarrow \mathcal{A}'$). □

QIM-compatibility is monotonic with respect to weakening (\rightsquigarrow).

Theorem 5.B.2. If $\mathcal{A} \rightsquigarrow \mathcal{A}'$ and $\mu \models \Diamond \mathcal{A}$, then $\mu \models \Diamond \mathcal{A}'$.

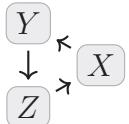
Theorem 5.B.2 is strictly stronger than Proposition 5.B.1 because a hyperarc with no targets is vacuous, so removing all targets of a hyperarc is equivalent to deleting it.

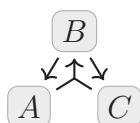
Theorem 5.B.2 explains why BNs and MRFs are arguably *anti*-monotonic: adding $X \rightarrow Y$ to a graph G means adding X to the *sources* the hyperarc whose target is Y , in \mathcal{A}_G .

As mentioned in the main body of the paper, the far more important consequence of this result is that it helps us begin to understand what QIM-compatibility means for cyclic hypergraphs. For the reader's convenience, we now restate the examples in the main text, which are really about monotonicity..

Example 8. Every $\mu(X, Y)$ is compatible with $[X] \not\rightarrow [Y]$. This is because this cycle is weaker than a hypergraph that can already represent any distribution, i.e., $\rightarrow [X] \rightarrow [Y] \rightsquigarrow [X] \not\rightarrow [Y]$. \triangle .

Example 9. What $\mu(X, Y, Z)$ are compatible with the 3-cycle shown, on the right? By monotonicity, among them must be all distributions consistent with a linear chain $\rightarrow X \rightarrow Y \rightarrow Z$. Thus, any distribution in which two variables are conditionally independent given the third is compatible with the 3-cycle. Are there any distributions that are *not* compatible with this hypergraph? It is not obvious. We return to this in Section 5.4. \triangle



 Because QIM-compatibility applies to cyclic structures, one might wonder if it also captures the independencies of undirected models.

Undirected edges $A-B$ are commonly identified with a (cyclic) pair of directed edges $\{A \rightarrow B, B \rightarrow A\}$, as we have implicitly done in defining \mathcal{A}_G . In this way, undirected graphs, too, naturally correspond to directed hypergraphs. For example, $G = A-B-C$ corresponds to the hypergraph \mathcal{A}_G shown on the left. Compatibility with \mathcal{A}_G , however, does not coincide with

any of the standard Markov properties corresponding to G [56]. This may appear to be a flaw in [Definition 5.2.2](#) (QIM-compatibility), but it is unavoidable. While both BNs and MRFs are monotonic, it is impossible to capture both classes with a monotonic definition.

Theorem 5.B.3. *It is possible to define a relation \models^\bullet between distributions μ and directed hypergraphs \mathcal{A} satisfying any two, but not all three, of the following.*

(monotonicity) *If $\mu \models^\bullet \mathcal{A}$ and $\mathcal{A} \rightsquigarrow \mathcal{A}'$, then $\mu \models^\bullet \mathcal{A}'$.*

(positive BN capture) *If μ satisfies the independencies $\mathcal{I}(G)$ of a dag G , then $\mu \models^\bullet \mathcal{A}_G$.*

(negative MRF capture) *If $\mu \models^\bullet \mathcal{A}_G$ for an undirected directed graph G , then μ has one of the Markov properties with respect to G .*

The proof is a direct and easy-to-visualize application of monotonicity ([Theorem 5.B.2](#)). Assume monotonicity and positive BN capture. Let $\mu_{xor}(A, B, C)$ be the joint distribution in which A and C are independent fair coins, and $B = A \oplus C$ is their parity. We then have:

$$\mu_{xor} \models \begin{array}{c} B \\ \downarrow \quad \nwarrow \quad \downarrow \\ A \quad C \end{array} \rightsquigarrow \begin{array}{c} B \\ \nwarrow \uparrow \searrow \\ A \quad C \end{array} = \mathcal{A}_{A-B-C}. \quad \text{But } \mu_{xor} \not\models A \perp\!\!\!\perp C \mid B. \quad \square$$

We emphasize that [Theorem 5.B.3](#) has implications for the qualitative semantics of *any* graphical model (even if one were to reject the definition QIM-compatibility). We now look into the implications for some lesser-known graphical models, which may appear not to comply with [Theorem 5.B.3](#).

Dependecny Networks To readers familiar with *dependency networks* (DNs) [41], [Theorem 5.B.3](#) may raise some conceptual issues. When G is an undirected graph, \mathcal{A}_G is the structure of a consistent DN. The semantics of such a DN, which intuitively describe an independent mechanism on each hyperarc, coincide with

the MRFs for G (at least for positive distributions). In more detail, DN semantics are given by the fixed point of a markov chain that repeatedly generates independent samples along the hyperarcs of \mathcal{A}_G for some (typically cyclic) directed graph G . The precise definition requires an order in which to do sampling. Although this choice doesn't matter for the “consistent DNs” that represent MRFs, it does in general. With a fixed sampling order, the DN is monotonic and captures MRFs, but can represent only BNs for which that order is a topological sort.

Theorem 5.B.3 shows that QIM-compatibility does not capture MRFs (at least, in the obvious way) at a purely observational level. Nevertheless, there is still a sense in which QIM-compatibility captures MRFs *causally*—that is, if we *intervene* instead of conditioning.

Proposition 5.B.4. *Let G be an undirected graph whose vertices correspond to variables \mathcal{X} .*

[link to
proof]

1. *Let $\mu(\mathcal{X})$ be a positive distribution (i.e., $\forall \mathbf{x} \in \mathcal{V}(\mathcal{X})$. $\mu(\mathcal{X}=\mathbf{x}) > 0$). If $\mu \models \mathcal{A}_G$, then for every witness $\bar{\mu}$ and causal model $\mathcal{M} \in \text{PSEM}_{\mathcal{A}_G}(\bar{\mu})$, whenever $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \subseteq \mathcal{X}$ are such that $\mathbf{X} \perp\!\!\!\perp_G \mathbf{Y} \mid \mathbf{Z}$, it is the case that $\bar{\mu} \models \mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \text{do}_{\mathcal{M}}(\mathbf{Z} = \mathbf{z})$.*
2. *Conversely, there exists some distribution $\mu(\mathcal{X})$ If $\mu \not\models \mathcal{A}_G$, then*

5.C Information Theory, PDGs, and QIM-Compatibility

5.C.1 More Detailed Primer on Information Theory

We now expand on the fundamental information quantities introduced at the beginning of [Section 5.4](#). Let μ be a probability distribution, and let X, Y, Z be (sets of) discrete random variables. The *entropy* of X is the uncertainty in X , when it is distributed according to μ , as measured by the number of bits of information needed (in expectation) needed to determine it, if the distribution μ is known. It is given by

$$H_\mu(X) := \sum_{x \in \mathcal{V}(X)} \mu(X=x) \log \frac{1}{\mu(X=x)} = -\mathbb{E}_\mu[\log \mu(X)],$$

and a few very important properties; chief among them, $H_\mu(X)$ is non-negative, and equal to zero iff X is a constant according to μ . The “joint entropy” $H(X, Y)$ is just the entropy of the combined variable (X, Y) whose values are pairs (x, y) for $x \in \mathcal{V}(X), y \in \mathcal{V}(Y)$; this is the same as the entropy of the variable $X \cup Y$ when X and Y are themselves sets of variables.

The *conditional entropy* of Y given X measures the uncertainty present in Y if one knows the value of X (think: the information in Y but not X), and is equivalently defined as any of the following three quantities:

$$H_\mu(Y|X) := \mathbb{E}_\mu[\log^{1/\mu(Y|X)}] = H_\mu(X, Y) - H_\mu(X) = \mathbb{E}_{x \sim \mu(X)} [H_{\mu|X=x}(Y)].$$

The *mutual information* $I(X; Y)$, and its conditional variant $I(X; Y|Z)$, are given, respectively, by

$$I_\mu(X; Y) := \mathbb{E}_\mu \left[\log \frac{\mu(X, Y)}{\mu(X)\mu(Y)} \right], \quad \text{and} \quad I(X; Y|Z) := \mathbb{E}_\mu \left[\log \frac{\mu(X, Y, Z)\mu(Z)}{\mu(X, Z)\mu(Y, Z)} \right].$$

The former is non-negative and equal to zero iff $\mu \models X \perp\!\!\!\perp Y$, and the latter is non-negative and equal to zero iff $\mu \models X \perp\!\!\!\perp Y | Z$. All of these quantities are purely “structural” or “qualitative” in the sense that they are invariant to relabelings of values, and

Just as conditional entropy can be written as a linear combination of unconditional entropies, so too can conditional mutual information be written as a linear combination of unconditional mutual informations: $I(X; Y|Z) = I(X; (Y, Z)) - I(X; Z)$. Thus conditional quantities are easily derived from the unconditional ones. But at the same time, the unconditional versions are clearly special cases of the conditional ones; for example, $H_\mu(X)$ is clearly the special case of $H(X|Z)$ when Z is a constant (e.g., $Z = \emptyset$). Furthermore, entropy and mutual information are also interdefinable and generated by linear combinations of one another. It is easy to verify that $I_\mu(X; Y) = H_\mu(X) + H_\mu(Y) - H(X, Y)$ and $I_\mu(X; Y|Z) = H_\mu(X|Z) + H_\mu(Y|Z) - H(X, Y|Z)$, and thus mutual information is derived from entropy. Yet on the other hand, $I_\mu(Y; Y) = H_\mu(Y)$ and $I_\mu(Y; Y|X) = H_\mu(Y|X)$ —thus entropy is a special case of mutual information.

5.C.2 Structural Deficiency: More Motivation, and Examples

To build intuition for $IDef$, which characterizes our bounds in [Section 5.4](#), we now visualize the vector \mathbf{v}_A for various example hypergraphs.

- Subfigures [5.C.1a](#), [5.C.1b](#), and [5.C.1c](#) show how adding hyperarcs makes distributions more deterministic. When A is the empty hypergraph, $IDef$ reduces to negative entropy, and so prefers distributions that are “maximally uncertain” (e.g., Subfigures [5.C.1a](#) and [5.C.1d](#)). For this empty but all

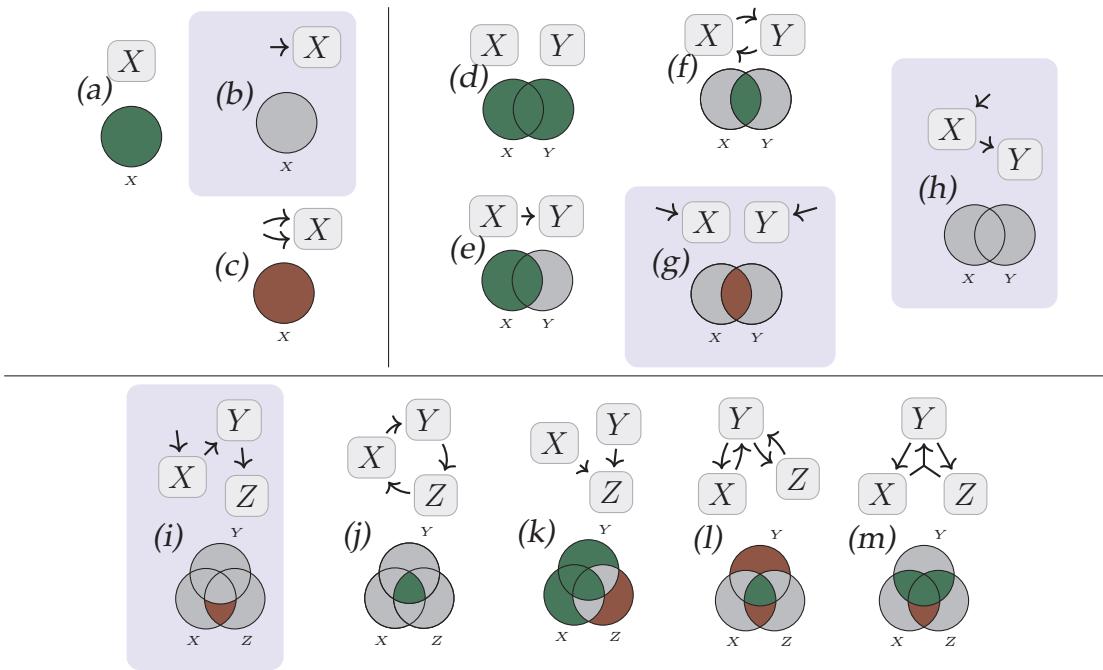


Figure 5.C.1: Illustrations of the structural deficiency $IDef_{\mathcal{A}}$ underneath drawn underneath their associated hypergraphs $\{G_i\}$. Each circle represents a variable; an area in the intersection of circles $\{C_j\}$ but outside of circles $\{D_k\}$ corresponds to information that is shared between all C_j 's, but not in any D_k . Variation of a candidate distribution μ in a green area makes its qualitative fit better (according to $IDef$), while variation in a red area makes its qualitative fit worse; grey is neutral. Only the boxed structures in blue, whose $IDef$ can be seen as measuring distance to a particular set of (conditional) independencies, are expressible as BNs.

distributions μ have negative $IDef_{\mathcal{A}}(\mu) \leq 0$. In the definition of $IDef$, each hyperarc $X \rightarrow Y$ is compiled to a “cost” $H(Y|X)$ for uncertainty in Y given X . One can see this visually in Figure 5.C.1 as a red crescent that’s added to the information profile as we move from 5.C.1d to 5.C.1e to 5.C.1f.

- Some hypergraphs (see Figures 5.C.1b and 5.C.1h) are *indiscriminate*, in the sense that every distribution gets the same score (of zero, because a point mass δ always has $SDef_{\mathcal{A}}(\delta) = 0$). Such a graph has a structure such that *any* distribution can be precisely encoded by the process in (b). As shown here and also in Richardson and Halpern [82], $IDef$ can also indicate independencies and conditional independencies, illustrated respectively in

Subfigures 5.C.1g and 5.C.1i.

- For more complex structures, structural information deficiency $IDef$ can represent more than independence and dependence. The cyclic structures in Examples 8 and 9, correspond to the structural deficiencies pictured in Subfigures 5.C.1f and 5.C.1j, respectively, which are functions that encourage shared information between the three variables.

5.C.3 Weights for SIM-Inc

Given \mathcal{A} and $|\mathcal{A}|+2$ positive weights $\lambda = (\lambda^{(a)}, \lambda^{(b)}, \{\lambda_a^{(c)}\}_{a \in \mathcal{A}})$, define the function

$$QIMInc_{\mathcal{A}, \lambda}(\mu) := \inf_{\nu(\mathcal{U}, \mathcal{X})} \left\{ \begin{array}{l} \lambda^{(a)} D(\nu(\mathcal{X}) \parallel \mu(\mathcal{X})) \\ + \lambda^{(b)} \left(-H_\nu(\mathcal{U}) + \sum_{a \in \mathcal{A}} H_\nu(U_a) \right) \\ + \sum_{a \in \mathcal{A}} \lambda_a^{(c)} H_\nu(T_a | S_a, U_a) \end{array} \right. . \quad (5.6)$$

When $\lambda = (\infty, 1, 1)$, we get the analogous quantity defined in (5.3) in the main text.

Here are some analogous results for this generalized version with weights. For a weighted hypergraph (\mathcal{A}, α) , here is a strengthening of Theorem 5.4.3, and the appropriate translation of the hypergraph. Given (\mathcal{A}, α) translate it to a new derandomized hypergraph $(\mathcal{A}, \alpha)^\dagger$ by replacing each weighted hyperarc

$$S_a \xrightarrow[\alpha_a]{a} T_a \quad \text{with the pair of weighted hyperarcs} \quad \begin{array}{c} a_0 \\ \xrightarrow[\alpha_a]{} \\ a_1 \end{array} \quad \begin{array}{c} U_a \\ \xrightarrow[\alpha_a]{} \\ T_a \end{array} .$$

⟨ INCOMPLETE ⟩

5.C.4 Counter-Examples to the Converse of Theorem 5.4.1

In light of [Example 11](#) and its connections to $SDef$ through [Theorem 5.4.1](#), one might hope this criterion is not just a bound, but a precise characterization of the distributions that are QIM-compatible with the 3-cycle. Unfortunately, it does not, and the converse of [Theorem 5.4.1](#) is false.

Example 12. Suppose $\mu(X, Y, Z) = \text{Unif}(X, Z)\delta\text{id}(Y|X)$ and $\mathcal{A} = \{\rightarrow X, \rightarrow Y\}$, where all variables are binary. Then $SDef_{\mathcal{A}}(\mu) = 0$, but X and Y are not independent. \triangle

Here is another counter-example, of a very different kind.

Example 13. Suppose A, B, C are binary variables. It can be shown by enumeration (see appendix) that no distribution supported on seven of the eight possible joint settings of $\mathcal{V}(A, B, C)$ can be QIM-compatible with the 3-cycle \mathcal{A}_{3o} . Yet it is easy to find examples of such distributions μ that have positive interaction information $I(A; B; C)$, and thus $SDef_{\mu}(\mathcal{A}_{3o}) \leq 0$ for such distributions. \triangle

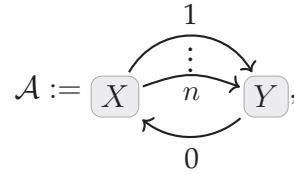
5.D QIM-Compatibility Constructions and Counterexamples

We now give a counterexample to a simpler previously conjectured strengthening of [Theorem 5.2.2](#), in which part (a) is an if-and-only-if. This may be surprising. In the unconditional case, it is true that, two arcs $\{\overset{1}{\rightarrow} X, \overset{2}{\rightarrow} X\}$ precisely encode that X is a constant, as illustrated by [Example 7](#). The following, slightly more general result, is an immediate corollary of [Theorem 5.2.2\(c\)](#).

Proposition 5.D.1. $\mu \models \mathcal{A} \sqcup \{\xrightarrow{1} X, \xrightarrow{2} X\}$ if and only if $\mu \models \mathcal{A}$ and $\mu \models \emptyset \twoheadrightarrow X$.

One can be forgiven for imagining that the conditional case would be analogous—that QIM-compatibility with a hypergraph that has two parallel arcs from X to Y would imply that Y is a function of X . But this is not the case. Furthermore, our counterexample also shows that neither of the two properties we consider in the main text (requiring that \mathcal{A} is partitional, or that the QIM-compatibility with μ is even) are enough to ensure this. That is, there are partitional graphs \mathcal{A} such that $\mu \models^e \mathcal{A}$ but $\mu \not\models \mathcal{A} \sqcup \{X \xrightarrow{1} Y, X \xrightarrow{2} Y\}$.

Example 14. We will construct a witness of SIM-compatibility for the hypergraph



in which Y is *not* a function of X , which for $n = 3$ will disprove the analogue of Theorem 5.2.2 for the partitional context \mathcal{A}' equal to the 2-cycle.

Let $\mathcal{U} = (U_0, U_1, \dots, U_n)$ be a vector of n mutually independent random coins, and A is one more independent random coin. For notational convenience, define the random vector $\mathbf{U} := (U_0, \dots, U_n)$ consisting of all variables U_i except for U_0 . Then, define variables X and Y according to:

$$X := (A \oplus U_1, \dots, A \oplus U_n, U_0 \oplus U_1, U_0 \oplus U_2, \dots, U_0 \oplus U_n)$$

$$= (A \oplus \mathbf{U}, U_0 \oplus \mathbf{U})$$

$$Y := (A, U_0 \oplus \mathbf{U}) = (A, U_0 \oplus U_1, U_0 \oplus U_2, \dots, U_0 \oplus U_n),$$

where and the operation $Z \oplus \mathbf{V}$ is element-wise xor (or addition in \mathbb{F}_2^n), after implicitly converting the scalar Z to a vector by taking n copies of it. Call the resulting distribution $\nu(X, Y, \mathcal{U})$.

If we now show that ν witnesses that its marginal on X, Y is QIM-compatible with \mathcal{A} , which is straightforward.

(b) \mathcal{U} are mutually independent by assumption;

(c.0) $Y = (A, \mathbf{B})$ and U_0 determine X according to:

$$\begin{aligned} g(A, \mathbf{B}, U_0) &= (A \oplus U_0 \oplus \mathbf{B}, \mathbf{B}) \\ &= (A \oplus U_0 \oplus U_0 \oplus \mathbf{U}, U_0 \oplus \mathbf{U}) \quad \text{since } \mathbf{B} = U_0 \oplus \mathbf{U} \\ &= (A \oplus \mathbf{U}, U_0 \oplus \mathbf{U}) = X \end{aligned}$$

(c.1-n) for $i \in \{1, \dots, n\}$, U_i and $X = (\mathbf{V}, \mathbf{B})$ together determine Y according to

$$f_i(\mathbf{V}, \mathbf{B}, U_i) := (V_i \oplus U_i, \mathbf{B}) = (A \oplus U_i \oplus U_i, U_0 \oplus \mathbf{U}) = Y.$$

In addition, this distribution $\nu(\mathcal{U}, X, Y)$ satisfies condition

(d) $\nu(X, Y \mid \mathcal{U}) = \frac{1}{2}\mathbb{1}[g(Y, U_0) = X] \prod_{i=1}^n \mathbb{1}[f_i(X, U_i) = Y]$, since, for all joint settings of \mathcal{U} , there are two possible values of (X, Y) , corresponding to the two values of A , and both happen with probability $\frac{1}{2}$.

Thus, we have constructed a distribution that witnessing the fact that $\mu(X, Y) \models^e \mathcal{A}$.

Yet, observe that X alone does not determine Y in this distribution, because X alone is not enough to determine A (without also knowing some U_i).

For those who are interested, observe that the bound of [Theorem 5.4.1](#) tells that we must satisfy

$$\begin{aligned} 0 &\geq IDef_{\mathcal{A}}(\mu) = -H_{\mu}(X, Y) + nH_{\mu}(Y \mid X) + H_{\mu}(X \mid Y) \\ &= -I_{\mu}(X; Y) + (n - 1)H_{\mu}(Y \mid X) \end{aligned}$$

Indeed, this distribution has information profile

$$H(X | Y) = 1 \text{ bit}, \quad I(X; Y) = n \text{ bits}, \quad H(Y | X) = 1 \text{ bit},$$

and so $IDef_{\mathcal{A}}(\mu) = -1$ bit. Intuitively, this one missing bit corresponds to the value of A that is not determined by the structure of \mathcal{A} . \triangle

5.E From Causal Models to Witnesses

We now return to the “easy” direction of the correspondence between QIM-compatibility witnesses and causal models, mentioned at the beginning of [Section 5.3.2](#). Given a (generalized) randomized PSEM \mathcal{M} , we now show that distributions $\nu \in \{\mathcal{M}\}$, are QIM-compatibility witness showing that the marginals of ν are QIM-compatible with the hypergraph $\mathcal{A}_{\mathcal{M}}$. More formally:

Proposition 5.E.1. *If $\mathcal{M} = (M=(\mathcal{U}, \mathcal{V}, \mathcal{F}), P)$ is a randomized PSEM, then every $\nu \in \{\mathcal{M}\}$ witnesses the QIM-compatibility of its marginal on its exogenous variables, with the dependency structure of \mathcal{M} . That is, for all $\nu \in \{\mathcal{M}\}$ and $\mathcal{Y} \subseteq \mathcal{U} \cup \mathcal{V}$, $\nu(\mathcal{Y}) \models \Diamond \mathcal{A}_{\mathcal{M}}$.*

The proof is straightforward: by definition, if $\nu \in \{\mathcal{M}\}$, then it must satisfy the equations, and so automatically fulfills condition (c). Condition (a) is also satisfied trivially, by assumption: the distribution we’re considering is defined to be a marginal of ν . Finally, (b) is also satisfied by construction: we assumed that $\mathcal{U}_{\mathcal{A}} = \{U_a\}_{a \in \mathcal{A}}$ are independent.

5.F An Algorithm for Finding Witnesses: The Null Value Construction

We have now built up a body of examples, but it is still not clear how to compute QIM-compatibility. In other words, it is still not clear how to solve the decision problem: given μ and \mathcal{A} , determine whether or not $\mu \models \mathcal{A}$. In this section, we discuss one approach to this problem.

If you start with a distribution $\nu(\mathcal{X})$, it's not at all obvious how to extend it with a conditional distribution $\nu(\mathcal{U}|\mathcal{X})$ such that the variables \mathcal{U} are *unconditionally* dependent, given that they cannot be independent of \mathcal{X} . It seems that the only way to ensure this unconditional independence is to start with a distribution $\nu(\mathcal{U}) = \prod_a \nu(U_a)$ and then figure out how to extend it to the variables \mathcal{X} .

To begin, for each $a \in \mathcal{A}$, take U_a to be a response variable, taking values $\mathcal{V}(U_a) = (\mathcal{V}T_a)^{(\mathcal{V}S_a)}$, just as in Section 5.3. But now we run into a problem: without carefully selecting the supports of the distributions over \mathcal{U} , it is entirely possible that there will be some joint setting $\mathbf{u} \in \mathcal{V}\mathcal{U}$ occurs with positive probability, but represents a collection of functions that has no fixed point. For example, take the graph

Example 15 (5, continued). By randomly selecting distributions $\Pr(U_1)$, $\Pr(U_2)$, and $\Pr(U_3)$ (see Section 5.F), one finds that the set of distributions that are consistent with this 3-cycle has larger dimension than the set of distributions that factorize according to $\Pr(X, Y, Z) \propto \phi_1(X, Y)\phi_2(Y, Z)\phi_3(Z, X)$.⁸ Thus, our definition does not coincide with the corresponding factor graph. \triangle

⁸see appendix for details.

Conjecture 5.F.1. If $\mu_0 \models \mathcal{A}$, and μ' lies on the path $\mu(t)$ of gradient flow minimizing $S\text{Def}_{\mathcal{A}}(\mu')$, starting at $\mu(0) = \mu_0$, then $\mu' \models \mathcal{A}$.

The following has emperical support.

Conjecture 5.F.2. The distribution (s?) $\hat{\mu} := \arg \min_{\mu: \mu \models \mathcal{A}} D(\mu \parallel \hat{\mu})$ have the same information profile as μ .

5.G Even Structural Compatibility

5.G.1 Even QIM-Compatibility

If \mathcal{M} is a cyclic or subpartitional PSEM, then $\{\mathcal{M}\}$ may contain many distributions. Still, so long as it is non-empty, there is still a unique distribution that, arguably, best describes the distribution of the PSEM (in the absence of interventions)—namely, the one that, for any given value $\mathbf{u} \in \mathcal{V}(\mathcal{U})$, treats all “fixed-points” $\mathbf{x} \in \mathcal{F}(\mathbf{u})$ of the equations \mathcal{F} symmetrically.

$$\left(\text{Recall that } \mathcal{F}(\mathbf{u}) := \{\mathbf{x} \in \mathcal{V}(\mathcal{X}) : \forall a \in \mathcal{A}. f_a(S_a(\mathbf{x}), u_a) = S_a(\mathbf{x})\}. \right)$$

For example, if \mathcal{M} has no exogenous variables ($\mathcal{U} = \emptyset$), endogenous variables $\mathcal{X} = [X_1, \dots, X_n]$ that are all binary, and equations $f_{X_i}(\mathcal{X} \setminus X_i) = X_{(i+1)\%n}$, then $\{\mathcal{M}\}$ is a 1-dimensional specturm of distributions supported on the two points $\{(0, \dots, 0), (1, \dots, 1)\}$. The distribution that gives the two an equal weight of $\frac{1}{2}$ is somehow special, in that it is the unique one that does not break the symmetry by preferring either $(0, \dots, 0)$ or $(1, \dots, 1)$. This intuition is made precise, and generalized to QIM-compatibility witnesses (rather than PSEMs), by the following definition.

Definition 5.G.1. We say a witness $\nu(\mathcal{U}, \mathcal{X})$ to $\mu \models \mathcal{A}$ is *even*, iff it satisfies properties (a,b) of Definition 5.2.2, and also the following strengthening of property (c):

$$(d) \quad \nu(\mathcal{X} \mid \mathcal{U}) \propto \mathbb{1} \left[\bigwedge_{a \in \mathcal{A}} f_a(S_a, U_a) = T_a \right],$$

for some set $\mathcal{F} = \{f_a : \mathcal{V}(S_a, U_a) \rightarrow \mathcal{V}(T_a)\}_{a \in \mathcal{A}}$ of equations. In this case, we say μ is *evenly QIM-compatible* (EQIM-compatible) with \mathcal{A} , write $\mu \models^e \mathcal{A}$, and call the pair (ν, \mathcal{F}) a witness of EQIM-compatibility. \square

EQIM-compatibility clearly implies QIM-compatibility, and thus is a stricter notion. Furthermore, EQIM-compatibility witnesses have an even sharper relationship to causal models: A witness $(\bar{\mu}(\mathcal{U}, \mathcal{X}), \mathcal{F})$ to EQIM-compatibility, can be equivalently viewed as a PSEM $\mathcal{M} = (\mathcal{U}, \mathcal{X}, \mathcal{F}, \nu(\mathcal{U}))$, because the rest of the distribution $\nu(\mathcal{X} \mid \mathcal{U})$ is determined by \mathcal{F} and property (d).

Proposition 5.G.1. • *There is a 1-1 correspondence between EQIM-compatibility witnesses and GRPSEMs \mathcal{M} in which $\{\mathcal{M}\} \neq \emptyset$.*

Proof. 1. Given a EQIM-compatibility witness $(\nu(\mathcal{X}, \mathcal{U}_A), \mathcal{F})$,

by Proposition 5.3.2 PSEMs $_{\mathcal{A}}(\nu)$ \square

Thus, PSEMs are in direct 1-1 correspondence with EQIM-compatibility witnesses when hypergraphs $\mathcal{A} = \mathcal{A}_G$ for some graph G .

We now verify that various results from the main text extend to EQIM-compatibility.

[Theorem 5.2.1] When G is acyclic (and, more generally, when $\mathcal{M} \models \mathcal{U} \twoheadrightarrow \mathcal{X}$), the extra

condition (d) holds trivially, and so EQIM-compatibility coincides with QIM-compatibility, and the analogue of Theorem 5.2.1 in which \models is replaced with \models^e also holds.

⋮

[Theorem 5.B.2] As we show in the proof of Theorem 5.B.2, EQIM-compatibility is also monotonic with respect to weakening (\rightsquigarrow).

The original scoring function $IDef$ is related to even QIM-compatibility.

Conjecture 5.G.2. *There is a constant $\kappa = \kappa(\mathcal{A}, \mathcal{V})$ depending on the hypergraph \mathcal{A} and the possible values \mathcal{V} that the variables can take, such that*

$$IDef_{\mathcal{A}}(\mu) \leq \kappa \iff \mu \models^e \mathcal{A}$$

Proof. (\Leftarrow). Suppose $\mu \models^e \mathcal{A}$. Then there is some witness $\bar{\mu}$ extending μ to independent variables $\mathcal{U}_{\mathcal{A}} = \{U_a\}_{a \in \mathcal{A}}$.

(\Rightarrow). Suppose that $IDef_{\mathcal{A}}(\mu) \leq \kappa$. \square

5.G.2 ESIM Compatibility Scoring Rules

We have now seen that, $IDef_{\mathcal{A}^\dagger}$ measures distance from being a witness to QIM-compatibility (Theorem 5.4.3(b)). Modulo a constant offset or limit, $IDef_{\mathcal{A}}$, i.e., the original scoring function applied to the original hypergraph

Let's now repeat the same approach as the previous section, by explicitly constructing a scoring function for EQIM-compatibility. Extend our previous

weight vector by one entry, so that $\lambda = (\lambda^{(a)}, \lambda^{(b)}, \{\lambda_a^{(c)}\}_{a \in \mathcal{A}}, \lambda^{(d)})$.

$$\begin{aligned} \text{EQIMInc}_{\mathcal{A}, \lambda}(\mu) &:= \inf_{\nu(\mathcal{X}, \mathcal{U})} \blacksquare \\ &\quad + \lambda^{(d)} \inf_{\mathcal{F}} \mathbb{E}_{\mathbf{u} \sim \nu(\mathcal{U})} \left[D(\nu(\mathcal{X} \mid \mathcal{U}=\mathbf{u}) \parallel \text{Unif}[\mathcal{F}(\mathbf{u})]) \right], \end{aligned}$$

where \blacksquare consists of everything but the infimum from Equation (5.3), \mathcal{F} ranges over sets of equations along \mathcal{A} (as in Definitions 5.3.1 and 5.G.1), and $\text{Unif}[\mathcal{F}(\mathbf{u})]$ is the uniform distribution over joint settings of \mathcal{X} that are consistent with the equations after fixing context $\mathcal{U} = \mathbf{u}$. Recall that the key step of constructing \mathcal{A}^\dagger was to add the hyperarc $\mathcal{U} \rightarrow \mathcal{X}$. But for even compatibility, we want to effectively do the opposite—that is, subject to satisfying the other constraints, we want to incentivize, rather than penalize the conditional entropy $H(\mathcal{X} \mid \mathcal{U})$. This is made precise by the following proposition.

Proposition 5.G.3. (a) For all $\lambda > 0$, $\text{EQIMInc}_{\mathcal{A}, \lambda}(\mu) \geq 0$ with equality iff $\mu \models^e \mathcal{A}$.

In other words, $IDef$ itself already measures distance from *even QIM-compatibility*, once we find the appropriate constant to make it non-negative. Although has an enormous benefit of not requiring an infimum.

⟨ TODO: There's an issue here I still need to
finish working out. ⟩

$$IDef_{\mathcal{A}}(\mu) \leq \text{QIMInc}_{\mathcal{A}}(\mu) \leq IDef_{\mathcal{A}^\dagger}(\nu^*) \leq IDef_{\mathcal{A}}(\mu) + \kappa().$$

$\kappa_{\mathcal{A}}$ is a (possibly infinite) piecewise constant function of μ with finitely many pieces, and finite when $\mu \models \mathcal{A}$.

5.G.3 Complete Derandomization for Cyclic Models

We have seen that a number of properties of causal models are simpler when $\mathcal{M} \models \mathcal{U} \twoheadrightarrow \mathcal{X}$. In some sense, the job of a causal model is model \mathcal{X} by adding variables \mathcal{U} that account for any uncertainty. When $\mathcal{M} \not\models \mathcal{U} \twoheadrightarrow \mathcal{X}$, this job is in some sense incomplete; cycles can create a new source of uncertainty. In this subsection, we explore the effects of adding one more variable U_0 to account for the remaining uncertainty, by explaining it as randomness. Technically speaking, this means looking into one way of converting a randomized PSEM \mathcal{M} to one in which $\mathcal{U} \twoheadrightarrow \mathcal{X}$.

Our construction is parameterized by a PSEM \mathcal{M} and a choice of $\nu \in \{\mathcal{M}\}$. In brief, we use a natural construction explained in the next subsection (5.A.1) to obtain a “maximally independent” derandomization of $\nu(\mathcal{X} \mid \mathcal{U})$. The result is a new generalized randomized PSEM, which we call $\mathcal{M}^{+\tilde{\nu}(\mathcal{X}^{\mathcal{U}})}$, differing from \mathcal{M} in that it has one extra hyperarc $\mathcal{U} \rightarrow \mathcal{X}$, and, correspondingly, an extra variable U_0 , and an extra equation $f_0 : \mathcal{U} \rightarrow \mathcal{X}$. This new causal model has two important properties:

1. Only ν can arise from $\mathcal{M}^{+\tilde{\nu}(\mathcal{X}^{\mathcal{U}})}$ (i.e., $\{\mathcal{M}^{+\tilde{\nu}(\mathcal{X}^{\mathcal{U}})}\} = \{\nu\}$), and
2. the exogenous variables determine the values endogenous ones (i.e., $\mathcal{M}^{+\tilde{\nu}(\mathcal{X}^{\mathcal{U}})} \models (\mathcal{U}, U_0) \twoheadrightarrow \mathcal{X}$).

Constructing the Causal Model $\mathcal{M}^{+\tilde{\nu}(\mathcal{X}^{\mathcal{U}})}$

We now apply the general technique above to obtain the causal model $\mathcal{M}^{+\tilde{\nu}(\mathcal{X}^{\mathcal{U}})}$ discussed in Section 5.G.3. Concretely, let $\mathcal{V}(U_0) := \prod_{u \in \mathcal{V}(\mathcal{U})} \mathcal{F}(u)$ consist of

all functions from $\mathcal{V}(\mathcal{U})$ to $\mathcal{V}(\mathcal{X})$ consistent with the equations \mathcal{F} , and add an equation $\mathcal{X} = U_0(\mathbf{u}) = f_0(U_0, \mathbf{u})$ along the hyperarc $\mathcal{U} \rightarrow \mathcal{X}$.

Given $\nu \in \{\mathcal{M}\}$, define $P(U_0=f) := \prod_{\mathbf{u} \in \mathcal{V}(\mathcal{U})} \nu(f(\mathbf{u}) \mid \mathbf{u})$. As shown more generally in ??,

1. this indeed a probability distribution, and
2. the joint distribution $P(\mathcal{U}, U_0) = P(\mathcal{U})P(U_0)$ extends uniquely along the function defined by U_0 , to a distribution $P(\mathcal{U}, U_0, \mathcal{X})$, and the marginal of that distribution on $(\mathcal{U}, \mathcal{X})$ equals ν .

We must also be careful about how to respect interventions. In the most general form, an intervention of the form $f_a \leftarrow g$, for some function $g : \mathcal{V}(S_a) \rightarrow \mathcal{V}(T_a)$, not only modifies the equation f_a by setting it equal to g , but also modifies the equation f_0 according to:

$$f_0(\mathbf{u}) := f_0(\mathbf{u})[]$$

⟨ INCOMPLETE ⟩

For instance, if \mathcal{M} is a PSEM, then to perform an intervention $\mathbf{X} \leftarrow \mathbf{x}$ on the causal model $\mathcal{M}^{+\tilde{\nu}(\mathcal{X}^{\mathcal{U}})}$, we mean not only to replace the equation for f_X , but also modify

$$f_0^{\text{new}}(\mathbf{u}) := f_0^{\text{old}}(\mathbf{u})[\mathbf{X} \mapsto \mathbf{x}]$$

⟨ INCOMPLETE ⟩

Proposition 5.G.4. 1. $\Pr_{\mathcal{M}^{+\tilde{\nu}(\mathcal{X}^{\mathcal{U}})}}(\varphi) = \Pr_{\mathcal{M}}(\varphi)$

Part II

A Universal Objective

CHAPTER 6

LOSS AS THE INCONSISTENCY OF A PDG: CHOOSE YOUR MODEL, NOT YOUR LOSS

RELATIVE ENTROPY SOUP

Oliver E. Richardson, Ph.D.

Cornell University 2024

In a world blessed with a great diversity of loss functions, we argue that that choice between them is not a matter of taste or pragmatics, but of model. Probabilistic dependency graphs (PDGs) are probabilistic models that come equipped with a measure of “inconsistency”. We prove that many standard loss functions arise as the inconsistency of a natural PDG describing the appropriate scenario, and use the same approach to justify a well-known connection between regularizers and priors. We also show that the PDG inconsistency captures a large class of statistical divergences, and detail benefits of thinking of them in this way, including an intuitive visual language for deriving inequalities between them. In variational inference, we find that the ELBO, a somewhat opaque objective for latent variable models, and variants of it arise for free out of uncontroversial modeling assumptions—as do simple graphical proofs of their corresponding bounds. Finally, we observe that inconsistency becomes the log partition function (free energy) in the setting where PDGs are factor graphs.

6.1 Introduction

Many tasks in artificial intelligence have been fruitfully cast as optimization problems, but often the choice of objective is not unique. For instance, a key component of a machine learning system is a loss function which the system must minimize, and a wide variety of losses are used in practice. Each implicitly represents different values and results in different behavior, so the choice between them can be quite important [94, 45]. Yet, because it's unclear how to choose a "good" loss function, the choice is usually made by empirics, tradition, and an instinctive calculus acquired through the practice—not by explicitly laying out beliefs. Furthermore, there is something to be gained by fiddling with these loss functions: one can add regularization terms, to (dis)incentivize (un)desirable behavior. But the process of tinkering with the objective until it works is often unsatisfying. It can be a tedious game without clear rules or meaning, while results so obtained are arguably overfitted and difficult to motivate.

By contrast, a choice of *model* admits more principled discussion, in part because models are testable; it makes sense to ask if a model is accurate. This observation motivates our proposal: instead of specifying a loss function directly, one articulates a situation that gives rise to it, in the (more interpretable) language of probabilistic beliefs and certainties. Concretely, we use the machinery of Probabilistic Dependency Graphs (PDGs), a particularly expressive class of graphical models that can incorporate arbitrary (even inconsistent) probabilistic information in a natural way, and comes equipped with a well-motivated measure of inconsistency [82].

A primary goal of this paper is to show that PDGs and their associated inconsistency measure can provide a "universal" model-based loss function. Towards

this end, we show that many standard objective functions—cross entropy, square error, many statistical distances, the ELBO, regularizers, and the log partition function—arise naturally by measuring the inconsistency of the appropriate underlying PDG. This is somewhat surprising, since PDGs were not designed with the goal of capturing loss functions at all. Specifying a loss function indirectly like this is in some ways more restrictive, but it is also more intuitive (it no technical familiarity with losses, for instance), and admits more grounded defense and criticism.

For a particularly powerful demonstration, consider the variational autoencoder (VAE), an enormously successful class of generative model that has enabled breakthroughs in image generation, semantic interpolation, and unsupervised feature learning [52]. Structurally, a VAE for a space X consists of a (smaller) latent space Z , a prior distribution $p(Z)$, a decoder $d(X|Z)$, and an encoder $e(Z|X)$. A VAE is not considered a “graphical model” for two reasons. The first is that the encoder $e(Z|X)$ has the same target variable as $p(Z)$, so something like a Bayesian Network cannot simultaneously incorporate them both (besides, they could be inconsistent with one another). The second reason: it is not a VAE’s structure, but rather its *loss function* that makes it tick. A VAE is typically trained by maximizing the “ELBO”, a somewhat difficult-to-motivate function of a sample x , originating in variational calculus. We show that $-\text{ELBO}(x)$ is also precisely the inconsistency of a PDG containing x and the probabilistic information of the autoencoder (p , d , and e). We can form such a PDG precisely because PDGs allow for inconsistency. Thus, PDG semantics simultaneously legitimize the strange structure of the VAE, and also justify its loss function, which can be thought of as a property of the model itself (its inconsistency), rather than some mysterious construction borrowed from physics.

Representing objectives as model inconsistencies, in addition to providing a principled way of selecting an objective, also has beneficial pedagogical side effects, because of the *structural* relationships between the underlying models. For instance, these relationships will allow us to derive simple and intuitive visual proofs of technical results, such as the variational inequalities that traditionally motivate the ELBO, and the monotonicity of Rényi divergence.

In the coming sections, we show in more detail how this concept of inconsistency, beyond simply providing a permissive and intuitive modeling framework, reduces exactly to many standard objectives used in machine learning and to measures of statistical distance. We demonstrate that this framework clarifies the relationships between them, by providing clear derivations of otherwise opaque inequalities.

6.2 Preliminaries

We generally use capital letters for variables, and lower case letters for their values. For variables X and Y , a conditional probability distribution (cpd) p on Y given X , written $p(Y|X)$, consists of a probability distribution on Y (denoted $p(Y|X = x)$ or $p(Y|x)$ for short), for each possible value x of X . If μ is a probability on outcomes that determine X and Y , then $\mu(X)$ denotes the marginal of μ on X , and $\mu(Y|X)$ denotes the conditional marginal of μ on Y given X . Depending on which we find clearer in context, we write either $\mathbb{E}_\mu f$ or $\mathbb{E}_{\omega \sim \mu} f(\omega)$ for expectation of $f : \Omega \rightarrow \mathbb{R}$ over a distribution μ with outcomes Ω . We write $D(\mu \| \nu) = \mathbb{E}_\mu \log \frac{\mu}{\nu}$ for the relative entropy (KL Divergence) of ν with respect to μ , we write $H(\mu) := \mathbb{E}_\mu \log \frac{1}{\mu}$ for the entropy of μ , $H_\mu(X) := H(\mu(X))$

for the marginal entropy on a variable X , and $H_\mu(Y \mid X) := \mathbb{E}_\mu \log 1/\mu(Y|X)$ for the conditional entropy of Y given X .

A *probabilistic dependency graph* (PDG) [82], like a Bayesian Network (BN), is a directed graph with cpds attached to it. While this data is attached to the *nodes* of a BN, it is attached to the *edges* of a PDG. For instance, a BN of shape $X \rightarrow Y \leftarrow Z$ contains a single cpd $\Pr(Y|X, Z)$ on Y given joint values of X and Z , while a PDG of the same shape contains two cpds $p(Y|X)$ and $q(Y|Z)$. The second approach is strictly more expressive, and can encode joint dependence with an extra variable. All information in a PDG can be expressed with variable confidence. We now restate the formal definition.

Definition 6.2.1. A Probabilistic Dependency Graph (PDG) is a tuple $m = (\mathcal{N}, \mathcal{A}, \mathcal{V}, \mathbb{P}, \alpha, \beta)$, where

- \mathcal{N} is a set of nodes, corresponding to variables;
- \mathcal{V} associates each node $X \in \mathcal{N}$ with a set $\mathcal{V}(X)$ of possible values that the variable X can take;
- \mathcal{A} is a set of labeled edges $\{X \xrightarrow{a} Y\}$, each with a source X and target Y from \mathcal{N} ;
- \mathbb{P} associates a cpd $p_a(Y|X)$ to each edge $\mathcal{A}LXY \in \mathcal{A}$;
- α associates to each edge $\mathcal{A}LXY$ a non-negative number α_L representing the modeler's confidence in the functional dependence of Y on X ;
- β associates to each edge L a number β_L , the modeler's confidence in the reliability of the cpd p_a . □

How should one choose parameters β and α ? A choice of $\beta_L = 0$ means that

the cpd p_L is effectively ignored, in the sense that such a PDG is equivalent to one in which the edge is attached to a different cpd $q \neq p_L$. On the other hand, a large value of β_L (or ∞) indicates high (or absolute) confidence in the cpd. By default, we suppose $\beta = 1$, which is just a convenient choice of units—what's important are the magnitudes of β relative to one another. The parameter α , typically in $[0, 1]$, represents certainty in the causal structure of the graph, and plays only a minor role in this paper.

Like other graphical models, PDGs have semantics in terms of joint distributions μ over all variables. Most directly, a PDG m determines two scoring functions on joint distributions μ . For the purposes of this paper, the more important of the two is the *incompatibility* of μ with respect to m , which measures the quantitative discrepancy between μ and m 's cpds, and is given by

$$Inc_m(\mu) := \sum_{X \xrightarrow{a} Y} \beta_L \mathbb{E}_{x \sim \mu(X)} D\left(\mu(Y|x) \parallel p_a(Y|x)\right). \quad (6.1)$$

Relative entropy $D(\mu \| p)$ measures divergence between μ and p , and can be viewed as the overhead (in extra bits per sample) of using codes optimized for p , when in fact samples are distributed according to μ [63]. But if one uses edges in proportion to the confidence one has in them, then inefficiencies for of high-confidence cpds are compounded, and hence more costly. So $Inc_m(\mu)$ measures the total excess cost of using m 's cpds in proportion to their confidences β , when worlds are distributed according to μ .

The *inconsistency* of m , denoted $\langle\!\langle m \rangle\!\rangle$, is the smallest possible incompatibility of m with any distribution: $\langle\!\langle m \rangle\!\rangle := \inf_\mu Inc_m(\mu)$. This quantity, which does not depend on α , is the primary focus of this paper.

The second scoring function defined by a PDG m , called the *Information Deficiency*, measures the qualitative discrepancy between m and μ , and is given

by

$$IDef_m(\mu) := -H(\mu) + \sum_{X \xrightarrow{a} Y} \alpha_L H_\mu(Y | X).$$

$IDef_m(\mu)$ can be thought of as the information needed to separately describe the target of each edge L given the value of its source (weighted by α_L) beyond the information needed to fully describe a sample from μ .

As shown by Richardson and Halpern [82], it is via these two scoring functions that PDGs capture other graphical models. The distribution specified by a BN \mathcal{B} is the unique one that minimizes both $Inc_{\mathcal{B}}$ and $IDef_{\mathcal{B}}$ (and hence every positive linear combination of the two), while the distribution specified by a factor graph Φ uniquely minimizes the sum $Inc_{\Phi} + IDef_{\Phi}$. In general, for any $\gamma > 0$, one can consider a weighted combination $\llbracket m \rrbracket_{\gamma}(\mu) := Inc_m(\mu) + \gamma IDef_m(\mu)$, for which there is a corresponding γ -inconsistency $\langle\!\langle m \rangle\!\rangle_{\gamma} := \inf_{\mu} \llbracket m \rrbracket_{\gamma}(\mu)$. In the limit as $\gamma \rightarrow 0$, there is always a unique best distribution whose score is $\langle\!\langle m \rangle\!\rangle$.

We now present some shorthand to clarify the presentation. We typically conflate a cpd's symbol with its edge label, thus drawing the PDG with a single edge attached to $f(Y|X)$ as $[X] \dashv f \dashv [Y]$. [Definition 6.2.1](#) is equivalent to one in which edge sources and targets are both *sets* of variables. This allows us to indicate joint dependence with multi-tailed arrows, joint distributions with multi-headed arrows, and unconditional distributions with nothing at the tail.

For instance, we draw

$$p(Y|X, Z) \text{ as } \begin{array}{c} Z \\ \nearrow p \\ X \end{array} \dashv Y \dashv \begin{array}{c} Z \\ \searrow q \\ B \end{array}, \text{ and } q(A, B) \text{ as } \begin{array}{c} A \\ \swarrow \\ X \end{array} \dashv \begin{array}{c} B \\ \searrow \\ Y \end{array}.$$

To emphasize that a cpd $f(Y|X)$ is degenerate (a function $f : X \rightarrow Y$), we will draw it with two heads, as in: $[X] \dashv f \dashv [Y]$. We identify an event $X = x$ with the degenerate unconditional distribution $\delta_x(X)$ that places all mass on x ; hence it may be associated to an edge and drawn simply as $\xrightarrow{x} [X]$. To

specify a confidence $\beta \neq 1$, we place the value near the edge, lightly colored and parenthesized, as in: $\xrightarrow[\beta]{p} [X]$, and we write (∞) for the limit of high confidence ($\beta \rightarrow \infty$).

Intuitively, believing more things can't make you any less inconsistent. Lemma 6.2.1 captures this formally: adding cpds or increasing confidences cannot decrease a PDG's inconsistency.

Lemma 6.2.1 (Monotonicity of $\langle\!\langle \cdot \rangle\!\rangle$). *Suppose PDGs m and m' differ only in their edges (resp. \mathcal{A} and \mathcal{A}') and confidences (resp. β and β'). If $\mathcal{A} \subseteq \mathcal{A}'$ and $\beta_L \leq \beta'_L$ for all $L \in \mathcal{A}$, then $\langle\!\langle m \rangle\!\rangle_\gamma \leq \langle\!\langle m' \rangle\!\rangle_\gamma$ for all γ .¹*

[link to proof]

As we will see, this tool is sufficient to derive many interesting relationships between loss functions.

6.3 Standard Metrics as Inconsistencies

Suppose you believe that X is distributed according to $p(X)$, and also that it (certainly) equals some value x . These beliefs are consistent if $p(X=x) = 1$ but become less so as $p(X=x)$ decreases. In fact, this inconsistency is equal to the information content $I_p[X=x] := -\log p(X=x)$, or *surprisal* [90], of the event $X=x$, according to p .² In machine learning, I_p is usually called “negative log likelihood”, and is perhaps the most popular objective for training generative models [35, 66].

¹All proofs can be found in Section 7.B.

²This construction requires the event $X=x$ to be measurable. One can get similar, but subtler, results for densities, where this is not the case; see Section 6.A.

[link to proof]

Proposition 6.3.1. Consider a distribution $p(X)$. The inconsistency of the PDG comprising p and $X=x$ equals the surprisal $I_p[X=x]$. That is,

$$I_p[X=x] = \left\langle\!\!\left\langle \xrightarrow{p} X \xleftarrow{x} \right\rangle\!\!\right\rangle.$$

(Recall that $\langle\!\langle m \rangle\!\rangle$ is the inconsistency of the PDG m .)

In some ways, this result is entirely unsurprising, given that (6.1) is a flexible formula built out of information theoretic primitives. Even so, note that the inconsistency of believing both a distribution and an event happens to be the standard measure of discrepancy between the two—and is even named after “surprise”, a particular expression of epistemic conflict.

Still, we have a ways to go before this amounts to any more than a curiosity. One concern is that this picture is incomplete; we train probabilistic models with more than one sample. What if we replace x with an empirical distribution over many samples?

Proposition 6.3.2. If $p(X)$ is a probabilistic model of X , and $\mathcal{D} = \{x_i\}_{i=1}^m$ is a dataset with empirical distribution $\Pr_{\mathcal{D}}$, then [link to proof]

$$\frac{1}{m} \sum_{i=1}^m I_p[X=x_i] = \left\langle\!\!\left\langle \xrightarrow{p} X \xleftarrow{\Pr_{\mathcal{D}}} \right\rangle\!\!\right\rangle + H(\Pr_{\mathcal{D}}).$$

Remark. The term $H(\Pr_{\mathcal{D}})$ is a constant depending only on the data, so is irrelevant for optimizing p .

Essentially the only choices we’ve made in specifying the PDG of Proposition 6.3.2 are the confidences. But $\text{CrossEntropy}(\Pr_{\mathcal{D}}, p)$ is the expected code length per sample from $\Pr_{\mathcal{D}}$, when using codes optimized for the (incorrect) distribution p . So implicitly, a modeler using cross-entropy has already articulated a

belief the data distribution $\Pr_{\mathcal{D}}$ is the “true one”. To get the same effect from a PDG, the modeler must make this belief explicit by placing infinite confidence in $\Pr_{\mathcal{D}}$.

Now consider an orthogonal generalization of [Proposition 6.3.1](#), in which the sample x is only a partial observation of (x, z) from a joint model $p(X, Z)$.

Proposition 6.3.3. *If $p(X, Z)$ is a joint distribution, then the information content of the partial observation $X = x$ is given by*

$$I_p[X=x] = \left\langle \begin{array}{c} Z \\ \nearrow p \\ X \end{array} \leftrightarrow^x \right\rangle. \quad (6.2)$$

Intuitively, the inconsistency of the PDG on the right side of (6.2) is localized to X , where the observation x conflicts with $p(X)$; other variables don’t make a difference. The multi-sample partial-observation generalization also holds; see [Section 6.B.3](#).

So far we have considered models of an unconditional distribution $p(X)$. Because they are unconditional, such models must describe how to generate a complete sample X without input, and so are called *generative*; the process of training them is called *unsupervised* learning [40]. In the (more common) *supervised* setting, we train *discriminative* models to predict Y from X , via labeled samples $\{(x_i, y_i)\}_i$. There, cross entropy loss is perhaps even more dominant—and it is essentially the inconsistency of a PDG consisting of the predictor $h(Y|X)$ together with high-confidence data.

Proposition 6.3.4 (Cross Entropy, Supervised). *The inconsistency of the PDG comprising a probabilistic predictor $h(Y|X)$, and a high-confidence empirical distribution $\Pr_{\mathcal{D}}$ of a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^m$ equals the cross-entropy loss (minus the empirical*

uncertainty in Y given X , a constant depending only on \mathcal{D}). That is,

$$\left\langle \Pr_{\mathcal{D}} \begin{array}{c} \xrightarrow{\text{---}} \\ X \end{array} \xrightarrow[h]{\quad} \begin{array}{c} \xrightarrow{\text{---}} \\ Y \end{array} \right\rangle = \frac{1}{m} \sum_{i=1}^m \log \frac{1}{h(y_i | x_i)} - H_{\Pr_{\mathcal{D}}}(Y|X).$$

Simple evaluation metrics, such as the accuracy of a classifier, and the mean squared error of a regressor, also arise naturally as inconsistencies.

Proposition 6.3.5 (Log Accuracy as Inconsistency). *Consider functions $f, h : X \rightarrow Y$ from inputs to labels, where h is a predictor and f generates the true labels. The inconsistency of believing f and h (with any confidences), and a distribution $D(X)$ with confidence β , is β times the log accuracy of h . That is,*

$$\left\langle \begin{array}{c} D \xrightarrow{\text{---}} \\ (\beta) \end{array} \begin{array}{c} \xrightarrow{h^{(r)}} \\ X \end{array} \xrightarrow[f^{(s)}]{\quad} \begin{array}{c} \xrightarrow{h^{(r)}} \\ Y \end{array} \right\rangle = -\beta \log \Pr_{x \sim D}(f(x) = h(x)) = \beta I_D[f = h]. \quad (6.3)$$

One often speaks of the accuracy of a hypothesis h , leaving the true labels f and empirical distribution D implicit. Yet Proposition 6.3.5 suggests that there is a sense in which $D(X)$ plays the primary role: the inconsistency in (6.3) is scaled by the confidence in D , and does not depend on the confidences in h or f . Why should this be this the case? Expressing (x, y) such that $y \neq f(x)$ with codes optimized for f is not just inefficient, but impossible. The same is true for h , so we can only consider μ such that $\mu(f = h) = 1$. In other words, the only way to form a joint distribution *at all* compatible with both the predictor h and the labels f , is to throw out samples that the predictor gets wrong—and the cost of throwing out samples scales with your confidence in D , not in h . This illustrates why accuracy gives no gradient information for training h . It is worth noting that this is precisely the opposite of what happened in Proposition 6.3.4: there we

link to proof

were unwilling to budge on the input distribution, and the inconsistency scaled with the confidence in h .

Observe how even properties of these simple metrics—relationships with one another and features of gradients—can be clarified by an underlying model.

When $Y \cong \mathbb{R}^n$, an estimator $h(Y|X)$ is referred to as a regressor instead of a classifier. In this setting, most answers are incorrect, but some more so than others. A common way of measuring incorrectness is with mean squared error (MSE): $\mathbb{E}|f(X) - Y|^2$. MSE is also the inconsistency of believing that the labels and predictor have Gaussian noise—often a reasonable assumption because of the central limit theorem.

Proposition 6.3.6 (MSE as Inconsistency).

[link to proof]

$$\left\langle\!\!\left\langle \begin{array}{c} D \xrightarrow{(\infty)} X \\ \xrightarrow{f} \mu_f \end{array} \right\rangle\!\!\right\rangle = \frac{1}{2} \mathbb{E}_D |f(X) - h(X)|^2$$

$$=: \text{MSE}_D(f, h),$$

where $\mathcal{N}_1(Y|\mu)$ is a unit Gaussian on Y with mean μ .

In the appendix, we treat general univariate Gaussian predictors, with arbitrary variances and confidences.

6.4 Regularizers and Priors as Inconsistencies

Regularizers are extra terms added to loss functions, which provide a source of inductive bias towards simple model parameters. There is a well-known correspondence between using a regularizer and doing maximum *a posteriori*

inference with a prior,³ in which L2 regularization corresponds to a Gaussian prior [78], while L1 regularization corresponds to a Laplacian prior [96]. Note that the ability to make principled modeling choices about regularizers is a primary benefit of this correspondence. Our approach provides a new justification of it.

Proposition 6.4.1. Suppose you have a parameterized model $p(Y|\Theta)$, a prior $q(\Theta)$, and a trusted distribution $D(Y)$. The inconsistency of also believing $\Theta = \theta$ is the cross entropy loss, plus the regularizer: $\log \frac{1}{q(\theta)}$ times your confidence in q . That is,

$$\left\langle \begin{array}{c} q \\ \xrightarrow{(\beta)} \end{array} \Theta \xrightarrow[p]{Y} D \uparrow_{(\infty)} \right\rangle = \mathbb{E}_{y \sim D} \log \frac{1}{p(y|\theta)} + \beta \log \frac{1}{q(\theta)} - H(D) \quad (6.4)$$

If our prior is $q(\theta) = \frac{1}{k} \exp(-\frac{1}{2}\theta^2)$, a (discretized) unit gaussian, then the right hand side of (6.4) becomes

$$\underbrace{\mathbb{E}_D \log \frac{1}{p(Y|\theta)}}_{\text{Cross entropy loss}} + \underbrace{\frac{\beta}{2} \theta_0}_{\substack{\text{L2 regularizer} \\ (\text{complexity cost of } \theta)}} + \underbrace{\beta \log k - H(D)}_{\text{constant in } p \text{ and } \theta},$$

which is the L2 regularized version of Proposition 6.3.2. Moreover, the regularization strength corresponds exactly to the confidence β . What about other priors? It is not difficult to see that if we use a (discretized) unit Laplacian prior, $q(\theta) \propto \exp(-|\theta|)$, the second term instead becomes $\beta|\theta_0|$, which is L1 regularization. More generally, to consider a complexity measure $U(\theta)$, we need only include the Gibbs distribution $\Pr_U(\theta) \propto \exp(-U(\theta))$ into our PDG. We remark that nothing here is specific to cross entropy; any of the objectives we describe can be regularized in this way.

link to proof

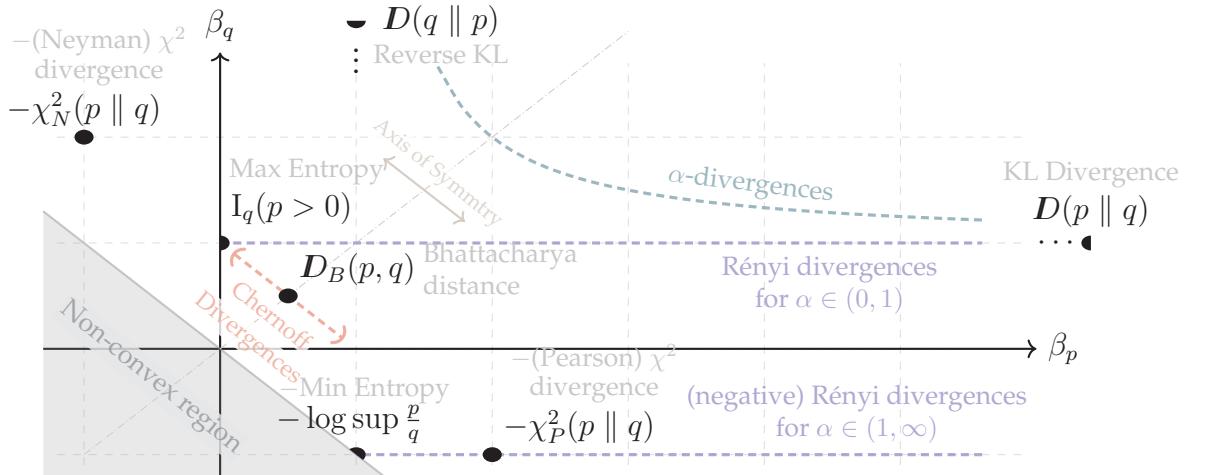


Figure 1: A map of the inconsistency of the PDG comprising $p(X)$ and $q(X)$, as we vary their respective confidences β_p and β_q . Solid circles indicate well-known named measures, semicircles indicate limiting values, and the heavily dashed lines are well-established classes.

6.5 Statistical Distances as Inconsistencies

Suppose you are concerned with a single variable X . One friend has told you that it is distributed according to $p(X)$; another has told you that it follows $q(X)$. You adopt both beliefs. Your mental state will be inconsistent if (and only if) $p \neq q$, with more inconsistency the more p and q differ. Thus the inconsistency of a PDG comprising p and q is a measure of divergence. Recall that a PDG also allows us to specify the confidences β_p and β_q of each cpd, so we can form a PDG divergence $D_{(r,s)}^{\text{PDG}}(p||q)$ for every setting (r,s) of (β_p, β_q) . It turns out that a large class of statistical divergences arise in this way. We start with a familiar one.

Proposition 6.5.1 (KL Divergence as Inconsistency). *The inconsistency of believing p with complete certainty, and also q with some finite certainty β , is β times the KL Divergence (or relative entropy) of q with respect to p . That is,*

$$\left\langle \overrightarrow{p}_{(\infty)} X \xleftarrow{(\beta)} \overleftarrow{q} \right\rangle = \beta D(p || q).$$

³A full account can be found in the appendix.

This result gives us an intuitive interpretation of the asymmetry of relative entropy / KL divergence, and a prescription about when it makes sense to use it. $D(p \parallel q)$ is the inconsistency of a mental state containing both p and q , when absolutely certain of p (and not willing to budge on it). This concords with the standard intuition that $D(p \parallel q)$ reflects the amount of information required to change q into p , which is why it is usually called the relative entropy “from q to p ”.

We now consider the general case of a PDG comprising $p(X)$ and $q(X)$ with arbitrary confidences.

Lemma 6.5.2. *The inconsistency $D_{(r,s)}^{\text{PDG}}(p \parallel q)$ of a PDG comprising $p(X)$ with confidence r and $q(X)$ with confidence s is given in closed form by*

$$\left\langle\left\langle \frac{p}{(r)} \rightarrow X \leftarrow \frac{q}{(s)} \right\rangle\right\rangle = -(r+s) \log \sum_x \left(p(x)^r q(x)^s \right)^{\frac{1}{r+s}}.$$

Of the many generalizations of KL divergence, Rényi divergences, first characterized by Alfréd Rényi 1961 are perhaps the most significant, as few others have found either application or an interpretation in terms of coding theory [91]. The Rényi divergence of order α between two distributions $p(X)$ and $q(X)$ is given by

$$D_\alpha(p \parallel q) := \frac{1}{1-\alpha} \log \sum_{x \in \mathcal{V}(X)} p(x)^\alpha q(x)^{1-\alpha}. \quad (6.5)$$

Rényi introduced this measure in the same paper as the more general class of f -divergences, but directs his attention towards those of the form (6.5), because they satisfy a natural weakening of standard postulates for Shannon entropy due to Fadeev [26]. Concretely, every symmetric, continuous measure that additively separates over independent events, and with a certain “mean-value property”, up to scaling, is of the form (6.5) for some α [79]. It follows from Lemma 6.5.2

that every Rényi divergence is a PDG divergence, and every (non-limiting) PDG divergence is a (scaled) Rényi divergence.

Corollary 6.5.2.1 (Rényi Divergences).

$$\begin{aligned} \left\langle\left\langle \frac{p}{(r)} \rightarrow [X] \leftarrow \frac{q}{(s)} \right\rangle\right\rangle &= s \cdot D_{\frac{r}{r+s}}(p \parallel q) \\ \text{and} \quad D_\alpha(p \parallel q) &= \left\langle\left\langle \frac{p}{(\frac{\alpha}{1-\alpha})} \rightarrow [X] \leftarrow \frac{q}{(1-\alpha)} \right\rangle\right\rangle \end{aligned}$$

However, the two classes are not identical, because the PDG divergences have extra limit points. One big difference is that the reverse KL divergence $D(q \parallel p)$ is not a Rényi divergence $D_\alpha(p \parallel q)$ for any value (or limit) of α . This lack of symmetry has led others [e.g., 18] to work instead with a symmetric variant called α -divergence, rescaled by an additional factor of $\frac{1}{\alpha}$. The relationships between these quantities can be seen in [Figure 1](#).

The Chernoff divergence measures the tightest possible exponential bound on probability of error [71] in Bayesian hypothesis testing. It also happens to be the smallest possible inconsistency of simultaneously believing p and q , with total confidence 1.

Corollary 6.5.2.2. *The Chernoff Divergence between p and q equals*

$$\inf_{\beta \in (0,1)} \left\langle\left\langle \frac{p}{(\beta)} \rightarrow [X] \leftarrow \frac{q}{(1-\beta)} \right\rangle\right\rangle.$$

One significant consequence of representing divergences as inconsistencies is that we can use [Lemma 6.2.1](#) to derive relationships between them. The following facts follow directly from [Figure 1](#), by inspection.

Corollary 6.5.2.3.

1. Rényi entropy is monotonic in its parameter α .
2. $D(p \parallel q) \geq 2D_B(p, q) \leq D(q \parallel p)$.
3. If $q(p > 0) < 1$ (i.e., $q \not\ll p$), then $D(q \parallel p) = \infty$.

These divergences correspond to PDGs with only two edges and one variable.

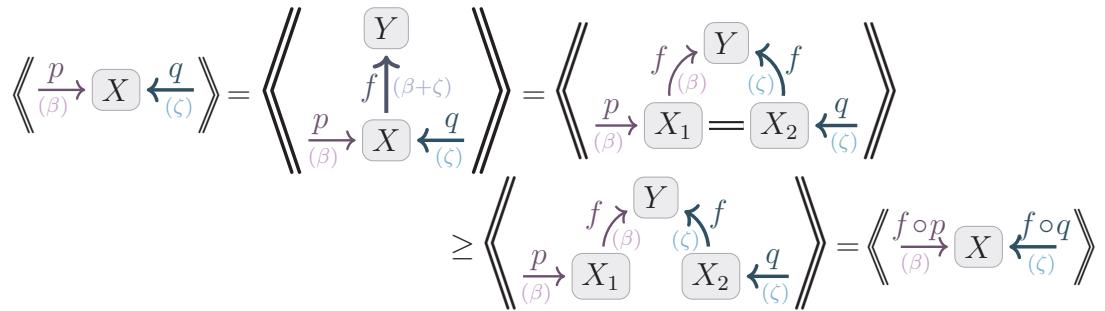


Figure 2: A visual, monotonicity-based proof of the data-processing inequality for all PDG divergences: $D_{(\beta,\zeta)}^{\text{PDG}}(p \parallel q) \geq D_{(\beta,\zeta)}^{\text{PDG}}(f \circ p \parallel f \circ q)$. In words: the cpd $f(Y|X)$ can always be satisfied, so adds no inconsistency. It is then equivalent to split f and the variable X into X_1 and X_2 with edges enforcing $X_1 = X_2$. But removing such edges can only decrease inconsistency. Finally, compose the remaining cpds to give the result. See the appendix for a full justification.

What about more complex graphs? For a start, conditional divergences

$$D_{(r,s)}^{\text{PDG}}\left(p(Y|X) \middle\| q(Y|X) \middle| r(X)\right) := \mathbb{E}_{x \sim r} D_{(r,s)}^{\text{PDG}}\left(p(Y|x) \middle\| q(Y|x)\right)$$

can be represented straightforwardly as

$$D_{(r,s)}^{\text{PDG}}(p \parallel q | r) = \left\langle \frac{r}{(\infty)} \rightarrow X \xrightarrow[p(r)]{q(s)} Y \right\rangle.$$

Other structures are useful intermediates. [Lemma 6.2.1](#), plus some structural manipulation, gives visual proofs of many divergence properties; [Figure 2](#) features such a proof of the data-processing inequality. And in general, PDG inconsistency can be viewed as a vast generalization of divergences to arbitrary structured objects.

6.6 Variational Objectives and Bounds

The fact that the incompatibility of \mathcal{M} with a *specific* joint distribution μ is an upper bound on the inconsistency is not a deep one, but it is of a variational

flavor. Here, we focus on the more surprising converse: PDG semantics capture general aspects of variational inference and provide a graphical proof language for it.

6.6.1 PDGs and Variational Approximations

We begin by recounting the standard development of the ‘Evidence Lower BOund’ (ELBO), a standard objective for training latent variable models [11, §2.2]. Suppose we have a model $p(X, Z)$, but only have access to observations of x . In service of adjusting $p(X, Z)$ to make our observations more likely, we would like to maximize $\log p(X = x)$, the “evidence” of x (Proposition 6.3.3). Unfortunately, computing $p(X) = \sum_z p(X, Z=z)$ requires summing over all of Z , which can be intractable. The variational approach is as follows: fix a family of distributions \mathcal{Q} that is easy to sample from, choose some $q(Z) \in \mathcal{Q}$, and define $\text{ELBO}_{p,q}(x) := \mathbb{E}_{z \sim q} \log \frac{p(x, z)}{q(z)}$. This is something we can estimate, since we can sample from q . By Jensen’s inequality,

$$\text{ELBO}_{p,q}(x) = \mathbb{E}_{p,q} \log \frac{p(x, Z)}{q(Z)} \leq \log \left[\mathbb{E}_q \frac{p(x, Z)}{q(Z)} \right] = \log p(x),$$

with equality if $q(Z) = p(Z)$. So to find p maximizing $p(x)$, it suffices to adjust p and q to maximize $\text{ELBO}_{p,q}(x)$,⁴ provided \mathcal{Q} is expressive enough.

The formula for the ELBO is somewhat difficult to make sense of.⁵ Nevertheless, it arises naturally as the inconsistency of the appropriate PDG.

Proposition 6.6.1. *The negative ELBO of x is the inconsistency of the PDG containing*

link to
proof

⁴or for many iid samples: $\max_{p,q} \sum_{x \in \mathcal{D}} \text{ELBO}_{p,q}(x)$.

⁵Especially if p, q are densities. See Section 6.A.

p, q , and $X=x$, with high confidence in q . That is,

$$-\text{ELBO}_{p,q}(x) = \left\langle \frac{q}{(\infty)} \rightarrow Z \xrightarrow[p]{\nwarrow} X \xleftarrow{x} \right\rangle.$$

Owing to its structure, a PDG is often more intuitive and easier to work with than the formula for its inconsistency. To illustrate, we now give a simple and visually intuitive proof of the bound traditionally used to motivate the ELBO, via Lemma 6.2.1:

$$\log \frac{1}{p(x)} = \left\langle Z \xrightarrow[p]{\nwarrow} X \xleftarrow{x} \right\rangle \leq \left\langle q \circledast Z \xrightarrow[p]{\nwarrow} X \xleftarrow{x} \right\rangle = -\text{ELBO}_{p,q}(x).$$

The first and last equalities are Propositions 6.6.1 and 6.3.3 respectively. Now to reap some pedagogical benefits. The second PDG has more edges so it is clearly at least as inconsistent. Furthermore, it's easy to see that equality holds when $q(Z)=p(Z)$: the best distribution for the left PDG has marginal $p(Z)$ anyway, so insisting on it incurs no further cost.

6.6.2 Variational Auto-Encoders and PDGs

An autoencoder is a probabilistic model intended to compress a variable X (e.g., an image) to a compact latent representation Z . Its structure is given by two conditional distributions: an encoder $e(Z|X)$, and a decoder $d(X|Z)$. Of course, not all pairs of cpds fill this role equally well. One important consideration is the *reconstruction error* (6.6): when we decode an encoded image, we would like it to resemble the original.

$$\text{Rec}(x) := \mathbb{E}_{z \sim e(Z|x)} \underbrace{\mathbf{I}_{d(X|z)}(x)}_{\begin{pmatrix} \text{additional bits required to} \\ \text{decode } x \text{ from its encoding } z \end{pmatrix}} = \sum_z e(z|x) \log \frac{1}{d(x|z)} \quad (6.6)$$

There are other desiderata as well. Perhaps good latent representations Z have uncorrelated components, and are normally distributed. We encode such wishful thinking as a belief $p(Z)$, known as a variational prior.

The data of a Variational Auto-Encoder [52, 80], or VAE, consists of $e(Z|X)$, $d(X|Z)$, and $p(Z)$. The encoder $e(Z|X)$ can be used as a variational approximation of Z , differing from $q(Z)$ of [Section 6.6.1](#) only in that it can depend on X . VAEs are trained with the analogous form of the ELBO:

$$\begin{aligned}\text{ELBO}_{p,e,d}(x) &:= \mathbb{E}_{z \sim e(Z|x)} \left[\log \frac{p(z)d(x|z)}{e(z|x)} \right] \\ &= -\text{Rec}(x) - D(e(Z|x) \| p).\end{aligned}$$

This gives us the following analog of [Proposition 6.6.1](#).

Proposition 6.6.2. *The VAE loss of a sample x is the inconsistency of the PDG comprising the encoder e (with high confidence, as it defines the encoding), decoder d , prior p , and x . That is,*

$$-\text{ELBO}_{p,e,d}(x) = \left\langle \overset{p}{\rightarrow} Z \xrightarrow[e]{(\infty)} X \xleftarrow{x} \right\rangle.$$

We now give a visual proof of the analogous variational bound. Let $\Pr_{p,d}(X, Z) := p(Z)d(X|Z)$ be the distribution that arises from decoding the prior. Then:

$$\log \frac{1}{\Pr_{p,d}(x)} = \left\langle \overset{p}{\downarrow} Z \xrightarrow[d]{x} X \right\rangle \leq \left\langle \overset{p}{\downarrow} Z \xrightarrow[e]{(\infty)} X \xleftarrow{x} \right\rangle = -\text{ELBO}_{p,e,d}(x).$$

The first and last equalities are [Propositions 6.6.2](#) and [6.3.3](#), and the inequality is [Lemma 6.2.1](#). See the appendix for multi-sample analogs of the bound and [Proposition 6.6.2](#).

6.6.3 The β -VAE Objective

The ELBO is not the only objective that has been used to train networks with a VAE structure. In the most common variant, due to Higgins et al. [42], one weights the reconstruction error (6.6) and the ‘KL term’ differently, resulting in a loss function of the form

$$\beta\text{-ELBO}_{p,e,d}(x) := -\text{Rec}(x) - \beta D(e(Z|x) \| p),$$

which, when $\beta=1$, is the ELBO as before. The authors view β as a regularization strength, and argue that it sometimes helps to have a stronger prior. Sure enough:

Proposition 6.6.3. $-\beta\text{-ELBO}_{p,e,d}(x)$ is the inconsistency of the same PDG, but with confidence β in $p(Z)$.

link to proof

6.7 Free Energy as Factor Graph Inconsistency

A weighted factor graph $\Psi = (\phi_J, \theta_J)_{J \in \mathcal{J}}$, where each θ_J is a real-valued weight, J is associated with a subset of variables \mathbf{X}_J , and $\phi_J : \mathcal{V}(\mathbf{X}_J) \rightarrow \mathbb{R}$, determines a distribution by

$$\Pr_\Psi(\mathbf{x}) = \frac{1}{Z_\Psi} \prod_{J \in \mathcal{J}} \phi_J(\mathbf{x}_J)^{\theta_J}.$$

Z_Ψ is the constant $\sum_{\mathbf{x}} \prod_{J \in \mathcal{J}} \phi_J(\mathbf{x}_J)^{\theta_J}$ required to normalize the distribution, and is known as the *partition function*. Computing $\log Z_\Psi$ is intimately related to probabilistic inference in factor graphs [62]. Following Richardson and Halpern [82], let $\mathit{pdg}(\Psi)$ be the PDG with edges $\{\xrightarrow{J} \mathbf{X}_J\}_{\mathcal{J}}$, cpds $p_J(\mathbf{X}_J) \propto \phi_J(\mathbf{X}_J)$, and

weights $\alpha_J, \beta_J := \theta_J$. There, it is shown that Pr_Ψ is the unique minimizer of $\llbracket \mathbf{pdg}(\Psi) \rrbracket_1$. But what about the corresponding inconsistency, $\langle\!\langle \mathbf{pdg}(\Psi) \rangle\!\rangle_1$?

If the factors are normalized and all variables are edge targets, then $Z_\Psi \leq 1$, so $\log \frac{1}{Z_\Psi} \geq 0$ measures how far the product of factors is from being a probability distribution. So in a sense, it measures Ψ 's inconsistency.

Proposition 6.7.1. *For all weighted factor graphs Ψ , we have that $\langle\!\langle \mathbf{pdg}(\Psi) \rangle\!\rangle_1 = -\log Z_\Psi$.*

link to
proof

The exponential families generated by weighted factor graphs are a cornerstone of statistical mechanics, where $-\log Z_\Psi$ is known as the (Heimholz) free energy. It is also an especially natural quantity to minimize: the principle of free-energy minimization has been enormously successful in describing of not only chemical and biological systems [16], but also cognitive ones [29].

6.8 Beyond Standard Losses: A Concrete Example

In contexts where a loss function is standard, it is usually for good reason—which is why we have focused on recovering standard losses. But most situations are non-standard, and even if they have standard sub-components, those components may interact with one another in more than one way. Correspondingly, there is generally more than one way to cobble standard loss functions together. How should you choose between them? By giving a principled model of the situation.

Suppose we want to train a predictor network $h(Y|X)$ from two sources of

information: partially corrupted data with distribution $d(X, Y)$, and a simulation with distribution $s(X, Y)$. If the simulation is excellent and the data unsalvageable, we would have high confidence in s and low confidence in d , in which case we would train with cross entropy with respect to s , $\mathcal{L}_{\text{sim}} := \mathbb{E}_s[\log 1/h(Y|X)]$. Conversely, if the simulation were bad and the data mostly intact, we would use \mathcal{L}_{dat} , the cross entropy with respect to d . What if we're not so confident in either?

One approach a practitioner might find attractive is to make a dataset from samples of both s and d , or equivalently, train with a convex combination of the two previous losses, $\mathcal{L}_1 := \lambda_s \mathcal{L}_{\text{sim}} + \lambda_d \mathcal{L}_{\text{dat}}$ for some $\lambda_s, \lambda_d > 0$ with $\lambda_s + \lambda_d = 1$. This amounts to training h with cross entropy with respect to the mixture $\lambda_s s + \lambda_d d$. Doing so treats d and s as completely unrelated, and so redundancy is not used to correct errors—a fact on display when we present the modeling choices in PDG form, such as

$$\mathcal{L}_1 = \left\langle \begin{array}{c} \xrightarrow{\lambda} \boxed{\begin{array}{c} Z \\ \bullet \quad \bullet \\ \text{sim dat} \end{array}} \\ \xrightarrow{(\infty)} \begin{array}{c} \text{dat} \mapsto d \\ \text{sim} \mapsto s \\ \hline (\infty) \end{array} \end{array} \right\rangle,$$

in which a switch variable Z with possible values $\{\text{sim}, \text{dat}\}$ controls whether samples come from s or d , and is distributed according to $\lambda(Z=\text{sim}) = \lambda_s$.

Our practitioner now tries a different approach: draw data samples $(x, y) \sim d$ but discount h 's surprisal when the simulator finds the point unlikely, via loss $\mathcal{L}_2 := \mathbb{E}_d[s(X, Y) \log 1/h(Y|X)]$. This is the cross entropy with respect to the (unnormalized) product density ds , which in many ways is appropriate. However, by this metric, the optimal predictor $h^*(Y|x) \propto d(Y|x)s(Y|x)$ is *uncalibrated* [22]. If the data and simulator agree ($d=s$), then we would want $h(Y|x)=s(Y|x)$ for all x , but instead we get $h^*(Y|x) \propto s(Y|x)^2$. So h^* is overconfident. What went wrong? \mathcal{L}_2 cannot be written as the (ordinary $\gamma=0$) inconsistency of a PDG containing

only s , h , and d , but for a large fixed γ , it is essentially the γ -inconsistency

$$\mathcal{L}_2 \approx C \left\langle \begin{array}{c} X \\ \downarrow h \\ Y \end{array} \right\rangle_{\gamma} + \text{const},$$

where C is the constant required to normalize the joint density sd , and const does not depend on h . However, the values of α in this PDG indicate an over-determination of XY (it is determined in two different ways), and so h^* is more deterministic than intended. By contrast,

$$\mathcal{L}_3 := \left\langle \begin{array}{c} X \\ \downarrow h \\ Y \end{array} \right\rangle,$$

does not have this issue: the optimal predictor h^* according to \mathcal{L}_3 is proportional to the λ -weighted geometric mean of s and d . It seems that our approach, in addition to providing a unified view of standard loss functions, can also suggest more appropriate loss functions in practical situations.

6.9 Reverse-Engineering a Loss Function?

Given an *arbitrary* loss function, can we find a PDG that gives rise to it? The answer appears to be yes—although not without making unsavory modeling choices. Without affecting its semantics, one may add the variable T that takes values $\{t, f\}$, and the event $T = t$, to any PDG. Now, given a cost function $c : \mathcal{V}(X) \rightarrow \mathbb{R}_{\geq 0}$, define the cpd $\hat{c}(T|X)$ by $\hat{c}(t|x) := e^{-c(x)}$. By threatening to generate the falsehood f with probability dependent on the cost of X , \hat{c} ties the value of X to inconsistency.

Proposition 6.9.1. $\left\langle \xrightarrow[p]{(\infty)} X \xrightarrow{\hat{c}} T \xleftarrow[t]{} \right\rangle = \mathbb{E}_{x \sim p}[c(x)].$

[link to proof]

Setting confidence $\beta_p := \infty$ may not be realistic since we're still training the model p , but doing so is necessary to recover $\mathbb{E}_p c$.⁶ Any mechanism that generates inconsistency based on the value of X (such as this one) also works in reverse: the PDG "squirms", contorting the probability of X to disperse the inconsistency. One cannot simply "emit loss" without affecting the rest of the model, as one does with utility in an Influence Diagram [44]. Even setting every $\beta := \infty$ may not be enough to prevent the squirming. To illustrate, consider a model \mathcal{S} of the supervised learning setting (predict Y from X), with labeled data \mathcal{D} , model h , and a loss function ℓ on pairs of output labels.

Concretely, define:

$$\mathcal{S} := \begin{array}{c} \Pr_{\mathcal{D}} \rightarrow Y \\ \downarrow \text{---} \quad \downarrow h \\ X \xrightarrow[\infty]{} Y' \end{array} \quad \text{and} \quad \mathcal{L} := \mathbb{E}_{\substack{(x,y) \sim \Pr_{\mathcal{D}} \\ y' \sim p(Y'|x)}} [\ell(y, y')].$$

Given Proposition 6.9.1, one might imagine $\langle\!\langle \mathcal{S} \rangle\!\rangle = \mathcal{L}$, but this is not so. In some ways, $\langle\!\langle \mathcal{S} \rangle\!\rangle$ is actually preferable. The optimal $h(Y'|X)$ according to \mathcal{L} is a degenerate cpd that places all mass on the label(s) y_X^* minimizing expected loss, while the optimal $h(Y'|X)$ according to $\langle\!\langle \mathcal{S} \rangle\!\rangle$ is $\Pr_{\mathcal{D}}(Y|X)$, which means that it is calibrated, unlike ℓ . If, in addition, we set $\alpha_p, \alpha_{\Pr_{\mathcal{D}}} := 1$ and strictly enforce the qualitative picture, finally no more squirming is possible, as we arrive at $\lim_{\gamma \rightarrow \infty} \langle\!\langle \mathcal{S} \rangle\!\rangle_\gamma = \mathcal{L}$.

In the process, we have given up our ability to tolerate inconsistency by setting all probabilistic modeling choices in stone. What's more, we've dragged in the global parameter γ , further handicapping our ability to compose this model with others. To summarize: while model inconsistency readily generates appropriate loss functions, the converse does not work as well. Reverse-engineering a

⁶If β_p were instead equal to 1, we would have obtained $-\log \mathbb{E}_p \exp(-c(X))$, with optimal distribution $\mu(X) \neq p(X)$.

loss may require making questionable modeling choices with absolute certainty, resulting in brittle models with limited potential for composition. In the end, we must confront our modeling choices; good loss functions come from good models.

6.10 Conclusions

We seen that that PDG semantics, in the same stroke by which they capture Bayesian Networks and Factor Graphs [82], also generate many standard loss functions, including some non-trivial ones. In each case, the appropriate loss arises simply by articulating modeling assumptions, and then measuring inconsistency. Viewing loss functions in this way also has beneficial side effects, including an intuitive visual proof language for reasoning about the relationships between them.

This “universal loss”, which provides a principled way of choosing an optimization objective, may be of particular interest to the AI alignment community.

Acknowledgements

Work supported in part by MURI grant W911NF-19-1-0217. Many thanks to my advisor, Joe Halpern, for his generous support, and for valuable critiques of many drafts. Thanks as well to my reviewers, who pushed me to better explain the confidence parameters, and to include a practical example ([Section 6.8](#)). Finally, thanks to my friends, particularly Varsha Kishore and Greg Yauney, for helping me to refine the presentation of these ideas.

APPENDICES FOR CHAPTER 6

6.A The Fine Print for Probability Densities

Densities and Masses. Many of our results (Propositions 6.3.1 to 6.B.5) technically require the distribution to be represented with a mass function (as opposed to a probability density function, or pdf). A PDG containing both pdf and a finitely supported distribution on the same variable will typically have infinite inconsistency—but this is not just a quirk of the PDG formalism.

Probability density is not dimensionless (like probability mass), but rather has inverse X -units (e.g., probability per meter), so depends on an arbitrary choice of scale (the pdf for probability per meter and per centimeter will yield different numbers). In places where the objective does not have units that cancel before we take a logarithm, the use of a probability density $p(X)$ becomes sensitive to this arbitrary choice of parameterization. For instance, the analog of surprisal, $-\log p(x)$ for a pdf p , or its expectation, called differential entropy, both depend on an underlying scheme of measurement (an implicit base measure).

On the other hand, this choice of scale ultimately amounts to an additive constant. Moreover, beyond a certain point, decreasing the discretization size k of a discretized approximation $\tilde{p}_k(X)$ also contributes a constant that depends only on k . But such constants are irrelevant for optimization, and so, even though such quantities are ill-defined and arguably meaningless in the continuous limit, the use of the continuous analogs as loss functions is still justified.

The bottom line is that all our results hold in a uniform way for every discretization size — yet in the limit as the discretization becomes smaller, an

inconsistency may diverge to infinity. However, this divergence stems from an additive constant that depends only on the discretization size, which is irrelevant to its employment as a loss function. As a result, using one of these “unbalanced” functions involving densities where the units do not work out properly, results in a morally equivalent loss function, except without a diverging constant.

Markov Kernels. In the more general setting of measurable spaces, one may want to adjust the definition of a cpd that we gave, so that one instead works with *Markov Kernels*. This imposes an additional constraint: suppose the variable Y takes values in the measurable space $(\mathcal{V}(Y), \mathcal{B})$. If $p(Y|X)$ is to be a *Markov Kernel*, then for every fixed measurable subset $B \in \mathcal{B}$ of the measure space, we must require that $x \mapsto \Pr(B|x)$ be a measurable function (with respect to the measure space in which X takes values). This too mostly does not bear on the present discussion, because the σ -algebras for all measure spaces of interest, are fine enough that one can get an arbitrarily close approximation of any cpd with a Markov Kernels. This means that the infimum defining the inconsistency of a PDG does not change.

6.B Further Results and Generalizations

6.B.1 Full Characterization of Gaussian Predictors

The inconsistency of a PDG containing two univariate Gaussian regressors of with arbitrary parameters and confidences, is most cleanly articulated in terms of the geometric and quadratic means.

Definition 6.B.1 (Weighted Power Mean). The weighted power mean $M_p^w(\mathbf{r})$ of

Name	p	Formula
Harmonic	($p = -1$):	$\text{HM}_w(\mathbf{r}) = \frac{1}{\sum_{i=1}^n w_i/r_i}$
Geometric	($\lim p \rightarrow 0$):	$\text{GM}_w(\mathbf{r}) = \prod_{i=1}^n r_i^{w_i}$
Arithmetic	($p = 1$):	$\text{AM}_w(\mathbf{r}) = \sum_{i=1}^n w_i r_i$
Quadratic	($p = 2$):	$\text{QM}_w(\mathbf{r}) = \sqrt{\sum_{i=1}^n w_i r_i^2}$

Table 6.B.1: special cases of the p -power mean $M_p^w(\mathbf{r})$

the collection of real numbers $\mathbf{r} = r_1, \dots, r_n$ with respect to the convex weights $w = w_1, \dots, w_n$ satisfying $\sum_i w_i = 1$, is given by

$$M_p^w(\mathbf{r}) := \left(\sum_{i=1}^n w_i (r_i)^p \right)^{\frac{1}{p}}.$$

We omit the superscript as a shorthand for the uniform weighting $w_i = 1/N$. \square

Many standard means, such as those in Table 6.B.1, are special cases. It is well known that $M_p^w(\mathbf{r})$ is increasing in p , and strictly so if not all elements of \mathbf{r} are identical. In particular, $\text{QM}_w(a, b) > \text{GM}_w(a, b)$ for all $a \neq b$ and positive weights w . We now present the result.

[link to proof]

Proposition 6.B.1. Consider a PDG containing two (distinct) conditional Gaussian distributions on a variable Y , whose parameters can both depend on a variable X . Its inconsistency takes the form

$$\left\langle \left. \begin{array}{c} \xrightarrow[D]{(\infty)} X \\ \xrightarrow[f]{ } \boxed{\begin{matrix} \mu_1 \\ \sigma_1 \end{matrix}} \\ \xrightarrow[s]{ } \boxed{\begin{matrix} \mu_1 \\ \sigma_1 \end{matrix}} \\ \xrightarrow[t]{ } \boxed{\begin{matrix} \sigma_2 \\ \mu_2 \end{matrix}} \\ \xrightarrow[h]{ } \boxed{\begin{matrix} \sigma_2 \\ \mu_2 \end{matrix}} \end{array} \right| Y \xleftarrow[\mathcal{N}]{ } \right\rangle = \mathbb{E}_D \left[(\beta_1 + \beta_2) \log \frac{\text{QM}_{\hat{\beta}}(\sigma_1, \sigma_2)}{\text{GM}_{\hat{\beta}}(\sigma_1, \sigma_2)} + \frac{1}{2} \frac{\beta_1 \beta_2}{\beta_1 + \beta_2} \left(\frac{\mu_1 - \mu_2}{\text{QM}_{\hat{\beta}}(\sigma_1, \sigma_2)} \right)^2 \right]$$

$$= \frac{1}{2} \mathbb{E}_{x \sim D} \left[\frac{\beta_1 \beta_2}{2} \frac{(f(x) - h(x))^2}{\beta_2 s(x)^2 + \beta_1 t(x)^2} + \frac{\beta_1 + \beta_2}{2} \log \frac{\beta_2 s(x)^2 + \beta_1 t(x)^2}{\beta_1 + \beta_2} - \beta_2 \log s(x) - \beta_1 \log t(x) \right]$$
(6.7)

where $\hat{\beta} = (\frac{\beta_2}{\beta_1 + \beta_2}, \frac{\beta_1}{\beta_1 + \beta_2})$ represents the normalized and reversed vector of confidences $\beta = (\beta_1, \beta_2)$ for the two distributions, and $\mu_1 = f(X)$, $\mu_2 = g(X)$, $\sigma_1 = s(X)$, $\sigma_2 = t(X)$ are random variables over X .

The PDG on the left is semantically equivalent to (and in particular has the same inconsistency as) the PDG

$$\xrightarrow[D]{(\infty)} X \xrightarrow[\mathcal{N}(h(x), t(x))]{ } Y \xleftarrow[\mathcal{N}(f(x), s(x))]{ } .$$

This illustrates an orthogonal point: that PDGs handle composition of functions as one would expect, so that it is equivalent to model an entire process as a single arrow, or to break it into stages, ascribing an arrow to each stage, with one step of randomization.

As a bonus, Proposition 6.B.1 also gives a proof of the inequality of the weighted geometric and quadratic means.

Corollary 6.B.1.1. For all σ_1 and σ_2 , and all weight vectors β , $\text{QM}_{\hat{\beta}}(\sigma_1, \sigma_2) \geq \text{GM}_{\hat{\beta}}(\sigma_1, \sigma_2)$.

6.B.2 Full-Dataset ELBO and Bounds

We now present the promised multi-sample analogs from Section 6.6.1.

Proposition 6.B.2. The following analog of Proposition 6.6.2 for a whole dataset \mathcal{D} holds:

$$-\mathbb{E}_{\Pr_{\mathcal{D}}} \text{ELBO}_{p,e,d}(X) = \left\langle \xrightarrow[p]{Z} \xleftarrow[e]{X} \xrightarrow[d]{\Pr_{\mathcal{D}}} \right\rangle + H(\Pr_{\mathcal{D}}).$$

Propositions 6.3.2 and 6.B.2 then give us an analog of the visual bounds in the body of the main paper (Section 6.6.1) for many i.i.d. datapoints at once, with only a single application of the inequality:

$$\begin{aligned} -\log \Pr(\mathcal{D}) &= -\log \prod_{i=1}^m (\Pr(x^{(i)})) = -\frac{1}{m} \sum_{i=1}^m \log \Pr(x^{(i)}) = \\ &H(\Pr_{\mathcal{D}}) + \left\langle \xrightarrow[p]{Z} \xrightarrow[d]{X} \xleftarrow[e]{\Pr_{\mathcal{D}}} \right\rangle \leq \left\langle \xrightarrow[p]{Z} \xleftarrow[e]{X} \xrightarrow[d]{\Pr_{\mathcal{D}}} \right\rangle + H(\Pr_{\mathcal{D}}) \\ &= -\mathbb{E}_{\Pr_{\mathcal{D}}} \text{ELBO}_{p,e,d}(X) \end{aligned}$$

We also have the following formal statement of Proposition 6.6.3.

Proposition 6.B.3. The negative β -ELBO objective for a prior $p(X)$, encoder $e(Z \mid X)$, decoder $d(X \mid Z)$, at a sample x , is equal to the inconsistency of the corresponding PDG,

where the prior has confidence equal to β . That is,

$$-\beta\text{-ELBO}_{p,e,d}(x) = \left\langle \begin{array}{c} p \\ \xrightarrow{(\beta)} \\ Z \end{array} \xrightarrow[d]{\curvearrowright} X \xleftarrow{x} \right\rangle$$

As a specific case (i.e., effectively by setting $\beta_p := 0$), we get the reconstruction error as the inconsistency of an autoencoder (without a variational prior):

Corollary 6.B.3.1 (reconstruction error as inconsistency).

$$-\text{Rec}_{ed,d}(x) := \mathbb{E}_{z \sim e(Z|x)} I_{d(X|z)}(x) = \left\langle \begin{array}{c} d \\ \curvearrowright \\ Z \end{array} \xrightarrow{\curvearrowright} X \xleftarrow{x} \right\rangle$$

6.B.3 More Variants of Cross Entropy Results

First, we show that our cross entropy results hold for all γ , in the sense that γ contributes only a constant.

Proposition 6.B.4. *Given a model determining a probability distribution with mass function $p(X)$, and samples $\mathcal{D} = \{x_i\}_{i=1}^m$ determining an empirical distribution $\Pr_{\mathcal{D}}$, the following are equal, for all $\gamma \geq 0$:*

link to
proof

1. The average negative log likelihood $\ell(p; \mathcal{D}) = -\frac{1}{m} \sum_{i=1}^m \log p(x_i)$
2. The cross entropy of p relative to $\Pr_{\mathcal{D}}$
3. $\llbracket p \rrbracket_{\gamma}(\Pr_{\mathcal{D}}) + (1 + \gamma) H(\Pr_{\mathcal{D}})$
4. $\left\langle \xrightarrow[p]{\Pr_{\mathcal{D}}} X \xleftarrow[(\infty)]{} \right\rangle_{\gamma} + (1 + \gamma) H(\Pr_{\mathcal{D}})$

As promised, we now give the simultaneous generalization of the surprisal result (Proposition 6.3.1) to both multiple samples (like in Proposition 6.3.2) and partial observations (as in Proposition 6.3.3).

Proposition 6.B.5. *The average marginal negative log likelihood $\ell(p; x) := -\frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \log \sum_z p(x, z)$ is the inconsistency of the PDG containing p and the data distribution $\Pr_{\mathcal{D}}$, plus the entropy of the data distribution (which is constant in p). That is,*

$$\ell(p; \mathcal{D}) = \left\langle \begin{array}{c} Z \\ \swarrow^p \\ X \\ \xleftarrow[\infty]{} \end{array} \right\rangle + H(\Pr_{\mathcal{D}}).$$

[link to proof]

6.C PROOFS

Lemma 6.2.1. Suppose PDGs \mathbf{m} and \mathbf{m}' differ only in their edges (resp. \mathcal{A} and \mathcal{A}') and confidences (resp. β and β'). If $\mathcal{A} \subseteq \mathcal{A}'$ and $\beta_L \leq \beta'_L$ for all $L \in \mathcal{A}$, then $\langle\!\langle \mathbf{m} \rangle\!\rangle_\gamma \leq \langle\!\langle \mathbf{m}' \rangle\!\rangle_\gamma$ for all γ .

Proof. For every μ , adding more edges only adds non-negative terms to (6.1), while increasing β results in larger coefficients on the existing (non-negative) terms of (6.1). So for every fixed distribution μ , we have $\llbracket \mathbf{m} \rrbracket_\gamma(\mu) \leq \llbracket \mathbf{m}' \rrbracket_\gamma(\mu)$. So it must also be the case that the infimum over μ , so we find that $\langle\!\langle \mathbf{m} \rangle\!\rangle \leq \langle\!\langle \mathbf{m}' \rangle\!\rangle$. \square

Proposition 6.3.1. Consider a distribution $p(X)$. The inconsistency of the PDG comprising p and $X=x$ equals the surprisal $I_p[X=x]$. That is,

$$I_p[X=x] = \langle\!\langle \xrightarrow{p} [X] \xleftarrow{x} \rangle\!\rangle.$$

(Recall that $\langle\!\langle \mathbf{m} \rangle\!\rangle$ is the inconsistency of the PDG \mathbf{m} .)

Proof. Any distribution $\mu(X)$ that places mass on some $x' \neq x$ will have infinite KL divergence from the point mass on x . Thus, the only possibility for a finite consistency arises when $\mu = \delta_x$, and so

$$\langle\!\langle \xrightarrow{p} [X] \xleftarrow{x} \rangle\!\rangle = \llbracket \xrightarrow{p} [X] \xleftarrow{x} \rrbracket(\delta_x) = D(\delta_x \parallel p) = \log \frac{1}{p(x)} = I_p(x).$$

\square

Proposition 6.B.4 is a generalization of Proposition 6.3.2, so we prove them at the same time.

Proposition 6.3.2. If $p(X)$ is a probabilistic model of X , and $\mathcal{D} = \{x_i\}_{i=1}^m$ is a dataset with empirical distribution $\Pr_{\mathcal{D}}$, then $\text{CrossEntropy}(\Pr_{\mathcal{D}}, p) =$

$$\frac{1}{m} \sum_{i=1}^m I_p[X=x_i] = \left\langle\left\langle \xrightarrow{p} X \xleftarrow[\infty]{\Pr_{\mathcal{D}}} \right\rangle\right\rangle + H(\Pr_{\mathcal{D}}).$$

Proposition 6.B.4. Given a model determining a probability distribution with mass function $p(X)$, and samples $\mathcal{D} = \{x_i\}_{i=1}^m$ determining an empirical distribution $\Pr_{\mathcal{D}}$, the following are equal, for all $\gamma \geq 0$:

1. The average negative log likelihood $\ell(p; \mathcal{D}) = -\frac{1}{m} \sum_{i=1}^m \log p(x_i)$

2. The cross entropy of p relative to $\Pr_{\mathcal{D}}$

3. $\llbracket p \rrbracket_{\gamma}(\Pr_{\mathcal{D}}) + (1 + \gamma) H(\Pr_{\mathcal{D}})$

4. $\left\langle\left\langle \xrightarrow{p} X \xleftarrow[\infty]{\Pr_{\mathcal{D}}} \right\rangle\right\rangle_{\gamma} + (1 + \gamma) H(\Pr_{\mathcal{D}})$

Proof. The equality of 1 and 2 is standard. The equality of 3 and 4 can be seen by the fact that in the limit of infinite confidence on $\Pr_{\mathcal{D}}$, the optimal distribution must also equal $\Pr_{\mathcal{D}}$, so the least inconsistency is attained at this value. Finally it remains to show that the first two and last two are equal:

$$\begin{aligned} \llbracket p \rrbracket_{\gamma}(\Pr_{\mathcal{D}}) + (1 + \gamma) H(\Pr_{\mathcal{D}}) &= D(\Pr_{\mathcal{D}} \parallel p) - \gamma H(\Pr_{\mathcal{D}}) + (1 + \gamma) H(\Pr_{\mathcal{D}}) \\ &= D(\Pr_{\mathcal{D}} \parallel p) + H(\Pr_{\mathcal{D}}) \\ &= \mathbb{E}_{\Pr_{\mathcal{D}}} \left[\log \frac{\Pr_{\mathcal{D}}}{p} + \log \frac{1}{\Pr_{\mathcal{D}}} \right] = \mathbb{E}_{\Pr_{\mathcal{D}}} \left[\log \frac{1}{p} \right], \end{aligned}$$

which is the cross entropy, as desired. \square

Proposition 6.3.3. If $p(X, Z)$ is a joint distribution, then the information content of the partial observation $X = x$ is given by

$$I_p[X=x] = \left\langle \begin{array}{c} Z \\ \nwarrow^p \\ X \end{array} \xrightarrow{x} \right\rangle. \quad (6.2)$$

Proof. As before, all mass of μ must be on x for it to have a finite score. Thus it suffices to consider joint distributions of the form $\mu(X, Z) = \delta_x(X)\mu(Z)$. We have

$$\begin{aligned} \left\langle \begin{array}{c} Z \\ \nwarrow^p \\ X \end{array} \xrightarrow{x} \right\rangle &= \inf_{\mu(Z)} \left[\begin{array}{c} Z \\ \nwarrow^p \\ X \end{array} \xrightarrow{x} \right] (\delta_x(X)\mu(Z)) \\ &= \inf_{\mu(Z)} D(\delta_x(X)\mu(Z) \parallel p(X, Z)) \\ &= \inf_{\mu(Z)} \mathbb{E}_{z \sim \mu} \log \frac{\mu(z)}{p(x, z)} = \inf_{\mu(Z)} \mathbb{E}_{z \sim \mu} \log \frac{\mu(z)}{p(x, z)} \frac{p(x)}{p(x)} \\ &= \inf_{\mu(Z)} \mathbb{E}_{z \sim \mu} \left[\log \frac{\mu(z)}{p(z \mid x)} + \log \frac{1}{p(x)} \right] \\ &= \inf_{\mu(Z)} [D(\mu(Z) \parallel p(Z \mid x))] + \log \frac{1}{p(x)} \\ &= \log \frac{1}{p(x)} = I_p(x) \end{aligned} \quad [\text{Gibbs Inequality}]$$

□

Proposition 6.B.5. *The average marginal negative log likelihood $\ell(p; x) := -\frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \log \sum_z p(x, z)$ is the inconsistency of the PDG containing p and the data distribution $\Pr_{\mathcal{D}}$, plus the entropy of the data distribution (which is constant in p). That is,*

$$\ell(p; \mathcal{D}) = \left\langle \begin{array}{c} Z \\ \nwarrow^p \\ X \end{array} \xrightarrow{\Pr_{\mathcal{D}}} \right\rangle + H(\Pr_{\mathcal{D}}).$$

Proof. The same idea as in Proposition 6.3.3, but a little more complicated.

$$\left\langle \begin{array}{c} Z \\ \nwarrow^p \\ X \end{array} \xrightarrow{\Pr_{\mathcal{D}}!} \right\rangle = \inf_{\mu(Z|X)} \left[\begin{array}{c} Z \\ \nwarrow^p \\ X \end{array} \xrightarrow{\Pr_{\mathcal{D}}!} \right] (\Pr_{\mathcal{D}}(X)\mu(Z \mid X))$$

$$\begin{aligned}
&= \inf_{\mu(Z|X)} D\left(\Pr_{\mathcal{D}}(X)\mu(Z|X) \parallel p(X, Z)\right) \\
&= \inf_{\mu(Z|X)} \mathbb{E}_{\substack{x \sim \Pr_{\mathcal{D}} \\ z \sim \mu}} \log \frac{\mu(z|x) \Pr_{\mathcal{D}}(x)}{p(x, z)} \\
&= \frac{1}{|\mathcal{D}|} \inf_{\mu(Z|X)} \sum_{x \in \mathcal{D}} \mathbb{E}_{z \sim \mu(Z|x)} \log \frac{\mu(z|x) \Pr_{\mathcal{D}}(x)}{p(x, z)} \frac{p(x)}{p(x)} \\
&= \frac{1}{|\mathcal{D}|} \inf_{\mu(Z|X)} \sum_{x \in \mathcal{D}} \left[\mathbb{E}_{z \sim \mu} \left[\log \frac{\mu(z|x)}{p(z|x)} \right] + \log \frac{1}{p(x)} - \log \frac{1}{\Pr_{\mathcal{D}}(x)} \right] \\
&= \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \left[\inf_{\mu(Z|x)} \mathbb{E}_{z \sim \mu} \left[\log \frac{\mu(z|x)}{p(z|x)} \right] + \log \frac{1}{p(x)} - \log \frac{1}{\Pr_{\mathcal{D}}(x)} \right] \\
&= \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \left[\inf_{\mu(Z)} [D(\mu(Z) \parallel p(Z|x))] + \log \frac{1}{p(x)} \right] - H(\Pr_{\mathcal{D}}) \\
&= \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \log \frac{1}{p(x)} - H(\Pr_{\mathcal{D}}) \\
&= \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} I_p(x) - H(\Pr_{\mathcal{D}})
\end{aligned}$$

($= D(\Pr_{\mathcal{D}} \parallel p)$)

□

Proposition 6.3.4. *The inconsistency of the PDG comprising a probabilistic predictor $h(Y|X)$, and a high-confidence empirical distribution $\Pr_{\mathcal{D}}$ of a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^m$ equals the cross-entropy loss (minus the empirical uncertainty in Y given X , a constant depending only on \mathcal{D}). That is,*

$$\left\langle\!\! \left\langle \begin{array}{c} \Pr_{\mathcal{D}} \\ \diagdown \quad \diagup \\ X \xrightarrow[h]{} Y \end{array} \right\rangle\!\! \right\rangle = \frac{1}{m} \sum_{i=1}^m \log \frac{1}{h(y_i|x_i)} - H_{\Pr_{\mathcal{D}}}(Y|X).$$

Proof. $\Pr_{\mathcal{D}}$ has high confidence, it is the only joint distribution μ with finite score. Since f is the only other edge, the inconsistency is therefore

$$\mathbb{E}_{x \sim \Pr_{\mathcal{D}}} D\left(\Pr_{\mathcal{D}}(Y|x) \parallel f(Y|x)\right) = \mathbb{E}_{x,y \sim \Pr_{\mathcal{D}}} \left[\log \frac{\Pr_{\mathcal{D}}(y|x)}{f(y|x)} \right]$$

$$\begin{aligned}
&= \mathbb{E}_{x,y \sim \Pr_{\mathcal{D}}} \left[\log \frac{1}{f(y|x)} - \log \frac{1}{\Pr_{\mathcal{D}}(y|x)} \right] \\
&= \frac{1}{|\mathcal{D}|} \sum_{(x,y) \in \mathcal{D}} \left[\log \frac{1}{f(y|x)} \right] - H_{\Pr_{\mathcal{D}}}(Y|X)
\end{aligned}$$

□

Proposition 6.3.5. Consider functions $f, h : X \rightarrow Y$ from inputs to labels, where h is a predictor and f generates the true labels. The inconsistency of believing f and h (with any confidences), and a distribution $D(X)$ with confidence β , is β times the log accuracy of h . That is,

$$\left\langle\!\!\left\langle \begin{array}{c} D \xrightarrow{\beta} X \\ f \xrightarrow{(s)} Y \end{array} \right\rangle\!\!\right\rangle = -\beta \log \Pr_{x \sim D}(f(x) = h(x)) = \beta I_D[f = h]. \quad (6.3)$$

Proof. Because f is deterministic, for every x in the support of a joint distribution μ with finite score, we must have $\mu(Y|x) = \delta_{f(x)}$, since if μ were to place any non-zero mass $\mu(x,y) = \epsilon > 0$ on a point (x,y) with $y \neq f(x)$ results in an infinite contribution to the KL divergence

$$D(\mu(Y|x) \| \delta_{f(x)}) = \mathbb{E}_{x,y \sim \mu} \log \frac{\mu(y|x)}{\delta_{f(x)}} \geq \mu(y,x) \log \frac{\mu(x,y)}{\mu(x) \cdot \delta_{f(x)}(y)} = \epsilon \log \frac{\epsilon}{0} = \infty.$$

The same holds for h . Therefore, for any μ with a finite score, and x with $\mu(x) > 0$, we have $\delta_{f(x)} = \mu(Y|x) = \delta_{h(x)}$, meaning that we need only consider μ whose support is a subset of those points on which f and h agree. On all such points, the contribution to the score from the edges associated to f and h will be zero, since μ matches the conditional marginals exactly, and the total incompatibility of such a distribution μ is equal to the relative entropy $D(\mu \| D)$, scaled by the confidence β of the empirical distribution D .

So, among those distributions $\mu(X)$ supported on an event $E \subset \mathcal{V}(X)$, which minimizes is the relative entropy of $D(\mu \| D)$? It is well known that the conditional distribution $D | E \propto \delta_E(X)D(X) = \frac{1}{D(E)}\delta_E(X)D(X)$ satisfies this property uniquely (see, for instance, [38]). Let $f=h$ denote the event that f and h agree. Then we calculate

$$\begin{aligned}
\left\langle \xrightarrow[D]{(\beta)} X \xrightarrow[h]{f} Y \right\rangle &= \inf_{\substack{\mu(X) \text{ s.t.} \\ \text{supp}(\mu) \subseteq [f=h]}} \beta D(\mu(X) \| D(X)) \\
&= \beta D(D | [f=h] \| D) \\
&= \beta \mathbb{E}_{D|f=h} \log \frac{\delta_{f=h}(X)D(X)}{D(f=h) \cdot D(X)} \\
&= \beta \mathbb{E}_{D|f=h} \log \frac{1}{D(f=h)} \quad \left[\begin{array}{l} \text{since } \delta_{f=h}(x) = 1 \text{ for all } x \text{ that} \\ \text{contribute to the expectation} \end{array} \right] \\
&= -\beta \log D(f=h) \\
&= -\beta \log (\text{accuracy}_{f,D}(h)) \\
&= \beta I_D[f=h]. \quad \left[\begin{array}{l} \text{since } D(f=h) \text{ is a constant} \end{array} \right]
\end{aligned}$$

□

Proposition 6.B.1. Consider a PDG containing two (distinct) conditional Gaussian distributions on a variable Y , whose parameters can both depend on a variable X . Its inconsistency takes the form

$$\left\langle \xrightarrow[D]{(\infty)} X \xrightarrow[f]{s} \begin{matrix} \mu_1 \\ \sigma_1 \end{matrix} \xrightarrow[(\beta_1)]{\mathcal{N}} Y \xrightarrow[h]{t} \begin{matrix} \sigma_2 \\ \mu_2 \end{matrix} \xrightarrow[(\beta_2)]{\mathcal{N}} \right\rangle = \mathbb{E}_D \left[(\beta_1 + \beta_2) \log \frac{\text{QM}_{\hat{\beta}}(\sigma_1, \sigma_2)}{\text{GM}_{\hat{\beta}}(\sigma_1, \sigma_2)} + \frac{1}{2} \frac{\beta_1 \beta_2}{\beta_1 + \beta_2} \left(\frac{\mu_1 - \mu_2}{\text{QM}_{\hat{\beta}}(\sigma_1, \sigma_2)} \right)^2 \right] \tag{6.7}$$

$$= \frac{1}{2} \mathbb{E}_{x \sim D} \left[\frac{\beta_1 \beta_2}{2} \frac{(f(x) - h(x))^2}{\beta_2 s(x)^2 + \beta_1 t(x)^2} + \frac{\beta_1 + \beta_2}{2} \log \frac{\beta_2 s(x)^2 + \beta_1 t(x)^2}{\beta_1 + \beta_2} - \beta_2 \log s(x) - \beta_1 \log t(x) \right]$$

where $\hat{\beta} = (\frac{\beta_2}{\beta_1 + \beta_2}, \frac{\beta_1}{\beta_1 + \beta_2})$ represents the normalized and reversed vector of confidences $\beta = (\beta_1, \beta_2)$ for the two distributions, and $\mu_1 = f(X)$, $\mu_2 = g(X)$, $\sigma_1 = s(X)$, $\sigma_2 = t(X)$ are random variables over X .

Proof. Let m denote the PDG in question. Since D has high confidence, we know any joint distribution μ with a finite score must have $\mu(X) = D(X)$. Thus,

$$\begin{aligned} \langle\langle m \rangle\rangle &= \inf_{\mu} \mathbb{E}_{x \sim D} \mathbb{E}_{y \sim \mu|x} \left[\beta_1 \log \frac{\mu(y|x)}{\mathcal{N}(y|f(x), s(x))} + \beta_2 \log \frac{\mu(y|x)}{\mathcal{N}(y|h(x), t(x))} \right] \\ &= \inf_{\mu} \mathbb{E}_{x \sim D} \mathbb{E}_{y \sim \mu|x} \left[\beta_1 \log \frac{\mu(y|x)}{\frac{1}{s(x)\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{y-f(x)}{s(x)}\right)^2\right)} + \beta_2 \log \frac{\mu(y|x)}{\frac{1}{t(x)\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{y-h(x)}{t(x)}\right)^2\right)} \right] \\ &= \inf_{\mu} \mathbb{E}_{x \sim D} \mathbb{E}_{y \sim \mu|x} \left[\log \mu(y|x)^{\beta_1 + \beta_2} + \frac{\beta_1}{2} \left(\frac{y-f(x)}{s(x)}\right)^2 + \frac{\beta_2}{2} \left(\frac{y-h(x)}{t(x)}\right)^2 + \beta_1 \log(s(x)\sqrt{2\pi}) + \beta_2 \log(t(x)\sqrt{2\pi}) \right]. \end{aligned} \tag{6.8}$$

At this point, we would like make use of the fact that the sum of two parabolas is itself a parabola, so as to combine the two terms on the top right of the previous equation. Concretely, we claim (?? 2, whose proof is at the end of the present one), that if we define

$$g(x) := \frac{\beta_1 t(x)^2 f(x) + \beta_2 s(x)^2 h(x)}{\beta_1 t(x)^2 + \beta_2 s(x)^2} \quad \text{and} \quad \tilde{\sigma}(x) := \frac{s(x)t(x)}{\sqrt{\beta_1 t(x)^2 + \beta_2 s(x)^2}},$$

then

$$\frac{\beta_1}{s(x)^2} (y - f)^2 + \frac{\beta_2}{t(x)^2} (y - h)^2 = \left(\frac{y - g}{\tilde{\sigma}} \right)^2 + \frac{\beta_1 \beta_2}{\beta_1 t(x)^2 + \beta_2 s(x)^2} (f - h)^2.$$

Applying this to (6.8) leaves us with:

$$\langle\!\langle m \rangle\!\rangle = \inf_{\mu} \mathbb{E}_{x \sim D} \mathbb{E}_{y \sim \mu|x} \left[\begin{array}{ll} \log \mu(y|x)^{\beta_1 + \beta_2} & + \frac{1}{2\tilde{\sigma}(x)^2} (y - g(x))^2 + \frac{1}{2} \frac{\beta_1 \beta_2}{\beta_1 t(x)^2 + \beta_2 s(x)^2} (f(x) - h(x))^2 \\ & + \beta_1 \log(s(x)\sqrt{2\pi}) + \beta_2 \log(t(x)\sqrt{2\pi}) \end{array} \right]$$

Pulling the term on the top right, which does not depend on Y , out of the expectation, and folding the rest of the terms back inside the logarithm (which in particular means first replacing the top middle term φ by $-\log(\exp(-\varphi))$), we obtain $\langle\!\langle m \rangle\!\rangle =$

$$\mathbb{E}_{x \sim D} \left[\begin{array}{l} \inf_{\mu(Y)} \mathbb{E}_{y \sim \mu} \left[\log \mu(y)^{\beta_1 + \beta_2} - \log \left(\frac{1}{\sqrt{2\pi}^{\beta_1 + \beta_2} s(x)^{\beta_1} t(x)^{\beta_2}} \exp \left\{ -\frac{1}{2} \left(\frac{y - g(x)}{\tilde{\sigma}(x)} \right)^2 \right\} \right) \right] \\ + \frac{1}{2} \frac{\beta_1 \beta_2}{\beta_1 t(x)^2 + \beta_2 s(x)^2} (f(x) - h(x))^2 \end{array} \right].$$

To simplify the presentation, let ψ be the term on the top right, and ξ be the term on the bottom. More explicitly, define

$$\begin{aligned} \psi(x, y) &:= \frac{1}{2} \frac{1}{\sqrt{2\pi}^{\beta_1 + \beta_2} s(x)^{\beta_1} t(x)^{\beta_2}} \exp \left\{ -\frac{1}{2} \left(\frac{y - g(x)}{\tilde{\sigma}(x)} \right)^2 \right\}, \\ \text{and } \xi(x) &:= \frac{\beta_1 \beta_2}{\beta_1 t(x)^2 + \beta_2 s(x)^2} (f(x) - h(x))^2, \end{aligned}$$

which lets us write the previous expression for $\langle\!\langle m \rangle\!\rangle$ as

$$\langle\!\langle m \rangle\!\rangle = \mathbb{E}_{x \sim D} \left[\inf_{\mu(Y)} \mathbb{E}_{y \sim \mu} [\log \mu(y)^{\beta_1 + \beta_2} - \log \psi(x, y)] + \xi(x) \right]. \quad (6.9)$$

Also, let $\hat{\beta}_1 := \frac{\beta_1}{\beta_1 + \beta_2}$, and $\hat{\beta}_2 := \frac{\beta_2}{\beta_1 + \beta_2}$. For reasons that will soon become clear, we are actually interested in $\psi^{\frac{1}{\beta_1 + \beta_2}}$, which we compute as

$$\begin{aligned} \psi(x, y)^{\frac{1}{\beta_1 + \beta_2}} &= (2\pi)^{-\frac{1}{2}} s(x)^{\left(\frac{-\beta_1}{\beta_1 + \beta_2}\right)} t(x)^{\left(\frac{-\beta_2}{\beta_1 + \beta_2}\right)} \exp \left\{ -\frac{1}{2} \left(\frac{y - g(x)}{\tilde{\sigma}(x)} \right)^2 \right\}^{\frac{1}{\beta_1 + \beta_2}} \\ &= \frac{1}{\sqrt{2\pi} s(x)^{\hat{\beta}_1} t(x)^{\hat{\beta}_2}} \exp \left\{ \frac{-1}{2(\beta_1 + \beta_2)} \left(\frac{y - g(x)}{\tilde{\sigma}(x)} \right)^2 \right\}. \end{aligned}$$

Recall that the Gaussian density $\mathcal{N}(y|g(x), \tilde{\sigma}(x)\sqrt{\beta_1 + \beta_2})$ of mean $g(x)$ and variance $\tilde{\sigma}(x)^2(\beta_1 + \beta_2)$ is given by

$$\mathcal{N}(y|g(x), \tilde{\sigma}(x)\sqrt{\beta_1 + \beta_2}) = \frac{1}{\sqrt{2\pi} \tilde{\sigma}(x)\sqrt{\beta_1 + \beta_2}} \exp \left\{ \frac{-1}{2(\beta_1 + \beta_2)} \left(\frac{y - g(x)}{\tilde{\sigma}(x)} \right)^2 \right\},$$

which is quite similar, and has an identical dependence on y . To facilitate converting one to the other, we explicitly compute the ratio:

$$\begin{aligned}
\frac{\psi(x, y)^{\frac{1}{\beta_1 + \beta_2}}}{\mathcal{N}(y | g(x), \tilde{\sigma}(x)\sqrt{\beta_1 + \beta_2})} &= \frac{\tilde{\sigma}\sqrt{2\pi(\beta_1 + \beta_2)}}{\sqrt{2\pi} s(x)^{\hat{\beta}_1} t(x)^{\hat{\beta}_2}} = \frac{\tilde{\sigma}\sqrt{\beta_1 + \beta_2}}{s(x)^{\hat{\beta}_1} t(x)^{\hat{\beta}_2}} \\
&= \left(\frac{s(x)t(x)}{\sqrt{\beta_1 t(x)^2 + \beta_2 s(x)^2}} \right) \frac{\sqrt{\beta_1 + \beta_2}}{s(x)^{\hat{\beta}_1} t(x)^{\hat{\beta}_2}} \quad [\text{expand defn of } \tilde{\sigma}(x)] \\
&= s(x)^{1-\hat{\beta}_1} t(x)^{1-\hat{\beta}_2} \sqrt{\frac{\beta_1 + \beta_2}{\beta_1 t(x)^2 + \beta_2 s(x)^2}} \\
&= s(x)^{1-\hat{\beta}_1} t(x)^{1-\hat{\beta}_2} \sqrt{\frac{1}{\hat{\beta}_1 t(x)^2 + \hat{\beta}_2 s(x)^2}} \quad [\text{defn of } \hat{\beta}_1, \hat{\beta}_2] \\
&= \frac{s(x)^{\hat{\beta}_2} t(x)^{\hat{\beta}_1}}{\sqrt{\hat{\beta}_1 t(x)^2 + \hat{\beta}_2 s(x)^2}} \quad [\text{since } \hat{\beta}_1 + \hat{\beta}_2 = 1]
\end{aligned}$$

Now, picking up from where we left off in (6.9), we have

$$\begin{aligned}
\langle\!\langle m \rangle\!\rangle &= \mathbb{E}_{x \sim D} \left[\inf_{\mu(Y)} \mathbb{E}_{y \sim \mu} [\log \mu(y)^{\beta_1 + \beta_2} - \log \psi(x, y)] + \xi(x) \right] \\
&= \mathbb{E}_{x \sim D} \left[\inf_{\mu(Y)} \mathbb{E}_{y \sim \mu} \left[\log \frac{\mu(y)^{\beta_1 + \beta_2}}{\psi(x, y)^{\frac{\beta_1 + \beta_2}{\beta_1 + \beta_2}}} \right] + \xi(x) \right] \\
&= \mathbb{E}_{x \sim D} \left[\inf_{\mu(Y)} \mathbb{E}_{y \sim \mu} \left[(\beta_1 + \beta_2) \log \frac{\mu(y)}{\psi(x, y)^{\frac{1}{\beta_1 + \beta_2}}} \right] + \xi(x) \right] \\
&= \mathbb{E}_{x \sim D} \left[\inf_{\mu(Y)} \mathbb{E}_{y \sim \mu} \left[(\beta_1 + \beta_2) \log \frac{\mu(y)}{\mathcal{N}(y | g(x), \tilde{\sigma}(x)\sqrt{\beta_1 + \beta_2}) \frac{s(x)^{\hat{\beta}_2} t(x)^{\hat{\beta}_1}}{\sqrt{\hat{\beta}_1 t(x)^2 + \hat{\beta}_2 s(x)^2}}} \right] + \xi(x) \right] \\
&= \mathbb{E}_{x \sim D} \left[\inf_{\mu(Y)} \mathbb{E}_{y \sim \mu} \left[(\beta_1 + \beta_2) \log \frac{\mu(y)}{\mathcal{N}(y | g(x), \tilde{\sigma}(x)\sqrt{\beta_1 + \beta_2})} \right] + (\beta_1 + \beta_2) \log \frac{\sqrt{\hat{\beta}_1 t(x)^2 + \hat{\beta}_2 s(x)^2}}{s(x)^{\hat{\beta}_2} t(x)^{\hat{\beta}_1}}
\right]
\end{aligned}$$

but now the entire left term is the infimum of a KL divergence, which is non-negative and equal to zero iff $\mu(y) = \mathcal{N}(y | g(x), \tilde{\sigma}(x)\sqrt{\beta_1 + \beta_2})$. So the infimum on the left is equal to zero.

$$\langle\!\langle m \rangle\!\rangle = \mathbb{E}_{x \sim D} \left[(\beta_1 + \beta_2) \log \frac{\sqrt{\hat{\beta}_1 t(x)^2 + \hat{\beta}_2 s(x)^2}}{s(x)^{\hat{\beta}_2} t(x)^{\hat{\beta}_1}} + \xi(x) \right] \quad (6.10)$$

$$\begin{aligned}
&= \mathbb{E}_{x \sim D} \left[(\beta_1 + \beta_2) \log \sqrt{\hat{\beta}_1 t(x)^2 + \hat{\beta}_2 s(x)^2} - (\beta_1 + \beta_2) \log \left(s(x)^{\hat{\beta}_2} t(x)^{\hat{\beta}_1} \right) + \xi(x) \right] \\
&= \mathbb{E}_{x \sim D} \left[(\beta_1 + \beta_2) \log \sqrt{\hat{\beta}_1 t(x)^2 + \hat{\beta}_2 s(x)^2} \begin{matrix} -\beta_2 \log s(x) \\ -\beta_1 \log t(x) \end{matrix} + \xi(x) \right] \\
&= \mathbb{E}_{x \sim D} \left[(\beta_1 + \beta_2) \log \sqrt{\frac{\beta_1 t(x)^2 + \beta_2 s(x)^2}{\beta_1 + \beta_2}} \begin{matrix} -\beta_2 \log s(x) \\ -\beta_1 \log t(x) \end{matrix} + \frac{1}{2} \frac{\beta_1 \beta_2}{\beta_1 t(x)^2 + \beta_2 s(x)^2} (f(x) - h(x))^2 \right]
\end{aligned} \tag{6.11}$$

Whew! Pulling the square root of the logarithm proves complex second half of the proposition. Now, we massage it into into a (slightly) more readable form.

To start, write σ_1 (the random variable) in place of $s(x)$ and σ_2 in place of $t(x)$. Let $\hat{\beta}$ without the subscript denote the vector $(\hat{\beta}_2, \hat{\beta}_1) = (\frac{\beta_2}{\beta_1 + \beta_2}, \frac{\beta_1}{\beta_1 + \beta_2})$, which we will use for weighted means. The $\hat{\beta}$ -weighted arithmetic, geometric ($p = 0$), and quadratic ($p = 2$) means of σ_1 and σ_2 are:

$$\text{GM}_{\hat{\beta}}(\sigma_1, \sigma_2) = (\sigma_1)^{\hat{\beta}_2} (\sigma_2)^{\hat{\beta}_1} \quad \text{and} \quad \text{QM}_{\hat{\beta}}(\sigma_1, \sigma_2) = \sqrt{\hat{\beta}_2 \sigma_1^2 + \hat{\beta}_1 \sigma_2^2}.$$

So, now we can write $\xi(x)$ as

$$\begin{aligned}
\frac{1}{2} \frac{\beta_1 \beta_2}{\beta_1 t(x)^2 + \beta_2 s(x)^2} (f(x) - h(x))^2 &= \frac{1}{2} \frac{\beta_1 \beta_2}{\beta_1 + \beta_2} \frac{\beta_1 + \beta_2}{\beta_1 t(x)^2 + \beta_2 s(x)^2} (f(x) - h(x))^2 \\
&= \frac{1}{2} \frac{\beta_1 \beta_2}{\beta_1 + \beta_2} \left(\frac{1}{\text{QM}_{\hat{\beta}}(\sigma_1, \sigma_2)} \right)^2 (f(x) - h(x))^2 \\
&= \frac{1}{2} \frac{\beta_1 \beta_2}{\beta_1 + \beta_2} \left(\frac{\mu_1 - \mu_2}{\text{QM}_{\hat{\beta}}(\sigma_1, \sigma_2)} \right)^2;
\end{aligned}$$

in the last step, we have replaced $f(x)$ and $g(x)$ with their respective random variables μ_1 and μ_2 . As a result, (6.10) can be written as

$$\langle\langle m \rangle\rangle = \mathbb{E}_D \left[(\beta_1 + \beta_2) \log \frac{\text{QM}_{\hat{\beta}}(\sigma_1, \sigma_2)}{\text{GM}_{\hat{\beta}}(\sigma_1, \sigma_2)} + \frac{1}{2} \frac{\beta_1 \beta_2}{\beta_1 + \beta_2} \left(\frac{\mu_1 - \mu_2}{\text{QM}_{\hat{\beta}}(\sigma_1, \sigma_2)} \right)^2 \right]$$

... which is perhaps more comprehensible, and proves the first half of our proposition. \square

Claim 2. *The sum of two functions that are unshifted parabolas as functions of y (i.e., both functions are of the form $k(y - a)^2$), is itself a (possibly shifted) parabola of y (and of the form $k'(y - a') + b'$). More concretely, and adapted to our usage above, the following algebraic relation holds:*

$$\frac{\beta_1}{\sigma_1^2}(y - f)^2 + \frac{\beta_2}{\sigma_2^2}(y - h)^2 = \left(\frac{y - g}{\tilde{\sigma}}\right)^2 + \frac{\beta_1\beta_2}{\beta_1\sigma_2^2 + \beta_2\sigma_1^2}(f - h)^2,$$

where

$$g := \frac{\beta_1\sigma_2^2 f + \beta_2\sigma_1^2 h}{\beta_1\sigma_2^2 + \beta_2\sigma_1^2} \quad \text{and} \quad \tilde{\sigma} := \left(\frac{\beta_1}{\sigma_1^2} + \frac{\beta_2}{\sigma_2^2}\right)^{-1/2} = \frac{\sigma_1\sigma_2}{\sqrt{\beta_1\sigma_2^2 + \beta_2\sigma_1^2}}.$$

Proof. Expand terms and complete the square. Starting from the left hand side, we have

$$\begin{aligned} & \frac{\beta_1}{\sigma_1^2}(y - f)^2 + \frac{\beta_2}{\sigma_2^2}(y - h)^2 \\ &= \frac{\beta_1}{\sigma_1^2}(y^2 - 2yf + f^2) + \frac{\beta_2}{\sigma_2^2}(y^2 - 2yh + h^2) \\ &= \left(\frac{\beta_1}{\sigma_1^2} + \frac{\beta_2}{\sigma_2^2}\right)y^2 - 2\left(\frac{\beta_1 f}{\sigma_1^2} + \frac{\beta_2 h}{\sigma_2^2}\right)y + \left(\frac{\beta_1 f^2}{\sigma_1^2} + \frac{\beta_2 h^2}{\sigma_2^2}\right) \\ &= \left(\frac{\beta_1}{\sigma_1^2} + \frac{\beta_2}{\sigma_2^2}\right)y^2 - 2\left(\frac{\beta_1 f}{\sigma_1^2} + \frac{\beta_2 h}{\sigma_2^2}\right)y + \left(\frac{\beta_1 f^2}{\sigma_1^2} + \frac{\beta_2 h^2}{\sigma_2^2} - \frac{\beta_1\beta_2(f - h)^2}{\beta_1\sigma_2^2 + \beta_2\sigma_1^2}\right) + \frac{\beta_1\beta_2(f - h)^2}{\beta_1\sigma_2^2 + \beta_2\sigma_1^2} \end{aligned} \tag{6.12}$$

where in the last step we added and removed the same term (i.e., the completion of the square, although at this point it may still be unclear that this quantity is the one we want). The third parenthesized quantity needs the most work. Isolating it and getting a common denominator gives us:

$$\begin{aligned} & \frac{\beta_1 f^2}{\sigma_1^2} + \frac{\beta_2 h^2}{\sigma_2^2} - \frac{\beta_1\beta_2(f - h)^2}{\beta_1\sigma_2^2 + \beta_2\sigma_1^2} \\ &= \frac{\beta_1 f^2(\beta_1\sigma_2^2 + \beta_2\sigma_1^2)\sigma_2^2}{\sigma_1^2(\beta_1\sigma_2^2 + \beta_2\sigma_1^2)\sigma_2^2} + \frac{\beta_2 h^2(\beta_1\sigma_2^2 + \beta_2\sigma_1^2)\sigma_1^2}{\sigma_2^2(\beta_1\sigma_2^2 + \beta_2\sigma_1^2)\sigma_1^2} - \frac{\beta_1\beta_2(f^2 - 2fh + h^2)(\sigma_1^2\sigma_2^2)}{(\beta_1\sigma_2^2 + \beta_2\sigma_1^2)(\sigma_1^2\sigma_2^2)} \\ &= \frac{\beta_1^2\sigma_2^4 f^2 + \cancel{\beta_1\beta_2\sigma_2^2\sigma_1^2 f^2} + \cancel{\beta_1\beta_2\sigma_1^2\sigma_2^2 h^2} + \beta_2^2\sigma_1^4 h^2 - \cancel{\beta_1\beta_2\sigma_1^2\sigma_2^2 f^2} + 2\beta_1\beta_2\sigma_1^2\sigma_2^2 fh - \cancel{\beta_1\beta_2\sigma_1^2\sigma_2^2 h^2}}{(\beta_1\sigma_2^2 + \beta_2\sigma_1^2)(\sigma_1^2\sigma_2^2)} \end{aligned}$$

$$= \frac{\beta_1^2 \sigma_2^4 f^2 + \beta_2^2 \sigma_1^4 h^2 + 2\beta_1 \beta_2 \sigma_1^2 \sigma_2^2 f h}{(\beta_1 \sigma_2^2 + \beta_2 \sigma_1^2)(\sigma_1^2 \sigma_2^2)}.$$

Substituting this expression into the third term of (6.12), while simultaneously computing common denominators for the first and second terms, yields

$$\left(\frac{\beta_1 \sigma_2^2 + \beta_2 \sigma_1^2}{\sigma_1^2 \sigma_2^2} \right) y^2 - 2 \left(\frac{\beta_1 \sigma_2^2 f + \beta_2 \sigma_1^2 h}{\sigma_1^2 \sigma_2^2} \right) y + \frac{\beta_1^2 \sigma_2^4 f^2 + \beta_2^2 \sigma_1^4 h^2 + 2\beta_1 \beta_2 \sigma_1^2 \sigma_2^2 f h}{(\beta_1 \sigma_2^2 + \beta_2 \sigma_1^2)(\sigma_1^2 \sigma_2^2)} + \frac{\beta_1 \beta_2 (f - h)^2}{\beta_1 \sigma_2^2 + \beta_2 \sigma_1^2}. \quad (6.13)$$

On the other hand, using the definitions of g and $\tilde{\sigma}$, we compute:

$$\begin{aligned} & \left(\frac{y - g}{\tilde{\sigma}} \right)^2 \\ &= \left(\frac{\beta_1 \sigma_2^2 + \beta_2 \sigma_1^2}{\sigma_1^2 \sigma_2^2} \right) \left(y - \frac{\beta_1 \sigma_2^2 f + \beta_2 \sigma_1^2 h}{\beta_1 \sigma_2^2 + \beta_2 \sigma_1^2} \right)^2 \\ &= \left(\frac{\beta_1 \sigma_2^2 + \beta_2 \sigma_1^2}{\sigma_1^2 \sigma_2^2} \right) \left(y^2 - 2y \frac{\beta_1 \sigma_2^2 f + \beta_2 \sigma_1^2 h}{\beta_1 \sigma_2^2 + \beta_2 \sigma_1^2} + \frac{\beta_1^2 \sigma_2^4 f^2 + \beta_2^2 \sigma_1^4 h^2 + 2\beta_1 \beta_2 \sigma_1^2 \sigma_2^2 f h}{(\beta_1 \sigma_2^2 + \beta_2 \sigma_1^2)^2} \right) \\ &= \left(\frac{\beta_1 \sigma_2^2 + \beta_2 \sigma_1^2}{\sigma_1^2 \sigma_2^2} \right) y^2 - 2 \left(\frac{\beta_1 \sigma_2^2 f + \beta_2 \sigma_1^2 h}{\sigma_1^2 \sigma_2^2} \right) y + \frac{\beta_1^2 \sigma_2^4 f^2 + \beta_2^2 \sigma_1^4 h^2 + 2\beta_1 \beta_2 \sigma_1^2 \sigma_2^2 f h}{(\beta_1 \sigma_2^2 + \beta_2 \sigma_1^2)(\sigma_1^2 \sigma_2^2)} \end{aligned}$$

... which is precisely the first 3 terms of (6.13). Putting it all together, we have shown that

$$\frac{\beta_1}{\sigma_1^2} (y - f)^2 + \frac{\beta_2}{\sigma_2^2} (y - h)^2 = \left(\frac{y - g}{\tilde{\sigma}} \right)^2 + \frac{\beta_1 \beta_2 (f - h)^2}{\beta_1 \sigma_2^2 + \beta_2 \sigma_1^2}$$

as desired. □

Proposition 6.3.6.

$$\left\langle \left. \begin{array}{ccc} D & \xrightarrow{f} & \mu_f \\ (\infty) & \searrow & \downarrow \mathcal{N}_1 \\ X & & Y \end{array} \right| \begin{array}{ccc} h & \nearrow & \mu_h \\ & & \mathcal{N}_1 \end{array} \right\rangle = \frac{1}{2} \mathbb{E}_D |f(X) - h(X)|^2 =: \text{MSE}_D(f, h),$$

where $\mathcal{N}_1(Y|\mu)$ is a unit Gaussian on Y with mean μ .

Proof. An immediate corollary of Proposition 6.B.1; simply set $s(x) = t(x) = \beta_1 = \beta_2 = 1$ □

Lemma 6.5.2. *The inconsistency $D_{(r,s)}^{\text{PDG}}(p\|q)$ of a PDG comprising $p(X)$ with confidence r and $q(X)$ with confidence s is given in closed form by*

$$\left\langle \left. \frac{p}{(r)} \rightarrow X \leftarrow \frac{q}{(s)} \right. \right\rangle = -(r+s) \log \sum_x \left(p(x)^r q(x)^s \right)^{\frac{1}{r+s}}.$$

Proof.

$$\begin{aligned} \left\langle \left. \frac{p}{(\beta:r)} \rightarrow X \leftarrow \frac{q}{(\beta:s)} \right. \right\rangle &= \inf_{\mu} \mathbb{E}_{\mu} \log \frac{\mu(x)^{r+s}}{p(x)^r q(x)^s} \\ &= (r+s) \inf_{\mu} \mathbb{E}_{\mu} \left[\log \frac{\mu(x)}{(p(x)^r q(x)^s)^{\frac{1}{r+s}}} \cdot \frac{Z}{Z} \right] \\ &= \inf_{\mu} (r+s) D\left(\mu \middle\| \frac{1}{Z} p^{\frac{r}{r+s}} q^{\frac{s}{r+s}} \right) - (r+s) \log Z \end{aligned}$$

where $Z := (\sum_x p(x)^r q(x)^s)^{\frac{1}{r+s}}$ is the constant required to normalize the denominator as a distribution. The first term is now a relative entropy, and the only usage of μ . $D(\mu \| \dots)$ achieves its minimum of zero when μ is the second distribution, so our formula becomes

$$\begin{aligned} &= -(r+s) \log Z \\ &= -(r+s) \log \sum_x \left(p(x)^r q(x)^s \right)^{\frac{1}{r+s}} \quad \text{as promised.} \quad \square \end{aligned}$$

Proposition 6.4.1. *Suppose you have a parameterized model $p(Y|\Theta)$, a prior $q(\Theta)$, and a trusted distribution $D(Y)$. The inconsistency of also believing $\Theta = \theta$ is the cross entropy loss, plus the regularizer: $\log \frac{1}{q(\theta)}$ times your confidence in q . That is,*

$$\left\langle \left. \begin{array}{c} \xrightarrow{q} \\ \xrightarrow{(\beta)} \\ \xrightarrow{\theta} \end{array} \Theta \xrightarrow{p} Y \right. \right\rangle = \mathbb{E}_{y \sim D} \log \frac{1}{p(y|\theta)} + \beta \log \frac{1}{q(\theta)} - H(D) \quad (6.4)$$

Proof. This is another case where there's only one joint distribution $\mu(\Theta, Y)$ that gets a finite score. We must have $\mu(Y) = D(Y)$ since D has infinite confidence,

which uniquely extends to the distribution $\mu(\Theta, Y) = D(Y)\delta_\theta(\Theta)$ for which deterministically sets $\Theta = \theta$.

The cpds corresponding to the edges labeled θ and D , then, are satisfied by this μ and contribute nothing to the score. So the two relevant edges that contribute incompatibility with this distribution are p and q . Letting \mathcal{M} denote the PDG in question, we compute:

$$\begin{aligned}\langle\langle \mathcal{M} \rangle\rangle &= \mathbb{E}_\mu \left[\log \frac{\mu(Y|\Theta)}{p(Y|\Theta)} + \beta \log \frac{\mu(\Theta)}{q(\Theta)} \right] \\ &= \mathbb{E}_{y \sim D} \left[\log \frac{D(y)}{p(y|\theta)} + \beta \log \frac{1}{q(\theta)} \right] \\ &= \mathbb{E}_{y \sim D} \left[\log \frac{1}{p(y|\theta)} + \beta \log \frac{1}{q(\theta)} + \log D(y) \right] \\ &= \mathbb{E}_{y \sim D} \left[\log \frac{1}{p(y|\theta)} \right] + \beta \log \frac{1}{q(\theta)} - H(D)\end{aligned}$$

as desired. \square

Proposition 6.6.1. *The negative ELBO of x is the inconsistency of the PDG containing p, q , and $X=x$, with high confidence in q . That is,*

$$-\text{ELBO}_{p,q}(x) = \left\langle\left\langle \begin{array}{c} q \xrightarrow{(\infty)} Z \\ \nwarrow^p \quad \nearrow \\ X \xleftarrow{x} \end{array} \right\rangle\right\rangle.$$

Proof. Every distribution that does marginalize to $q(Z)$ or places any mass on $x' \neq x$ will have infinite score. Thus the only distribution that could have a finite score is $\mu(X, Z)$. Thus,

$$\begin{aligned}\left\langle\left\langle \begin{array}{c} q \xrightarrow{(\infty)} Z \\ \nwarrow^p \quad \nearrow \\ X \xleftarrow{x} \end{array} \right\rangle\right\rangle &= \inf_\mu \left[\begin{array}{c} q \xrightarrow{(\infty)} Z \\ \nwarrow^p \quad \nearrow \\ X \xleftarrow{x} \end{array} \right](\mu) \\ &= \left[\begin{array}{c} q \xrightarrow{(\infty)} Z \\ \nwarrow^p \quad \nearrow \\ X \xleftarrow{x} \end{array} \right](\delta_x(X)q(Z))\end{aligned}$$

$$= \mathbb{E}_{\substack{x' \sim \delta_x \\ z \sim q}} \log \frac{\delta_x(x')q(z)}{p(x', z)} = - \mathbb{E}_{z \sim q} \frac{p(x, z)}{q(z)} = -\text{ELBO}_{p,q}(x).$$

□

We prove both Proposition 6.6.2 and Proposition 6.B.2 at the same time.

Proposition 6.6.2. *The VAE loss of a sample x is the inconsistency of the PDG comprising the encoder e (with high confidence, as it defines the encoding), decoder d , prior p , and x . That is,*

$$-\text{ELBO}_{p,e,d}(x) = \left\langle \xrightarrow{p} Z \xrightarrow[d]{e} X \xleftarrow{x} \right\rangle.$$

Proposition 6.B.2. *The following analog of Proposition 6.6.2 for a whole dataset \mathcal{D} holds:*

$$-\mathbb{E}_{\Pr_{\mathcal{D}}} \text{ELBO}_{p,e,d}(X) = \left\langle \xrightarrow{p} Z \xrightarrow[d]{e} X \xleftarrow[\Pr_{\mathcal{D}}]{(\infty)} \right\rangle + H(\Pr_{\mathcal{D}}).$$

Proof. The two proofs are similar. For Proposition 6.6.2, the optimal distribution must be $\delta_x(X)e(Z \mid X)$, and for Proposition 6.B.2, it must be $\Pr_{\mathcal{D}}(X)e(Z \mid X)$, because e and the data both have infinite confidence, so any other distribution gets an infinite score. At the same time, d and p define a joint distribution, so the inconsistency in question becomes

$$D\left(\delta_x(X)e(Z \mid X) \parallel p(Z)d(X \mid Z)\right) = \mathbb{E}_{z \sim e|x} \left[\log \frac{p(z)d(x \mid z)}{e(z \mid x)} \right] = \text{ELBO}_{p,e,d}(x)$$

in the first, case, and

$$D\left(\Pr_{\mathcal{D}}(X)e(Z \mid X) \parallel p(Z)d(X \mid Z)\right) = \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \mathbb{E}_{z \sim e|x} \left[\log \frac{p(z)d(x \mid z)}{e(z \mid x)} + \log \frac{1}{\Pr_{\mathcal{D}}(x)} \right]$$

$$= \text{ELBO}_{p,e,d}(x) - H(\Pr_{\mathcal{D}})$$

in the second. \square

Now, we formally state and prove the more general result for β -VAEs.

Proposition 6.B.3. *The negative β -ELBO objective for a prior $p(X)$, encoder $e(Z \mid X)$, decoder $d(X \mid Z)$, at a sample x , is equal to the inconsistency of the corresponding PDG, where the prior has confidence equal to β . That is,*

$$-\beta\text{-ELBO}_{p,e,d}(x) = \left\langle \left. \begin{array}{c} p \\ (\beta) \end{array} \right\rightarrow Z \overset{d}{\curvearrowright} X \left\leftarrow \begin{array}{c} x \\ e \\ (\infty) \end{array} \right\right\rangle$$

Proof.

$$\begin{aligned} \left\langle \left. \begin{array}{c} p \\ (\beta) \end{array} \right\rightarrow Z \overset{d}{\curvearrowright} X \left\leftarrow \begin{array}{c} x \\ e \\ (\infty) \end{array} \right\right\rangle &= \inf_{\mu} \left[\left. \begin{array}{c} p \\ (\beta) \end{array} \right\rightarrow Z \overset{d}{\curvearrowright} X \left\leftarrow \begin{array}{c} x \\ e \\ (\infty) \end{array} \right\right] (\mu) \\ &= \inf_{\mu} \mathbb{E}_{\mu(X,Z)} \left[\beta \log \frac{\mu(Z)}{p(Z)} + \log \frac{\mu(X, Z)}{\mu(Z)d(X \mid Z)} \right] \end{aligned}$$

As before, the only candidate for a joint distribution with finite score is $\delta_x(X)e(Z \mid X)$. Note that the marginal on Z for this distribution is itself, since $\int_x \delta_x(X)e(Z \mid X) dx = e(Z \mid x)$. Thus, our equation becomes

$$\begin{aligned} &= \mathbb{E}_{\delta_x(X)e(Z \mid X)} \left[\beta \log \frac{e(Z \mid x)}{p(z)} + \log \frac{\delta_x(X)e(Z \mid X)}{e(Z \mid x)d(x \mid Z)} \right] \\ &= \mathbb{E}_{e(Z \mid x)} \left[\beta \log \frac{e(Z \mid x)}{p(Z)} + \log \frac{1}{d(x \mid Z)} \right] \\ &= D(e(Z \mid x) \parallel p) + \text{Rec}_{e,d}(x) \\ &= -\beta\text{-ELBO}_{p,e,d}(x). \end{aligned}$$

\square

Proposition 6.7.1. *For all weighted factor graphs Ψ , we have that $\langle\!\langle \text{pdg}(\Psi) \rangle\!\rangle_1 = -\log Z_\Psi$.*

Proof. In the main text, we defined $\text{pdg}(\Psi)$ to be the PDG with edges $\{\xrightarrow{J} \mathbf{X}_J\}_{\mathcal{J}}$, cpds $p_J(\mathbf{X}_J) \propto \phi_J(\mathbf{X}_J)$, and weights $\alpha_J, \beta_J := \theta_J$. Let the be a function that extracts the unique element singleton set, so that $\text{the}(\{x\}) = x$. It was shown by Richardson and Halpern [82] (Corollary 4.4.1) that

$$\text{the}[\![\mathbf{m}_\Psi]\!]^* = \Pr_{\Phi, \theta}(\mathbf{x}) = \frac{1}{Z_\Psi} \prod_J \phi_J(\mathbf{x}_J)^{\theta_J}.$$

Recall the statement of Prop 4.6 from Richardson and Halpern [82]:

$$[\![\mathbf{m}]\!]_\gamma(\mu) = \mathbb{E}_{\mathbf{w} \sim \mu} \left\{ \sum_{X \xrightarrow{L} Y} \left[\beta_L \log \frac{1}{\mathbb{P}_L(y^{\mathbf{w}}|x^{\mathbf{w}})} + (\gamma \alpha_L - \beta_L) \log \frac{1}{\mu(y^{\mathbf{w}}|x^{\mathbf{w}})} \right] - \gamma \log \frac{1}{\mu(\mathbf{w})} \right\}, \quad (6.14)$$

where $x^{\mathbf{w}}$ and $y^{\mathbf{w}}$ are the respective values of the variables X and Y in the world \mathbf{w} . Note that if $\gamma = 1$, and α, β are both equal to θ in $\text{pdg}(\Psi)$, the **middle term (in purple)** is zero. So in our case, since the edges are $\{\xrightarrow{J} \mathbf{X}_J\}$ and $\mathbb{P}_J(\mathbf{X}_J) = \phi_J(\mathbf{X}_J)$, (6.14) reduces to the standard variational free energy

$$\begin{aligned} VFE_\Psi(\mu) &= \mathbb{E}_\mu \left[\sum_{J \in \mathcal{J}} \theta_J \log \frac{1}{\phi_J(\mathbf{X}_J)} \right] - H(\mu) \\ &= \mathbb{E}_\mu \langle \boldsymbol{\varphi}, \boldsymbol{\theta} \rangle_{\mathcal{J}} - H(\mu), \quad \text{where } \varphi_J(\mathbf{X}_J) := \log \frac{1}{\phi_J(\mathbf{X}_J)}. \end{aligned} \quad (6.15)$$

By construction, \Pr_Ψ uniquely minimizes VFE . The 1-inconsistency, $\langle\!\langle \mathbf{m}_\Psi \rangle\!\rangle$ is the minimum value attained. We calculate:

$$\begin{aligned} \langle\!\langle \mathbf{m} \rangle\!\rangle_1 &= VFE_\Psi(\Pr_\Psi) \\ &= \mathbb{E}_{\mathbf{x} \sim \mu} \left\{ \sum_{J \in \mathcal{J}} \left[\theta_J \log \frac{1}{\phi_J(\mathbf{x}_J)} \right] - \log \frac{1}{\Pr_{\Phi, \theta}(\mathbf{x})} \right\} \quad [\text{by (6.15)}] \\ &= \mathbb{E}_{\mathbf{x} \sim \mu} \left\{ \sum_{J \in \mathcal{J}} \left[\theta_J \log \frac{1}{\phi_J(\mathbf{x}_J)} \right] - \log \frac{Z_\Psi}{\prod_{J \in \mathcal{J}} \phi_J(\mathbf{x}_J)^{\theta_J}} \right\} \quad [\text{definition of } \Pr_\Psi] \end{aligned}$$

$$\begin{aligned}
&= \mathbb{E}_{\mathbf{x} \sim \mu} \left\{ \sum_J \left[\theta_J \log \frac{1}{\phi_J(\mathbf{x}_J)} \right] - \sum_{J \in \mathcal{J}} \left[\theta_J \log \frac{1}{\phi_J(\mathbf{x}_J)} \right] - \log Z_\Psi \right\} \\
&= \mathbb{E}_{\mathbf{x} \sim \mu} [-\log Z_\Psi] \\
&= -\log Z_\Psi \quad \left[\begin{array}{l} Z_\Psi \text{ is constant in } \mathbf{x} \end{array} \right]
\end{aligned}$$

Proposition 6.9.1. $\left\langle \left\langle \xrightarrow[p]{(\infty)} X \xrightarrow{\hat{c}} T \xleftarrow{t} \right\rangle \right\rangle = \mathbb{E}_{x \sim p}[c(x)].$

Proof. Since p has high confidence, and T is always equal to t , the only joint distribution on (X, T) with finite score is $\mu(X, T) = p(X)\delta_t(T)$. We compute its score directly:

$$\begin{aligned}
\left\langle \left\langle \xrightarrow[p]{(\infty)} X \xrightarrow{\hat{c}} T \xleftarrow{t} \right\rangle \right\rangle &= \mathbb{E}_\mu \log \frac{\mu(X, T)}{\hat{c}(t | X)} = \mathbb{E}_p \log \frac{1}{\hat{c}(t | X)} = \mathbb{E}_p \log \frac{1}{\exp(-c(X))} \\
&= \mathbb{E}_p \log \exp(c(X)) = \mathbb{E}_p c(X) = \mathbb{E}_{x \sim p} c(x). \quad \square
\end{aligned}$$

6.C.1 Additional Proofs for Unnumbered Claims

Details on the Data Processing Inequality Proof

We now provide more details on the proof of the Data Processing Equality that appeared in [Figure 2](#) of the main text. We repeat it now for convenience, with labeled PDGs (m_1, \dots, m_5) and numbered (in)equalities.

$$\begin{array}{ccc}
m_1 & m_2 & m_3 \\
\left\langle \left. \frac{p}{(\beta)} \rightarrow X \leftarrow \frac{q}{(\zeta)} \right. \right\rangle \stackrel{(1)}{=} \left\langle \left. \begin{array}{c} Y \\ f \uparrow (\beta+\zeta) \\ p \xrightarrow{(\beta)} X \leftarrow \frac{q}{(\zeta)} \end{array} \right. \right\rangle \stackrel{(2)}{=} \left\langle \left. \begin{array}{c} f \nearrow Y \\ (\beta) \\ p \xrightarrow{(\beta)} X_1 = X_2 \leftarrow \frac{q}{(\zeta)} \\ f \searrow \\ (\zeta) \end{array} \right. \right\rangle \\
& & \stackrel{(3)}{\geq} \left\langle \left. \begin{array}{c} f \nearrow Y \\ (\beta) \\ p \xrightarrow{(\beta)} X_1 \\ f \searrow \\ (\zeta) \\ X_2 \leftarrow \frac{q}{(\zeta)} \end{array} \right. \right\rangle \stackrel{(4)}{=} \left\langle \left. \begin{array}{c} f \circ p \\ (\beta) \\ X \leftarrow \frac{f \circ q}{(\zeta)} \end{array} \right. \right\rangle \\
m_4 & & m_5
\end{array}$$

We now enumerate the (in)equalities to prove them.

1. Let $\mu(X)$ denote the (unique) optimal distribution for m_1 . Now, the joint distribution $\mu(X, Y) := \mu(X)f(Y|X)$ has incompatibility with m_2 equal to

$$\begin{aligned}
Inc_{m_2}(\mu(X, Y)) &= \beta D(\mu(X) \parallel p(X)) + \zeta D(\mu(X) \parallel q(X)) + (\beta + \zeta) \mathbb{E}_{x \sim \mu} [D(\mu(Y|x) \parallel f(Y|x))] \\
&= Inc_{m_1}(\mu(X)) + (\beta + \zeta) \mathbb{E}_{x \sim \mu} D(\mu(Y|x) \parallel f(Y|x)) \\
&= \langle\!\langle m_1 \rangle\!\rangle \quad \begin{bmatrix} \text{as } \mu(Y|x) = f(Y|x) \text{ wherever } \mu(x) > 0, \\ \text{and } \mu(X) \text{ minimizes } Inc_{m_1} \end{bmatrix}.
\end{aligned}$$

So $\mu(X, Y)$ witnesses the fact that $\langle\!\langle m_2 \rangle\!\rangle \leq Inc_{m_2}(\mu(X, Y)) = \langle\!\langle m_1 \rangle\!\rangle$. Furthermore, every joint distribution $\nu(X, Y)$ must have at least this incompatibility, as it must have some marginal $\nu(X)$, which, even by itself, already gives rise to incompatibility of magnitude $Inc_{m_1}(\nu(X)) \geq Inc_{m_1}(\mu(X)) = \langle\!\langle m_1 \rangle\!\rangle$. And since this is true for all $\nu(X, Y)$, we have that $\langle\!\langle m_2 \rangle\!\rangle \geq \langle\!\langle m_1 \rangle\!\rangle$. So $\langle\!\langle m_2 \rangle\!\rangle = \langle\!\langle m_1 \rangle\!\rangle$.

2. The equals sign in m_3 may be equivalently interpreted as a cpd $eq(X_1|X_2) := x_2 \mapsto \delta_{x_2}(X_1)$, a cpd $eq'(X_2|X_1) := x_1 \mapsto \delta_{x_1}(X_2)$, or both at once; in each case, the effect is that a joint distribution μ with support on an outcome for which $X_1 \neq X_2$ gets an infinite penalty, so a minimizer $\mu(X_1, X_2, Y)$ of Inc_{m_3} must be isomorphic to a distribution $\mu'(X, Y)$.

Furthermore, it is easy to verify that $Inc_{m_2}(\mu'(X, Y)) = Inc_{m_3}(\mu(X, X, Y))$.

More formally, we have:

$$\langle\!\langle \mathbf{m}_3 \rangle\!\rangle = \inf_{\mu(X_1, X_2, Y)} \mathbb{E}_{\mu} \left[\begin{array}{ccc} \beta \log \frac{\mu(X_1)}{p(X_1)} & +\zeta \log \frac{\mu(X_2)}{q(X_2)} & + \log \frac{\mu(X_1|X_2)}{eq(X_1, X_2)} \\ +\beta \log \frac{\mu(Y|X_1)}{f(Y|X_1)} & +\zeta \log \frac{\mu(Y|X_2)}{f(Y|X_2)} & \end{array} \right]$$

but if X_1 always equals X_2 (which we call simply X), as it must for the optimal distribution μ , this becomes

$$\begin{aligned} &= \inf_{\mu(X_1=X_2=X, Y)} \mathbb{E}_{\mu} \left[\beta \log \frac{\mu(X)}{p(X)} + \zeta \log \frac{\mu(X)}{q(X)} + \beta \log \frac{\mu(Y|X)}{f(Y|X)} + \zeta \log \frac{\mu(Y|X)}{f(Y|X)} \right] \\ &= \inf_{\mu(X, Y)} \mathbb{E}_{\mu} \left[\beta \log \frac{\mu(X)}{p(X)} + \zeta \log \frac{\mu(X)}{q(X)} + (\beta + \zeta) \log \frac{\mu(Y|X)}{f(Y|X)} \right] \\ &= \inf_{\mu(X, Y)} Incm_2(\mu) \\ &= \langle\!\langle \mathbf{m}_2 \rangle\!\rangle. \end{aligned}$$

3. Eliminating the edge or edges enforcing the equality ($X_1 = X_2$) cannot increase inconsistency, by Lemma 6.2.1.

4. Although this final step of composing the edges with shared confidences looks intuitively like it should be true (and it is!), its proof may not be obvious. We now provide a rigorous proof of this equality.

To ameliorate subscript pains, we henceforth write X for X_1 , and Z for X_2 .

We now compute:

$$\begin{aligned} \langle\!\langle \mathbf{m}_4 \rangle\!\rangle &= \inf_{\mu(X, Z, Y)} \mathbb{E}_{\mu} \left[\beta \log \frac{\mu(X) \mu(Y|X)}{p(X) f(Y|X)} + \zeta \log \frac{\mu(Z) \mu(Y|Z)}{q(Z) f(Y|Z)} \right] \\ &= \inf_{\mu(X, Z, Y)} \mathbb{E}_{\mu} \left[\beta \log \frac{\mu(Y) \mu(X|Y)}{p(X) f(Y|X)} + \zeta \log \frac{\mu(Y) \mu(Z|Y)}{q(Z) f(Y|Z)} \right] \quad [\text{apply Bayes Rule in numerators}] \end{aligned}$$

By the chain rule, every distribution $\mu(X, Z, Y)$ may be specified as $\mu(Y)\mu(X|Y)\mu(Z|X, Y)$, so we can rewrite the formula above as

$$\langle\!\langle \mathbf{m}_4 \rangle\!\rangle = \inf_{\mu(Y)} \inf_{\mu(X|Y)} \inf_{\mu(Z|Y, X)} \mathbb{E}_{y \sim \mu(Y)} \mathbb{E}_{x \sim \mu(X|y)} \mathbb{E}_{z \sim \mu(Z|y, x)} \left[\beta \log \frac{\mu(y) \mu(x|y)}{p(x) f(y|x)} + \zeta \log \frac{\mu(y) \mu(z|y)}{q(z) f(y|z)} \right],$$

where $\mu(Z|Y)$ is defined in terms of the primitives $\mu(X|Y)$ and $\mu(Z|X, Y)$ as $\mu(Z|Y) := y \mapsto \mathbb{E}_{x \sim \mu(X|y)} \mu(Z|y, x)$, and is a valid cpd, since it is a mixture distribution. Since the first term (with β) does not depend on z , we can take it out of the expectation, so

$$\langle\!\langle m_4 \rangle\!\rangle = \inf_{\mu(Y)} \inf_{\mu(X|Y)} \inf_{\mu(Z|Y, X)} \mathbb{E}_{y \sim \mu(Y)} \mathbb{E}_{x \sim \mu(X|y)} \left[\beta \log \frac{\mu(y) \mu(x|y)}{p(x) f(y|x)} + \zeta \mathbb{E}_{z \sim \mu(Z|y, x)} \left[\log \frac{\mu(y) \mu(z|y)}{q(z) f(y|z)} \right] \right];$$

we can split up $\mathbb{E}_{\mu(X|y)}$ by linearity of expectation, to get

$$\langle\!\langle m_4 \rangle\!\rangle = \inf_{\mu(Y)} \inf_{\mu(X|Y)} \inf_{\mu(Z|Y, X)} \mathbb{E}_{y \sim \mu(Y)} \left[\beta \mathbb{E}_{x \sim \mu(X|y)} \left[\log \frac{\mu(y) \mu(x|y)}{p(x) f(y|x)} \right] + \zeta \mathbb{E}_{\substack{x \sim \mu(X|y) \\ z \sim \mu(Z|y, x)}} \left[\log \frac{\mu(y) \mu(z|y)}{q(z) f(y|z)} \right] \right]$$

Note that the quantity inside the second expectation does not depend on x . Therefore, the second expectation is just an explicit way of sampling z from the mixture distribution $\mathbb{E}_{x \sim \mu(X|y)} \mu(Z|x, y)$, which is the definition of $\mu(Z|y)$. Once we make this replacement, it becomes clear that the only feature of $\mu(Z|Y, X)$ that matters is the mixture $\mu(Z|Y)$. Simplifying the second expectation in this way, and replacing the infimum over $\mu(Z|X, Y)$ with one over $\mu(Z|Y)$ yields:

$$\langle\!\langle m_4 \rangle\!\rangle = \inf_{\mu(Y)} \inf_{\mu(X|Y)} \inf_{\mu(Z|Y)} \mathbb{E}_{y \sim \mu(Y)} \left[\beta \mathbb{E}_{x \sim \mu(X|y)} \left[\log \frac{\mu(y) \mu(x|y)}{p(x) f(y|x)} \right] + \zeta \mathbb{E}_{z \sim \mu(Z|y)} \left[\log \frac{\mu(y) \mu(z|y)}{q(z) f(y|z)} \right] \right]$$

Now, a cpd $\mu(X|Y)$ is just⁷ a (possibly different) distribution $\nu_y(X)$ for every value of Y . Observe that, inside the expectation over $\mu(Y)$, the cpds $\mu(X|Y)$ and $\mu(Z|Y)$ are used only for the *present* value of y , and do not reference, say, $\mu(X|y')$ for $y' \neq y$. Because there is no interaction between the choice of cpd $\mu(X|y)$ and $\mu(X|y')$, it is not necessary to jointly optimize over entire cpds $\mu(X|Y)$ all at once. Rather, it is equivalent to take the infimum over $\nu(X)$, separately for each y . Symmetrically, we may as well take the infimum over $\lambda(Z)$ separately for each y , rather than jointly finding the optimal $\mu(Z|Y)$ all at once. Operationally, this

⁷modulo measurability concerns that do not affect the infimum; see [Section 6.A](#)

means we can pull the infima inside the expectation over Y . And since the first term doesn't depend on Z and the second doesn't depend on X , we get:

$$\langle\!\langle m_4 \rangle\!\rangle = \inf_{\mu(Y)} \mathbb{E}_{y \sim \mu(Y)} \left[\inf_{\nu(X)} \beta \mathbb{E}_{\nu(X)} \left[\log \frac{\mu(y) \nu(X)}{p(X) f(y|X)} \right] + \inf_{\lambda(Z)} \zeta \mathbb{E}_{\lambda(Z)} \left[\log \frac{\mu(y) \lambda(Z)}{q(Z) f(y|Z)} \right] \right]$$

Next, we pull the same trick we've used over and over: find constants so that we can regard the dependence as a relative entropy with respect to the quantity being optimized. Grouping the quantities apart from $\nu(X)$ on the left term and normalizing them (and analogously for $\lambda(Z)$ on the right), we find that

$$\langle\!\langle m_4 \rangle\!\rangle = \inf_{\mu(Y)} \mathbb{E}_{y \sim \mu(Y)} \left[\begin{array}{l} \beta \inf_{\nu(X)} D\left(\nu(X) \parallel \frac{1}{C_1(y)} p(X) \frac{f(y|X)}{\mu(y)}\right) - \beta \log C_1(y) \\ + \zeta \inf_{\lambda(Z)} D\left(\lambda(Z) \parallel \frac{1}{C_2(y)} q(Z) \frac{f(y|Z)}{\mu(y)}\right) - \zeta \log C_2(y) \end{array} \right],$$

where

$$C_1(y) = \sum_x p(x) \frac{f(y|x)}{\mu(y)} = \frac{1}{\mu(y)} \mathbb{E}_{p(X)} f(y|X)$$

and

$$C_2(y) = \sum_z q(z) \frac{f(y|z)}{\mu(y)} = \frac{1}{\mu(y)} \mathbb{E}_{q(Z)} f(y|Z)$$

are the constants required to normalize the distributions. Both relative entropies are minimized when their arguments match, at which point they contribute zero, so we have

$$\begin{aligned} \langle\!\langle m_4 \rangle\!\rangle &= \inf_{\mu(Y)} \mathbb{E}_{y \sim \mu(Y)} \left[\beta \log \frac{1}{C_1(y)} + \zeta \log \frac{1}{C_2(y)} \right] \\ &= \inf_{\mu(Y)} \mathbb{E}_{y \sim \mu(Y)} \left[\beta \log \frac{\mu(y)}{\mathbb{E}_{p(X)} f(y|X)} + \zeta \log \frac{\mu(y)}{\mathbb{E}_{q(Z)} f(y|Z)} \right] \\ &= \inf_{\mu(Y)} \mathbb{E}_{\mu} \left[\beta D(\mu \parallel f \circ p) + \zeta D(\mu \parallel f \circ q) \right] \\ &= \langle\!\langle m_5 \rangle\!\rangle. \end{aligned}$$

Details for Claims made in Section 6.8

First, the fact that

$$\mathcal{L}_1 = \lambda_d \mathcal{L}_{\text{dat}} + \lambda_s \mathcal{L}_{\text{sim}} = \left\langle \begin{array}{c} \xrightarrow[\infty]{\lambda} Z \\ \text{sim dat} \end{array} \right\rangle_{\text{sim} \mapsto s} \xrightarrow[\infty]{\text{dat} \mapsto d} \begin{array}{c} X \\ h \\ Y \end{array} \right\rangle,$$

where $\lambda(Z = \text{sim}) = \lambda_s$ and $\lambda(Z = \text{dat}) = \lambda_d$ is immediate. The two cpds with infinite confidence ensure that the only joint distribution with a finite score is $\lambda_s s + \lambda_d d$, and the inconsistency with h is its surprisal, so the inconsistency of this PDG is

$$\mathbb{E}_{\lambda_s s + \lambda_d d} \left[\log \frac{1}{h(Y|X)} \right] = -\lambda_s \mathbb{E}_s [\log h(Y|X)] - \lambda_d \mathbb{E}_d [\log h(Y|X)] = \lambda_d \mathcal{L}_{\text{dat}} + \lambda_s \mathcal{L}_{\text{sim}} = \mathcal{L}_1, \quad \text{as pr}$$

The second correspondence is the least straightforward. Let $C = \int sd$ be the normalization constant required to normalize the joint density sd . We claim that, for large fixed γ , we have

$$\mathcal{L}_2 \approx C \left\langle \begin{array}{c} s \\ h \\ d \end{array} \right\rangle_\gamma + \text{const},$$

where const does not depend on h . To see this, let m_2 be the PDG above, and compute

$$\begin{aligned} \langle m_2 \rangle_\gamma &= \inf_{\mu(X,Y)} \mathbb{E}_\mu \left[\underbrace{\gamma \log \frac{\mu(XY)}{s(XY)} \frac{\mu(XY)}{d(XY)}}_{Inc(\mu)} + \log \frac{\mu(Y|X)}{h(Y|X)} + \gamma \log \frac{1}{s(XY)} \frac{1}{d(XY)} - \gamma \log \frac{1}{\mu(XY)} \right] \\ &= \inf_{\mu(X,Y)} \mathbb{E}_\mu \left[\gamma \log \frac{\mu(XY)}{s(XY)} \frac{\mu(XY)}{d(XY)} \frac{1}{\mu(XY)} \frac{1}{\mu(XY)} \frac{\mu(XY)}{1} + \log \frac{\mu(Y|X)}{h(Y|X)} \right] \\ &= \inf_{\mu(X,Y)} \mathbb{E}_\mu \left[\gamma \log \frac{\mu(XY)}{s(XY)d(XY)} + \log \frac{\mu(Y|X)}{h(Y|X)} \right] \\ &= \inf_{\mu(X,Y)} \mathbb{E}_\mu \left[\gamma \log \frac{\mu(XY)C}{s(XY)d(XY)} - \gamma \log C + \log \frac{\mu(Y|X)}{h(Y|X)} \right] \end{aligned}$$

$$= \inf_{\mu(X,Y)} \gamma D\left(\mu \middle\| \frac{1}{C}sd\right) + \mathbb{E}_{\mu} \left[\log \frac{\mu(Y|X)}{h(Y|X)} \right] - \gamma \log C$$

D is (γm) -strongly convex in a region around its minimizer for some $m > 0$ that depends only on s and d . Together with our assumption that h is positive, we find that when γ becomes large, the first term dominates, and the optimizing μ quickly approaches the normalized density $\nu := \frac{1}{C}sd$. Plugging in ν , we find that the value of the infimum approaches

$$\begin{aligned} \langle\!\langle \mathbf{m}_2 \rangle\!\rangle &\approx \mathbb{E}_{\nu} \left[\log \frac{1}{h(Y|X)} \right] - H_{\nu}(Y|X) - \gamma \log C \\ &= \int_{XY} \frac{1}{C} \log \frac{1}{h(Y|X)} s(X, Y) d(X, Y) - H_{\nu}(Y|X) - \gamma \log C \\ &= \frac{1}{C} \mathbb{E}_s \left[d(X, Y) \log \frac{1}{h(Y|X)} \right] - H_{\nu}(Y|X) - \gamma \log C \\ &= \frac{1}{C} \mathcal{L}_2 - H_{\nu}(Y|X) - \gamma \log C, \end{aligned}$$

$$\begin{aligned} \text{and therefore } \mathcal{L}_2 &= C \langle\!\langle \mathbf{m}_2 \rangle\!\rangle + C H_{\nu}(Y|X) - \gamma C \log C \\ &= C \langle\!\langle \mathbf{m}_2 \rangle\!\rangle + \text{const.} \end{aligned}$$

Finally, we turn to

$$\mathcal{L}_3 := \left\langle\!\left\langle \frac{s}{(\lambda_s)} \xrightarrow{\quad h \downarrow \quad} \begin{array}{c} X \\ \downarrow \\ Y \end{array} \xleftarrow{\quad d \quad (\lambda_d) \quad} \right\rangle\!\right\rangle.$$

To see the why the optimal distribution $\mu^*(XY)$ is the λ -weighted geometric mean of s and d , let us first consider the same PDG, except without h . From Lemma 6.5.2, we have this loss without h in closed form, and from the proof of Lemma 6.5.2, we see that the optimizing distribution in this case is the λ -weighted geometric distribution $\mu^* \propto s(XY)^{\lambda_s} d(XY)^{\lambda_d}$. Now (Lemma 6.2.1), including h cannot make the PDG any less inconsistent. In particular, by choosing

$$h^*(Y|X) := \mu^*(Y|X) \propto (Y|X)^{\lambda_s} d(Y|X)^{\lambda_d},$$

to be already compatible with this joint distribution, the inconsistency does not change, while choosing a different h would cause the inconsistency to increase. Thus, the optimal classifier h^* by this metric is indeed as we claim. Finally, it is easy to see that this loss is calibrated: if $s = d$, then the optimal joint distribution is equal to s and to d , and the optimal classifier is $h(Y|X) = s(Y|X) = d(Y|X)$. So \mathcal{L}_3 is calibrated.

Details for Claims made in Section 6.9

Distortion Due to Inconsistency. In the footnote on [Page 157](#), we claimed that if the model confidence β_p were 1 rather than ∞ , we would have obtained an inconsistency of $-\log \mathbb{E}_{x \sim p} \exp(-c(x))$, and that the optimal distribution would not have been $p(X)$.

$$\begin{aligned} \left\langle \xrightarrow{p} [X] \xrightarrow{\hat{c}} [\mathbf{T}] \xleftarrow{\mathbf{t}} \right\rangle &= \inf_{\mu(X)} \mathbb{E}_{x \sim \mu} \left[\log \frac{\mu(x)}{p(x)} + \log \frac{\mu(\mathbf{t}|x)}{\hat{c}(\mathbf{t}|x)} \right] \\ &= \inf_{\mu(X)} \mathbb{E}_{x \sim \mu} \left[\log \frac{\mu(x)}{p(x)} + \log \frac{1}{\hat{c}(\mathbf{t}|x)} \right] \\ &= \inf_{\mu(X)} \mathbb{E}_{x \sim \mu} \left[\log \frac{\mu(x)}{p(x) \exp(-c(x))} \cdot \frac{Z}{Z} \right] \end{aligned}$$

where $Z = \sum_x p(x) \exp(-c(x)) = \mathbb{E}_p \exp(-c(X))$ is the constant required to normalize the distribution

$$\begin{aligned} &= \inf_{\mu(X)} D\left(\mu \middle\| \frac{1}{Z} p(X) \exp(-c(X))\right) - \log Z \\ &= -\log Z \\ &= -\log \mathbb{E}_{x \sim p} \exp(-c(x)) \end{aligned}$$

as promised. Note also that in the proof, we showed that the optimal distribution is proportional to $p(X) \exp(-c(X))$ which means that it equals $p(X)$ if and only

if $c(X)$ is constant in X .

Enforcing the Qualitative Picture. We also claimed without careful proof in Section 6.9 that, if $\alpha_h = \alpha_{\Pr_D} = 1$, then

$$\lim_{\gamma \rightarrow \infty} \left\langle \begin{array}{c} \Pr_D \xrightarrow{\ell} Y \\ \downarrow \text{---} \xrightarrow{\phi} X \xrightarrow{h} Y' \\ \uparrow \text{---} \xrightarrow{\hat{\ell}} \mathbf{T} \end{array} \right\rangle_\gamma = \mathbb{E}_{\substack{(x,y) \sim \Pr_D \\ y' \sim p(Y'|x)}} [\ell(y, y')]$$

Why is this? For such a setting of α , which intuitively articulates a causal picture where X, Y is generated from \Pr_D , and Y' generated by $h(Y'|X)$, the information deficiency $IDef_S(\mu(X, Y, Y'))$ of a distribution μ is

$$\begin{aligned} IDef_S(\mu(X, Y, Y')) &= -H_\mu(X, Y, Y') + H(X, Y) + H(Y'|X) \\ &= H_\mu(Y'|X) - H_\mu(Y'|X, Y) \\ &= I_\mu(Y; Y'|X). \end{aligned}$$

Both equalities of the derivation above standard information theoretic identities [See, for instance, 63], and the final quantity $I_\mu(Y; Y'|X)$ is the *conditional mutual information* between Y and Y' given X , and is a non-negative number that equals zero if and only if Y and Y' are conditionally independent given X .

As a result, as $\gamma \rightarrow \infty$ any distribution that for which Y' and Y are not independent given X will incur infinite cost. Since the confidences in h and \Pr_D are also infinite, so will a violation of either cpd. There is only one distribution that has both cpds and also this independence; that distribution is $\mu(X, Y, Y') := \Pr_D(X, Y)h(Y'|X)$. Now the argument of Proposition 6.9.1 applies: all other cpds must be matched, and the inconsistency is the expected incompatibility of $\hat{\ell}$, which equals

$$\mathbb{E}_{\substack{(x,y) \sim \Pr_D \\ y' \sim p(Y'|x)}} \log \frac{1}{\hat{\ell}(\mathbf{t}|y, y')} = \mathbb{E}_{\substack{(x,y) \sim \Pr_D \\ y' \sim p(Y'|x)}} \log \frac{1}{\exp(-\ell(y, y'))} = \mathbb{E}_{\substack{(x,y) \sim \Pr_D \\ y' \sim p(Y'|x)}} [\log \exp(\ell(y, y'))] = \mathbb{E}_{\substack{(x,y) \sim \Pr_D \\ y' \sim p(Y'|x)}} [\ell(y, y')] = \mathcal{L}$$

6.D More Notes

6.D.1 Maximum A Posteriori and Priors

The usual telling of the correspondence between regularizers and priors is something like the following. Suppose you have a parameterized family of distributions $\Pr(X|\Theta)$ and have observed evidence X , but do not know the parameter Θ .

The maximum-likelihood estimate of Θ is then

$$\theta^{\text{MLE}}(X) := \arg \max_{\theta \in \Theta} \Pr(X|\theta) = \arg \max_{\theta \in \Theta} \log \Pr(X|\theta).$$

The logarithm is a monotonic transformation, so it does not change the argmax, but it has nicer properties, so that function is generally used instead. (Many of the loss functions in main body of the paper are log-likelihoods also.)

In some sense, better than estimating the maximum likelihood, is to perform a Bayesian update with the new information, to get a *distribution* over Θ . If that's too expensive, we could simply take the estimate with the highest posterior probability, which is called the Maximum A Posteriori (MAP) estimate. For any given θ , the Bayesian reading of Bayes rule states that

$$\text{posterior } \Pr(\Theta|X) = \frac{\text{likelihood } \Pr(X|\Theta) \cdot \text{prior } \Pr(\Theta)}{\text{evidence } \Pr(X) = \sum_{\theta'} \Pr(X|\theta') \Pr(\theta')}.$$

So taking a logarithm,

$$\text{log-posterior } \log \Pr(\Theta|X) = \text{log-likelihood } \log \Pr(X|\Theta) + \text{log-prior } \log \Pr(\Theta) - \text{log-evidence } \log$$

The final term does not depend on θ , so it is not relevant for finding the optimal θ by this metric. Swapping the signs so that we are taking a minimum rather than a maximum, the MAP estimate is then given by

$$\theta^{\text{MAP}}(X) := \arg \min_{\theta \in \Theta} \left\{ \log \frac{1}{\Pr(X|\theta)} + \log \frac{1}{\Pr(\theta)} \right\}.$$

Note that if negative log likelihood (or surprisal, $-\log \Pr(X|\theta)$) was our original loss function, we have now added an arbitrary extra term, as a function of Θ , to our loss function. It is in this sense that priors classically correspond to regularizers.

CHAPTER 7

THE LOCAL INCONSISTENCY RESOLUTION (LIR) ALGORITHM

RELATIVE ENTROPY SOUP

Oliver E. Richardson, Ph.D.

Cornell University 2024

We present a generic algorithm for learning and approximate inference across a broad class of statistical models, that unifies many approaches in the literature. Our algorithm, called local inconsistency resolution (LIR), has an intuitive epistemic interpretation. It is based on the theory of probabilistic dependency graphs (PDGs), an expressive class of graphical models rooted in information theory, which can capture inconsistent beliefs.

7.1 Introduction

What causes a person to change their mind? According to some, it is a response to internal conflict: the result of discovering new information that contradicts our beliefs, or becoming aware of discrepancies between beliefs we already hold [27]. Inconsistencies can be difficult to detect, however [85], and indeed can only be resolved once we are aware of them. Some things are also beyond our control; for example, we might receive conflicting information from two trusted sources and be unable to resolve their disagreement. So in practice, we resolve inconsistencies *locally*—little by little, and looking at only a small part of the picture at a time.

This can have externalities; fixing one inconsistency can easily create others out of view. Furthermore, some inconsistencies are not local in nature, and can only be seen when considering many components at once. Yet despite its imperfections, this process of locally resolving inconsistency can be quite useful. As we shall soon see, it is a powerful recipe for learning and approximate inference. We formalize the process in the language of probability and convex optimization, and show how that many popular techniques in the literature arise naturally as instances of it.

Our approach leans heavily on the theory of Probabilistic Dependency Graphs (PDGs), which are very flexible graphical models that allow for arbitrary—even inconsistent—probabilistic information, weighted by confidence [82]. There is a natural way to measure how inconsistent a PDG is, and many standard loss functions can be viewed as measuring the inconsistency of a PDG that describes the appropriate situation [81]. Recently, techniques have been developed to calculate this inconsistency in polynomial time for bounded tree-width, although

it scales exponentially with the tree-width of the graph [83]. As we move to variables that are continuous variables of large dimension, it becomes intractable to calculate this global inconsistency even for small graphs—the log evidence of a latent variable model [81] can be represented as a PDG inconsistency, for example. We introduce an algorithm to operationalize the process of adjusting parameters to resolve this inconsistency.

In general, even just calculating a PDG’s degree of inconsistency is intractable. Much of variational inference can be understood as adopting extra beliefs to minimize an overapproximation of it that is easier to calculate [81]. Our approach can capture this, but also enables the opposite: focusing on small parts of the graph at a time to address tractable underapproximations of the global inconsistency. This makes it more suitable for distributed settings, and more amenable to parallelization. The algorithm, which we call *local inconsistency resolution* (LIR), is quite expressive, and naturally reduces to a wide variety of learning and inference algorithms in the literature. This observation suggests a generic approach to learning and inference in models with arbitrary structure.

7.2 Mathematical Preliminaries

We write $\mathcal{V}X$ for the set of values that a variable X can take on, and $\Delta\mathcal{V}X$ for the set of distributions over $\mathcal{V}X$. A conditional probability distribution (cpd) is a map $p(Y|X) : \mathcal{V}X \rightarrow \Delta\mathcal{V}Y$. A *directed hypergraph* (N, \mathcal{A}) is a set of nodes N and a set of arcs \mathcal{A} , each $a \in \mathcal{A}$ of which is associated with a set $S_a \subseteq N$ of source nodes, and $T_a \subseteq N$ target nodes. We also write $S \xrightarrow{a} T \in \mathcal{A}$ to specify an arc a together with its sources $S = S_a$ and targets $T = T_a$.

Geometry. We will need various parameter spaces Θ . To simplify the presentation, assume that each Θ is a convex subset of \mathbb{R}^n (not necessarily of the same dimension). A *vector field* over Θ is a differentiable map X assigning to each $\theta \in \Theta$ a vector $X_\theta \in \mathbb{R}^n$. The *gradient* of a twice differentiable map $f : \Theta \rightarrow \mathbb{R}$, which we write $\nabla_\Theta f(\Theta)$, is a vector field. Given a vector field X and an initial point $\theta_0 \in \Theta$, there is a unique trajectory $y(t)$ that solves the ODE $\{\frac{d}{dt}y(t) = X_{y(t)}, y(0) = \theta_0\}$, and we adopt the notation $\exp_{\theta_0}(X) := y(1)$ for a compact description of it. At first glance, \exp only gives us access to $y(1)$, but it is easily verified that $\exp_{\theta_0}(tX) = y(t)$. So altogether, the map $t \mapsto \exp_\theta(t\nabla_\Theta f(\Theta))$ is the smooth path beginning at θ that follows the gradient of f . It is known as *gradient flow*.

Given a manifold Θ and a differentiable map $P : \Theta \rightarrow \Delta \mathcal{V}X$, the Fisher Information Matrix $\mathcal{I}(\theta)$ at each $\theta \in \Theta$ gives rise to a Riemannian metric; thus the mere fact that Θ parameterizes a family of probability distributions is enough to make it a Riemannian manifold. Moreover, $\mathcal{I}(\theta)$ is particularly natural in a probabilistic context; up to a multiplicative constant, it is the *only* such metric on Θ that is invariant under sufficient statistics, [?]. [3]

Probabilistic Dependency Graphs. A PDG is a directed graph whose arcs carry probabilistic and causal information, weighted by confidence [82]. We now introduce an equally expressive variant, whose explicit parametric nature will prove useful for our purposes.

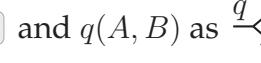
Definition 7.2.1. A *Parametric Probabilistic Dependency Graph* (PPDG) $m(\Theta) = (\mathcal{X}, \mathcal{A}, \Theta, \mathbb{P}, \alpha, \beta)$ is a directed hypergraph $(\mathcal{X}, \mathcal{A})$ whose nodes correspond to variables, each arc $a \in \mathcal{A}$ of which is associated with:

- a parameter space $\Theta_a \subseteq \mathbb{R}^n$, with a default value θ_a^{init} .

- a map $\mathbb{P}_a : \Theta_a \times \mathcal{V}S_a \rightarrow \Delta \mathcal{V}T_a$ that gives a cpd $\mathbb{P}_a^\theta(T_a|S_a)$ over a 's targets given its sources, for every $\theta \in \Theta_a$,
- confidences $\alpha_a \in \mathbb{R}$ in the functional dependence of T_a on S_a expressed by a , and $\beta_a \in [0, \infty]$ in the cpd \mathbb{P}_a .

A PDG is the object obtained by fixing the parameters; thus, a choice of $\theta \in \Theta := \prod_{a \in \mathcal{A}} \Theta_a$ yields a PDG $\mathbf{m} = \mathbf{m}(\theta)$. \square

Clearly, a PDG is the special case of a PPDG in which every $\Theta_a = \{\theta_a^{\text{init}}\}$ is a singleton. Conversely, a PPDG may be viewed as a PDG by adding each Θ_a as a variable, as illustrated in Figure 1. We often identify the label a with the cpd \mathbb{P}_a , and specify (P)PDGs in graphical notation, drawing

a cpd $p(Y|X, Z)$ as  and $q(A, B)$ as .

Unless otherwise specified, take $\beta, \alpha=1$ by default. We write $\mathbf{m}_1 + \mathbf{m}_2$ for the PDG that has the arcs of both \mathbf{m}_1 and \mathbf{m}_2 , and represents their combined information.

PDG Semantics and Inconsistency. The power of PDGs comes from their semantics, which sew their (possibly inconsistent) cpds and confidences together into joint probabilistic information. A PDG contains two kinds of information: structural information about causal mechanisms, (the graph \mathcal{A} and weights α), and observational data (the cpds \mathbb{P} and confidences β). With respect to a PDG \mathbf{m} , the *observational incompatibility* of a joint probability measure $\mu \in \Delta \mathcal{V}\mathcal{X}$ is given by a weighted sum of relative entropies

$$OInc_m(\mu) := \sum_{S \xrightarrow{a} T \in \mathcal{A}} \beta_a D\left(\mu(T, S) \parallel \mathbb{P}_a(T|S)\mu(S)\right), \quad (7.1)$$

and can be thought of as the excess cost of using codes optimized for each cpd, weighted by the confidence we have in them, if $\mathcal{X} \sim \mu$. If \mathbf{m} 's observational

confidences are positive ($\beta > 0$), then $OInc_m(\mu) = 0$ if and only if μ has every conditional marginal described by \mathbb{P} .

We can also score μ by its incompatibility with the structural information (\mathcal{A}, α) . This *structural deficiency* is given by:¹

$$SDef_m(\mu) := \mathbb{E}_{\mu} \left[\log \frac{\mu(\mathcal{X})}{\lambda(\mathcal{X})} \prod_{S \xrightarrow{a} T} \left(\frac{\lambda(T|S)}{\mu(T|S)} \right)^{\alpha_a} \right], \quad (7.2)$$

and, roughly, measures μ 's failure to arise as a result of independent causal mechanisms along each edge. If \mathcal{A} is a qualitative Bayesian Network, for instance, then $SDef_{\mathcal{A}}(\mu) \geq 0$ with equality iff μ has the independencies of \mathcal{A} . We encourage the reader to consult previous work for further details.

With confidence $\gamma \geq 0$ in the structural information overall, the γ -inconsistency of m is the smallest possible overall incompatibility of any distribution with m , and denoted

$$\langle\!\langle m \rangle\!\rangle_{\gamma} := \inf_{\mu} \left(OInc_m(\mu) + \gamma SDef_m(\mu) \right). \quad (7.3)$$

Richardson [81] argues that this inconsistency measure (7.3) is a “universal” loss function, largely showing how it specializes to standard loss functions in a wide variety of situations. It follows that, at an abstract level, much of machine learning can be viewed as inconsistency resolution. We take this idea a few steps further, by operationalizing the resolution process, and allowing it to be done locally.

PDG inference is fixed-parameter tractable: assuming a graph with bounded tree-width, the can be computed in polynomial time. and it appears that [83]. Several important algorithms can already be seen as inconsistency reduction. When viewed as a graphical model

¹In (7.2), λ is base measure, a property of \mathcal{X} . The precise choice is not important, but think: uniform or an appropriate analogue.

7.3 Local Inconsistency Resolution (LIR)

Attention and Control. There are two distinct senses in which inconsistency resolution can be *local*: we can restrict what we can see, or what we can do about it. Correspondingly, there are two “focus” knobs for our algorithm: one that restricts our attention to the inconsistency of a subset of arcs $A \subseteq \mathcal{A}$, and another that restricts our control to (only) the parameters of arcs $C \subseteq \mathcal{A}$ as we resolve that inconsistency. The former makes for an underestimate of the inconsistency that is easier to calculate, while the latter makes for an easier optimization problem. These restrictions are not just cheap approximations, though: they are also appropriate modeling assumptions for actors that cannot see and control everything at once.

Attention and control need not be black or white. A more general approach is to choose an *attention mask* $\varphi \in \mathbb{R}^{\mathcal{A}}$ and a *control mask* $\chi \in [0, \infty]^{\mathcal{A}}$. Large $\varphi(a)$ makes a salient, while $\varphi(a) = 0$ keeps it out of the picture. Similarly, large $\chi(a)$ gives significant freedom to change a ’s parameters, small $\chi(a)$ affords only minor adjustments, and $\chi(a) = 0$ prevents change altogether. Either mask can then be applied to a tensor that has an axis corresponding to \mathcal{A} , via pointwise multiplication (\odot).

The Algorithm. LIR modifies the parameters θ of a PPDG $\mathcal{M}(\Theta)$ so as to make it more consistent with its context. It proceeds as follows. First, receive context in the form of a PDG Ctx , and initialize mutable memory $\mathcal{M}(\Theta)$. In each iteration, choose γ (which can be viewed as attention to structure), an attention mask φ over the arcs of $\mathcal{M}(\Theta) + Ctx$, and a control mask χ over the arcs of $\mathcal{M}(\Theta)$. Calculate $\langle\langle \varphi \odot (\mathcal{M}(\theta) + Ctx) \rangle\rangle_{\gamma}$, the inconsistency of the combined context and memory, weighted by attention. (For discrete PDGs, this can be done with the

methods of Richardson et al. [83].) Then mitigate this local inconsistency by updating mutable memory θ via (an approximation to) gradient flow, changing a 's parameters in proportion to control $\chi(a)$. The procedure is fully formalized in [Algorithm 1](#).

Algorithm 1 Local Inconsistency Resolution (LIR)

Input: context Ctx , mutable memory $m(\Theta)$.
 Initialize $\theta^{(0)} \leftarrow \theta^{\text{init}}$;
for $t = 0, 1, 2, \dots$ **do**
 $Ctx \leftarrow \text{REFRESH}(Ctx)$; *//optional*
 $\varphi, \chi, \gamma \leftarrow \text{REFOCUS}()$;
 $\theta^{(t+1)} \leftarrow \exp_{\theta^{(t)}} \left\{ -\chi \odot \nabla_\Theta \langle\!\langle \varphi \odot (Ctx + m(\Theta)) \rangle\!\rangle_\gamma \right\}$;

In order to execute this procedure, we must say something about how the choice of (φ, χ, γ) is made. Thus, we must supply an additional procedure REFOCUS to select attention and control masks. We focus mostly on the case where γ is fixed, and REFOCUS chooses non-deterministically from a fixed set of attention/control mask pairs $(\varphi, \chi) \in \mathbf{F}$, which we call *foci*. [Algorithm 1](#) also allows us to select a second procedure, REFRESH, which makes it easier to model receiving new information in online settings.

The ODE on the last line of [Algorithm 1](#), which is an instance of gradient flow, may be approximated with an inner loop running an iterative gradient-based optimization algorithm. Alternatively, if REFOCUS produces small χ , then it is well-approximated by a single gradient descent step of size χ . At the other extreme: if χ is infinite in every component, then, so long as the parameterizations \mathbb{P} are log-concave, the final line reduces to

$$\theta^{(t+1)} \leftarrow \arg \min_\theta \langle\!\langle \varphi \odot (Ctx + m(\theta)) \rangle\!\rangle_\gamma, \quad \text{because of}$$

Theorem 7.3.1. If \mathbb{P} is log-concave, then for small enough γ , the map $\theta \mapsto \langle\!\langle \varphi \odot (\mathcal{C}tx + m(\theta)) \rangle\!\rangle_\gamma$ is convex.²

In the remaining sections, we give a sample of some historically important algorithms that are instances of LIR.

If m is a PDG with discrete variables \mathcal{X} and we regard $\mu(X)$ $m(\Theta)$ consists of a single joint distribution $\mu(\mathcal{X})$ parameterized as a vector $[0, 1]^{\mathcal{V}\mathcal{X}}$, and REFOCUS always produces constant χ, γ , then $\text{LIR}(\mathcal{C}tx, m)$ equals $[\chi \odot \mathcal{C}tx]_\gamma^*$, the optimal distribution, albeit in a roundabout manner.

7.4 LIR in the Classification Setting

Consider a parametric classifier $p_\theta(Y|X)$, perhaps arising from a neural network whose final layer is a softmax. Suppose $\mathcal{V}Y$ is a finite set of classes. If $\mathcal{V}X$ is itself a manifold (such as the space of images), we can regard a value $x \in \mathcal{V}X$ as parameterizing a deterministic cpd, written $\xrightarrow{x} \boxed{X}$. Together with a labeled sample (x, y) , we get a PPDG $m(\theta) := \xrightarrow{x} \boxed{X} \xrightarrow{p_\theta} \boxed{Y} \leftarrow y$ whose observational inconsistency is $\langle\!\langle m \rangle\!\rangle_0 = -\log p_\theta(y|x)$, the standard training objective for such a classifier [81]. Each cpd plays major role in this inconsistency.

What happens when we resolve this inconsistency by modifying the parameters associated to different arcs?

- Adjusting θ amounts to training the network in the standard way. In this case, the value χ of the control mask corresponds roughly to the product of the learning rate and the number of optimization iterations.

²All proofs can be found in the appendix.



Figure 1: Two illustrations of adversarial training. Left: the PPDG obtained by including a perturbed input x' and target y' to the classification setting. Right: the PDG obtained by making the parameters for p explicit, together with a Gaussian prior $\Theta_p \sim \mathcal{N}(0, 1)$ over them. Both are colored with two foci: the blue focus trains the network, and the green one creates adversarial examples. Dashes indicate control.

- Adjusting y is like a forward pass, in that it adjusts y to match distribution $p_\theta(Y|x)$.
- Adjusting x creates an adversarial example. That is, it makes incremental changes to the input x until the (fixed) network assigns it label y .

Stochastic Gradient Descent (SGD). Take the mutable state to be the classifier p as before. Define REFRESH so that it draws a batch of samples $\{(x_i, y_i)\}_{i=1}^m$, and returns a PDG with a single arc describing their empirical distribution $d(X, Y)$; let REFOCUS be such that $\varphi(d) = \infty$ (reflecting high confidence in the data). If $\eta := \chi(p)\varphi(p)$ is small, then LIR is SGD with batch size m and learning rate η .

Adversarial training. Suppose we want to slightly alter x to obtain x' that is classified as y' instead of y . By adding arcs corresponding to x' and y' to M , and relaxing the cpd \mathbb{P}_x associated with x to be a Gaussian centered x rather than a point mass, we get the PPDG on the left of Figure 1. An iteration of LIR whose focus is the edges marked in green (with control over the dashed green edge) is then an adversarial attack with Euclidean distance [10]. The blue focus, by contrast, “patches” the adversarial example by adjusting the model parameters to again classify it correctly. Thus, LIR that alternates between the two foci, in which REFRESH selects a fresh $(x, y, x' = x)$ from the dataset and target label y' ,

is adversarial training, a standard defense to adversarial attacks [33].

The ML community's focus on adversarial examples may appear to be a cultural phenomenon, but mathematically, it is no accident. At this level of abstraction, there is no difference between model parameters and inputs. Indeed, if we make the parameterization of p explicit and add L2 regularization (i.e., a Gaussian prior over Θ_p), the symmetry becomes striking (Figure 1, right). This may help explain why, even outside of adversarial contexts, it can be just as sensible to train an input, as a model [53].

7.5 The EM Algorithm as LIR

Suppose we have a generative model $p(Z, X|\Theta)$ describing the probability over an observable variable X and a latent one Z . Given an observation $X=x$, the standard approach for trying to learn the parameters despite the missing data is called the EM algorithm. It iteratively computes

$$\theta_{\text{EM}}^{(t+1)} = \arg \max_{\theta} \mathbb{E}_{z \sim p(Z|x, \theta_{\text{EM}}^{(t)})} [\log p(x, z|\theta)].$$

Proposition 7.5.1. *LIR* $\left(\xrightarrow{x} \boxed{X}, \boxed{X} \xrightarrow[p]{\downarrow} \boxed{Z} \xleftarrow[q]{\leftarrow \infty} \right)$ in which REFOCUS fixes $\varphi = \mathbf{1}$ and alternates between full control of p and q implements EM, in that $\theta_{\text{EM}}^{(t)} = \theta_{\text{LIR}}^{(2t)}$.

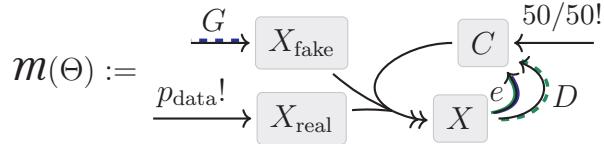
This result is closely related to one due to Neal and Hinton [68], who view it as an intuitive explanation of why the EM algorithm works. Indeed, it is obvious in this form that every adjustment reduces the overall inconsistency. The result can also be readily adapted to an entire dataset by replacing x with a high confidence empirical distribution, or batched with the same technique in Section 7.4. It also

captures fractional EM when $\chi < \infty$.

This form of the EM algorithm is closely related to variational inference. Indeed, analogous choices applied to the analysis of Richardson [81] yields the usual training algorithm for variational autoencoders (VAEs).

7.6 Generative Adversarial Training as LIR

LIR also subsumes more complex training procedures such as the one used to train GANs [32]. The goal is to train a network G to generate images that cannot be distinguished from real ones. More precisely, define X to be either an image $X_{\text{fake}} \sim G$ or from a dataset $X_{\text{real}} \sim p_{\text{data}}$, based on a fair coin C . A discriminator D then predicts C from X . The generator also has a belief that, even given X , the coin is equally likely heads as tails (call this e). This state of affairs is summarized below.



The GAN objective is typically written as a 2-player minimax game: $\min_G \min_D \mathcal{L}^{\text{GAN}}(G, D)$, where

$$\mathcal{L}^{\text{GAN}}(G, D) = \mathbb{E}_{x \sim p_{\text{data}}} [\log D(x)] + \mathbb{E}_{x' \sim G} [\log(1 - D(x'))].$$

The Discriminator's Focus. The discriminator has full control over D , and attends to everything but e . That inconsistency of this PDG is what might be called the discriminator's objective: the expected KL divergence from D to the optimal discriminator. If D also disbelieves that any image is equally likely to be fake as real (by choosing $\varphi(e) = -1$), then the inconsistency becomes $-\mathcal{L}^{\text{GAN}}$.

The Generator’s Focus. The generator has control over G . If it ignores D attends only to e , the inconsistency is the Jenson-Shanon Divergence between G and p_{data} . If the generator also disbelieves the discriminator D (i.e., $\varphi(D) = -1$), then the inconsistency becomes $+\mathcal{L}^{\text{GAN}}$.

Standard practice is to use small $\chi(G)$ and large $\chi(D)$, so that the discriminator is well-adapted to the generator.

7.7 Message Passing Algorithms as LIR

Nearly every standard graphical model can be viewed as a factor graph, and correspondingly admits an (approximate) inference procedure known variously as (loopy) belief propagation [56], the generalized distributive law [2], and the sum-product algorithm [58]. It also turns out to be the special case of LIR specialized to factor graphs.

A *factor graph* over a set of variables \mathcal{X} is a set of factors $\Phi = \{\phi_a : \mathbf{X}_a \rightarrow \mathbb{R}_{\geq 0}\}_{a \in \mathcal{A}}$, where each $\mathbf{X}_a \subseteq \mathcal{X}$ is called the *scope* of a . Conversely, for $X \in \mathcal{X}$, let $\partial X := \{a \in \mathcal{A} : X \in \mathbf{X}_a\}$ be the set of factors with X in scope. Φ specifies a distribution $\Pr_\Phi(\mathcal{X}) \propto \prod_a \phi_a(\mathbf{X}_a)$, and corresponds to a PDG

$$m_\Phi = \left\{ \xrightarrow[\alpha, \beta=1]{} \boxed{\mathbf{X}_a} \right\}_{a \in \mathcal{A}}$$

that specifies the same joint distribution \Pr_Φ , when observation and structure are weighted equally (i.e., $\gamma = 1$).

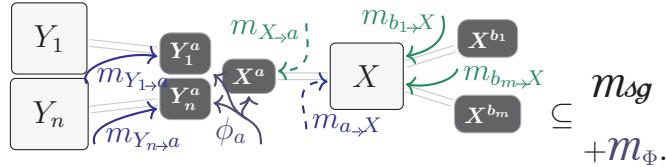
Sum-product belief propagation [58] aims to approximate marginals of \Pr_Θ with only local computations: messages sent between factors and the variables they have in scope. Its state consists of pairs of “messages” $\{m_{X \rightarrow a}, m_{a \rightarrow X}\}$, both

(unnormalized) distributions over X , for each pair (a, X) with $a \in \partial X$, which together form a PDG \mathcal{M}_{Φ} in the same way as the original factor graph. After initialization, belief propagation repeatedly recomputes:

$$m_{X \rightarrow a}(x) := \prod_{b \in \partial X \setminus a} m_{b \rightarrow X}(x) \quad (7.4)$$

$$m_{a \rightarrow X}(x) := \sum_{y \in \mathcal{V}(\mathbf{X}_a \setminus X)} \phi_a(x, y) \prod_{Y \in \mathbf{X}_a \setminus X} m_{Y \rightarrow a}(Y(y)), \quad (7.5)$$

where $Y(y)$ is the value of Y in the joint setting y . Finally, variable marginals $\{b_X\}_{X \in \mathcal{X}}$, which we regard as another PDG, \mathcal{B} , are computed from the messages according to $b_X(x) \propto \prod_{a \in \partial X} m_{a \rightarrow X}(x)$. Observe that every calculation is a (marginal of) a product of factors, and thus amounts to inference in some “local” factor graph. The traditional depiction of messages moving between variables and factors (see [Section 7.A](#)) is not so different from the PDG



Indeed, it can be shown that (7.4,7.5) minimize inconsistency of the dotted components in their appropriate contexts (shown in green and blue above, and formalized in [Section 7.A](#)).

Proposition 7.7.1. *If REFOCUS selects a focus non-deterministically from $\{a \rightarrow X, X \rightarrow a, X\}_{X \in \mathcal{X}, a \in \partial X}$ (details in [Section 7.A](#)), then the possible runs of LIR($\mathcal{M}_{\Phi}, \mathcal{M}_{\Phi} + \mathcal{B}$) are precisely those of BP for different message schedules.*

There are many established variants of this algorithm. Some of them are generated different by clustering factors together—in the language of Koller and Friedman [56], that is to say choosing something other than the Bethe cluster

graph as the basis for message passing. Our analysis immediately applies to these other cluster graphs.

Minka [65] offers a different perspective, in which a broader class of message passing algorithms can be viewed as iteratively adjusting some local context to minimize an α -divergence. We suspect that LIR generalizes these procedure as well—not only because it is similar in spirit, but also because these divergences can be viewed as the degree of inconsistency of a PDG containing two distributions [81].

7.8 Discussion and Future Work

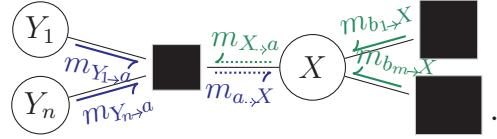
These examples are only the beginning. Our initial investigations suggest that opinion dynamics models, the training process for diffusion models, and much more, are all naturally captured by LIR. The surprising generality of LIR begs some theoretical questions. What assumptions are needed to prove that it reduces overall inconsistency, as is often the case? What are the simplest choices we could make to produce an efficient non-standard algorithm? How expressive is this mode of computation?

It also suggests a novel approach to structured generative modeling: haphazardly assemble a PDG with many variables, existing models, priors, constraints, and data of all shapes and sizes. Then, train new models to predict variables from one another, using LIR (with random refocusing, say). Is this effective? We are excited to find out!

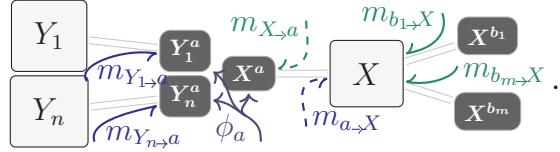
APPENDICES FOR CHAPTER 7

7.A Details on Belief Propagation

The usual schematic illustration of belief propagation [58] looks something like:



This is only a schematic, but the PDG $M_{\delta g}$ can be made to look similar to it. Adding a variable X^a for every pair (X, a) with $X \in \mathbf{X}_a$ along with edges asserting that $X^a = X$, we obtain the equivalent PDG in the main body of the paper:



We now define the views. Modulo a small subtlety, the following is essentially true: Equation (7.4) adjusts the parameters of $C_{X \rightarrow a} := \{m_{X \rightarrow a}\}$ so as to minimize 1-inconsistency in context $A_{X \rightarrow a} := \{m_{b \rightarrow X}\}_{b \in \partial X \setminus a} \cup \{m_{X \rightarrow a}\}$, while (7.5) adjusts $C_{a \rightarrow X} := \{m_{a \rightarrow X}\}$ so as to minimize the 1-inconsistency in context $A_{a \rightarrow X} := \{\phi_a, m_{a \rightarrow X}\} \cup \{m_{Y \rightarrow a}\}_{Y \in \mathbf{X}_a \setminus X}$.

The only wrinkle is that we do not want to attend to the structural aspect of a message e that we are updating—that is, we must select φ so as to ignore its causal weight α_e . Intuitively: when we are updating some message e , we are interested in summarizing information in the other messages (both observational and causal information), purely with an observation.

More precisely, the foci

$$\mathbf{F} := \left\{ (\varphi_j, \chi_j) : j \in \bigcup_{\substack{a \in \mathcal{A} \\ X \in \mathbf{X}_a}} \left\{ a \rightarrow X, X \rightarrow a, X \right\}, \right\}$$

are indexed by messages and variables, and defined as follows. The attention mask φ_j is given by:

$$\varphi_j(a) := \begin{cases} \begin{pmatrix} 1 \\ 1 \end{pmatrix} & \text{if } a \in A_j \setminus C_j \\ \begin{pmatrix} 1 \\ 0 \end{pmatrix} & \text{if } a \in C_j \\ \begin{pmatrix} 0 \\ 0 \end{pmatrix} & \text{otherwise} \end{cases},$$

where $\begin{pmatrix} \phi_1 \\ \phi_2 \end{pmatrix}$ scales β_a by ϕ_1 and α_a by ϕ_2 . Finally, full control over C_j means defining

$$\chi_j(a) := \begin{cases} \infty & \text{if } a \in C_j \\ 0 & \text{otherwise.} \end{cases}$$

With these definitions, [Proposition 7.7.1](#) follows easily. Continue to [Section 7.B](#) for proofs!

7.B Proofs

First, some extra details for [Theorem 7.3.1](#). By parameteriations \mathbb{P} log-concave, we mean that, for every $a \in \mathcal{A}$, and $(s, t) \in \mathcal{V}(S_a, T_a)$, the function

$$\theta \mapsto -\log \mathbb{P}_a^\theta(T_a = t \mid S_a = a) : \Theta_a \rightarrow [0, \infty]$$

is convex. This is true for many families of distributions of interest. For example, if S_a, T_a is discrete, and the cpd is parameterized by stochastic matrices $\mathbf{P} = [p_{s,t}] \in [0, 1]^{\mathcal{V}(S_a, T_a)}$, then

$$-\log \mathbb{P}_a^\mathbf{P}(T_a = t \mid S_a = s) = -\log(p_{s,t})$$

which is clearly convex in \mathbf{P} .

To take another example: if \mathbb{P}_a is linear Gaussian, i.e., $\mathbb{P}_a(T|S) = \mathcal{N}(T|\mathbf{A}s + b, \sigma^2)$, parameterized by $(\mathbf{A}, b, 1/\sigma^2)$, then

$$-\log \mathbb{P}_a^{(\mathbf{A}, b, \sigma^2)}(t|s) = -\frac{1}{2} \log \frac{2\pi}{\sigma^2} + \frac{1}{2} \left(\frac{t - \mathbf{A}s + b}{\sigma} \right)^2$$

which is convex in $(\mathbf{A}, b, \frac{1}{\sigma^2})$. Now, for the proof.

Theorem 7.3.1. *If \mathbb{P} is log-concave, then for small enough γ , the map $\theta \mapsto \langle\!\langle \varphi \odot (\mathcal{C}tx + m(\theta)) \rangle\!\rangle_\gamma$ is convex.*

Proof. By definition,

$$\langle\!\langle \varphi \odot (\mathcal{C}tx + m(\theta)) \rangle\!\rangle_\gamma = \inf_\mu \left\{ OInc_{ctx}(\mu) + \gamma SDef_{ctx}(\mu) + \gamma SDef_{m(\theta)}(\mu) + OInc_{m(\theta)}(\mu) \right\}.$$

Only the final term actually depends on θ , though—recall that $SDef_{m(\theta)}$ depends only on the structure of the hypergraph (and the weights α), and not the parameters of the cpds. Thus, we can write $F(\mu)$ for the first three terms.

For all of our examples, and indeed, if γ is chosen small enough, the sum of the two terms is convex in μ [82]. Then we have

$$\begin{aligned}\langle\varphi \odot (Ct\mathbf{x} + \mathbf{m}(\theta)) \rangle_\gamma &= \inf_{\mu} \left(F(\mu) + \mathbb{E}_{\mu} \left[\sum_{S \xrightarrow{a} T} \beta_a \log \frac{\mu(T|S)}{\mathbb{P}_a^\theta(T|S)} \right] \right) \\ &= \inf_{\mu} \left(F(\mu) + \mathbb{E}_{\mu} \left[\sum_{S \xrightarrow{a} T} \beta_a \log \frac{\mu(T|S)}{\lambda(T|S)} \right] + \underbrace{\mathbb{E}_{\mu} \left[\sum_{S \xrightarrow{a} T} \beta_a \log \frac{\lambda(T|S)}{\mathbb{P}_a^\theta(T|S)} \right]}_{\text{third term}} \right)\end{aligned}$$

The second term is then entropy (relative to the base distribution), which is convex in μ . The first term, $F(\mu)$, is convex in μ as well, and neither depend on θ . The final term is linear in μ . Since \mathbb{P} is log-convex in θ , we know that $(\log \frac{\lambda(t|s)}{\mathbb{P}_a^\theta(t|s)})$ is convex in θ . It follows that the third term is a positive linear combination of expectations that are all convex in θ , and hence itself convex in θ . Because the first two terms do not depend on θ and are convex in μ , they are jointly convex in (μ, θ) . And, as we have seen, the third term is linear in μ and convex in θ , so it is also jointly convex in (μ, θ) . Thus, the sum of all three terms in the infimum is jointly convex in (θ, μ) . Taking an infimum over μ pointwise, the result is still convex in θ [13]. \square

Proposition 7.7.1. *If REFOCUS selects a view non-deterministically from $\{a \rightarrow X, X \rightarrow a, X\}_{X \in \mathcal{X}, a \in \partial X}$ with φ, χ as above, and $\gamma = 1$, then the possible runs of LIR($\mathbf{m}_\Phi, \mathbf{msg} + \mathcal{B}$) are precisely those of BP for different message schedules.*

Proof. When $\gamma = 1$, and $\alpha, \beta = 1$ for all of the input factors, then the optimal distribution μ^* that realizes the infimum is just the product of factors. It follows that any distribution that has those marginals will minimize the observational inconsistency.

The different orders that the (7.4), and (7.5) can be ordered for different adjacent pairs (a, X) correspond to both the message passing schedules, and to the

possible view selections of LIR.

□

Part III

Algorithms, Logic, and Complexity

CHAPTER 8

INFERENCE FOR PDGS, VIA EXPONENTIAL CONIC PROGRAMMING

8.1 Introduction

Probabilistic dependency graphs (PDGs) [82], form a very general class of probabilistic graphical models, that includes not only Bayesian Networks (BNs) and Factor Graphs (FGs), but also more recent statistical models built out of neural networks, such as Variational Autoencoders (VAEs) [52]. ■■■■■ PDGs can also capture inconsistent beliefs, and provide a useful way to measure the degree of this inconsistency; for a VAE, this is the loss function used in training [81]. PDGs have some significant advantages over other representations of probabilistic information. Their flexibility allows them to model beliefs that BNs cannot, such as information from independent studies of the same variable (perhaps with different controls, yielding probabilistic observations $p(Y|X)$ and $q(Y|Z)$). PDGs can deal gracefully with conflicting information from multiple sources. Every subcomponent of a PDG has probabilistic meaning, independent of the other components; compared to FGs, this makes PDGs more interpretable. But up to now, there has been no practical way to do inference for PDGs—that is, to answer questions of the form “what is the probability of Y given X ?”. This paper presents the first algorithm to do so.

Before discussing our algorithm, we must discuss what it even means to do inference for a PDG. A BN or FG represents a unique joint distribution. Thus, for example, when we ask “what is the probability of Y given that $X=x$?”, in a BN, we mean “what is $\mu(Y|X=x)$?” for the probability measure μ that the BN represents. But a PDG might, in general, represent more than just one

distribution.

Like a BN, a PDG encodes two types of information: “structural” information about the independence of causal mechanisms, and “observational” information about conditional probabilities. Unlike in a BN, the two can conflict in a PDG. Corresponding to these two types of information, a PDG has two loss functions, which quantify how far a distribution μ is from modeling the information of each type. Given a number $\hat{\gamma} \in [0, 1]$ indicating the importance of structure relative to observation, we take the $\hat{\gamma}$ -semantics of a PDG to be the set of distributions that minimize the appropriate convex combination of losses. We also consider the 0^+ -semantics: the limiting case that arises as $\hat{\gamma}$ goes to zero (which focuses on observation, using structure only to break ties). This set can be shown to contain precisely one distribution for PDGs satisfying a mild regularity condition (required by definition by Richardson and Halpern); we call such PDGs *proper*. Thus, we have a parameterized family of inference notions: to do $\hat{\gamma}$ -inference, for $\hat{\gamma} \in [0, 1] \cup \{0^+\}$, is to answer queries in a way that is true of all distributions in the $\hat{\gamma}$ -semantics.

If there are distributions fully consistent with both the observational and the structural information in a PDG m , then for $\hat{\gamma} \in (0, 1) \cup \{0^+\}$, all notions of $\hat{\gamma}$ -inference coincide. ■ If m is also proper, this means there is a single distribution μ_m that minimizes both loss functions, in which case we want to answer queries with respect to μ_m no matter how we weight observational and structural information. Moreover, if m represents a BN, then μ_m is the distribution represented by the BN. However, if there is no distribution that is consistent with both types of information, then the choice of $\hat{\gamma}$ matters.

Since PDGs subsume BNs, and inference for BNs is already NP-hard, the same

must be true of PDGs. At a high level, the best we could hope for would be tractability on the restricted class of models on which inference has traditionally been tractable—that is, a polynomial algorithm for models whose underlying structure has *bounded treewidth* (see [Section 8.2](#) for formal definitions). That is indeed what we have. More precisely, we show that 0^+ -inference and $\hat{\gamma}$ -inference for small $\hat{\gamma}$ can be done for discrete PDGs of bounded treewidth containing N variables in $\tilde{O}(N^4)$ time.

Our algorithm is based on a line of recent work in convex programming that establishes polynomial-time for a class of optimization problems called *exponential conic programs* [5, 88, 69]. Our contribution is to show that the problem of inference in a PDG of bounded treewidth can be efficiently converted to a (sequence of) exponential conic program(s), at which point it can be solved with a commercial solver (e.g., ApS [4]) in polynomial time. The direct appeal to a solver allows us to benefit from the speed and reliability of such highly optimized solvers, and also from future improvements in exponential conic optimization. Thus, our result is not only a theoretical one, but practical as well.

Beyond its role as a probabilistic model, a PDG is also of interest for its degree of inconsistency—that is, the minimum value of its loss function. As shown by Richardson [81], many loss functions and statistical divergences can be viewed as measuring the inconsistency of a PDG that models the context appropriately. This makes calculating this minimum value of interest—but up to now, there has been no way to do so. There is a deep connection between this problem and PDG inference; for now, we remark that this number is a byproduct of our techniques.



Contributions. We provide the first algorithm for inference in a PDG; in addi-

tion, it calculates a PDG’s degree of inconsistency. We prove that our algorithm is correct, and also fixed-parameter tractable: for PDGs of bounded treewidth, it runs in polynomial time. We also prove that PDG inference and inconsistency calculation are equivalent problems. Our algorithm reduces inference in PDGs to exponential conic programming in a way that can be offloaded to powerful existing solvers. We provide an implementation of this reduction in a standard convex optimization framework, giving users an interface between such solvers and the standard PDG Python library. Finally, we evaluate our implementation. The results suggest our method is faster and significantly more reliable than simple baseline approaches.

8.2 Preliminaries & Related Work

Vector Notation. For us, a vector is a map from a finite set S , called its *shape*, to the extended reals $\bar{\mathbb{R}} := \mathbb{R} \cup \{\infty\}$. We write $\mathbf{u} := [u_i]_{i \in S}$ to define a vector \mathbf{u} by its components. ■ Vectors of the same shape can be added (+), partially ordered (\leq), or multiplied (\odot) pointwise as usual. $\mathbf{1}$ denotes an all-ones vector, of a shape implied by context. ■

■

Probabilities. We write ΔS to denote the set of probability distributions over a finite set S . Every variable X can take on values from a finite set $\mathcal{V}X$ of possible values. We can regard sets of variables \mathbf{X} as variables themselves, with $\mathcal{V}\mathbf{X} = \Pi_{X \in \mathbf{X}} \mathcal{V}X$. A conditional probability distribution (cpd) $p(Y|X)$ is a map $p : \mathcal{V}X \rightarrow \Delta \mathcal{V}Y$ assigning to each $x \in \mathcal{V}X$ a probability distribution $p(Y|x) \in \Delta \mathcal{V}Y$, which is shorthand for $p(Y|X=x)$. Given a distribution μ over

(the values of) a set of variables including X and Y , we write $\mu(X)$ for its marginal on X , and $\mu(Y|X)$ for the cpd obtained by conditioning on X and marginalizing to Y . We also refer to μ 's entropy $H(\mu) := \mathbb{E}_\mu[\log \frac{1}{\mu}]$ and conditional entropy $H_\mu(Y|X) := \mathbb{E}_\mu[\log \frac{1}{\mu(Y|X)}]$ of Y given X .

Hypergraphs and Treewidth. A hypergraph (V, \mathcal{E}) is a set V of vertices and a set \mathcal{E} of *hyperedges*, which correspond to subsets of V . An ordinary graph may be viewed as the special case in which every hyperedge contains two vertices.

Definition 8.2.1. A *directed hypergraph* (N, \mathcal{A}) is a set of nodes N , and a set of (*hyper*)arcs \mathcal{A} , each $a \in \mathcal{A}$ of which is associated with a set of source nodes $S_a \subseteq N$, and target nodes $T_a \subseteq N$. We also write $S^a \rightarrow T \in \mathcal{A}$ to specify an arc a together with its sources $S = S_a$ and targets $T = T_a$. □

A directed hypergraph can be viewed as a hypergraph by joining each source and target set, thereby “forgetting” the direction of the arrow. Thus, notions defined for undirected hypergraphs (like that of treewidth, which we now review), can be applied to directed hypergraphs as well.

Many problems that are intractable for general graphs are tractable for trees, and some graphs are closer to being trees than others. A tree decomposition of a (hyper)graph $G = (V, \mathcal{E})$ is a tree $(\mathcal{C}, \mathcal{T})$ whose vertices $C \in \mathcal{C}$, called *clusters*, are subsets of V such that:

1. every vertex $v \in V$ and every hyperedge $E \in \mathcal{E}$ is contained in at least one cluster, and
2. every cluster D along the unique path from C_1 to C_2 in \mathcal{T} , contains $C_1 \cap C_2$.

The *width* of a tree decomposition is one less than the size of its largest cluster,

and the *treewidth* of a (hyper)graph G is the smallest possible width of any tree decomposition of G . It is NP-hard to determine the tree-width of a graph, but if the tree-width is known to be bounded above, a tree decomposition may be constructed in linear time [12]. For graphs of bounded tree-width, many problems (indeed, any problem expressible in a certain second-order logic [19]) can be solved in linear time. This is also true of inference in standard graphical models.

Graphical Models and Inference. A *graphical model structure* is a (directed) (hyper)graph whose vertices \mathcal{X} are variables, and whose (hyper)edges somehow indicate local influences between variables. A *probabilistic graphical model*, or simply “graphical model”, is a graphical model structure together with quantitative information about these local influences. Semantically, a graphical model \mathcal{M} typically represents a joint probability distribution $\Pr_{\mathcal{M}} \in \Delta^{\mathcal{V}\mathcal{X}}$ over its variables. ■ Inference for \mathcal{M} is then the ability to calculate cpds $\Pr_{\mathcal{M}}(Y|X=x)$, where $X, Y \subset \mathcal{X}$ and $x \in \mathcal{V}X$.



Many inference algorithms (such as belief propagation), when applied to tree-like graphical models, run in linear time and are provably correct. If the same algorithms are naïvely applied to graphs with cycles (as in loopy belief propagation), then they may not converge, and even if they do, may give incorrect (or even inconsistent) answers [93]. Nearly all exact inference algorithms (including variable elimination [9], message-passing with [60] and without division [86], among others [92]) effectively construct a tree decomposition, and can be viewed as running on a tree [56, §9-11]. Indeed, under widely believed assumptions, every class of graphical models for which (exact) inference is *not* NP-hard has

bounded treewidth [14].

Given a tree decomposition $(\mathcal{C}, \mathcal{T})$ of the underlying model structure, many of these algorithms use a standard data structure that we will call a *tree marginal*, which is a collection $\mu = \{\mu_C(C)\}_{C \in \mathcal{C}}$ of probabilities over the clusters [56, §10]. A tree marginal μ is said to be *calibrated* if neighboring clusters' distributions agree on the variables they share. In this case, μ determines a joint distribution by

$$\Pr_{\mu}(\mathcal{X}) = \prod_{C \in \mathcal{C}} \mu_C(C) / \prod_{(C-D) \in \mathcal{T}} \mu_C(C \cap D), \quad (8.1)$$

which has the property that $\Pr_{\mu}(C) = \mu_C$ for all $C \in \mathcal{C}$. ■ A calibrated tree marginal summarizes the answers to queries about \Pr_{μ} [see 56, §10.3.3]. Therefore, to answer probabilistic queries with respect to a distribution μ , it suffices to find a calibrated tree marginal μ that represents μ , and appeal to standard algorithms.

Probabilistic Dependency Graphs. Our presentation of PDGs is slightly different from (but equivalent to) that of Richardson and Halpern [82], which the reader is encouraged to consult for more details and intuition. At a high level, a PDG is just a collection of cpds and causal assertions, weighted by confidence. More precisely:

Definition 8.2.2. A PDG $\mathcal{M} = (\mathcal{X}, \mathcal{A}, \mathbb{P}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ is a directed hypergraph $(\mathcal{X}, \mathcal{A})$ whose nodes are variables, together with probabilities \mathbb{P} and confidence vectors $\boldsymbol{\alpha} = [\alpha_a]_{a \in \mathcal{A}}$ and $\boldsymbol{\beta} = [\beta_a]_{a \in \mathcal{A}}$, so that each $S \xrightarrow{a} T \in \mathcal{A}$ is associated with:

- a conditional probability distribution $\mathbb{P}_a(T|S)$ on the target variables given values of the source variables,
- a weight $\beta_a \in \bar{\mathbb{R}}$ indicating the modeler's confidence in the cpd $\mathbb{P}_a(T|S)$, ■

and

- a weight $\alpha_a \in \mathbb{R}$ indicating the modeler's confidence in the functional dependence of T on S expressed by a . ■

■ If $\beta \geq 0$ and $\alpha_a > 0$ implies $\beta_a > 0$, we write $\beta \gg \alpha$ and call m *proper*. Note that $\beta \gg \alpha$ if $\beta > 0$. □

One significant advantage of PDGs is their modularity: we can combine the information in m_1 and m_2 by taking the union of their variables and the disjoint union of their arcs (and associated data) to get a new PDG, denoted $m_1 + m_2$.

■■ As mentioned in the introduction, a PDG contains two types of information: “structural” information, in the hypergraph \mathcal{A} and weights α , and “observational” data, in the cpds \mathbb{P} and weights β . PDG semantics are based on two scoring functions that quantify discrepancy between each type of information and a distribution $\mu \in \Delta \mathcal{VX}$ over its variables.

The *observational incompatibility* of μ with m , which can be thought of as a “distance” between μ and the cpds of m , is given by the weighted sum of relative entropies:

$$OInc_m(\mu) := \sum_{S \xrightarrow{a} T \in \mathcal{A}} \beta_a D\left(\mu(T, S) \parallel \mathbb{P}_a(T|S)\mu(S)\right).$$

Under a standard interpretation of the relative entropy $D(\mu \parallel p) = \mathbb{E}_\mu[\log \frac{\mu}{p}]$, $OInc_m$ measures the excess cost of using codes optimized for the cpds of m (weighted by their confidences), when reality is distributed according to μ .

The second scoring function measures the extent to which μ is incompatible with a causal picture consisting of independent mechanisms along each hyperarc.

This is captured by the *structural incompatibility* (of μ with \mathcal{M}), and given by

$$SInc_{\mathcal{M}}(\mu) := \left(\sum_{S \xrightarrow{a} T \in \mathcal{A}} \alpha_a H_{\mu}(T|S) \right) - H(\mu).$$

Note that $SInc_{\mathcal{M}}$ does not depend on the cpds of \mathcal{M} , nor the possible values of the variables; it is defined purely in terms of the weighted hypergraph structure $(\mathcal{A}, \boldsymbol{\alpha})$.

If the observational and structural information conflict, then the distribution(s) that best represent a PDG will depend on the importance of structure relative to observation, ■ as captured by a trade-off parameter $\hat{\gamma} \in [0, 1]$ that controls the convex combination $(1 - \hat{\gamma}) OInc + \hat{\gamma} SInc$. So as to simplify the math and match the notation in previous work (2021, 2022), we mostly use a rescaled variant with a different parameterization. Using $\gamma := \hat{\gamma}/(1 - \hat{\gamma}) \in [0, \infty]$, define the overall scoring function:

$$\begin{aligned} \llbracket \mathcal{M} \rrbracket_{\gamma}(\mu) &:= OInc_{\mathcal{M}}(\mu) + \gamma SInc_{\mathcal{M}}(\mu) \\ &= \frac{1}{1 - \hat{\gamma}} \left((1 - \hat{\gamma}) OInc_{\mathcal{M}}(\mu) + \hat{\gamma} SInc_{\mathcal{M}}(\mu) \right) \\ &= \mathbb{E}_{\mu} \left[\sum_{S \xrightarrow{a} T \in \mathcal{A}} \log \frac{\mu(T|S)^{\beta_a - \gamma \alpha_a}}{\mathbb{P}_a(T|S)^{\beta_a}} \right] - \gamma H(\mu). \end{aligned} \tag{8.2}$$

Let $\llbracket \mathcal{M} \rrbracket_{\gamma}^* := \arg \min_{\mu} \llbracket \mathcal{M} \rrbracket_{\gamma}(\mu)$ denote the set of optimal distributions at a particular value γ . One natural conception of inference in PDGs is then parameterized by $\hat{\gamma}$: to do $\hat{\gamma}$ -inference in \mathcal{M} is to respond to probabilistic queries in a way that is sound with respect to every $\mu \in \llbracket \mathcal{M} \rrbracket_{\gamma}^*$. It is not too difficult to see that when $\beta \geq \gamma \alpha$, (8.2) is strictly convex, which ensures that $\llbracket \mathcal{M} \rrbracket_{\gamma}^*$ is a singleton. This paper demonstrates that $\hat{\gamma}$ -inference is tractable for such PDGs. ■

The limiting behavior of the $\hat{\gamma}$ -semantics as $\hat{\gamma} \rightarrow 0$, which we denote $\llbracket \mathcal{M} \rrbracket_{0^+}^*$ and call the 0^+ -semantics, has some special properties. If \mathcal{M} is proper, then $\llbracket \mathcal{M} \rrbracket_{0^+}^*$ contains precisely one distribution. This distribution intuitively reflects an

extreme empirical view: observational data trumps causal structure. Note that in the absence of a causal picture ($\alpha = 0$), this corresponds to the well-established practice of selecting the maximum entropy distribution consistent with some observational constraints [48]. One should be careful to distinguish $[\![\mathcal{M}]\!]_{0+}^*$ from $[\![\mathcal{M}]\!]_0^*$, the set of distributions that minimize $OInc_m$; the latter set includes $[\![\mathcal{M}]\!]_{0+}^*$ [82, Prop 3.4], but may also contain other distributions. ■ This paper also shows how to efficiently answer queries with respect to the unique distribution in $[\![\mathcal{M}]\!]_{0+}^*$, which we call 0^+ -*inference*.

Given a PDG \mathcal{M} , the smallest possible value of its scoring function, $\langle\!\langle \mathcal{M} \rangle\!\rangle_\gamma := \inf_\mu [\![\mathcal{M}]\!]_\gamma(\mu)$, is known as its γ -inconsistency and is interesting in its own right: $\langle\!\langle \cdot \rangle\!\rangle_\gamma$ is arguably a “universal” loss function [81].

Interior-Point Methods and Convex Optimization. Interior-point methods provide an iterative way of approximately solving linear programs in polynomial time [50]. With the theory of “symmetric cones”, these methods were extended in the 1990s to handle second-order cone programs (SOCPs) and semidefinite programs (SDPs), which allow more expressive constraints. But the constraints that these methods can handle are insufficient for our purposes. We need what have been called *exponential cone constraints*. The *exponential cone* is the convex set

$$\begin{aligned} K_{\text{exp}} := & \{(x_1, x_2, x_3) : x_1 \geq x_2 e^{x_3/x_2}, x_2 > 0\} \\ & \cup \{(x_1, 0, x_3) : x_1 \geq 0, x_3 \leq 0\} \quad \subset \bar{\mathbb{R}}^3. \end{aligned}$$

■ Let $K := K_{\text{exp}}^p \times [0, \infty]^q \subset \bar{\mathbb{R}}^n$ be a product of p exponential cones and $q = n - 3k$ non-negative orthants. An *exponential conic program* is then an optimization problem of the form

$$\underset{\mathbf{x}}{\text{minimize}} \quad \mathbf{c}^\top \mathbf{x} \quad \text{subject to} \quad A\mathbf{x} = \mathbf{b}, \quad \mathbf{x} \in K, \tag{8.3}$$

where $\mathbf{c} \in \bar{\mathbb{R}}^n$ is some cost vector, the function $\mathbf{x} \mapsto \mathbf{c}^\top \mathbf{x}$ is called the *objective*,

and $\mathbf{b} \in \bar{\mathbb{R}}^m$, $\mathbf{A} \in \bar{\mathbb{R}}^{m \times n}$ encode linear constraints. Nesterov, Todd, and Ye [69] first established that such problems can be solved in polynomial time, but incur double the memory and eight times the time, compared to the symmetric counterparts. These drawbacks were eliminated in [88]. The algorithm that seems to display the best empirical performance [21], however, was only recently shown to run in polynomial time [5].

Disciplined Convex Programming [34] is a compositional approach to convex optimization that imposes certain restrictions on how problems can be specified. A problem conforming to those rules is said to be *dcp*, and can be efficiently compiled to a standard form [1], which in our case is an exponential conic program. Only two rules are relevant to us: a constraint of the form $(x, y, z) \in K_{\text{exp}}$ is dcp iff x, y , and z are affine transformations of the optimization variables, and a linear program augmented with dcp constraints is dcp. Because all the optimization problems that we give are of this form, we can easily compile them to exponential conic programs even if they do not exactly conform to (8.3).



8.3 Inference as a Convex Program

Here is an obvious, if inefficient, way of calculating $\Pr_{\mathcal{M}}(Y|X=x)$ in a probabilistic model \mathcal{M} . First compute an explicit representation of the joint distribution $\Pr_{\mathcal{M}} \in \Delta \mathcal{V}\mathcal{X}$, then marginalize to $\Pr_{\mathcal{M}}(X, Y)$ and condition on $X=x$. For a factor graph or BN, each step is straightforward; the problem is the exponential time and space required to represent $\Pr_{\mathcal{M}}(\mathcal{X})$ explicitly. A key feature of inference algorithms for BNs and FGs is that they do not represent joint distributions in this

way. For PDGs, though, it is not obvious that we can calculate the $\hat{\gamma}$ -semantics, even if we know it is unique, and we ignore the space required to represent it (as we do in this section). Note that $\hat{\gamma}$ -inference is already an optimization problem by definition:

$$\underset{\mu}{\text{minimize}} \quad \llbracket \mathbf{m} \rrbracket_{\hat{\gamma}}(\mu) \quad \text{subject to} \quad \mu \in \Delta \mathcal{VX}.$$

For small enough γ , it is even convex. But can we solve it efficiently? With exponential cone constraints, the answer is yes, as we show in [Section 8.3.2](#). Moreover, we can compute the 0^+ -semantics with a sequence of two exponential conic programs ([Section 8.3.3](#)). To give a flavor of our constructions and ease into the more complicated ones, we begin by minimizing $OInc$, the simpler of the two scoring functions.

8.3.1 Minimizing Incompatibilty ($\gamma = 0$)

When $\gamma = 0$, we want to find minimizers of $OInc$, which is a weighted sum of conditional relative entropies. There is a straightforward connection between the exponential cone and relative entropy: if $\mathbf{m}, \mathbf{p} \in \Delta\{1, \dots, n\} \subset \mathbb{R}^n$ are points on a probability simplex, then $(-\mathbf{u}, \mathbf{m}, \mathbf{p}) \in K_{\exp}^n$ if and only if \mathbf{u} is an upper bound on $\mathbf{m} \log \frac{\mathbf{m}}{\mathbf{p}}$, the pointwise contribution to relative entropy at each outcome. Thus, perhaps unsurprisingly, we can use an exponential conic program to find minimizers of $OInc$. If all beliefs are unconditional and over the same space, the construction is standard; we review it here, so that we can build upon it.

Warm-up. Consider a PDG with only one variable X with $\mathcal{VX} = \{1, \dots, n\}$.

■ Suppose further that every arc $j \in \mathcal{A} = \{1, \dots, k\}$ has $T_j = \{X\}$ and $S_j = \emptyset$. Then each $\mathbb{P}_j(X)$ can be identified with a vector $\mathbf{p}_j \in [0, 1]^n$, and all k of them can conjoined to form a matrix $\mathbf{P} = [p_{ij}] \in [0, 1]^{n \times k}$. Similarly, a candidate

distribution μ can be identified with $\mathbf{m} \in [0, 1]^n$. Now consider a matrix $\mathbf{U} = [u_{i,j}] \in \bar{\mathbb{R}}^{n \times k}$ that, intuitively, gives an upper bound on the contribution to $OInc$ due to each edge and value of X . Observe that

$$\begin{aligned}
& (-\mathbf{U}, [\mathbf{m}, \dots, \mathbf{m}], \mathbf{P}) \in K_{\exp}^{n \times k} \\
\iff & \forall i, j. \ u_{ij} \geq m_i \log(m_i/p_{ij}) \\
\implies & \forall j. \sum_i u_{ij} \geq D(\mu \| p_j) \\
\implies & \sum_{i,j} \beta_j u_{ij} \geq \sum_j \beta_j D(\mu \| p_j) \\
\iff & \mathbf{1}^\top \mathbf{U} \boldsymbol{\beta} \geq OInc(\mu).
\end{aligned} \tag{8.4}$$

So now, if (\mathbf{U}, \mathbf{m}) is a solution to the convex program

$$\begin{aligned}
& \underset{\mathbf{m}, \mathbf{U}}{\text{minimize}} \quad \mathbf{1}^\top \mathbf{U} \boldsymbol{\beta} \quad \text{subject to} \quad \mathbf{1}^\top \mathbf{m} = 1, \\
& \quad \quad \quad (-\mathbf{U}, [\mathbf{m}, \dots, \mathbf{m}], \mathbf{P}) \in K_{\exp}^{n \times k},
\end{aligned}$$

then (a) the objective value $\mathbf{1}^\top \mathbf{U} \boldsymbol{\beta}$ equals the inconsistency $\langle\!\langle \mathcal{M} \rangle\!\rangle_0$, and (b) $\mu \in \llbracket \mathcal{M} \rrbracket_0^*$, meaning μ minimizes $OInc_{\mathcal{M}}$.

The General Case. We now show how the same construction can be used to find a distribution $\mu \in \llbracket \mathcal{M} \rrbracket_0^*$ for an arbitrary PDG $\mathcal{M} = (\mathcal{X}, \mathcal{A}, \mathcal{P}, \boldsymbol{\alpha}, \boldsymbol{\beta})$. ■ To further simplify the presentation, for each arc $a \in \mathcal{A}$, let $\mathcal{V}_a := \mathcal{V}(S_a, T_a)$ denote all joint settings of a 's source and target variables, and write $\mathcal{VA} := \sqcup_{a \in A} \mathcal{V}_a = \{(a, s, t) : a \in \mathcal{A}, (s, t) \in \mathcal{V}(S_a, T_a)\}$ for the set of all choices of an arc together with values of its source and target. For each $a \in \mathcal{A}$, we can regard $\mu(T_a, S_a)$ and $\mu(S_a) \mathbb{P}_a(T_a | S_a)$, both distributions over $\{S_a, T_a\}$, as vectors of shape \mathcal{V}_a . As before, we introduce an optimization variable \mathbf{u} that packages together all of the relevant pointwise upper bounds. To that end, consider a vector $\mathbf{u} = [u_{a,s,t}] \in \bar{\mathbb{R}}^{\mathcal{VA}}$ in the

optimization problem ■

$$\begin{aligned} & \underset{\mu, \mathbf{u}}{\text{minimize}} \quad \sum_{(a,s,t) \in \mathcal{V}\mathcal{A}} \beta_a u_{a,s,t} \\ & \text{subject to} \quad \mu \in \Delta \mathcal{V}\mathcal{X}, \\ & \forall a \in \mathcal{A}. \left(-\mathbf{u}_a, \mu(T_a, S_a), \mathbb{P}_a(T_a | S_a) \mu(S_a) \right) \in K_{\text{exp}}^{\mathcal{V}a}. \end{aligned} \tag{8.5}$$

where $\mathbf{u}_a = [u_{a,s,t}]_{(s,t) \in \mathcal{V}a}$ consists of those components of \mathbf{u} associated with arc a . Note that the marginals $\mu(S_a, T_a)$ and $\mu(S_a)$ are affine transformations of μ , so (8.5) is dcp. A straightforward generalization of the logic in (8.4) gives us:

Proposition 8.3.1. If (μ, \mathbf{u}) is a solution to (8.5), then $\mu \in [\![\mathbf{m}]\!]_0^*$ and

$$\sum_{(a,s,t) \in \mathcal{V}\mathcal{A}} \beta_a u_{a,s,t} = \langle\!\langle \mathbf{m} \rangle\!\rangle_0.$$

Thus, a solution to (8.5) encodes a distribution that minimizes $OInc$, and the (0-)inconsistency of \mathbf{m} . This is a start, but to do 0^+ -inference, among the minimizers of $OInc$ we must find the unique distribution in $[\![\mathbf{m}]\!]_{0+}^*$, while for $\hat{\gamma}$ -inference ($\hat{\gamma} > 0$), we need to find the optimizers of $[\![\mathbf{m}]\!]_\gamma^*$. Either way, we must consider $SInc$ in addition to $OInc$.

8.3.2 γ -Inference for small $\gamma > 0$

When $\gamma > 0$ is small enough, the scoring function (8.2) is not only convex, but admits a straightforward representation as an exponential conic program. To see this, note that (8.2) can be rewritten [82, Prop 4.6] as:

$$\begin{aligned} [\![\mathbf{m}]\!]_\gamma(\mu) = & -\gamma H(\mu) - \sum_{a \in \mathcal{A}} \beta_a \mathbb{E}_{\mu} \log \mathbb{P}_a(T_a | S_a) \\ & + \sum_{a \in \mathcal{A}} (\gamma \alpha_a - \beta_a) H_{\mu}(T_a | S_a). \end{aligned} \tag{8.6}$$

The first term, $-\gamma H(\mu)$, is strictly convex and has a well-known translation into an exponential cone constraint; the second one linear in μ . If $0 < \gamma \leq \min_a \frac{\beta_a}{\alpha_a}$, then every summand of the last term is a negative conditional entropy, and can be captured by an exponential cone constraint. The only wrinkle is that it is possible for a user to specify that some $\mathbb{P}_a(t | s) = 0$, in which case the linear term is undefined. The result is a requirement that $\mu(s, t) = 0$ at such points, which we can instead encode directly with linear constraints. To do this formally, divide \mathcal{VA} into two parts: $\mathcal{VA}^+ := \{(a, s, t) \in \mathcal{VA} : \mathbb{P}_a(t|s) > 0\}$ and $\mathcal{VA}^0 := \{(a, s, t) \in \mathcal{VA} : \mathbb{P}_a(t|s) = 0\}$. Armed with this notation, consider upper bound vectors $\mathbf{u} = [u_{a,s,t}]_{(a,s,t) \in \mathcal{VA}}$ and $\mathbf{v} = [v_w]_{w \in \mathcal{VX}}$, in the following optimization problem:

$$\begin{aligned} \underset{\mu, \mathbf{u}, \mathbf{v}}{\text{minimize}} \quad & \sum_{(a,s,t) \in \mathcal{VA}} (\beta_a - \alpha_a \gamma) u_{a,s,t} + \gamma \sum_{w \in \mathcal{VX}} v_w \\ & - \sum_{(a,s,t) \in \mathcal{VA}^+} \alpha_a \gamma \mu(S_a=s, T_a=t) \log \mathbb{P}_a(t|s) \\ \text{subject to} \quad & \mu \in \Delta \mathcal{VX}, \quad (-\mathbf{v}, \mu, \mathbf{1}) \in K_{\text{exp}}^{\mathcal{VX}}, \\ & \forall a \in \mathcal{A}. \left(-\mathbf{u}_a, \mu(T_a, S_a), \mathbb{P}_a(T_a|S_a) \mu(S_a) \right) \in K_{\text{exp}}^{\mathcal{V}_a}, \\ & \forall (a, s, t) \in \mathcal{VA}^0. \mu(S_a=s, T_a=t) = 0. \end{aligned} \tag{8.7}$$

This optimization problem may look complex, but it falls out of (8.6) fairly directly, and gives us what we wanted.

Proposition 8.3.2. *If $(\mu, \mathbf{u}, \mathbf{v})$ is a solution to (8.7), and $\beta \geq \gamma \alpha$, then μ is the unique element of $[\![m]\!]_{\gamma}^*$, and $\langle\!\langle m \rangle\!\rangle_{\gamma}$ equals the objective of (8.7) evaluated at $(\mu, \mathbf{u}, \mathbf{v})$.*

[link to
proof]

8.3.3 Calculating the 0^+ -semantics ($\gamma \rightarrow 0$)

Section 8.3.1 shows how to find a distribution ν that minimizes $OInc$ —but to do 0^+ -inference, we need to find the minimizer that, uniquely among them, best minimizes $SInc$. It turns out this can be done by using ν to construct a second optimization problem. The justification requires two more results; we start by characterizing the minimizers of $OInc$.

Proposition 8.3.3. *If \mathbf{m} has arcs \mathcal{A} and $\beta \geq 0$, the minimizers of $OInc_m$ all have the same conditional marginals along \mathcal{A} . That is, for all $\mu_1, \mu_2 \in [\![\mathbf{m}]\!]_0^*$ and all $S \xrightarrow{a} T \in \mathcal{A}$ with $\beta_a > 0$, we have $\mu_1(T, S)\mu_2(S) = \mu_2(T, S)\mu_1(S)$.¹*

[link to
proof]

As a result, once we find one minimizer ν of $OInc_m$ (e.g., via (8.5)), it suffices to optimize $SInc$ among distributions that have the same conditional marginals along \mathcal{A} that ν does. This presents another problem: $SInc$ is typically not convex. Fortunately, if we constrain to distributions that minimize $OInc$, then it is. Moreover, on this restricted domain, it can be represented with dcp exponential cone constraints.

Proposition 8.3.4. *If $\mu \in [\![\mathbf{m}]\!]_0^*$, then*

$$SInc_m(\mu) = \sum_{w \in \mathcal{VX}} \mu(w) \log \left(\frac{\mu(w)}{\prod_{a \in \mathcal{A}} \nu(T_a(w) | S_a(w))^{\alpha_a}} \right), \quad (8.8)$$

where $\{\nu(T_a | S_a)\}_{a \in \mathcal{A}}$ are the marginals along the arcs \mathcal{A} shared by all distributions in $[\![\mathbf{m}]\!]_0^*$ (per Proposition 8.3.3), and $S_a(w), T_a(w)$ are the values of variables S_a and T_a in w .

¹Intuitively, this asserts $\mu_1(T_a | S_a) = \mu_2(T_a | S_a)$, but also handles cases where some $\mu_1(S_a = s)$ or $\mu_2(S_a = s)$ equals zero.

If we already know a distribution $\nu \in \llbracket m \rrbracket_0^*$, perhaps by solving (8.5), then the denominator of (8.8) does not depend on μ and so is constant in our search for minimizers of $SInc$. For ease of exposition, aggregate these values into a vector

$$\mathbf{k} := \left[\prod_{a \in \mathcal{A}} \nu(T_a(w)|S_a(w))^{\alpha_a} \right]_{w \in \mathcal{VX}}. \quad (8.9)$$

We can now capture $\llbracket m \rrbracket_{0+}^*$ with a convex program.

Proposition 8.3.5. *If $\nu \in \llbracket m \rrbracket_0^*$ and (μ, \mathbf{u}) solves the problem*

$$\begin{aligned} & \underset{\mu, \mathbf{u}}{\text{minimize}} \quad \mathbf{1}^\top \mathbf{u} \\ & \text{subject to} \quad (-\mathbf{u}, \mu, \mathbf{k}) \in K_{\text{exp}}^{\mathcal{VX}}, \quad \mu \in \Delta \mathcal{VX}, \\ & \quad \forall S \xrightarrow{a} T \in \mathcal{A}. \quad \mu(S, T) \nu(S) = \mu(S) \nu(S, T), \end{aligned} \quad (8.10)$$

then $\llbracket m \rrbracket_{0+}^* = \{\mu\}$ and $\mathbf{1}^\top \mathbf{u} = SInc_m(\mu)$.

Running (8.10) through a convex solver gives rise to the first algorithm that can reliably find $\llbracket m \rrbracket_{0+}^*$.

8.4 Polynomial-Time Inference Under Bounded Treewidth

We have now seen how $\hat{\gamma}$ -inference (for small $\hat{\gamma}$) can be reduced to convex optimization over joint distributions μ —but μ grows exponentially with the number of variables in the PDG, so we do not yet have a tractable inference algorithm. We now show how μ can be replaced with a tree marginal over the PDG’s structure. What makes this possible is a key independence property of traditional graphical models, which we now prove holds for PDGs as well.

[link to
proof]

Theorem 8.4.1 (Markov Property for PDGs). *If \mathbf{m}_1 and \mathbf{m}_2 are PDGs over sets \mathcal{X}_1 and \mathcal{X}_2 of variables, respectively, \blacksquare then \mathcal{X}_1 and \mathcal{X}_2 are conditionally independent given $\mathcal{X}_1 \cap \mathcal{X}_2$ in every $\mu \in [\![\mathbf{m}_1 + \mathbf{m}_2]\!]_{\gamma}^*$, for all $\gamma > 0$ and $\gamma = 0^+$.*

For the remainder of this section, fix a PDG \mathbf{m} and a tree decomposition $(\mathcal{C}, \mathcal{T})$ of \mathbf{m} 's hypergraph. One significant consequence of Theorem 8.4.1 is that, in the search for optimizers of (8.2), we need consider only distributions that satisfy those independencies, all of which can be represented as a tree marginal $\boldsymbol{\mu} = \{\mu_C \in \Delta\mathcal{V}(C)\}_{C \in \mathcal{C}}$ over $(\mathcal{C}, \mathcal{T})$.

Corollary 8.4.1.1. *If \mathbf{m} is a PDG with arcs \mathcal{A} , $(\mathcal{C}, \mathcal{T})$ is a tree decomposition of \mathcal{A} , $\gamma > 0$, and $\mu \in [\![\mathbf{m}]\!]_{\gamma}^*$, then there exists a tree marginal $\boldsymbol{\mu}$ over $(\mathcal{C}, \mathcal{T})$ such that $\Pr_{\boldsymbol{\mu}} = \mu$.*

[link to proof]

For convenience, let $\mathcal{VC} := \{(C, c) : C \in \mathcal{C}, c \in \mathcal{V}(C)\}$ be the set of all choices of a cluster together with a setting of its variables. Like before, we start by optimizing $OInc$, this time over calibrated tree marginals $\boldsymbol{\mu}$, which we identify with vectors $\boldsymbol{\mu} \cong [\mu_C(C=c)]_{(C,c) \in \mathcal{VC}}$. We need the conditional marginals $\Pr_{\boldsymbol{\mu}}(T_a | S_a)$ of $\boldsymbol{\mu}$ along every arc a in order to calculate $OIncm(\Pr_{\boldsymbol{\mu}})$; fortunately, they are readily available. Since $(\mathcal{C}, \mathcal{T})$ is a tree decomposition, we know S_a and T_a lie entirely within some cluster $C_a \in \mathcal{C}$, and $\Pr_{\boldsymbol{\mu}}(T_a | S_a) = \mu_{C_a}(T_a | S_a)$ if $\boldsymbol{\mu}$ is calibrated. For $\mathbf{u} \in \bar{\mathbb{R}}^{\mathcal{VA}}$, consider the problem

$$\underset{\boldsymbol{\mu}, \mathbf{u}}{\text{minimize}} \quad \sum_{(a,s,t) \in \mathcal{VA}} \beta_a u_{a,s,t} \tag{8.11}$$

$$\text{subject to} \quad \forall C \in \mathcal{C}. \mu_C \in \Delta\mathcal{V}(C),$$

$$\forall a \in \mathcal{A}. (-\mathbf{u}_a, \mu_{C_a}(S_a, T_a), \mu_{C_a}(S_a) \mathbb{P}_a(T_a | S_a)) \in K_{\text{exp}}^{\mathcal{V}a}$$

$$\forall (C, D) \in \mathcal{T}. \mu_C(C \cap D) = \mu_D(C \cap D),$$

where again \mathbf{u}_a is the restriction of \mathbf{u} to components associated with a . Problem

(8.11) is similar to (8.5), except that it requires local marginal constraints to restrict our search to calibrated tree marginals. It is analogous to problem CTREE-OPTIMIZE-KL of Koller and Friedman [56, pg. 384].

Proposition 8.4.2. *If (μ, \mathbf{u}) is a solution to (8.11), then*

link to
proof

- (a) μ is a calibrated, with $\Pr_\mu \in \llbracket m \rrbracket_0^*$, and
- (b) the objective of (8.11) evaluated at \mathbf{u} equals $\langle\!\langle m \rangle\!\rangle_0$.

We can now find a minimizer of $OInc$ and compute $\langle\!\langle m \rangle\!\rangle_0$ without storing a joint distribution. ■ But to do anything else, we must deviate from the template laid out in Section 8.3.

Dealing with Joint Entropy. In the construction of (8.11), we rely heavily on the fact that each term of $OInc_m$ depends only on local marginal distributions $\mu_{C_a}(T_a, S_a)$ and $\mu_{C_a}(S_a)$. The same is not true of $SInc$, which depends on the joint entropy $H(\Pr_\mu)$ of the entire distribution. At this point we should point out an important reason to restrict our focus to trees: it allows the joint entropy to be expressed in terms of the cluster marginals [93], by

$$-H(\Pr_\mu) = -\sum_{C \in \mathcal{C}} H(\mu_C) + \sum_{(C, D) \in \mathcal{T}} H_\mu(C \cap D). \quad (8.12)$$

Even so, it is not obvious that (8.12) can be captured with dcp exponential cone constraints. (Exponential conic programs can minimize negative entropy, but not positive entropy, which is concave.) We now describe how this can be done.

Choose a root node C_0 of the tree decomposition, and orient each edge of \mathcal{T} so that it points away from C_0 . Each cluster $C \in \mathcal{C}$, except for C_0 , then has a parent cluster $\text{Par}(C)$; define $\text{Par}(C_0) := \emptyset$ to be an empty cluster, since C_0 has no parent. Finally, for each $C \in \mathcal{C}$, let $VCP_C := C \cap \text{Par}(C)$ denote the the set of variables

that cluster C has in common with its parent cluster.² As \mathcal{T} is now a directed tree, this definition allow us to express (8.12) in a more useful form:

$$\begin{aligned} -H(\Pr_{\boldsymbol{\mu}}) &= -H(\mu_{C_0}) - \sum_{(C \rightarrow D) \in \mathcal{T}} H_{\Pr_{\boldsymbol{\mu}}}(D | C) \\ &= \sum_{C \in \mathcal{C}} \sum_{c \in \mathcal{V}(C)} \mu_C(C=c) \log \frac{\mu_C(C=c)}{\mu_C(VCP_C(c))}, \end{aligned} \quad (8.13)$$

where $VCP_C(c)$ is the restriction of the joint value $c \in \mathcal{V}(C)$ to the variables $VCP_C \subseteq C$. Crucially, the denominator of (8.13) is an affine transformation of μ_C . The upshot: we have rewritten the joint entropy as a sum of functions of the clusters, each of which can be captured with a dcp exponential cone constraint. This gives us analogues of the problems in Sections 8.3.2 and 8.3.3 that operate on tree marginals.

Finding tree marginals for $\hat{\gamma}$ -inference. The ability to decompose the joint entropy as in (8.13) allows us to adapt (8.7) to operate on calibrated tree marginals, rather than joint distributions. Beyond the changes already present in (8.11), the key is to replace the exponential cone constraint $(-\mathbf{v}, \mu, \mathbf{1}) \in K_{\text{exp}}^{\mathcal{V}\mathcal{X}}$, which captures the entropy of μ , with

$$(-\mathbf{v}, \boldsymbol{\mu}, [\mu_C(VCP_C(c))]_{(C,c) \in \mathcal{VC}}) \in K_{\text{exp}}^{\mathcal{VC}},$$

which captures the entropy of $\boldsymbol{\mu}$, by (8.13). Over vectors $\mathbf{v}, \boldsymbol{\mu} \in \bar{\mathbb{R}}^{\mathcal{VC}}$ and $\mathbf{u} \in \bar{\mathbb{R}}^{\mathcal{VA}}$, the problem becomes:

$$\begin{aligned} \underset{\boldsymbol{\mu}, \mathbf{u}, \mathbf{v}}{\text{minimize}} \quad & \sum_{(a,s,t) \in \mathcal{VA}} (\beta_a - \alpha_a \gamma) u_{a,s,t} + \gamma \sum_{(C,c) \in \mathcal{VC}} v_{C,c} \\ & - \sum_{(a,s,t) \in \mathcal{VA}^+} \alpha_a \gamma \mu_{C_a}(S_a=s, T_a=t) \log \mathbb{P}_a(T_a=t | s) \\ \text{subject to} \quad & \forall C \in \mathcal{C}. \mu_C \in \Delta \mathcal{V}(C), \\ \forall a \in \mathcal{A}. \quad & (-\mathbf{u}_a, \mu_{C_a}(S_a, T_a), \mu_{C_a}(S_a) \mathbb{P}_a(T_a | S_a)) \in K_{\text{exp}}^{\mathcal{V}a}, \end{aligned} \quad (8.14)$$

$$\begin{aligned} \forall(a, s, t) \in \mathcal{VA}^0. \mu_{C_a}(S_a=s, T_a=t) = 0, \\ \forall(C, D) \in \mathcal{T}. \mu_C(C \cap D) = \mu_D(C \cap D), \\ (-\mathbf{v}, \boldsymbol{\mu}, [\mu_C(VCP_C(c))]_{(C,c) \in \mathcal{VC}}) \in K_{\text{exp}}^{\mathcal{VC}}. \end{aligned}$$

Proposition 8.4.3. If $(\boldsymbol{\mu}, \mathbf{u}, \mathbf{v})$ is a solution to (8.14) and $\beta \geq \gamma\alpha$, then $\Pr_{\boldsymbol{\mu}}$ is the unique element of $[\![m]\!]_{\gamma}^*$, and the objective of (8.14) at $(\boldsymbol{\mu}, \mathbf{u}, \mathbf{v})$ equals $\langle\!\langle m \rangle\!\rangle_{\gamma}$.

[link to proof]

■ A related use of (8.13) is to enable an analogue of (8.10) that searches over tree marginals (rather than joint distributions), to find a compact representation of $[\![m]\!]_{0+}^*$. We begin with a straightforward adaptation of the relevant machinery in Section 8.3.3. Suppose that $\nu = \{\nu_C : C \in \mathcal{C}\}$ is a calibrated tree marginal over the tree decomposition $(\mathcal{C}, \mathcal{T})$ representing a distribution $\Pr_{\nu} \in [\![m]\!]_0^*$, say obtained by solving (8.11). For $C \in \mathcal{C}$, let $\mathcal{A}_C := \{a \in \mathcal{A} : C_a = C\}$ be the set of arcs assigned to cluster C , and let

$$\mathbf{k} := \left[\prod_{a \in \mathcal{A}_C} \nu_C(T_a(c) | S_a(c))^{\alpha_a} \right]_{(C,c) \in \mathcal{VC}} \in \bar{\mathbb{R}}^{\mathcal{VC}}$$

be the analogue of (8.9) for a cluster tree. Once again, consider $\mathbf{u} := [u_{(C,c)}]_{(C,c) \in \mathcal{VC}}$ in the optimization problem

$$\begin{aligned} & \underset{\boldsymbol{\mu}, \mathbf{u}}{\text{minimize}} \quad \mathbf{1}^T \mathbf{u} \tag{8.15} \\ & \text{subject to} \quad \forall C \in \mathcal{C}. \mu_C \in \Delta \mathcal{V}(C), \\ & \quad (-\mathbf{u}, \boldsymbol{\mu}, \mathbf{k} \odot [\mu_C(VCP_C(c))]_{(C,c) \in \mathcal{VC}}) \in K_{\text{exp}}^{\mathcal{VC}}, \\ & \quad \forall a \in \mathcal{A}. \mu_{C_a}(S_a, T_a) \nu_{C_a}(S_a) = \mu_{C_a}(S_a) \nu_{C_a}(S_a, T_a) \\ & \quad \forall(C, D) \in \mathcal{T}. \mu_C(C \cap D) = \mu_D(C \cap D). \end{aligned}$$

²Different choices of C_0 yield different definitions of VCP , and ultimately optimization problems of different sizes; the optimal choice can be found with Edmund's Algorithm [17], which computes a directed analogue of the minimum spanning tree.

The biggest change is in the second constraint: the upper bounds $[u_{(C,c)}]_{c \in \mathcal{V}_C}$ for cluster C now account only for the additional entropy not already modeled by C 's ancestors.

Proposition 8.4.4. *If (μ, u) is a solution to (8.15), then μ is a calibrated tree marginal and $[\![m]\!]_{0^+}^* = \{\Pr_\mu\}$.*

[link to proof]

At this point, standard algorithms can use μ to answer probabilistic queries about \Pr_μ in polynomial time [56, §10.3.3]. ■ From Propositions 8.4.3 and 8.4.4, it follows that $\hat{\gamma}$ -inference (for small $\hat{\gamma}$, and for 0^+) can be reduced to a (pair of) convex optimization problem(s) with a polynomial number of variables and constraints. All that remains is to show that such a problem can be solved in polynomial time. For this, we turn to interior-point methods. As (8.14) and (8.15) are dcp, they can be transformed via established methods [1] into a standard form that can be solved in polynomial time by commercial solvers [4, 24]. Threading the details of our constructions through the analyses of Dahl and Andersen [21] and Nesterov et al. [69] results in our main theorem.



Theorem 8.4.5. *Let $m = (\mathcal{X}, \mathcal{A}, \mathbb{P}, \alpha, \beta)$ be a proper discrete PDG with $N = |\mathcal{X}|$ variables each taking at most V values and $A = |\mathcal{A}|$ arcs, in which each component of $\beta \in \mathbb{R}^{\mathcal{A}}$ and $\mathbb{P} \in \mathbb{R}^{\mathcal{V}^{\mathcal{A}}}$ is specified in binary with at most k bits. Suppose that $\gamma \in \{0^+\} \cup (0, \min_{a \in \mathcal{A}} \frac{\beta_a}{\alpha_a}]$. If $(\mathcal{C}, \mathcal{T})$ is a tree decomposition of $(\mathcal{X}, \mathcal{A})$ of width T and $\mu^* \in \mathbb{R}^{\mathcal{V}^{\mathcal{C}}}$ is the unique calibrated tree marginal over $(\mathcal{C}, \mathcal{T})$ that represents the $\hat{\gamma}$ -semantics of m , then*

[link to proof]

- (a) *Given m , γ , and $\epsilon > 0$, we can find a calibrated tree marginal ϵ close in ℓ_2 norm*

to μ^* in time¹

$$\begin{aligned} & O\left(|\mathcal{VA} + \mathcal{VC}|^4 \left(\log |\mathcal{VA} + \mathcal{VC}| + \log \frac{1}{\epsilon}\right) k^2 \log k\right) \\ & \subseteq \tilde{O}\left(k^2 |\mathcal{VA} + \mathcal{VC}|^4 \log^{1/\epsilon}\right) \\ & \subseteq \tilde{O}\left(k^2 (N+A)^4 V^{4(T+1)} \log^{1/\epsilon}\right). \end{aligned}$$

(b) The unique tree marginal closest to μ^* in which every component is represented with a k -bit binary number, can be calculated in time¹

$$\tilde{O}\left(k^2 |\mathcal{VA} + \mathcal{VC}|^4\right) \subseteq \tilde{O}\left(k^2 (N+A)^4 V^{4(T+1)}\right).$$

Observe that the dependence on the precision is $\log(1/\epsilon)$, which is optimal in the sense that, in general, it takes time $\Omega(\log 1/\epsilon)$ to write down the binary representation of any number within ϵ of a given value.³ In practice, this procedure can be used as if it were an exact algorithm, with no more overhead than that incurred by floating point arithmetic.

8.5 Experiments

We have given the first algorithm to provably do inference in polynomial time, but that does not mean that it is the best way of answering queries in practice; it also makes sense to use black-box optimization tools such as Adam [51] or L-BFGS [28] to find minimizers of $\|\mathbf{m}\|_\gamma$. Indeed, this scoring function has several properties that make it highly amenable to such methods: it is infinitely differentiable, γ -strongly convex, and its derivatives have simple closed-form expressions. So it may seem surprising that $\|\mathbf{m}\|_\gamma$ poses a challenge to standard

³More precisely: if a value x is chosen uniformly from $[0, 1]$, then with probability $1 - \sqrt{\epsilon}$ the binary representation of every $y \in [x - \epsilon, x + \epsilon]$ has at least $\lfloor \frac{1}{2} \log_2 1/\epsilon \rfloor - 1$ bits.

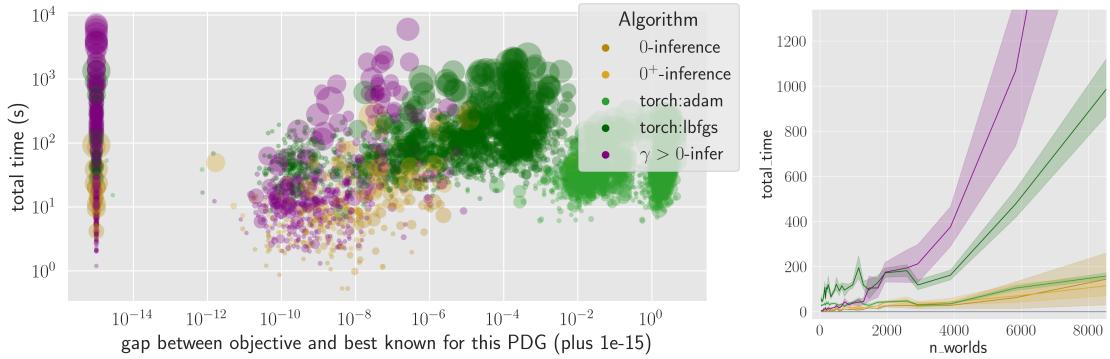


Figure 1: Accuracy and resource costs for the methods in Section 8.3. Left: a scatter plot of several algorithms on random PDGs of ≈ 10 variables. The x-axis is the difference in scores $\|\mathbf{m}\|_\gamma(\mu) - \|\mathbf{m}\|_\gamma(\mu^*) + 10^{-15}$, where μ is the method’s output, and μ^* achieves best (smallest) known value of $\|\mathbf{m}\|_\gamma$. (Thus, the best solutions lie on the far left.) The y axis is the time required to compute μ . Our methods are in gold (0^+ -inference) and violet ($\hat{\gamma}$ -inference, for $\hat{\gamma} > 0$); the baselines (black-box optimizers applied directly to (8.2)) are in green. The area of each circle is proportional to the size of the optimization problem, as measured by $n_worlds := |\mathcal{V}\mathcal{X}|$. Right: how the same methods scale in run time, as $|\mathcal{V}\mathcal{X}|$ increases.

optimization tools—but it does, even when we optimize directly over joint distributions.

Synthetic Experiment 1 (over joint distributions). Repeatedly do the following. First, randomly generate a small PDG \mathbf{m} containing at most 10 variables and 15 arcs. Then for various values of $\gamma \in \{0, 0^+, 10^{-8}, \dots, \min_a \frac{\beta_a}{\alpha_a}\}$, optimize $\|\mathbf{m}\|_\gamma(\mu)$ over joint distributions μ , in one of two ways.

- (a) Use cvxpy [23] to feed one of problems (8.5, 8.7, 8.10) to the MOSEK solver [4], or
- (b) Choose a learning rate and a representation of μ in terms of optimization variables $\theta \in \mathbb{R}^n$. Then run a standard optimizer (Adam or L-BFGS) built into pytorch [72] to optimize θ until μ_θ converges to a minimizer of $\|\mathbf{m}\|_\gamma$ (or a time limit is reached). Keep only the best result across all learning rates.

The results are shown in Figure 1. Observe that the convex solver (gold, violet)

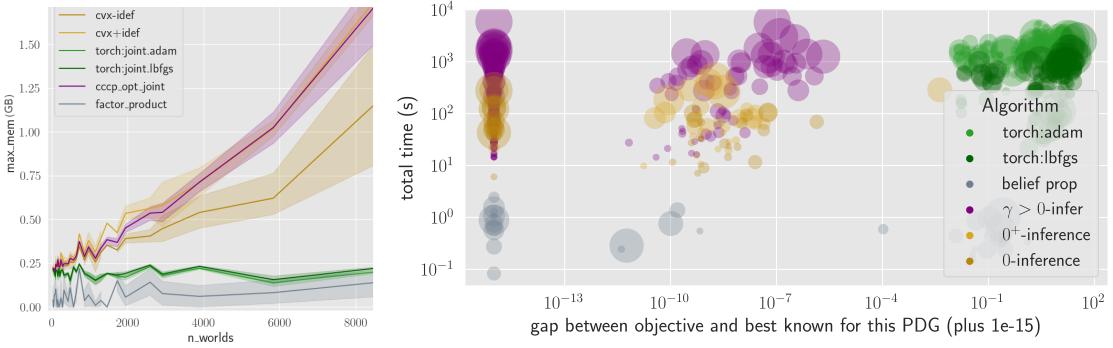


Figure 2: Left: Memory footprint. The convex solver (violet, gold) requires more memory than baselines (green). Right: Analogue of Figure 1 for the cluster setting. Here there is even more separation between exponential conic optimization (gold, violet) and black-box optimization (greens). The grey points represent belief propagation, which is fastest and most accurate—but only applies in the special case when $\beta = \gamma\alpha$.

is significantly more accurate than the baselines, and also much faster for small PDGs. Our implementation of 0^+ -inference (gold) also appears to scale better than L-BFGS in this regime, although that of $\hat{\gamma}$ -inference (purple) seems to scale much worse. We suspect that the difference comes from cvxpy’s compilation process, because the two use similar amounts of memory (Figure 2), and so are problems of similar sizes.

Synthetic Experiment 2 (over tree marginals). For PDGs of bounded treewidth, ?? 8.4.1.1 allows us to express these optimization problems compactly not just for the convex solver, but for the black-box baseline approaches as well. We adapt the previous experiment for tree marginals as follows. First randomly sample a maximal graph G of tree-width k , called a k -tree [73]; then generate a PDG m whose hyperarcs lie within cliques of G . This ensures that the maximal cliques of G form a tree-decomposition $(\mathcal{C}, \mathcal{T})$ of m ’s underlying hypergraph. We can now proceed as before: either encode (8.11,8.14,8.15) as disciplined convex programs in cvxpy, or use torch to directly minimize $\|m\|_\gamma(\Pr_\mu)$ amongst tree marginals μ over $(\mathcal{C}, \mathcal{T})$.

In the latter case, however, there is now an additional difficulty: it is not easy to strictly enforce the calibration constraints with the black-box methods. Common practice is to instead add extra loss terms to “encourage” calibration—but it can still be worthwhile for the optimizer to simply incur that loss in order to violate the constraints. Thus, for fairness, we must recalibrate the tree marginals returned by all methods before evaluation. The result is an even more significant advantage for the convex solver; see [Figure 2](#).

Evaluation on BNs. We also applied the procedure of the Synthetic Experiment 2 to the smaller BNs in the `bnlearn` repository, and found similar results (but with fewer examples; see [Section 8.C.3](#)). But for a PDG that happens to also be a BN, it is possible to use belief propagation, which is much faster and at least as accurate.

Explicit details about all of our experiments, and many more figures, can be found in [Section 8.C](#).

8.6 Discussion and Conclusion

In this paper, we have provided the first practical algorithm for inference in PDGs. In more detail, we have defined a parametric family of PDG inference notions, given a fixed-parameter tractable inference algorithm for a subset of these parameters, proven our algorithm correct, implemented it, and shown our code to empirically outperform baselines. Yet many questions about PDG inference remain open.

Asymptotically, there may be a lot of room for improvement. Our imple-

mentation runs in time $\tilde{O}(N^4)$, and our analysis suggests one of time $\tilde{O}(N^{2.872})$. But assuming bounded tree-width, most graph problems, including inference inference for BNs and FGs, can be solved in time $O(N)$.

Furthermore, we have shown how to do inference for only a subset of possible parameter values, specifically, when either $\beta \geq \gamma\alpha$ or $\beta \gg \alpha$. The remaining cases are also of interest, and likely require different techniques. When $\beta = 0$ and (\mathcal{A}, α) encodes the structure of a BN, for instance, inference is about characterizing the BN’s independencies. While we do not know how to tackle the inference problem in the general setting, our methods can be augmented with the convex-concave procedure [97] to obtain an inference algorithm that applies slightly more broadly; see [Section 8.B](#). We imagine that this extension could also be useful for computing with PDGs beyond the specific inference problem considered in this paper.

Given the long history of improvements to our understanding of inference for Bayesian networks, we are optimistic that faster and more general inference algorithms for PDGs are possible. Our analysis does not resolve these problems, but it does shed light on some of them. The 0-semantics, for instance, is characterized by [Propositions 8.4.2 and 8.3.3](#). Also, when $\llbracket m \rrbracket_\gamma$ is not convex, we can still find an optimal distribution with the concave-convex procedure [97], which we do in [Section 8.B](#)—but this only suffices for inference if we already know there’s a unique optimal distribution. In some cases, this might actually allow us to do inference—say, if we happen to know for external reasons that $\llbracket m \rrbracket_\gamma^*$ is pseudo-convex (although we loose polynomial time guarantees and have no ability to automatically recognize such situations). In any case, we have implemented this, and describe it in [Section 8.B](#).

APPENDICES FOR CHAPTER 8

8.A Proofs

Our results fall broadly into three categories:

1. Foundational results about PDGs that we needed to prove to get an inference procedure, but which are likely to be generally useful for anyone working with PDGs ([Section 8.A.1](#));
2. Correctness and efficiency results, showing that the optimization problems we present in the main paper give the correct answers, and that they can be formulated and solved in polynomial time; ([Section 8.A.2](#))
3. Hardness results, i.e., [Theorem 9.3.2](#) and the constructions and lemmas needed to support it ([Section 9.4](#)).

8.A.1 Novel Results about PDGs

Proposition 8.3.3. *If \mathcal{M} has arcs \mathcal{A} and $\beta \geq 0$, the minimizers of $OInc_{\mathcal{M}}$ all have the same conditional marginals along \mathcal{A} . That is, for all $\mu_1, \mu_2 \in [\![\mathcal{M}]\!]_0^*$ and all $S \xrightarrow{a} T \in \mathcal{A}$ with $\beta_a > 0$, we have $\mu_1(T, S)\mu_2(S) = \mu_2(T, S)\mu_1(S)$.*

Proof. For contradiction, suppose that $\mu_1, \mu_2 \in [\![\mathcal{M}]\!]_0^*$, but there is some $(\hat{a}, \hat{s}, \hat{t}) \in \mathcal{V}\mathcal{A}$ such that $\beta_{\hat{a}} > 0$ and

$$\mu_1(T_{\hat{a}}=\hat{t}, S_{\hat{a}}=\hat{s})\mu_2(S_{\hat{a}}=\hat{s}) \neq \mu_2(T_{\hat{a}}=\hat{t}, S_{\hat{a}}=\hat{s})\mu_1(S_{\hat{a}}=\hat{s}).$$

For $t \in [0, 1]$, let $\mu_t := (1 - t)\mu_0 + t\mu_1$ as before. Then define

$$F(t) := D\left(\mu_t(S_a, T_a) \parallel \mu_t(S_a)\mathbb{P}_a(T_a|S_a)\right).$$

Since $\mu_0(S_a, T_a)$ and $\mu_1(S_a, T_a)$ are joint distributions over two variables, with different conditional marginals, as above, Lemma 8.A.2 applies, and so $F(t)$ is strictly convex.

Let

$$OInc_{m \setminus \hat{a}} := \sum_{a \neq \hat{a}} \beta_a D(\mu(T_a, S_a) \parallel \mathbb{P}_a(T_a|S_a)\mu(S_a))$$

be the observational incompatibility loss, but without the term corresponding to edge \hat{a} . Since $OInc_{m \setminus \hat{a}}$ is convex in its argument, it is in particular convex along the segment from μ_0 to μ_1 ; that is, for $t \in [0, 1]$, the function $t \mapsto OInc_{m \setminus \hat{a}}(\mu_t)$ is convex. Therefore, we know that the function

$$G(t) := OInc_m(\mu_t) = OInc_{m \setminus \hat{a}}(\mu_t) + \beta_{\hat{a}} F(t),$$

is *strictly* convex. But then this means $\mu_{1/2}$ satisfies

$$OInc_m(\mu_{1/2}) < OInc_m(\mu_0),$$

contradicting the premise that μ_0 minimizes $OInc_m$ (i.e., $\mu_0 \in [\![m]\!]_0^*$). Therefore, it must be the case that all distributions in $[\![m]\!]_0^*$ have the same conditional marginals, as promised. \square

Proposition 8.3.4. If $\mu \in [\![\mathbf{m}]\!]_0^*$, then

$$\frac{SInc}{m}(\mu) = \sum_{w \in \mathcal{VX}} \mu(w) \log \left(\frac{\mu(w)}{\prod_{a \in \mathcal{A}} \nu(T_a(w)|S_a(w))^{\alpha_a}} \right), \quad (8.8)$$

where $\{\nu(T_a|S_a)\}_{a \in \mathcal{A}}$ are the marginals along the arcs \mathcal{A} shared by all distributions in $[\![\mathbf{m}]\!]_0^*$ (per Proposition 8.3.3), and $S_a(w), T_a(w)$ are the values of variables S_a and T_a in w .

Proof. This is mostly a simple algebraic manipulation. By definition:

$$\begin{aligned} SInc_m(\mu) &= -H(\mu) + \sum_{a \in \mathcal{A}} \alpha_a H_\mu(T_a|S_a) \\ &= \mathbb{E}_\mu \left[-\log \frac{1}{\mu} + \sum_{a \in \mathcal{A}} \alpha_a \log \frac{1}{\mu(T_a|S_a)} \right] \\ &= \sum_{w \in \mathcal{VX}} \mu(w) \left[\log \mu(w) + \sum_{a \in \mathcal{A}} \log \frac{1}{\mu(T_a(w)|S_a(w))^{\alpha_a}} \right] \\ &= \sum_{w \in \mathcal{VX}} \mu(w) \log \left(\frac{\mu(w)}{\prod_{a \in \mathcal{A}} \mu(T_a(w)|S_a(w))^{\alpha_a}} \right) \end{aligned}$$

But, by Proposition 8.3.3, if we restrict $\mu \in [\![\mathbf{m}]\!]_0^*$, then the conditional marginals in the denominator do not depend on the particular choice of μ ; they're shared among all $\nu \in [\![\mathbf{m}]\!]_0^*$. \square

Theorem 8.4.1. If \mathbf{m}_1 and \mathbf{m}_2 are PDGs over sets \mathcal{X}_1 and \mathcal{X}_2 of variables, respectively, ■ then \mathcal{X}_1 and \mathcal{X}_2 are conditionally independent given $\mathcal{X}_1 \cap \mathcal{X}_2$ in every $\mu \in [\![\mathbf{m}_1 + \mathbf{m}_2]\!]_\gamma^*$, for all $\gamma > 0$ and $\gamma = 0^+$.

Or symbolically: $\mathbf{m}_1 + \mathbf{m}_2 \models \mathcal{X}_1 \perp\!\!\!\perp \mathcal{X}_2 \mid \mathcal{X}_1 \cap \mathcal{X}_2$.

Proof. Note that, save for the joint entropy, every summand the scoring function $[\![\mathbf{m}_1 + \mathbf{m}_2]\!]_\gamma : \Delta(\mathcal{VX}_1 \times \mathcal{VX}_2)$, is a function of the conditional marginal of μ along

some edge. In particular, those terms that correspond to edges of m_1 can be computed from the marginal $\mu(\mathcal{X}_1)$, while those that correspond to edges of m_2 can be computed from the marginal $\mu(\mathcal{X}_2)$. Therefore, there are functions f and g such that:

$$[\![m_1 + m_2]\!]_\gamma(\mu) = f(\mu(\mathcal{X}_1)) + g(\mu(\mathcal{X}_2)) - \gamma H(\mu).$$

To make this next step extra clear, let $\mathbf{X} := \mathcal{X}_1 \setminus \mathcal{X}_2$ and $\mathbf{Z} := \mathcal{X}_2 \setminus \mathcal{X}_1$, be the variables unique to each PDG, and $\mathbf{S} := \mathcal{X}_1 \cap \mathcal{X}_2$ be the set of variables they have in common, so that $(\mathbf{X}, \mathbf{S}, \mathbf{Z})$ is a partition of all variables $\mathbf{X}_1 \cup \mathbf{X}_2$. Now define a new distribution $\mu' \in \Delta(\mathcal{V}\mathcal{X}_1 \times \mathcal{V}\mathcal{X}_2)$ by

$$\mu'(\mathbf{X}, \mathbf{S}, \mathbf{Z}) := \mu(\mathbf{S})\mu(\mathbf{Z} \mid \mathbf{S})\mu(\mathbf{X} \mid \mathbf{S}) \quad \left(= \mu(\mathbf{X}, \mathbf{S})\mu(\mathbf{Z} \mid \mathbf{S}) = \mu(\mathbf{Z}, \mathbf{S})\mu(\mathbf{X} \mid \mathbf{S}) \right).$$

One can easily verify that \mathbf{X} and \mathbf{Z} are independent given \mathbf{S} in μ' (by construction), and the alternate forms on the right make it easy to see that $\mu(\mathcal{X}_1) = \mu'(\mathcal{X}_1)$ and $\mu(\mathcal{X}_2) = \mu'(\mathcal{X}_2)$. Furthermore, for any $\nu'(\mathbf{X}, \mathbf{S}, \mathbf{Z})$, we can write

$$\begin{aligned} H(\nu) &= H_\nu(\mathbf{X}, \mathbf{S}, \mathbf{Z}) = H_\nu(\mathbf{X}, \mathbf{S}) + H_\nu(\mathbf{Z} \mid \mathbf{X}, \mathbf{S}) \\ &= H_\nu(\mathbf{X}, \mathbf{S}) + H_\nu(\mathbf{Z} \mid \mathbf{X}, \mathbf{S}) - H_\nu(\mathbf{Z} \mid \mathbf{S}) + H_\nu(\mathbf{Z} \mid \mathbf{S}) \\ &= H_\nu(\mathbf{X}, \mathbf{S}) + H_\nu(\mathbf{Z} \mid \mathbf{S}) - I_\nu(\mathbf{Z}; \mathbf{X} \mid \mathbf{S}), \end{aligned}$$

where $I_\nu(\mathbf{X}; \mathbf{Z} \mid \mathbf{S})$, the conditional mutual information between \mathbf{X} and \mathbf{Z} given \mathbf{S} (in ν), is non-negative, and equal to zero if and only if \mathbf{X} and \mathbf{Z} are conditionally independent given \mathbf{S} [see, for instance, 63, §1]. So $I_{\mu'}(\mathbf{X}; \mathbf{Z} \mid \mathbf{S}) = 0$, and $H_{\mu'} = H_{\mu'}(\mathbf{X}, \mathbf{S}) + H_{\mu'}(\mathbf{Z} \mid \mathbf{S})$. Because μ and μ' share marginals on \mathcal{X}_1 and \mathcal{X}_2 , while the terms $H(\mathbf{X}, \mathbf{S})$ and $H(\mathbf{Z} \mid \mathbf{S})$ depend only on these marginals, respectively, we also know that $H_\mu(\mathbf{X}, \mathbf{S}) = H_{\mu'}(\mathbf{X}, \mathbf{S})$ and $H_\mu(\mathbf{Z} \mid \mathbf{S}) = H_{\mu'}(\mathbf{Z} \mid \mathbf{S})$; thus we have

$$\begin{aligned} H(\mu) &= H_\mu(\mathbf{X}, \mathbf{S}) + H_\mu(\mathbf{Z} \mid \mathbf{S}) - I_\mu(\mathbf{Z}; \mathbf{X} \mid \mathbf{S}) \\ &= H(\mu') - I_\mu(\mathbf{Z}; \mathbf{X} \mid \mathbf{S}). \end{aligned}$$

Therefore,

$$\begin{aligned}
\llbracket \mathbf{m}_1 + \mathbf{m}_2 \rrbracket_\gamma(\mu) &= f(\mu(\mathcal{X}_1)) + g(\mu(\mathcal{X}_2)) - \gamma H(\mu) \\
&= f(\mu'(\mathcal{X}_1)) + g(\mu'(\mathcal{X}_2)) - \gamma H(\mu') + \gamma I_\mu(\mathbf{Z}; \mathbf{X}|\mathbf{S}) \\
&= \llbracket \mathbf{m}_1 + \mathbf{m}_2 \rrbracket_\gamma(\mu') + \gamma I_\mu(\mathbf{Z}; \mathbf{X}|\mathbf{S}).
\end{aligned}$$

But conditional mutual information is non-negative, and by assumption, $\llbracket \mathbf{m}_1 + \mathbf{m}_2 \rrbracket_\gamma(\mu)$ is minimal. Therefore, it must be the case that

$$I_\mu(\mathbf{Z}; \mathbf{X}|\mathbf{S}) = I_\mu(\mathcal{X}_1; \mathcal{X}_2 \mid \mathcal{X}_1 \cap \mathcal{X}_2) = 0,$$

showing that \mathcal{X}_1 and \mathcal{X}_2 are conditionally independent given the variables that they have in common.

(The fact that $I_\mu(\mathbf{Z}; \mathbf{X}|\mathbf{S}) = I_\mu(\mathcal{X}_1; \mathcal{X}_2 \mid \mathcal{X}_1 \cap \mathcal{X}_2)$ is both easy to show and an instance of a well-known identity; see CIRV2 in Theorem 4.4.4 of Halpern [39], for instance.) \square

?? 8.4.1.1. *If \mathbf{m} is a PDG with arcs \mathcal{A} , $(\mathcal{C}, \mathcal{T})$ is a tree decomposition of \mathcal{A} , $\gamma > 0$, and $\mu \in \llbracket \mathbf{m} \rrbracket_\gamma^*$, then there exists a tree marginal $\boldsymbol{\mu}$ over $(\mathcal{C}, \mathcal{T})$ such that $\Pr_{\boldsymbol{\mu}} = \mu$.*

Proof. The set of distributions that can be represented by a calibrated tree marginal over $(\mathcal{C}, \mathcal{T})$ is the same as the set of distributions that can be represented by a factor graph for which $(\mathcal{C}, \mathcal{T})$ is a tree decomposition. One direction holds because any such product of factors “calibrated”, via message passing algorithms such as belief propagation, to form a tree marginal. The other direction holds because $\Pr_{\boldsymbol{\mu}}$ itself is a product of factors that decomposes over $(\mathcal{C}, \mathcal{T})$.

Alternatively, this same set of distributions that satisfy the independencies of the Markov Network obtained by connecting every pair of variables that share a

cluster. More formally, this network is the graph $G := (\mathcal{X}, E := \{(X - Y) : \exists C \in \mathcal{C}. \{X, Y\} \subseteq C\})$. Also, G happens to chordal as well, which we prove at the end.

Using only the PDG Markov property (Theorem 8.4.1), we now show that every independence described by G also holds in every distribution $\mu \in \llbracket m \rrbracket_{\gamma}^*$. Suppose that, for sets of variables $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \subseteq \mathcal{X}$, $I(\mathbf{X}; \mathbf{Y} | \mathbf{Z})$ is an independence described by G . This means [56, Defn 4.8] that if $X \in \mathbf{X}$, $Y \in \mathbf{Y}$, and π is a path in G between them, then some node along π lies in \mathbf{Z} .

Let \mathcal{T}' be the graph that results from removing each edge $(C - D) \in \mathcal{T}$ that satisfies $C \cap D \subseteq \mathbf{Z}$, which is a disjoint union $\mathcal{T}' = \mathcal{T}_1 \sqcup \dots \sqcup \mathcal{T}_n$ of subtrees that have no clusters in common. To parallel this notation, let $\mathcal{C}_1, \dots, \mathcal{C}_n$ be their respective vertex sets. Note that for every edge $e = (C - D) \in \mathcal{T}'$, there must by definiton be some variable $U_e \in (C \cap D) \setminus \mathbf{Z}$.

We claim that no subtree \mathcal{T}_i can have both a cluster D_X containing a variable $X \in \mathbf{X} \setminus \mathbf{Z}$ and also a cluster D_Y containing a variable $Y \in \mathbf{Y} \setminus \mathbf{Z}$. Suppose that it did. Then the (unique) path in \mathcal{T} between D_X and D_Y , which we label

$$D_X = D_0 \xrightarrow{e_1} D_1 \xrightarrow{e_2} \dots \xrightarrow{e_{m-1}} D_{m-1} \xrightarrow{e_m} D_m = D_Y ,$$

would lie entirely within $\mathcal{T}_i \subseteq \mathcal{T}'$. This gives rise to a corresponding path in G :

$$\begin{array}{ccccccccccc} X & \xlongequal{\quad} & U_{e_1} & \xlongequal{\quad} & U_{e_2} & \xlongequal{\quad} & \cdots & \xlongequal{\quad} & U_{e_{n-1}} & \xlongequal{\quad} & U_{e_n} & \xlongequal{\quad} & Y \\ \cap & & \cap & & \cap & & \cdots & & \cap & & \cap & & \cap & , \\ D_0 & & D_0 \cap D_1 & & D_1 \cap D_2 & & & & D_{n-2} \cap D_{n-1} & & D_{n-1} \cap D_n & & D_n \end{array}$$

and moreover, this path is disjoint from \mathbf{Z} . This contradicts our assumption that every path in G between a member of \mathbf{X} and a member of \mathbf{Y} must intersect with \mathbf{Z} , and so no subtree can have both a cluster containing a variable $X \in \mathbf{X} \setminus \mathbf{Z}$ and also one containing $Y \in \mathbf{Y} \setminus \mathbf{Z}$.

We can now partition the clusters as $\mathcal{C} = \mathcal{C}_{\mathbf{X}} \sqcup \mathcal{C}_{\mathbf{Y}}^+$, where $\mathcal{C}_{\mathbf{X}}$ is the set of the

clusters that belong to subtrees \mathcal{T}_i with a cluster containing some $X \in \mathbf{X} \setminus \mathbf{Z}$, and its \mathcal{C}_Y^+ is its complement, which in particular contains those subtrees have some $Y \in \mathbf{Y} \setminus \mathbf{Z}$. Or, more formally, we define

$$\mathcal{C}_{\mathbf{X}} := \bigcup_{\substack{i \in \{1, \dots, n\} \\ (\cup C_i) \cap (\mathbf{X} \setminus \mathbf{Z}) \neq \emptyset}} C_i \quad \text{and} \quad \mathcal{C}_{\mathbf{Y}}^+ := \bigcup_{\substack{i \in \{1, \dots, n\} \\ (\cup C_i) \cap (\mathbf{X} \setminus \mathbf{Z}) = \emptyset}} C_i .$$

Let $\mathcal{X}_{\mathbf{X}} := \cup \mathcal{C}_{\mathbf{X}}$ set of all variables appearing in the clusters $\mathcal{C}_{\mathbf{X}}$; symmetrically, define $\mathcal{X}_{\mathbf{Y}}^+ := \cup \mathcal{C}_{\mathbf{Y}}^+$.

We claim that $\mathcal{X}_{\mathbf{X}} \cap \mathcal{X}_{\mathbf{Y}}^+ \subset \mathbf{Z}$. Choose any variable $U \in \mathcal{X}_{\mathbf{X}} \cap \mathcal{X}_{\mathbf{Y}}^+$. From the definitions of $\mathcal{X}_{\mathbf{X}}$ and $\mathcal{X}_{\mathbf{Y}}^+$, this means U is a member of some cluster $C \in \mathcal{C}_{\mathbf{X}}$, and also a member of a cluster $D \in \mathcal{C}_{\mathbf{Y}}^+$. Recall that the clusters of each disjoint subtree \mathcal{T}_i either fall entirely within $\mathcal{C}_{\mathbf{X}}$ or entirely within $\mathcal{C}_{\mathbf{Y}}^+$ by construction. This means that C and D , which are on opposite sides of the partition, must have come from distinct subtrees. So, some edge $e = (C' - D') \in \mathcal{T}$ along the (unique) path from C to D must have been removed when forming \mathcal{T}' , which by the definition of \mathcal{T}' , means that $(C' \cap D') \subset Z$. But by the running intersection property (tree marginal property 2), every cluster along the path from C to D must contain $C \cap D$ —in particular, this must be true of both C' and D' . Therefore,

$$U \in C \cap D \subset C' \cap D' \subset \mathbf{Z}.$$

So $\mathcal{X}_{\mathbf{X}} \cap \mathcal{X}_{\mathbf{Y}}^+ \subset \mathbf{Z}$, as promised. We will rather use it in the equivalent form $(\mathcal{X}_{\mathbf{X}} \cap \mathcal{X}_{\mathbf{Y}}^+) \cup \mathbf{Z} = \mathbf{Z}$.

Next, since $(\mathcal{C}, \mathcal{T})$ is a tree decomposition of \mathcal{A} , each hyperarc $a \in \mathcal{A}$ can be assigned to some cluster C_a that contains all of its variables; this allows us to lift the cluster partition $\mathcal{C} = \mathcal{C}_{\mathbf{X}} \sqcup \mathcal{C}_{\mathbf{Y}}^+$ to a partition $\mathcal{A} = \mathcal{A}_{\mathbf{X}} \sqcup \mathcal{A}_{\mathbf{Y}}^+$ of edges, and consequently, a partition of PDGs $\mathcal{m} = \mathcal{m}_{\mathbf{X}} + \mathcal{m}_{\mathbf{Y}}^+$. Concretely: let $\mathcal{m}_{\mathbf{X}}$ be the sub-PDG of \mathcal{m} induced by restricting to the variables $\mathcal{X}_{\mathbf{X}} \subseteq \mathcal{X}$ arcs

$\mathcal{A}_X = \{a \in \mathcal{A} : C_a \in \mathcal{C}_X\} \subseteq \mathcal{A}$; define m_Y^+ symmetrically. (To be explicit: the other data of m_X and m_Y^+ are given by restricting each of $\{\mathbb{P}, \alpha, \beta\}$ to \mathcal{A}_X and \mathcal{A}_Y^+ , respectively.)

This partition of m allows us to use the PDG Markov property. Suppose that for some $\gamma > 0$ that $\mu \in \llbracket m \rrbracket_\gamma^* = \llbracket m_X + m_Y^+ \rrbracket_\gamma^*$. We can then apply [Theorem 8.4.1](#), to find that \mathcal{X}_X and \mathcal{X}_Y^+ are independent given $\mathcal{X}_X \cap \mathcal{X}_Y^+$. We use standard properties of random variable independence [CIRV1-5 of [39](#), Theorem 4.4.4] to find that μ must satisfy:

$$\begin{aligned} & \mathcal{X}_X \perp\!\!\!\perp \mathcal{X}_Y^+ \mid \mathcal{X}_X \cap \mathcal{X}_Y^+ \\ \implies & (\mathcal{X}_X \setminus Z) \perp\!\!\!\perp (\mathcal{X}_Y^+ \setminus Z) \mid (\mathcal{X}_X \cap \mathcal{X}_Y^+) \cup Z & [\text{CIRV3}] \\ \implies & (X \setminus Z) \perp\!\!\!\perp (Y \setminus Z) \mid (\mathcal{X}_X \cap \mathcal{X}_Y^+) \cup Z & [\text{by CIRV2, as } X \subseteq \mathcal{X}_X \text{ and } Y \subseteq \mathcal{X}_Y^+] \\ \implies & (X \setminus Z) \perp\!\!\!\perp (Y \setminus Z) \mid Z & [\text{since } (\mathcal{X}_X \cap \mathcal{X}_Y^+) \cup Z = Z] \\ \iff & X \perp\!\!\!\perp Y \mid Z & [\text{standard; e.g., Exercise 4.18 of Halpern [39]}] \end{aligned}$$

Using only the PDG Markov property, we have now shown that every independence modeled by the Markov Network G also holds in every distribution $\mu \in \llbracket m \rrbracket_\gamma^*$. Moreover, G is chordal (as we will prove momentarily), and it is well-known that distributions that have the independencies of a chordal graph can be represented by tree marginals [[56](#), Theorem 4.12]. Therefore, there is a tree marginal μ representing every $\mu \in \llbracket m \rrbracket_\gamma^*$.

Claim 8.A.0.1. G is chordal.

Proof. Suppose that G contains a loop $X - Y - Z - W - X$. Suppose further, for contradiction, that neither X and Z nor Y and W share a cluster. Given a variable

V , it is easy to see that property (2) of the tree decomposition ensures that the subtree $\mathcal{T}(V) \subseteq \mathcal{T}$ induced by the clusters $C \in \mathcal{C}$ that contain V , is connected. By assumption, $\mathcal{T}(Y)$ and $\mathcal{T}(W)$ must be disjoint. There is an edge between Y and Z , so some cluster must contain both variables, meaning $\mathcal{T}(Y) \cap \mathcal{T}(Z)$ is non-empty. Similarly, $\mathcal{T}(Z) \cap \mathcal{T}(W)$ is non-empty because of the edge between Z and W . This creates an (indirect) connection in \mathcal{T} between $\mathcal{T}(Y)$ and $\mathcal{T}(W)$. Because \mathcal{T} is a tree, and $\mathcal{T}(Y) \cap \mathcal{T}(W) = \emptyset$, every path from a cluster $C_1 \in \mathcal{T}(Y)$ to a cluster $C_2 \in \mathcal{T}(W)$ must pass through $\mathcal{T}(Z)$, which is not part of $\mathcal{T}(Y)$ or $\mathcal{T}(W)$. $\mathcal{T}(X)$ and $\mathcal{T}(Y)$ intersect as well, meaning that, for any $C \in \mathcal{T}(X)$, there is a (unique) path from C to that point of intersection, then across edges of $\mathcal{T}(Y)$, then edges of $\mathcal{T}(Z)$, and finally connects to the clusters of $\mathcal{T}(W)$. And also, since \mathcal{T} is a tree, that path must be unique. The problem is that there is also an edge between X and W , so there's some cluster that contains X and W ; let's call it C_0 . It's distinct from the cluster D_0 that contains Z and W , since no cluster contains both X and Z by assumption. The unique path from C_0 to D_0 intersects with $\mathcal{T}(Y)$. But now $W \in C_0 \cap D_0$, and by the running intersection property, every node along this unique path must contain W as well. But this contradicts our assumption that W is disjoint from Y ! So G is chordal. \square

\square

8.A.2 Correctness and Efficiency of Inference via Exponential Conic Programming

Proposition 8.3.1. *If (μ, \mathbf{u}) is a solution to (8.5), then $\mu \in \llbracket \mathbf{m} \rrbracket_0^*$, and $\sum_{(a,s,t) \in \mathcal{VA}} \beta_a u_{a,s,t} = \langle\!\langle \mathbf{m} \rangle\!\rangle_0$.*

Proof. Suppose that (μ, \mathbf{u}) is a solution to (8.5). The exponential cone constraints ensure that, for every $(a, s, t) \in \mathcal{VA}$,

$$u_{a,s,t} \geq \mu(s, t) \log \frac{\mu(s, t)}{\mathbb{P}_a(t|s)\mu(s)}$$

where $\mu(s, t)$ and $\mu(s)$, as usual, are shorthand for $\mu(S_a=s, T_a=t)$ and $\mu(S_a=s)$, respectively.

Suppose, for contradiction, that one of these inequalities is strict at some index $(a', s', t') \in \mathcal{VA}$ for which $\beta_{a'} > 0$. Explicitly, this means

$$u_{a',s',t'} > \mu(s_0, t_0) \log \frac{\mu(s', t')}{\mathbb{P}_{a'}(t'|s')\mu(s')}.$$

In that case, we can define a vector $\mathbf{u}' = [u'_{a,s,t}]_{(a,s,t) \in \mathcal{VA}}$ which is identical to \mathbf{u} , except that at (a', s', t') , it is halfway between the two quantities described as different above. More precisely:

$$u'_{a',s',t'} = \frac{1}{2}u_{a',s',t'} + \frac{1}{2}\log\mu(s', t') \log \frac{\mu(s', t')}{\mathbb{P}_a(t'|s')\mu(s')}.$$

Note that $u'_{a',s',t'} < u_{a',s',t'}$, and also that, by construction, (μ, \mathbf{u}') also satisfies the constraints of (8.5). In more detail: for (a', s', t') it doesn't violate the associated exponential cone constraint, as

$$\left(\text{formally: } u'_{a',s',t'} = \frac{1}{2}u_{a',s',t'} + \frac{1}{2}\log\mu(s', t') \log \frac{\mu(s', t')}{\mathbb{P}_{a'}(t'|s')\mu(s')} > \mu(s', t') \log \frac{\mu(s', t')}{\mathbb{P}_{a'}(t'|s')\mu(s')} \right),$$

and \mathbf{u}' remains unchanged at the other indices, and so satisfies the constraints at those indices, because \mathbf{u} does. But now, because $u'_{a',s',t'} < u_{a',s',t'}$, and $\beta_{a'} > 0$, we also have

$$\sum_{(a,s,t) \in \mathcal{VA}} \beta_a u'_{a,s,t} > \sum_{(a,s,t) \in \mathcal{VA}} \beta_a u'_{a,s,t}.$$

Thus the objective value at (μ, \mathbf{u}') is strictly smaller than the one at (μ, \mathbf{u}) , both of which are feasible points. This contradicts the assumption that (μ, \mathbf{u}) is optimal.

We therefore conclude that none of these inequalities can be strict at points where $\beta_a > 0$. This can be compactly written as:

$$\begin{aligned} \forall (a, s, t) \in \mathcal{VA}. \quad \beta_a u_{a,s,t} &= \beta_a \mu(s, t) \log \frac{\mu(s, t)}{\mathbb{P}_a(t|s)\mu(s)} \\ \implies \sum_{(a,s,t) \in \mathcal{VA}} \beta_a u_{a,s,t} &= \sum_{(a,s,t) \in \mathcal{VA}} \beta_a \mu(s, t) \log \frac{\mu(s, t)}{\mathbb{P}_a(t|s)\mu(s)} = OInc_m(\mu). \end{aligned}$$

In other words, the objective of problem (8.5) at (μ, \mathbf{u}) is equal to the observational incompatibility $OInc_m(\mu)$ of μ with \mathbf{m} . And, because (μ, \mathbf{u}) minimizes this value among all joint distributions, μ must be a minimum of $OInc_m$.

More formally: assume for contradiction that μ is not a minimizer of $OInc_m$. Then there would be some other distribution μ' for which $OInc_m(\mu') < OInc_m(\mu)$. Let $\mathbf{u}'' := [\mu'(s, t) \log \frac{\mu'(s, t)}{\mathbb{P}_a(t|s)\mu'(s)}]_{(a,s,t) \in \mathcal{VA}}$. Clearly (μ', \mathbf{u}'') satisfies the constraints of the problem, and moreover,

$$\sum_{(a,s,t) \in \mathcal{VA}} \beta_a u_{a,s,t} = OInc_m(\mu) > OInc_m(\mu') = \sum_{(a,s,t) \in \mathcal{VA}} \beta_a u'_{a,s,t},$$

contradicting the assumption that the (μ, \mathbf{u}) is optimal for problem (8.5). Thus, μ is a minimizer of $OInc_m$, and the objective value is $\inf_\mu OInc_m(\mu) = \langle\!\langle \mathbf{m} \rangle\!\rangle_0$, as desired. \square

Proposition 8.3.2. *If $(\mu, \mathbf{u}, \mathbf{v})$ is a solution to (8.7), and $\boldsymbol{\beta} \geq \gamma \boldsymbol{\alpha}$, then μ is the unique element of $[\mathbf{m}]^*_\gamma$, and $\langle\!\langle \mathbf{m} \rangle\!\rangle_\gamma$ equals the objective of (8.7) evaluated at $(\mu, \mathbf{u}, \mathbf{v})$.*

For convenience, we repeat problem (8.7) (left) and an equivalent variant of it that we implement (right) below.

$$\underset{\mu, \mathbf{u}, \mathbf{v}}{\text{minimize}} \sum_{(a,s,t) \in \mathcal{VA}} (\beta_a - \alpha_a \gamma) u_{a,s,t} + \gamma \sum_{w \in \mathcal{VX}} v_w$$

(8.7)

$$- \sum_{(a,s,t) \in \mathcal{VA}^+} \alpha_a \gamma \mu(S_a=s, T_a=t) \log \mathbb{P}_a(t|s)$$

subject to $\mu \in \Delta \mathcal{VX}$, $(-\mathbf{v}, \mu, \mathbf{1}) \in K_{\text{exp}}^{\mathcal{VX}}$,

$$\forall a \in \mathcal{A}. (-\mathbf{u}_a, \mu(T_a, S_a), \mathbb{P}_a(T_a|S_a)\mu(S_a)) \in K_{\text{exp}}^{\mathcal{VX}}$$

$$\forall (a, s, t) \in \mathcal{VA}^0. \mu(S_a=s, T_a=t) = 0;$$

$$\underset{\mu, \mathbf{u}, \mathbf{v}}{\text{minimize}} \sum_{(a,s,t) \in \mathcal{VA}} (\beta_a - \alpha_a \gamma) u_{a,s,t} + \gamma \sum_{w \in \mathcal{VX}} v_w$$

(8.7b)

$$- \sum_{(a,s,t) \in \mathcal{VA}^+} \beta_a \mu(S_a=s, T_a=t) \log \mathbb{P}_a(t|s)$$

subject to $\mu \in \Delta \mathcal{VX}$, $(-\mathbf{v}, \mu, \mathbf{1}) \in K_{\text{exp}}^{\mathcal{VX}}$,

$$\forall a \in \mathcal{A}. (-\mathbf{u}_a, \mu(T_a, S_a), [\mu(S_a=s)]_{(s,t) \in \mathcal{VX}^+}) \in K_{\text{exp}}^{\mathcal{VX}}$$

$$\forall (a, s, t) \in \mathcal{VA}^0. \mu(S_a=s, T_a=t) = 0.$$

Proof. We start with the problem on the left, which is (8.7) from the main text.

Suppose that $(\mu, \mathbf{u}, \mathbf{v})$ is a solution to (8.7). The exponential constraints ensure that

$$\forall (a, s, t) \in \mathcal{VA}. u_{a,s,t} \geq \mu(s, t) \log \frac{\mu(t|s)}{\mathbb{P}_a(t|s)} \quad \text{and} \quad \forall w \in \mathcal{VX}. v_w \geq \mu(w) \log \mu(w).$$

As in the previous proof, we claim that these must hold with equality (except possibly for $u_{a,s,t}$ at indices satisfying $\beta_a = \gamma \alpha_a$, when it doesn't matter). This is because otherwise one could reduce the value of a component of u or v while still satisfying all of the constraints, to obtain a strictly smaller objective, contradicting the assumption that $(\mu, \mathbf{u}, \mathbf{v})$ minimizes it.

Thus, \mathbf{v} is a function of μ , as is every value of \mathbf{u} that affects the objective value

of (8.7), meaning that this objective value can be written as a function of μ alone:

$$\begin{aligned}
& \sum_{(a,s,t) \in \mathcal{VA}} (\beta_a - \alpha_a \gamma) u_{a,s,t} + \gamma \sum_{w \in \mathcal{VX}} v_w - \sum_{(a,s,t) \in \mathcal{VA}^+} \alpha_a \gamma \mu(s, t) \log \mathbb{P}_a(t|s) \\
&= \sum_{(a,s,t) \in \mathcal{VA}} (\beta_a - \alpha_a \gamma) \left(\mu(s, t) \log \frac{\mu(t|s)}{\mathbb{P}_a(t|s)} \right) + \gamma \sum_{w \in \mathcal{VX}} \mu(w) \log \mu(w) - \sum_{(a,s,t) \in \mathcal{VA}^+} \alpha_a \gamma \mu(s, t) \log \mathbb{P}_a(t|s) \\
&= \sum_{a \in \mathcal{A}} (\beta_a - \alpha_a \gamma) \sum_{(s,t) \in \mathcal{VA}} \left(\mu(s, t) \log \frac{\mu(t|s)}{\mathbb{P}_a(t|s)} \right) - \gamma H(\mu) - \sum_{a \in \mathcal{A}} \alpha_a \gamma \sum_{(s,t) \in \mathcal{VA}} \mu(s, t) \log \mathbb{P}_a(t|s) \\
&= \sum_{a \in \mathcal{A}} (\beta_a - \alpha_a \gamma) \sum_{(s,t) \in \mathcal{VA}} \mu(s, t) \left(\log \frac{1}{\mathbb{P}_a(t|s)} - \log \frac{1}{\mu(t|s)} \right) - \gamma H(\mu) - \sum_{a \in \mathcal{A}} \alpha_a \gamma \mathbb{E}_{\mu} [\log \mathbb{P}_a(T_a | S_a)] \\
&= \sum_{a \in \mathcal{A}} (\beta_a - \alpha_a \gamma) \mathbb{E}_{\mu} [-\log \mathbb{P}_a(T_a | S_a)] - \sum_{a \in \mathcal{A}} (\beta_a - \alpha_a \gamma) H_{\mu}(T_a | S_a) - \gamma H(\mu) - \sum_{a \in \mathcal{A}} \alpha_a \gamma \mathbb{E}_{\mu} [\log \mathbb{P}_a(T_a | S_a)] \\
&= \sum_{a \in \mathcal{A}} \left(-\alpha_a \gamma - (\beta_a - \alpha_a \gamma) \right) \mathbb{E}_{\mu} [\log \mathbb{P}_a(T_a | S_a)] + \sum_{a \in \mathcal{A}} (\alpha_a \gamma - \beta_a) H_{\mu}(T_a | S_a) - \gamma H(\mu) \\
&= -\sum_{a \in \mathcal{A}} \beta_a \mathbb{E}_{\mu} [\log \mathbb{P}_a(T_a | S_a)] + \sum_{a \in \mathcal{A}} (\alpha_a \gamma - \beta_a) H_{\mu}(T_a | S_a) - \gamma H(\mu).
\end{aligned}$$

(In the third step, we were able to convert \mathcal{VA}^+ to \mathcal{VA} because, as usual in when dealing with information-therotic quantities, we interpret $0 \log \frac{1}{0}$ as equal to zero, which is its limit.)

The algebra, for the right side variant (8.7b) is slightly simpler. In this case the middle conic constraint is almost the same, except for that $\mathbb{P}_a(t|s)$ has been replaced with 1, and so it ensures that $u_{a,s,t} = \mu(s, t) \log \mu(t | s)$ (i.e., the same as before, but without the probability in the denominator). So,

$$\begin{aligned}
& \sum_{(a,s,t) \in \mathcal{VA}} (\beta_a - \alpha_a \gamma) u_{a,s,t} + \gamma \sum_{w \in \mathcal{VX}} v_w - \sum_{(a,s,t) \in \mathcal{VA}^+} \beta_a \mu(s, t) \log \mathbb{P}_a(t|s) \\
&= \sum_{(a,s,t) \in \mathcal{VA}} (\beta_a - \alpha_a \gamma) \mu(s, t) \log \mu(t|s) + \gamma \sum_{w \in \mathcal{VX}} \mu(w) \log \mu(w) - \sum_{(a,s,t) \in \mathcal{VA}^+} \beta_a \mu(s, t) \log \mathbb{P}_a(t|s) \\
&= \sum_{a \in \mathcal{A}} (\beta_a - \alpha_a \gamma) \sum_{(s,t) \in \mathcal{VA}} \mu(s, t) \log \mu(t|s) - \gamma H(\mu) - \sum_{a \in \mathcal{A}} \beta_a \sum_{(s,t) \in \mathcal{VA}} \mu(s, t) \log \mathbb{P}_a(t|s) \\
&= \sum_{a \in \mathcal{A}} (\alpha_a \gamma - \beta_a) H_{\mu}(T_a | S_a) - \gamma H(\mu) - \sum_{a \in \mathcal{A}} \beta_a \mathbb{E}_{\mu} [\log \mathbb{P}_a(T_a | S_a)].
\end{aligned}$$

In either case, the objective value is equal to $\llbracket m \rrbracket_\gamma(\mu)$, by (8.6). Because $(\mu, \mathbf{u}, \mathbf{v})$ is optimal for this problem, we know that μ is a minimizer of $\llbracket m \rrbracket_\gamma(\mu)$, and that the objective value equals $\langle\!\langle m \rangle\!\rangle_\gamma$. \square

Lemma 8.A.1. *The gradient and Hessian of the conditional relative entropy are given by*

$$\begin{aligned} \left[\nabla_\mu D(\mu(X, Y) \parallel \mu(X)p(Y|X)) \right]_u &= \log \frac{\mu(Yu|Xu)}{p(Yu|Xu)} \\ \left[\nabla_\mu^2 D(\mu(X, Y) \parallel \mu(X)p(Y|X)) \right]_{u,v} &= \frac{\mathbb{1}[Xu=Xv \wedge Yu=Yv]}{\mu(Yu, Xu)} - \frac{\mathbb{1}[Xv=Xu]}{\mu(Xu)}, \end{aligned}$$

where $Xu = X(u)$ it the value of the variable X in the joint setting $u \in \mathcal{V}\mathcal{X}$ of all variables.

Proof. Represent μ as a vector $[\mu_w]_{w \in \mathcal{V}\mathcal{X}}$. We will make repeated use of the following facts:

$$\begin{aligned} \frac{\partial}{\partial \mu_u} [\mu(X=x)] &= \frac{\partial}{\partial \mu_u} [\mu(x)] = \sum_w \frac{\partial}{\partial \mu_u} [\mu_w] \mathbb{1}[Xw=x] = \mathbb{1}[Xu=x]; \quad \text{and} \\ \frac{\partial}{\partial \mu_u} [\mu(y|x)] &= \frac{\partial}{\partial \mu_u} \left[\frac{\mu(x,y)}{\mu(x)} \right] \\ &= \mu(x,y) \frac{\partial}{\partial \mu_u} \left[\frac{1}{\mu(x)} \right] + \frac{1}{\mu(x)} \frac{\partial}{\partial \mu_u} [\mu(x,y)] \\ &= -\mu(x,y) \frac{\mathbb{1}[Xu=x]}{\mu(x)^2} + \frac{1}{\mu(x)} \mathbb{1}[Xu=x \wedge Yu=y] \\ &= \frac{\mathbb{1}[Xu=x]}{\mu(x)} \left(\mathbb{1}[Yu=y] - \mu(y|x) \right). \end{aligned}$$

We now apply this to the (conditional) relative entropy:

$$\begin{aligned} \frac{\partial}{\partial \mu_u} [D(\mu(X, Y) \parallel \mu(X)p(Y|X))] \\ = \frac{\partial}{\partial \mu_u} \left[\sum_w \mu_w \log \frac{\mu(Yw|Xw)}{p(Yw|Xw)} \right] \end{aligned}$$

$$\begin{aligned}
&= \sum_w \mathbb{1}[u=w] \log \frac{\mu(Yw|Xw)}{p(Yw|Xw)} + \sum_w \mu_w \frac{\partial}{\partial \mu_u} \left[\log \frac{\mu(Yw|Xw)}{p(Yw|Xw)} \right] \\
&= \log \frac{\mu(Yu|Xu)}{p(Yu|Xu)} + \sum_w \mu_w \frac{p(Yw|Xw)}{\mu(Yw|Xw)} \frac{\partial}{\partial \mu_u} \left[\frac{\mu(Yw|Xw)}{p(Yw|Xw)} \right] \\
&= \log \frac{\mu(Yu|Xu)}{p(Yu|Xu)} + \sum_w \mu_w \frac{1}{\mu(Yw|Xw)} \frac{\partial}{\partial \mu_u} \left[\mu(Yw|Xw) \right] \\
&= \log \frac{\mu(Yu|Xu)}{p(Yu|Xu)} + \sum_w \mu_w \frac{1}{\mu(Yw|Xw)} \frac{\mathbb{1}[Xu = Xw]}{\mu(Xw)} \left(\mathbb{1}[Yu = Yw] - \mu(Yw|Xw) \right) \\
&= \log \frac{\mu(Yu|Xu)}{p(Yu|Xu)} + \sum_w \mu_w \frac{\mathbb{1}[Xu = Xw \wedge Yu = Yw]}{\mu(Xw, Yw)} - \sum_w \mu_w \frac{\mathbb{1}[Xu = Xw]}{\mu(Xw)} \\
&= \log \frac{\mu(Yu|Xu)}{p(Yu|Xu)} + \frac{1}{\mu(Xu, Yu)} \sum_w \mu_w \mathbb{1}[Xu = Xw \wedge Yu = Yw] - \frac{1}{\mu(Xu)} \sum_w \mu_w \mathbb{1}[Xu = Xw] \\
&= \log \frac{\mu(Yu|Xu)}{p(Yu|Xu)} + \frac{\mu(Xu, Yu)}{\mu(Xu, Yu)} - \frac{\mu(Xu)}{\mu(Xu)} \\
&= \log \frac{\mu(Yu|Xu)}{p(Yu|Xu)} + 1 - 1 \\
&= \log \frac{\mu(Yu|Xu)}{p(Yu|Xu)}.
\end{aligned}$$

This allows us to compute the Hessian of the conditional relative entropy, whose components are

$$\begin{aligned}
\frac{\partial^2}{\partial \mu_u \partial \mu_v} \left[D(\mu(XY) \parallel \mu(X)p(Y|X)) \right] &= \frac{\partial}{\partial \mu_v} \left[\log \frac{\mu(Yu|Xu)}{p(Yu|Xu)} \right] \\
&= \frac{p(Yu|Xu)}{\mu(Yu|Xu)} \frac{1}{p(Yu|Xu)} \frac{\partial}{\partial \mu_v} \left[\mu(Yu|Xu) \right] \\
&= \frac{1}{\mu(Yu|Xu)} \frac{\mathbb{1}[Xv = Xu]}{\mu(Xu)} \left(\mathbb{1}[Yv = Yu] - \mu(Yu|Xu) \right) \\
&= \frac{\mathbb{1}[Xu = Xv \wedge Yu = Yv]}{\mu(Yu, Xu)} - \frac{\mathbb{1}[Xv = Xu]}{\mu(Xu)}.
\end{aligned}$$

□

Lemma 8.A.2. *Let $p(Y|X)$ be a cpd, and suppose that $\mu_0, \mu_1 \in \Delta\mathcal{V}(X, Y)$ are joint distributions that have different conditional marginals on Y given X ; that is, that there exist $(x, y) \in \mathcal{V}(X, Y)$ such that $\mu_0(x, y)\mu_1(x) \neq \mu_1(x, y)\mu_0(x)$. Then the conditional*

relative entropy $D(\mu(X, Y) \parallel \mu(X)p(Y|X))$ is strictly convex in μ along the line segment from μ_0 to μ_1 . More precisely, for $t \in [0, 1]$, if we define $\mu_t := (1 - t)\mu_0 + t\mu_1$, then the function

$$t \mapsto D(\mu_t(X, Y) \parallel \mu_t(X)p(Y|X)) \quad \text{is strictly convex.}$$

Proof. The function of interest can fail to be strictly convex only if the direction δ along $\mu_1 - \mu_0$ is in the null-space of the Hessian matrix $\mathbf{H}(\mu)$ of the (conditional) relative entropy. By Lemma 8.A.1,

$$\mathbf{H}_{(xy), (x'y')} = \frac{\mathbb{1}[x=x' \wedge y=y']}{\mu(x, y)} - \frac{\mathbb{1}[x=x']}{\mu(x)}.$$

Consider a function $\delta : \mathcal{V}(X, Y) \rightarrow \mathbb{R}$ that is not identically zero, which can be viewed as a vector $\boldsymbol{\delta} = [\delta(x, y)]_{(x,y) \in \mathcal{V}(X, Y)} \in \mathbb{R}^{\mathcal{V}(X, Y)}$. We can also view δ as a (signed) measure on $\mathcal{V}(X, Y)$, that has marginals in the usual sense. In particular, we use the analogous notation

$$\delta(x) := \sum_{y \in \mathcal{V}Y} \delta(x, y).$$

We then compute

$$\begin{aligned} (\mathbf{H}(\mu) \boldsymbol{\delta})_{x,y} &= \sum_{x',y'} \delta(x', y') \left(\frac{\mathbb{1}[x=x' \wedge y=y']}{\mu(x, y)} - \frac{\mathbb{1}[x=x']}{\mu(x)} \right) \\ &= \frac{\delta(x, y)}{\mu(x, y)} - \frac{\delta(x)}{\mu(x)}. \end{aligned}$$

and also

$$\begin{aligned}
\boldsymbol{\delta}^\top \mathbf{H}(\mu) \boldsymbol{\delta} &= \sum_{x,y} \delta(x,y) (\mathbf{H}(\mu) \boldsymbol{\delta})_{x,y} \\
&= \sum_{x,y} \delta(x,y) \left(\frac{\delta(x,y)}{\mu(x,y)} - \frac{\delta(x)}{\mu(x)} \right) \\
&= \sum_{x,y} \frac{\delta(x,y)^2}{\mu(x,y)} - \sum_x \frac{\delta(x)}{\mu(x)} \sum_y \delta(x,y) \\
&= \sum_{x,y} \frac{\delta(x,y)^2}{\mu(x,y)} - \sum_x \frac{\delta(x)^2}{\mu(x)} \\
&= \sum_x \frac{\delta(x)^2}{\mu(x)} \left(\sum_y \frac{\delta(x,y)^2}{\delta(x)^2 \mu(y|x)} - 1 \right). \tag{8.16}
\end{aligned}$$

Now consider another discrete measure $|\delta|$, whose value at each component is the absolute value of the value of δ at that component, i.e., $|\delta|(x,y) := |\delta(x,y)|$. By construction, $|\delta|$ is now an unnormalized probability measure: $|\delta| = kq(X,Y)$, where $k = \sum_{x,y} |\delta(x,y)| > 0$ and $q \in \Delta \mathcal{V}(X,Y)$.

Note also that $|\delta|(x)^2 = (\sum_y |\delta(x,y)|)^2 \geq (\sum_y \delta(x,y))^2$, and strictly so if there are y, y' such that $\delta(x,y) < 0 < \delta(x,y')$. In other words, the vector $\boldsymbol{\delta}_x = [\delta(x,y)]_{y \in \mathcal{V}Y}$ is either non-negative or non-positive: $\boldsymbol{\delta}_x \geq 0$ or $\boldsymbol{\delta}_x \leq 0$ for each x . Meanwhile, $|\delta|(x,y)^2 = \delta(x,y)^2$ is unchanged. Thus, for every $x \in \mathcal{V}X$, we have:

$$\begin{aligned}
\sum_y \frac{\delta(x,y)^2}{\delta(x)^2 \mu(y|x)} - 1 &\geq \sum_y \frac{|\delta|(x,y)^2}{|\delta|(x)^2 \mu(y|x)} - 1 \\
&= \sum_y \frac{k^2 q(x,y)^2}{k^2 q(x)^2 \mu(y|x)} - 1 \\
&= \sum_y \frac{q(y|x)^2}{\mu(y|x)} - 1 \\
&= \chi^2(q(Y|x) \parallel \mu(Y|x)) \geq 0.
\end{aligned}$$

The final line depicts the χ^2 divergence between the distributions $q(Y|x)$ and

$\mu(Y|x)$, both distributions over Y . Since it is a divergence, this quantity is non-negative and equals zero if and only if $q(Y|x) = \mu(Y|x)$.

Picking up where we left off, we have:

$$\begin{aligned}\boldsymbol{\delta}^\top \mathbf{H}(\mu) \boldsymbol{\delta} &= \sum_x \frac{\delta(x)^2}{\mu(x)} \left(\sum_y \frac{\delta(x,y)^2}{\delta(x)^2 \mu(y|x)} - 1 \right) \\ &\geq \sum_x \frac{\delta(x)^2}{\mu(x)} \left(\sum_y \frac{|\delta|(x,y)^2}{|\delta|(x)^2 \mu(y|x)} - 1 \right) \\ &= \sum_x \frac{\delta(x)^2}{\mu(x)} \chi^2(q(Y|x) \parallel \mu(Y|x)) \geq 0.\end{aligned}$$

As a non-negatively weighted sum of non-negative numbers, this final quantity is non-negative, and equals zero if and only if, for each $x \in \mathcal{V}X$, we have either $q(Y|x) = \mu(Y|x)$, or $\delta(x) = 0$. Furthermore, if $\boldsymbol{\delta}^\top \mathbf{H}(\mu) \boldsymbol{\delta} = 0$, then *both* inequalities hold with equality. Therefore, we know that if $\delta(x) \neq 0$, then $\delta_x \geq 0$ or $\delta_x \leq 0$. These two conditions are also sufficient to show that $\boldsymbol{\delta}^\top \mathbf{H}(\mu) \boldsymbol{\delta} = 0$. To summarize what we know so far:

$$\begin{aligned}\boldsymbol{\delta}^\top \mathbf{H}(\mu) \boldsymbol{\delta} = 0 \iff \forall x \in \mathcal{V}X. \text{ either } (\delta_x \geq 0 \text{ or } \delta_x \leq 0) \text{ and } |\delta|(Y|x) = \mu(Y|x) \\ \text{or } \delta(x) = 0.\end{aligned}$$

The second possibility, however, is a mirage: it cannot occur. Let's now return to the expression we had in (8.16) before considering $|\delta|$. We've already shown that the contribution to the sum at each value of x is non-negative, so if $\boldsymbol{\delta}^\top \mathbf{H}(\mu) \boldsymbol{\delta}$ is equal to zero, each summand which depends on x must be zero as well. So if x is a value of X for which $\delta(x) = 0$, then

$$0 = \frac{1}{\mu(x)} \left(\sum_y \frac{\delta(x,y)^2}{\mu(y|x)} - \delta(x)^2 \right) = \frac{1}{\mu(x)} \sum_y \frac{\delta(x,y)^2}{\mu(y|x)} = \sum_y \frac{\delta(x,y)^2}{\mu(x,y)},$$

which is only possible if $\delta(x,y) = 0$ for all y . This allows us to compute, more

simply, that

$$\boldsymbol{\delta}^\top \mathbf{H}(\mu) \boldsymbol{\delta} = 0 \iff (\forall x. \boldsymbol{\delta}_x \geq \mathbf{0} \text{ or } \boldsymbol{\delta}_x \leq \mathbf{0}) \quad \text{and} \quad \forall (x, y) \in \mathcal{V}(X, Y). \delta(x, y)\mu(x) = \delta(x, y)\mu_0(x)$$

Finally, we are in a position to prove the lemma. Suppose that $\mu_0, \mu_1 \in \Delta\mathcal{V}(X, Y)$ and $(x^*, y^*) \in \mathcal{V}(X, Y)$ are such that $\mu_0(x^*, y^*)\mu_1(x^*) \neq \mu_1(x^*, y^*)\mu_0(x^*)$. So, the quantity

$$gap := \mu_1(x^*, y^*)\mu_0(x^*) - \mu_0(x^*, y^*)\mu_1(x^*) \quad \text{is nonzero.}$$

Then for all $t \in (0, 1)$ the intermediate point $\mu_t = (1-t)\mu_0 + t\mu_1$ must have different conditional marginals from both μ_0 and μ_1 , as

$$\begin{aligned} & \mu_t(x^*, y^*)\mu_0(x^*) - \mu_0(x^*, y^*)\mu_t(x^*) \\ &= \underbrace{(1-t)\mu_0(x^*, y^*)\mu_0(x^*)}_{=} + t\mu_1(x^*, y^*)\mu_0(x^*) - \underbrace{(1-t)\mu_0(x^*, y^*)\mu_0(x^*)}_{=} - t\mu_0(x^*, y^*)\mu_1(x^*) \\ &= t(\mu_1(x^*, y^*)\mu_0(x^*) - \mu_0(x^*, y^*)\mu_1(x^*)) \\ &= t \cdot gap \neq 0, \end{aligned}$$

and analogously for μ_1 ,

$$\begin{aligned} & \mu_t(x^*, y^*)\mu_1(x^*) - \mu_1(x^*, y^*)\mu_t(x^*) \\ &= (1-t)\mu_0(x^*, y^*)\mu_1(x^*) + \underbrace{t\mu_1(x^*, y^*)\mu_1(x^*)}_{=} - (1-t)\mu_1(x^*, y^*)\mu_0(x^*) - \underbrace{t\mu_1(x^*, y^*)\mu_1(x^*)}_{=} \\ &= (1-t)(\mu_0(x^*, y^*)\mu_1(x^*) - \mu_1(x^*, y^*)\mu_0(x^*)) \\ &= -(1-t) \cdot gap \neq 0. \end{aligned}$$

Then for any direction $\delta := k(\mu_0 - \mu_1)$ parallel to the segment between μ_0 and μ_1 (intuitively a tangent vector at μ_t , although this fact doesn't affect the

computation), of nonzero length ($k \neq 0$), we have:

$$\begin{aligned}
& \mu_t(x^*, y^*)\delta(x^*) - \delta(x^*, y^*)\mu_t(x^*) \\
&= k \mu_t(x^*, y^*) (\mu_0(x^*) - \mu_1(x^*)) - k (\mu_0(x^*, y^*) - \mu_1(x^*, y^*))\mu_t(x^*) \\
&= k \left(\mu_t(x^*, y^*)\mu_0(x^*) - \mu_t(x^*, y^*)\mu_1(x^*) - \mu_0(x^*, y^*)\mu_t(x^*) + \mu_1(x^*, y^*)\mu_t(x^*) \right) \\
&= k \left((\mu_t(x^*, y^*)\mu_0(x^*) - \mu_0(x^*, y^*)\mu_t(x^*)) + (\mu_1(x^*, y^*)\mu_t(x^*) - \mu_t(x^*, y^*)\mu_1(x^*)) \right) \\
&= k (+t \text{ gap} + (1-t) \text{ gap}) \\
&= k \text{ gap} \quad \neq 0.
\end{aligned}$$

So at every t , directions parallel to the segment are not in the null space of $\mathbf{H}(\mu_t)$, meaning that $\boldsymbol{\delta}^\top \mathbf{H}(\mu_t) \boldsymbol{\delta} > 0$ and so our function is strictly convex along this segment. \square

Proposition 8.3.5. *If $\nu \in [\![\mathcal{M}]\!]_0^*$ and (μ, \mathbf{u}) solves the problem*

$$\begin{aligned}
& \underset{\mu, \mathbf{u}}{\text{minimize}} \quad \mathbf{1}^\top \mathbf{u} \\
& \text{subject to} \quad (-\mathbf{u}, \mu, \mathbf{k}) \in K_{\text{exp}}^{\mathcal{V}\mathcal{X}}, \quad \mu \in \Delta \mathcal{V}\mathcal{X}, \\
& \quad \forall S \xrightarrow{a} T \in \mathcal{A}. \quad \mu(S, T) \nu(S) = \mu(S) \nu(S, T),
\end{aligned} \tag{8.10}$$

then $[\![\mathcal{M}]\!]_{0+}^* = \{\mu\}$ and $\mathbf{1}^\top \mathbf{u} = SInc_m(\mu)$.

Proof. Suppose that $(-\mathbf{u}, \mu, \mathbf{k})$ is a solution to problem (8.10). The second constraint, by Proposition 8.3.3, ensures that $\mu \in [\![\mathcal{M}]\!]_0^*$. Then

$$\begin{aligned}
(-\mathbf{u}, \mu, \mathbf{k}) \in K^{\mathcal{V}\mathcal{X}} \quad \implies \quad \forall w \in \mathcal{V}\mathcal{X}. \quad u_w \geq \mu(w) \log \frac{\mu(w)}{k_w} \\
&= \mu(w) \log \left(\mu(w) / \prod_{a \in \mathcal{A}} \mu(T_a(w) | S_a(w))^{\alpha_a} \right).
\end{aligned}$$

The same logic as in the proofs of Propositions 8.3.1 and 8.3.2 shows that this inequality must be tight, or else $(-\mathbf{u}, \mu, \mathbf{k})$ would not be optimal for (8.10). So, \mathbf{u} is a function of μ . Also, by Proposition 8.3.4, the problem objective satisfies

$$\mathbf{1}^\top \mathbf{u} = \sum_{w \in \mathcal{VX}} u_w = SInc_m(\mu).$$

Finally, because μ is optimal, it must be the unique distribution $[\![\mathbf{m}]\!]^*$, which among those distributions that minimize $OInc_m$, also minimizes $SInc_m$, meaning $\mu = [\![\mathbf{m}]\!]^*$. \square

Proposition 8.4.2. *If (μ, \mathbf{u}) is a solution to (8.11), then*

- (a) μ is a calibrated, with $\Pr_\mu \in [\![\mathbf{m}]\!]_0^*$, and
- (b) the objective of (8.11) evaluated at \mathbf{u} equals $\langle\!\langle \mathbf{m} \rangle\!\rangle_0$.

Proof. The final constraints alone are enough to ensure that μ is calibrated. Much like before, the exponential conic constraints tell us that

$$\forall (a, s, t) \in \mathcal{VA}. \quad u_{a,s,t} \geq \mu_{C_a}(s, t) \log \frac{\mu_{C_a}(s, t)}{\mu_{C_a}(s) \mathbb{P}_a(t|s)}$$

and they hold with equality (at least at those indices where $\beta_a > 0$) because \mathbf{u} is optimal. So

$$\begin{aligned} \sum_{(a,s,t) \in \mathcal{VA}} \beta_a u_{a,s,t} &= \sum_{(a,s,t) \in \mathcal{VA}} \beta_a \mu_{C_a}(s, t) \log \frac{\mu_{C_a}(s, t)}{\mu_{C_a}(s) \mathbb{P}_a(t|s)} \\ &= \sum_a \beta_a \sum_{(s,t) \in \mathcal{V}a} \mu_{C_a}(s, t) \log \frac{\mu_{C_a}(s, t)}{\mu_{C_a}(s) \mathbb{P}_a(t|s)} \\ &= OInc_m(\Pr_\mu). \end{aligned}$$

Because μ is optimal, it is the choice of calibrated tree marginal that minimizes this quantity. By ?? 8.4.1.1, the distribution $[\![\mathbf{m}]\!]^*$ can be represented by such a tree

marginal, and by Richardson and Halpern [82, Prop. 3.4], this distribution minimizes $OInc_m$. All this is to say that there exist tree marginals of this form whose corresponding distributions attain the minimum value $OInc_m(\Pr_\mu) = \langle\!\langle m \rangle\!\rangle_0$. So μ must be one of them, as it minimizes $OInc(\Pr_\mu)$ among such tree marginals by assumption. Thus $\Pr_\mu \in \llbracket m \rrbracket_0^*$ and the objective value of (8.11) equals $\langle\!\langle m \rangle\!\rangle_0$. \square

Proposition 8.4.3. *If $(\mu, \mathbf{u}, \mathbf{v})$ is a solution to (8.14) and $\beta \geq \gamma\alpha$, then \Pr_μ is the unique element of $\llbracket m \rrbracket_\gamma^*$, and the objective of (8.14) at $(\mu, \mathbf{u}, \mathbf{v})$ equals $\langle\!\langle m \rangle\!\rangle_\gamma$.*

Proof. Suppose that $(\mu, \mathbf{u}, \mathbf{v})$ is a solution to (8.14). The first and fourth lines of constraints ensures that μ is indeed a calibrated tree marginal. The second line of constraints, plays exactly the same role that it did in the previous problems, most directly in the variant (8.11) for $\gamma = 0$. In particular, it tells says

$$\forall (a, s, t) \in \mathcal{VA}. \quad u_{a,s,t} \geq \mu_{C_a}(s, t) \log \frac{\mu_{C_a}(s, t)}{\mu_{C_a}(s) \mathbb{P}_a(t|s)}$$

as before, this holds with equality (at least at those indices where $\beta_a > \alpha_a \gamma$) because \mathbf{u} is optimal. Because $\beta \geq \gamma\alpha$ by assumption, either $\beta_a > \gamma\alpha_a$ or the two are equal, for every $a \in \mathcal{A}$. Either way, the argument used at this point in the proof of Proposition 8.4.2 goes through, giving us:

$$\begin{aligned} \sum_{(a,s,t) \in \mathcal{VA}} (\beta_a - \alpha_a \gamma) u_{a,s,t} &= \sum_{(a,s,t) \in \mathcal{VA}} ((\beta_a - \alpha_a \gamma) \mu_{C_a}(s, t) \log \frac{\mu_{C_a}(s, t)}{\mu_{C_a}(s) \mathbb{P}_a(t|s)}) \\ &= \sum_a (\beta_a - \alpha_a \gamma) \sum_{(s,t) \in \mathcal{V}a} \mu_{C_a}(s, t) \log \frac{\mu_{C_a}(s, t)}{\mu_{C_a}(s) \mathbb{P}_a(t|s)} \\ &= \sum_a (\beta_a - \alpha_a \gamma) D\left(\mu_{C_a}(S_a, T_a) \parallel \mu_{C_a}(S_a) \mathbb{P}_a(T_a|S_a)\right) \end{aligned}$$

This time, though, that's not the problem objective. In this regard, our problem (8.14) is more closely related to (8.14).

Before we get to that, we have to first bring in the final collection of exponential constraints, which show that

$$\forall C \in \mathcal{C}. \forall c \in \mathcal{V}(C). \quad v_{C,c} \geq \mu_C(c) \log \frac{\mu_C(c)}{VCP_C(c)},$$

and yet again these constraints hold with equality, for otherwise \mathbf{v} would not be optimal (since we assumed $\gamma > 0$). Therefore,

$$\sum_{(C,c) \in \mathcal{VC}} v_{C,c} = \sum_{(C,c) \in \mathcal{VC}} \mu_C(c) \log \frac{\mu_C(c)}{VCP_C(c)} = -H(\Pr_\mu) \quad \text{by Equation (8.13).}$$

The objective of our problem (8.14) is essentially the same as that of (8.7), so the analysis in the proof of Proposition 8.3.2 applies with only a handful of superficial modifications. Using that proof to take a shortcut, the objective of (8.14) must equal

$$\begin{aligned} & \sum_{(a,s,t) \in \mathcal{VA}} (\beta_a - \alpha_a \gamma) u_{a,s,t} + \gamma \sum_{(C,c) \in \mathcal{VC}} v_{C,c} - \sum_{(a,s,t) \in \mathcal{VA}^+} \alpha_a \gamma \mu_{C_a}(s, t) \log \mathbb{P}_a(t|s) \\ &= \sum_{(a,s,t) \in \mathcal{VA}} (\beta_a - \alpha_a \gamma) \mu_{C_a}(s, t) \log \frac{\mu_{C_a}(s, t)}{\mu_{C_a}(s) \mathbb{P}_a(t|s)} - \gamma H(\Pr_\mu) - \sum_{(a,s,t) \in \mathcal{VA}^+} \alpha_a \gamma \mu_{C_a}(s, t) \log \mathbb{P}_a(t|s) \\ &= \sum_{a \in \mathcal{A}} \beta_a \mathbb{E}_{\mu_{C_a}} [\log \mathbb{P}_a(T_a | S_a)] + \sum_{a \in \mathcal{A}} (\alpha_a \gamma - \beta_a) H_{\Pr_\mu}(T_a | S_a) - \gamma H(\Pr_\mu) \\ &= \llbracket \mathbf{m} \rrbracket_\gamma(\Pr_\mu), \quad . \end{aligned}$$

Finally, since μ is such that this quantity is minimized, and because its unique minimizer can be represented as a cluster tree (by ?? 8.4.1.1), we conclude that μ must be the cluster tree representation of it. Therefore, \Pr_μ is the unique element of $\llbracket \mathbf{m} \rrbracket_\gamma^*$, and the objective at $(\mu, \mathbf{u}, \mathbf{v})$ equals $\langle\!\langle \mathbf{m} \rangle\!\rangle_\gamma$, as desired. \square

Proposition 8.4.4. *If (μ, \mathbf{u}) is a solution to (8.15), then μ is a calibrated tree marginal and $\llbracket \mathbf{m} \rrbracket_{0^+}^* = \{\Pr_\mu\}$.*

Proof. Suppose that (μ, \mathbf{u}) is a solution to (8.15). The exponential cone constraints state that

$$\begin{aligned} \forall C \in \mathcal{C}, \forall c \in \mathcal{V}(C). \quad u_{C,c} &\geq \mu_C(c) \log \frac{\mu_C(c)}{k_{C,c} VCP_C(c)} \\ &= \mu_C(c) \log \frac{\mu_C(c)}{VCP_C(c)} - \mu_C(c) \log \prod_{a \in \mathcal{A}_C} \nu_C(T_a(c)|S_a(c))^{\alpha_a} \\ &= \mu_C(c) \log \frac{\mu_C(c)}{VCP_C(c)} - \mu_C(c) \sum_{a \in \mathcal{A}_C} \alpha_a \log \nu_C(T_a(c)|S_a(c)), \end{aligned}$$

and once again this holds with equality, as each $u_{C,c}$ is minimal with this property.

The third line of constraints

$$\forall a \in \mathcal{A}. \quad \mu_{C_a}(S_a, T_a) \nu_{C_a}(S_a) = \mu_{C_a}(S_a) \nu_{C_a}(S_a, T_a)$$

and the assumption that $\Pr_{\nu} \in \llbracket m \rrbracket_0^*$, suffice to ensure that $\Pr_{\mu} \in \llbracket m \rrbracket_0^*$ by Proposition 8.3.3. They also allow us to replace each $\nu_{C_a}(T_a(c)|S_a(c))$ with $\nu_{C_a}(T_a(c)|S_a(c))$, in cases where $S_a(c) \neq 0$. Therefore, we calculate the objective to be:

$$\begin{aligned} \mathbf{1}^\top \mathbf{u} &= \sum_{C \in \mathcal{C}} \sum_{c \in \mathcal{V}(C)} \left(\mu_C(c) \log \frac{\mu_C(c)}{VCP_C(c)} - \mu_C(c) \sum_{a \in \mathcal{A}_C} \alpha_a \log \nu_C(T_a(c)|S_a(c)) \right) \\ &= \sum_{C \in \mathcal{C}} \sum_{c \in \mathcal{V}(C)} \mu_C(c) \log \frac{\mu_C(c)}{VCP_C(c)} - \sum_{C \in \mathcal{C}} \sum_{c \in \mathcal{V}(C)} \mu_C(c) \sum_{a \in \mathcal{A}} \mathbb{1}[C = C_a] \alpha_a \log \nu_C(T_a(c)|S_a(c)) \\ &= -H(\Pr_{\mu}) - \sum_{a \in \mathcal{A}} \alpha_a \sum_{C \in \mathcal{C}} \mathbb{1}[C = C_a] \sum_{c \in \mathcal{V}(C)} \mu_C(c) \log \nu_C(T_a(c)|S_a(c)) \quad [\text{by (8.13)}] \\ &= -H(\Pr_{\mu}) - \sum_{a \in \mathcal{A}} \alpha_a \sum_{c \in \mathcal{V}(C)_a} \mu_{C_a}(c) \log \nu_{C_a}(T_a(c)|S_a(c)) \\ &= -H(\Pr_{\mu}) - \sum_{a \in \mathcal{A}} \alpha_a \sum_{c \in \mathcal{V}(C)_a} \mu_{C_a}(c) \log \mu_{C_a}(T_a(c)|S_a(c)) \quad \left[\text{since } \mu_{C_a}(S_a(c)) > 0 \text{ whenever } \mu_{C_a}(S_a(c)) \neq 0 \right] \\ &= -H(\Pr_{\mu}) + \sum_{a \in \mathcal{A}} \alpha_a H_{\Pr_{\mu}}(T_a|S_a) \\ &= SInc_m(\Pr_{\mu}). \end{aligned}$$

To summarize: \Pr_{μ} minimizes $SInc_m(\Pr_{\mu})$ among calibrated tree marginals

with conditional marginals matching those of ν . Since we know that there is a unique distribution that minimizes $SInc_m$ among the elements $[\![\mathcal{M}]\!]_0^*$, and also that this distribution can be represented by a tree marginal (by ?? 8.4.1.1), we conclude that μ must represent this distribution. Thus, $\Pr_\mu = [\![\mathcal{M}]\!]^*$ as desired. \square

The next lemma packages the results of Dahl and Andersen [21], Nesterov et al. [69] in a precise form that we will be able to make use of.

Lemma 8.A.3. *Fix integers $n_o, n_e \in \mathbb{N}$, and let $n := 3n_e + n_o$. Suppose that $K = \mathbb{R}_{\geq 0}^{n_o} \times K_{\text{exp}}^{n_e} \subset \mathbb{R}^n$ is a product cone, consisting of n_o copies of the non-negative orthant and n_e copies of the exponential cone. Let $\mathbf{c} \in [-1, 1]^n$ and $\mathbf{b} \in [-1, 1]^m$ be vectors, and $A \in [-1, 1]^{m \times n}$ be a matrix, defining an exponential conic program*

$$\underset{\mathbf{x} \in K}{\text{minimize}} \quad \mathbf{c}^\top \mathbf{x} \quad \text{subject to} \quad A\mathbf{x} = \mathbf{b}, . \quad (8.3)$$

If this program is strictly feasible (i.e., if there exists $\mathbf{x} \in \text{int } K$ such that $A\mathbf{x} = \mathbf{b}$), as is its dual problem

$$\underset{\mathbf{s} \in K^*, \mathbf{y} \in \mathbb{R}^m}{\text{maximize}} \quad \mathbf{b}^\top \mathbf{y} \quad \text{subject to} \quad A^\top \mathbf{y} + \mathbf{s} = \mathbf{c},$$

(i.e, if there exists $\mathbf{s} \in \text{int } K_$ such that $A^\top \mathbf{y} + \mathbf{s} = \mathbf{c}$), then both can be simultaneously solved to precision ϵ in $O(n(m+n)^\omega \log \frac{n+m}{\epsilon})$ time, where ω is the smallest exponent such that a linear system of k variables and equations can be solved in $O(k^\omega)$ time. Furthermore, MOSEK solves this problem in $O(n(m+n)^3 \log \frac{n+m}{\epsilon})$ time.*

Proof. For this, we begin by appealing to the algorithm and analysis of Badenbroek and Dahl [5], threading details through for this specific choice of cone K . To finish the proof, however, we will also need to supplement that analysis with

some other well-established results of Nesterov et al. [69] that the authors were no doubt familiar with, but did not bother referencing.

First, we'll need some background material from convex optimization. A *logarithmically homogeneous self-concordant barrier* with parameter ν (ν -LHSCB) for a cone K is a thrice differentiable strictly convex function $F : \text{int } K \rightarrow \mathbb{R}$ satisfying $F(tx) = F(x) - \nu \log t$ for all $t > 0$ and $x \in \text{int } K$. In some sense, the point of such a barrier function is to augment the optimization objective so that we remain within the cone during the optimization process.

For the positive orthant cone $\mathbb{R}_{\geq 0}$, the function $x \mapsto -\log x$ is a 1-LHSCB. We now fill in some background facts about exponential cones. The *dual* of the exponential cone is

$$\begin{aligned} K_{\exp}^* := & \{(s_1, s_2, s_3) \in \mathbb{R}^3 : \forall (x_1, x_2, x_3) \in K_{\exp}. x_1 s_1 + x_2 s_2 + x_3 s_3 \geq 0\} \\ = & \{(s_1, s_2, s_3) : -s_1 \log(-s_1/s_3) + s_1 - s_2 \leq 0, s_1 \leq 0, s_3 \geq 0\}. \end{aligned}$$

Consider points $x = (x_1, x_2, x_3) \in K_{\exp}$. The function

$$F_{\exp}(x) := -\log \left(x_2 \log \frac{x_1}{x_2} - x_3 \right) - \log x_1 x_2 \quad (8.17)$$

is a 3-LHSCB for K_{\exp} , since

$$\begin{aligned} F_{\exp}(tx) &= -\log \left(tx_2 \log \frac{tx_1}{tx_2} - tx_3 \right) - \log(t^2 x_1 x_2) \\ &= -\log \left(t \left(\log \frac{x_1}{x_2} - x_3 \right) \right) - \log(x_1 x_2) - 2 \log t \\ &= F_{\exp}(x) - 3 \log t \end{aligned}$$

Such barrier functions can be combined to act on product cones by summation. Concretely, suppose that for each $i \in \{1, \dots, k\}$, we have a ν_i -LHSCB $F_i : \text{int } K_i \rightarrow \bar{\mathbb{R}}$. For $x = (x_i)_{i=1}^k \in \prod_i K_i$, the function $F(x) := \sum_{i=1}^k F_i(x_i)$ is a

$(\sum_i \nu_i)$ -LHSCB for $\prod_i K_i$, since

$$F(tx) = \sum_{i=1}^k F_i(tx_i) = \sum_{i=1}^k (F(x_i) - \nu_i \log t) = F(x) - \sum_{i=1}^k \nu_i.$$

In this way, our product cone $K = \mathbb{R}_{\geq 0}^{n_o} \times K_{\text{exp}}^{n_e}$ admits a LHSCB F with parameter $\nu = n_o + 3n_e = n$. Furthermore it can be evaluated in $O(n)$ time, as can each component of its gradient $F'(x)$ and Hessian $F''(x) \in \mathbb{R}^{n \times n}$ at x , all of which can be expressed analytically. In addition, the convex conjugate of F also has a known analytic form.

Generally speaking, the idea behind primal-dual interior point methods [70] such as the one behind MOSEK, is to maintain both a point $x \in K$ and a dual point $s \in K_*$ (as well as $y \in \mathbb{R}^m$) and iteratively update them, as we slowly relax the barrier and approach a point on the boundary of the cone. The quantity $\mu(z) := \langle s, x \rangle / \nu \geq 0$, called the complementarity gap, is a measure of how close the process is to converging.

Because the initial points may not satisfy the constraints, instead the standard algorithms work with “extended points” $\bar{x} = (x, \tau)$ and $\bar{s} = (s, \kappa)$, for which the analogous complementarity gap is $\mu^e(\bar{x}, \bar{s}) := (\langle x, s \rangle + \kappa\tau) / (\nu + 1)$. Altogether, the data at each iteration may be summarized as a point $z = (y, x, \tau, s, \kappa) \in \mathbb{R} \times (K \times \mathbb{R}_{\geq 0}) \times (K_* \times \mathbb{R}_{\geq 0})$. The primary object of interest is then something called the *homogenous self-dual* model. Originally due to Nesterov et al. [69] and also used by others [88], it can be defined as a linear operator:

$$G : \bar{\mathbb{R}}^{m+2n+2} \rightarrow \bar{\mathbb{R}}^{n+m+1}$$

$$G(y, x, \tau, s, \kappa) := \begin{bmatrix} 0 & A & -b \\ -A^\top & 0 & c \\ b^\top & -c^\top & 0 \end{bmatrix} \begin{bmatrix} y \\ x \\ \tau \end{bmatrix} - \begin{bmatrix} 0 \\ s \\ k \end{bmatrix}.$$

The reason for our interest is that if z is such that $G(z) = 0$ and $\tau > 0$, then (x/τ) is a solution to the primal problem, and $(y, s)/\tau$ is a solution to the dual problem [88, Lemma 1], while if $G(z) = 0$ and $\kappa > 0$, then at least one of the two problems is infeasible.

We now are in a better position to describe the algorithm. According to the MOSEK documentation [21], for the exponential cone, begins with an initial point

$$\mathbf{v} := (1.291, 0.805, -0.828) \in (K_{\text{exp}} \cap K_{\text{exp}}^*)$$

for this particular cone K , the algorithm begins at the initial point

$$z_0 := (y_0, x_0, \tau_0, s_0, \kappa_0) \quad \text{where} \quad x_0 = s_0 = (\overbrace{1, \dots, 1}^{n_o \text{ copies}}, \overbrace{\mathbf{v}, \dots, \mathbf{v}}^{n_e \text{ copies}}) \in (\mathbb{R}_{\geq 0})^{n_o} \times (K_{\text{exp}} \cap K_{\text{exp}}^*)^{n_e},$$

$$y_0 = \mathbf{0} \in \mathbb{R}^m, \quad \tau_0 = \kappa_0 = 1.$$

At each iteration, the first step is to predict a direction for which Badenbroek and Dahl [5] compute a scaling matrix W . To describe it, we first need to define *shadow iterates*

$$\tilde{x} := -F'_*(s) \quad \text{and} \quad \tilde{s} := -F'(x).$$

which are in a sense reflections of s and x across their barrier functions, and can be computed in $O(n)$ time. The analogous notion of complementarity can then be defined as $\tilde{\mu}(z) := \langle \tilde{x}, \tilde{s} \rangle / \nu$. The scaling matrix, which we do not interpret here, can then be calculated as:

$$W := \mu F''(x) + \frac{ss^\top}{\nu\mu} - \frac{\mu\tilde{s}\tilde{s}^\top}{\nu} + \frac{(s - \mu\tilde{s})(s - \mu\tilde{s})^\top}{(s - \mu\tilde{s})^\top(x - \mu\tilde{x})} - \frac{\mu[F''(x)\tilde{x} - \tilde{\mu}\tilde{s}][F''(x)\tilde{x} - \tilde{\mu}\tilde{s}]^\top}{\tilde{x}^\top F''(x)\tilde{x} - \nu\tilde{\mu}^2} \quad (8.18)$$

Doing so requires $O(n^2)$ steps (although it may be parallelized). The first four terms clearly require $O(n^2)$ steps, since each one is an outer product resulting in

a $n \times n$ matrix. The last term computes a matrix-vector product (which requires $O(n^2)$ steps), and computes an outer product with the resulting vector, which takes $O(n^2)$ steps as well.

The next step involves finding a solution $\Delta z^{\text{aff}} = (\dots)$ to the system of equations

$$G(\Delta z^{\text{aff}}) = -G(z) \quad (8.19\text{a})$$

$$\tau \Delta \kappa^{\text{aff}} + \Delta \tau^{\text{aff}} = -\tau \kappa \quad (8.19\text{b})$$

$$W \Delta x^{\text{aff}} + \Delta s^{\text{aff}} = -s. \quad (8.19\text{c})$$

(8.19a-c) describe a system of $(n+m+1) + 1 + (n) = 2n+m+2$ equations and equally many unknowns, and solved in $O((n+m)^\omega)$ steps. It may be possible to exploit the sparsity of G to do better.

The next step is to center that search direction so that it lies on the central path. This is done by finding a solution Δz^{cen} to

$$G(\Delta z^{\text{cen}}) = G(z) \quad (8.20\text{a})$$

$$\tau \Delta \kappa^{\text{cen}} + \kappa \Delta \tau^{\text{cen}} = \mu^e \quad (8.20\text{b})$$

$$W \Delta x^{\text{cen}} + \Delta s^{\text{cen}} = \mu^e \tilde{s}, \quad (8.20\text{c})$$

which again can be done in $O((n+m)^3)$ steps with Gaussian elimination, or with a fancier solver in $O((n+m)^2 \cdot 332)$ steps. The two updates are then applied to the current point z to obtain

$$z_+ = (y_+, x_+, \tau_+, s_+, \kappa_+) := z + \alpha(\Delta z^{\text{aff}} + \gamma \Delta z^{\text{cen}}).$$

Finally, a “correction step”, which is the primary innovation of Badenbroek and Dahl [5] and used in MOSEK’s algorithm, is a third direction Δz_+^{cor} , which is

found by solving the system of equations

$$G(\Delta z^{\text{cor}}) = 0 \quad (8.21\text{a})$$

$$\tau_+ \Delta \kappa^{\text{cor}} + \kappa_+ \Delta \tau^{\text{cor}} = 0 \quad (8.21\text{b})$$

$$W_+ \Delta x_+^{\text{cor}} + \Delta s^{\text{cen}} = \mu^e \tilde{s}, \quad (8.21\text{c})$$

where W_+ is defined the same way that W is, except that it uses the components of z_+ instead of z . After adding the correction step Δz_+^{cor} to z , we repeat the entire process. The full algorithm, then, is summarized as follows:

```

 $z \leftarrow (y_0, x_0, \tau_0, s_0, \kappa_0);$ 
while do
    Compute scaling matrix  $W$  as in (8.18);
    Find the solution  $\Delta z^{\text{aff}}$  to (8.19a-c), and the solution  $\Delta z^{\text{cen}}$  to (8.20a-c);
     $z_+ \leftarrow z + \alpha(\Delta z^{\text{aff}} + \gamma \Delta z^{\text{cen}});$ 
    Compute the saling matrix  $W_+$ ;
    Find the solution  $\Delta z_+^{\text{cor}}$  to (8.21a-c);
     $z \leftarrow z_+ + \Delta z_+^{\text{cor}};$ 

```

We have verified that each iteration of this process can be done in $O((n+m)^\omega)$ time. Their main result [5, Theorem 3], states that for every $\epsilon \in (0, 1)$, the algorithm results in a solution z satisfying

$$\mu^e(z) \leq \epsilon \quad \text{and} \quad \|G(z)\| \leq \epsilon \|G(z_0)\|$$

in $O(n \log(1/\epsilon))$ iterations, for a total cost of $O(n(m+n)^3 \log(1/\epsilon))$ time with Gaussian elimination, or $O(n(m+n)^{2.332} \log(1/\epsilon))$ time using the linear solver with best known asymptotic complexity as of 2022 [25].

Verifying that the solution is approximately optimal. What we have at this point is not quite enough: simply because the residual quantity $G(z)$ is

approximately zero (so that we have approximately solved the homogenous model), does not mean that we've approximately solved the original problem. Specifically, it's entirely possible a priori that the parameter τ goes to zero at the same rate as everything else, and the quantity (x/τ) does not converge to a solution to the primal problem. To address this issue, we must also trace the analysis of the seminal work of Nesterov et al. [69], who use slightly different quantities, conflicting with the notation we have been using thus far.

Following Nesterov et al. [69, pg. 231], fix an initial point z_0 , and let *shifted feasible set* $\mathcal{F} := \{z \in \mathbb{R} \times K \times \mathbb{R}_{\geq 0} \times K^* \times \mathbb{R}_{\geq 0} : G(z) = G(z_0)\}$ be the collection of all points that have the same residual as z_0 . **Nesterov, Todd, and Ye** also refer to a complementary gap by $\mu(z)$ and define it identically, but the meaning of this parameter is different, because the set \mathcal{F} on which it's defined is quite distinct from (if closely related to) the iterates of **Badenbroek and Dahl**'s algorithm. In the service of clarity, will call this quantity $\mu^N(z^N)$, for $z^N = (y^N, x^N, \tau^N, s^N, \kappa^N) \in \mathcal{F}$.

Although we made a point of emphasizing that the two are distinct, the actual relationship between them is straightforward. Let $z = (y, x, \tau, s, \kappa)$ be the final output of Badenbroek and Dahl [5]. In proving their main theorem, they also prove that $G(z) = \epsilon G(z_0)$, and $\mu^e = \epsilon$; because G is linear, we know that $G(z/\epsilon) = G(z_0)$. This means that $z^N := z/\epsilon \in \mathcal{F}$. Therefore,

$$\mu^N(z^N) = \frac{1}{\nu + 1} \left(\left\langle \frac{s}{\epsilon}, \frac{x}{\epsilon} \right\rangle + \frac{\tau \kappa}{\epsilon \epsilon} \right) = \frac{1}{\epsilon^2} \mu^e(z) = \frac{1}{\epsilon}.$$

So, roughly speaking, μ^N and μ^e are reciprocals. **Badenbroek and Dahl** also prove that, every iterate z satisfies their assumption (A2): for a fixed constant β (equal to 0.9 in their analysis), $\beta \mu^e(z) \leq \tau \kappa$. Consequently, it happens that the same inequality holds with Nesterov's notation:

$$\tau^N \kappa^N = \frac{\tau \kappa}{\epsilon \epsilon} = \frac{\tau \kappa}{\epsilon^2} \geq \frac{\beta \epsilon}{\epsilon^2} = \frac{\beta}{\epsilon} = \beta \mu^N(z^N).$$

This witnesses that $z^N = \frac{z}{\epsilon}$ satisfies equation (81) of [Nesterov et al.](#), which allows us to apply one of their main theorems, which addresses these issues. Supposing that the original problem is solvable, let (x^*, s^*) be any solution to the primal and dual problems, and define the value $\psi := 1 + \langle s_0, x^* \rangle + \langle s^*, x_0 \rangle \geq 1$, which depends only on the problem and the choice of initialization. Then [Theorem 1](#), part 1 of [Nesterov, Todd, and Ye](#), allows us to conclude that

$$\frac{\kappa}{\epsilon} \leq \psi \quad \text{and} \quad \frac{\tau}{\epsilon} \geq \frac{\beta}{\epsilon\psi} \iff \kappa \leq \epsilon\psi \quad \text{and} \quad \tau \geq \frac{\beta}{\psi}.$$

Finally, the original theorem guarantees that $\|G(x)\| \leq \epsilon\|G(z_0)\|$, meaning that

$$\left\| A\left(\frac{x}{\tau}\right) - b \right\|_\tau + \left\| A^\top\left(\frac{y}{\tau}\right) - \frac{s}{\tau} - c \right\|_\tau + \left\| b^\top\left(\frac{y}{\tau}\right) - c^\top\left(\frac{x}{\tau}\right) - \frac{\kappa}{\tau} \right\|_\tau \leq \epsilon\|G(z_0)\|.$$

Since the Euclidean norm is an upper bound on the deviation in any component ($\|v\| := \sqrt{\sum_i v_i^2} \geq \sqrt{\max_i v_i^2} = \max_i v_i =: \|v\|_\infty$), this means that in light of our bound on τ above, we have

$$\left\| A\left(\frac{x}{\tau}\right) - b \right\|_\infty + \left\| A^\top\left(\frac{y}{\tau}\right) + \frac{s}{\tau} - c \right\|_\infty + \left\| b^\top\left(\frac{y}{\tau}\right) - c^\top\left(\frac{x}{\tau}\right) - \frac{\kappa}{\tau} \right\|_\infty \leq \epsilon \frac{\beta\|G(z_0)\|}{\psi}.$$

The first two components show that the total constraint violation (in the primal and dual problems, respectively) is at most $\epsilon\beta/\psi\|G(z_0)\|$. Meanwhile, the final component shows that the duality gap $gap = b^\top\left(\frac{y}{\tau}\right) - c^\top\left(\frac{x}{\tau}\right)$, which is positive and an upper bound on the difference between the objective at x/τ and the optimal objective value, satisfies

$$gap \leq gap + \frac{\kappa}{\tau} \leq \frac{\epsilon\beta\|G(z_0)\|}{\psi}.$$

Thus x/τ is an $(\epsilon\|G(z_0)\|)$ -approximate solution to the original exponential conic problem. Since also $\psi \geq 1$, we may freely drop it to get a looser bound. All that remains is to investigate $\|G(z_0)\|$, the residual norm of the initial point chosen by the MOSEK solver, which equals:

$$\|G(z_0)\| = \|Ax_0 - b\| + \|A^\top y_0 + s_0 - c\| + |c^\top x - b^\top y + 1|.$$

Making use of our assumption that every component of A , b , and c is at most one, we find that

$$\begin{aligned}\|Ax_0 - b\|^2 &= \sum_j (\sum_i A_{j,i}(1.3) - b_j)^2 \leq m(1.3n + 1)^2 \in O(mn^2) \subset O((m+n)^3) \\ \|A^\top y_0 + s_0 - c\|^2 &= \sum_i (\sum_j (A_{j,i})^2) \leq n(m+2)^2 \in O(nm^2) \subset O((m+n)^3) \\ |c^\top x - b^\top y + 1|^2 &\leq (1.3n + m + 1)^2 \in O((n+m)^2) \subset O((n+m)^3).\end{aligned}$$

Therefore, the residual of the initial point is $G(z_0) \in O((n+m)^{3/2})$.

To obtain a solution at most ϵ_0 away from the true solution in any coordinate, we need to select ϵ small enough that the final output of the algorithm z satisfies

$$\frac{\epsilon}{\epsilon} \|G(z_0)\| \leq \epsilon_0 \iff \frac{1}{\epsilon} \geq \frac{1}{\epsilon_0} \|G(z_0)\|$$

It therefore suffices to choose $\frac{1}{\epsilon} \in O(\frac{1}{\epsilon_0}(n+m)^{3/2})$, leading to $\log \frac{1}{\epsilon} = O(\log \frac{n+m}{\epsilon_0})$ iterations.

Thus, we arrive at our total advertised asymptotic complexity of time

$$O\left(n(n+m)^\omega \log \frac{n+m}{\epsilon_0}\right).$$

In particular, to attain machine precision, we can fix ϵ_0 to be the smallest gap between numbers representable (say with 64-bit floats, leading to $\epsilon_0 = 10^{-78}$ in the worst case), and omit the dependance on ϵ_0 for the price of relatively small constant (78, for 64-bit floats). \square



Having combed through all of the details of the analysis of Badenbroek and Dahl [5] and Nesterov et al. [69] for exponential conic programs as we have

defined them, we are ready to show that this algorithm solves the problems presented in [Section 8.4](#) within polynomial time.

In the results that follow, we use the symbol $O_{\text{BP}}(\cdot)$ to describe the complexity under the *bounded precision* assumption: the numerical values of $(\alpha, \beta, \mathbb{P})$ that describe the PDG, as well as γ , lie within a fixed range, e.g., are 64-bit floating point numbers. Correspondingly, we use $\tilde{O}_{\text{BP}}(\cdot)$ to describe the complexity under the same assumption, but hiding logarithmic factors for parameters on which the complexity also depends polynomially.

Lemma 8.A.4. *Problem (8.11) can be solved to ϵ precision in time*

$$O\left((\mathcal{VA} + \mathcal{VC})^{1+\omega} \left(\log \frac{|\mathcal{VA}| + |\mathcal{VC}|}{\epsilon} + \log \frac{\beta^{\max}}{\beta^{\min}} \right)\right) \subset \tilde{O}_{\text{BP}}\left(|\mathcal{VA} + \mathcal{VC}|^4 \log \frac{1}{\epsilon}\right),$$

where $\beta^{\max} := \max_{a \in \mathcal{A}} \beta_a$ is the largest value of β , and $\beta^{\min} := \min_{a \in \mathcal{A}} \{\beta_a : \beta_a > 0\}$ is the smallest positive one.

Proof. Problem (8.11) can be translated via the DCP framework to the following exponential conic program, which has:

- variables $x = (\mathbf{u}, \mathbf{v}, \mathbf{w}, \boldsymbol{\mu}) \in K_{\text{exp}}^{\mathcal{VA}} \times \mathbb{R}_{\geq 0}^{\mathcal{VC}}$, where
 - $\mathbf{u}, \mathbf{v}, \mathbf{w} \in \bar{\mathbb{R}}^{\mathcal{VA}}$ are all vectors over \mathcal{VA} , that at index $\iota = (a, s, t) \in \mathcal{VA}$, have components u_ι, v_ι , and w_ι , respectively;
 - $\boldsymbol{\mu} = [\mu_C(C=c)]_{C \in \mathcal{C}, c \in \mathcal{V}(C)} \in \bar{\mathbb{R}}^{\mathcal{VC}}$ is a vector representation of a tree marginal over clusters \mathcal{C} ;
- constraints as follows:

- two linear constraints for every $(a, s, t) \in \mathcal{VA}$ to ensure that

$$v_{a,s,t} = \mu_{C_a}(s, t) \quad \left(= \sum_{\bar{c} \in \mathcal{V}(C_a \setminus \{S_a, T_a\})} \mu_{C_a}(\bar{c}, s, t) \right)$$

$$\text{and} \quad w_{a,s,t} = \mu_{C_a}(S_a=s) \mathbb{P}_a(T_a=t \mid S_a=s) \quad \left(= \mathbb{P}_a(T_a=t \mid S_a=s) \sum_{\bar{c} \in \mathcal{V}(C_a \setminus \{S_a\})} \mu_{C_a}(\bar{c}, s, t) \right)$$

- for every edge $(C-D) \in \mathcal{T}$, and every value $\omega \in \mathcal{V}(C \cap D)$ of the variables that clusters C and D have in common, a linear constraint

$$\sum_{\bar{c} \in \mathcal{V}(C \setminus D)} \mu_C(\bar{c}, \omega) = \sum_{\bar{d} \in \mathcal{V}(D \setminus C)} \mu_D(\bar{d}, \omega);$$

- and one constraint for each cluster $C \in \mathcal{C}$ to ensure that μ_C lies on the probability simplex, i.e.,

$$\sum_{c \in \mathcal{V}(C)} \mu_C(c) = 1.$$

Altogether this means that we have an exponential conic program in the form of Lemma 8.A.3, with $n = 3|\mathcal{VA}| + |\mathcal{VC}|$ variables, and $m = 2|\mathcal{VA}| + |\mathcal{VT}| + |\mathcal{C}|$ constraints, where $\mathcal{VT} = \{(C-D, \omega) : C-D \in \mathcal{T}, \omega \in \mathcal{V}(C \cap D)\}$. Since we can simply disregard variables whose value sets are singletons, we can assume $\mathcal{V}(C) > 1$; summing over all clusters yields $|\mathcal{VC}| > |\mathcal{C}|$. At the same time, since $|\mathcal{VT}| \leq |\mathcal{VC}|$, we have

$$m, n, (m+n) \in O(\mathcal{VA}, +\mathcal{VC}).$$

We now give the explicit construction of the data (A, b, c) of the exponential conic program that (8.11) compiles to. The variables are indexed by tuples of the form $i = (\ell, a, s, t)$ for $(a, s, t) \in \mathcal{VA}$ and $\ell \in \{u, v, w\}$, or by tuples of the form (C, c) , for $c \in \mathcal{V}(C)$ and $C \in \mathcal{C}$, while the constraints are indexed by tuples of the form $j = (\ell, a, s, t)$ for $(a, s, t) \in \mathcal{VA}$ and $\ell \in \{v, w\}$, of the form $(C-D, \omega)$, for an edge $(C-D) \in \mathcal{T}$ and $\omega \in \mathcal{V}(C \cap D)$, or simply by (C) , the name of a cluster

$C \in \mathcal{C}$. The problem data $A = [A_{j,i}], b = [b_j], c = [c_i]$ of this program are zero, except (possibly) for the components:

$$c_{(u,a,s,t)} = \beta_a$$

$$A_{(v,a,s,t),(C,c)} = \mathbb{1}[C=C_a \wedge S_a(c)=s \wedge T_a(c)=t]$$

$$A_{(w,a,s,t),(C,c)} = \mathbb{P}_a(T_a=t \mid S_a=s) \mathbb{1}[C=C_a \wedge S_a(c)=s]$$

$$A_{(w,a,s,t),(w,a,s,t)} = -1$$

$$A_{(v,a,s,t),(v,a,s,t)} = -1$$

$$A_{(C-D,\omega),(C',c)} = \mathbb{1}[C=C'] - \mathbb{1}[C'=D]$$

$$A_{(C),(C,c)} = 1$$

$$b_{(C)} = 1,$$

where $\mathbb{1}[\varphi]$ is equal to 1 if φ is true, and zero if φ is false. We note that we can equivalently divide each β_a by $\max_a \beta_a$ without affecting the problem, although this could affect the approximation accuracy by the same factor. Thus, we get another factor of

$$\log(\max\{1\} \cup \{\beta_a : a \in \mathcal{A}\}) \subseteq O(\log(1 + \max_a \beta_a)).$$

Finally, to find a point that is ϵ -close (say, in 2-norm) to the limiting point μ^* on the central path, as opposed to simply one that for which the suboptimality gap is at most ϵ , we can appeal to strong concavity of the objective function. (Conditional) relative entropy is 1-strongly convex, and each relative entropy term is scaled by β_a . Furthermore, we're only considering marginal conditional entropy, so this convexity may not hold in all directions. Still, if the next step direction δ is not far from the gradient (as is the case if the interior point method has nearly converged), then, in that direction, the objective will be at least $(\min_a \{\beta_a : \beta_a > 0\})$ -strongly convex. Therefore, by multiplying the requested

precision by an additional factor of $\min_a \{\beta_a : \beta_a > 0\}$, we can guarantee that our point is ϵ -close to μ^* , and not just in complementarity gap.

To summarize, applying Lemma 8.A.3, we find that we can solve problem (8.11) in time

$$O\left((|\mathcal{VA}| + |\mathcal{VC}|)^{1+\omega} \left(\log \frac{|\mathcal{VA}| + |\mathcal{VC}|}{\epsilon} + \log \frac{\beta_{\max}}{\beta_{\min}} \right) \right) \subset \tilde{O}_{BP}\left((|\mathcal{VA}| + |\mathcal{VC}|)^4 \log \frac{1}{\epsilon}\right).$$

The factor of $\log \frac{\beta_{\max}}{\beta_{\min}}$ can be treated as a constant under the bounded precision assumption. \square

We now quickly step through the analogous construction for problems (8.14) and (8.15), which solve the $\hat{\gamma}$ -inference problem, and 0^+ -inference, respectively.

Lemma 8.A.5. *Problem (8.14) is solved to precision ϵ in time*

$$O\left(|\mathcal{VA} + \mathcal{VC}|^{1+\omega} \left(\log \frac{|\mathcal{VA}| + |\mathcal{VC}|}{\epsilon} + \log(1 + \|\boldsymbol{\beta}\|_\infty) + \log \log \frac{1}{p^{\min}} \right) \right) \subset \tilde{O}_{BP}\left(|\mathcal{VA} + \mathcal{VC}|^4 \log \frac{1}{\epsilon}\right)$$

where p^{\min} is the smallest nonzero probability in the PDG.

Proof. Problem (8.14) has

- variables $x = (\mathbf{u}, \mathbf{y}, \mathbf{w}, \mathbf{v}, \boldsymbol{\mu}, \mathbf{z}) \in K_{\exp}^{\mathcal{VA}} \times K_{\exp}^{\mathcal{VC}}$ where
 - $\mathbf{u}, \mathbf{y}, \mathbf{w} \in \bar{\mathbb{R}}^{\mathcal{VA}}$ are all vectors over \mathcal{VA} that at index $\iota = (a, s, t) \in \mathcal{VA}$, have components u_ι, v_ι , and w_ι , respectively;
 - Meanwhile, $\mathbf{v}, \boldsymbol{\mu}, \mathbf{z} \in \bar{\mathbb{R}}^{\mathcal{VC}}$ are all vectors over \mathcal{VC} which at index $(C, c) \in \mathcal{VC}$, have components $v_{C,c}, \mu_C(c)$, and $z_{C,c}$, respectively. Once again, $\boldsymbol{\mu} = [\mu_C(C=c)]_{C \in \mathcal{C}, c \in \mathcal{V}(C)} \in \bar{\mathbb{R}}^{\mathcal{VC}}$ is intended to be a vector representation of a tree marginal.

► constraints as follows:

- two linear constraints for each $(a, s, t) \in \mathcal{VA}$, to ensure that

$$y_{a,s,t} = \mu_{C_a}(s, t) \quad \text{and} \quad w_{a,s,t} = \mu_{C_a}(S_a=s) \mathbb{P}_a(T_a=t \mid S_a=s),$$

- for every edge $(C-D) \in \mathcal{T}$, and every value $\omega \in \mathcal{V}(C \cap D)$ of the variables that clusters C and D have in common, a linear constraint

$$\sum_{\bar{c} \in \mathcal{V}(C \setminus D)} \mu_C(\bar{c}, \omega) = \sum_{\bar{d} \in \mathcal{V}(D \setminus C)} \mu_D(\bar{d}, \omega)$$

- for every $(a, s, t) \in \mathcal{VA}^0$, a linear constraint that ensures

$$0 = \mu_{C_a}(S_a=s, T_a=t) \quad \left(= \sum_{\bar{c} \in \mathcal{V}(C \setminus \{S_a, T_a\})} \mu_{C_a}(\bar{c}, s, t) \right)$$

■

- a linear constraint for every value $c \in \mathcal{V}(C)$ of every cluster $C \in \mathcal{C}$, to ensure that

$$z_{C,c} = \mu_C(VCP_C(c)) \quad \left(= \sum_{\bar{c} \in \mathcal{V}(C \setminus VCP_C)} \mu_C(\bar{c}, VCP_C(c)) \right)$$

- and one constraint for each cluster $C \in \mathcal{C}$ to ensure that μ_C lies on the probability simplex, i.e.,

$$\sum_{c \in \mathcal{V}(C)} \mu_C(c) = 1.$$

So in total, there are $n = |\mathcal{VA}| + |\mathcal{VC}|$ variables, and $m = 2|\mathcal{VA}| + |\mathcal{VT}| + |\mathcal{VA}^0| + |\mathcal{VC}| + |\mathcal{C}|$ constraints. The same arguments made in [Lemma 8.A.4](#) show that both $n, m \in O(|\mathcal{VA}| + |\mathcal{VC}|)$.

Also like before, it is easy to see that the components of A and b are all at most 1. However, we will need to rescale the objective c in order for

each of its components to be most 1. We can do this by dividing it by $\max\{-\beta_a \log p_a(t|s)\}_{(a,s,t) \in \mathcal{VA}} \cup \{1\}$.

Finally, to ensure that we have a solution that is ϵ -close to the end of the central path, as opposed to one that is merely ϵ -close in complementarity gap, we must appeal to convexity. As in the proof of Lemma 8.A.4, this amounts to reducing the target accuracy by a factor of the smallest possible coefficient of strong convexity, along the next step direction. In this case, the bound is simpler: because negative entropy is (unconditionally) 1-strongly convex, and since $\beta \geq \alpha\gamma$, the remaining terms are convex, this could be, at worst, $\frac{1}{\gamma}$.

This gives rise to our result: problem (8.14) can be solved in

$$\begin{aligned} O\left(|\mathcal{VA} + \mathcal{VC}|^{1+\omega} \left(\log \frac{|\mathcal{VA} + \mathcal{VC}|}{\epsilon} + \log \frac{1}{\gamma} \left(1 + \max_{(a,s,t) \in \mathcal{VA}} \beta_a \log \frac{1}{\mathbb{P}_a(t|s)} \right) \right) \right) \\ \subset O\left(|\mathcal{VA} + \mathcal{VC}|^{1+\omega} \left\{ \log \frac{|\mathcal{VA} + \mathcal{VC}|}{\epsilon} + \log \frac{\beta^{\max}}{\gamma} + \log \log \frac{1}{p^{\min}} \right\} \right) \\ \subset \tilde{O}_{\text{BP}}\left(|\mathcal{VA} + \mathcal{VC}|^{1+\omega} \left(\log \frac{1}{\epsilon} \right)\right) \end{aligned}$$

operations, where p is the smallest nonzero probability in the PDG, and β^{\max} is the largest confidence in the PDG larger than 1. \square

Lemma 8.A.6. *Problem (8.15) is solved to precision ϵ in*

$$O\left(|\mathcal{VC}||\mathcal{VA} + \mathcal{VC}|^\omega \log \frac{|\mathcal{VA} + \mathcal{VC}|}{\epsilon}\right) \subset \tilde{O}_{\text{BP}}\left(|\mathcal{VC} + \mathcal{VA}|^4 \log \frac{1}{\epsilon}\right) \text{ time.}$$

Proof. Problem (8.15) is slightly more straightforward; having done Lemmas 8.A.4 and 8.A.5 in depth, we do this one more quickly. In the standard form, problem (8.15), has variables $x = (\mathbf{u}, \boldsymbol{\mu}, \mathbf{w}) \in K_{\text{exp}}^{\mathcal{VC}}$. The constraints are:

- one linear constraint for each $(C, c) \in \mathcal{VC}$, to ensure that

$$w_{C,c} = k_{(C,c)} \mu_C(VCP_C(c)) \quad \left(= \sum_{\bar{c} \in \mathcal{V}(C \setminus VCP_C)} \mu_C(\bar{c}, VCP_C(c)) \right)$$

- for every edge $(C-D) \in \mathcal{T}$, and every value $\omega \in \mathcal{V}(C \cap D)$ of the variables that clusters C and D have in common, a linear constraint

$$\sum_{\bar{c} \in \mathcal{V}(C \setminus D)} \mu_C(\bar{c}, \omega) = \sum_{\bar{d} \in \mathcal{V}(D \setminus C)} \mu_D(\bar{d}, \omega)$$

- for every $(a, s, t) \in \mathcal{VA}$, a linear constraint that ensures

$$\mu_{C_a}(S_a=s, T_a=t) \nu_{C_a}(S_a=s) = \nu_{C_a}(S_a=s, T_a=t) \mu_{C_a}(S_a=s).$$

This is linear, because recall that ν is a constant in this optimization problem, found by having previously solved (8.11).

- and one constraint for each cluster $C \in \mathcal{C}$ to ensure that μ_C lies on the probability simplex.

So in total, there are $n = 3|\mathcal{VC}|$ variables, and $m = |\mathcal{VC}| + |\mathcal{VT}| + |\mathcal{VA}| + |\mathcal{C}|$ constraints. Once again the components of A and b are all at most one, and now the components of the cost function $c = \mathbf{1}$ are identically one. Furthermore, our objective is 1-strongly convex, so no additional multiplicative terms are required to convert an ϵ -close solution in the sense of suboptimality, to an ϵ -close solution in the sense of proximity to the true solution.

Therefore (8.15) can be solved in

$$O\left(|\mathcal{VC}||\mathcal{VA} + \mathcal{VC}|^\omega \log \frac{|\mathcal{VA} + \mathcal{VC}|}{\epsilon}\right) \subset \tilde{O}_{\text{BP}}(|\mathcal{VC} + \mathcal{VA}|^4 \log \frac{1}{\epsilon})$$

operations. □

Theorem 8.4.5. Let $\mathbf{m} = (\mathcal{X}, \mathcal{A}, \mathbb{P}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ be a proper discrete PDG with $N = |\mathcal{X}|$ variables each taking at most V values and $A = |\mathcal{A}|$ arcs, in which each component of $\boldsymbol{\beta} \in \mathbb{R}^{\mathcal{A}}$ and $\mathbb{P} \in \mathbb{R}^{\mathcal{V}\mathcal{A}}$ is specified in binary with at most k bits. Suppose that $\gamma \in \{0^+\} \cup (0, \min_{a \in \mathcal{A}} \frac{\beta_a}{\alpha_a}]$. If $(\mathcal{C}, \mathcal{T})$ is a tree decomposition of $(\mathcal{X}, \mathcal{A})$ of width T and $\boldsymbol{\mu}^* \in \mathbb{R}^{\mathcal{V}\mathcal{C}}$ is the unique calibrated tree marginal over $(\mathcal{C}, \mathcal{T})$ that represents the $\hat{\gamma}$ -semantics of \mathbf{m} , then

- (a) Given \mathbf{m} , γ , and $\epsilon > 0$, we can find a calibrated tree marginal ϵ close in ℓ_2 norm to $\boldsymbol{\mu}^*$ in time

$$\begin{aligned} O\left(|\mathcal{V}\mathcal{A} + \mathcal{V}\mathcal{C}|^4 \left(\log |\mathcal{V}\mathcal{A} + \mathcal{V}\mathcal{C}| + \log \frac{1}{\epsilon}\right) k^2 \log k\right) \\ \subseteq \tilde{O}\left(k^2 |\mathcal{V}\mathcal{A} + \mathcal{V}\mathcal{C}|^4 \log^{1/\epsilon}\right) \\ \subseteq \tilde{O}\left(k^2 (N + A)^4 V^{4(T+1)} \log^{1/\epsilon}\right). \end{aligned}$$

- (b) The unique tree marginal closest to $\boldsymbol{\mu}^*$ in which every component is represented with a k -bit binary number, can be calculated in time¹

$$\tilde{O}\left(k^2 |\mathcal{V}\mathcal{A} + \mathcal{V}\mathcal{C}|^4\right) \subseteq \tilde{O}\left(k^2 (N + A)^4 V^{4(T+1)}\right).$$

Proof. Suppose that the PDG has N variables (each of which can take at most V distinct values), and A hyperarcs, which together form a structure has tree-width T . Then each cluster (of which there are at most N) can have at most $T + 1$ variables, and so can take at most V^T values. Therefore, $|\mathcal{V}\mathcal{C}| \leq NV^{T+1}$. Since each arc must be entirely contained within some cluster, $|\mathcal{V}\mathcal{A}| \leq AV^T$. So, $|\mathcal{V}\mathcal{A} + \mathcal{V}\mathcal{C}| \leq (N + A)V^{T+1}$.

By Lemma 8.A.5, we know that, for $\gamma \in (0, \min_a \frac{\beta_a}{\alpha_a}]$, a tree marginal ϵ -close (in ℓ_2 norm) to the one that represents the unique distribution in the $\hat{\gamma}$ -semantics can

be found in time in time

$$O\left((N+A)^4 V^{4T+4} \log\left(V^{T+1}(N+A) \frac{1}{\epsilon} \frac{\beta^{\max}}{\gamma} + \log \frac{1}{p^{\min}}\right)\right).$$

Similarly, by Lemmas 8.A.4 and 8.A.6 a tree marginal ϵ -close to the one representing the 0^+ semantics can be found in time

$$\begin{aligned} & O\left(|\mathcal{VC} + \mathcal{VA}|^4 \log \frac{|\mathcal{VC} + \mathcal{VA}|}{\epsilon}\right) + O\left(|\mathcal{VC} + \mathcal{VA}|^4 \log \frac{|\mathcal{VC} + \mathcal{VA}| \beta^{\max}}{\epsilon \beta^{\min}}\right) \\ & \subseteq O\left((N+A)^4 V^{4(T+1)} \log\left(V^{T+1}(N+A) \frac{1}{\epsilon} \frac{\beta^{\max}}{\beta^{\min}}\right)\right). \end{aligned}$$

Either way, a tree marginal ϵ -close to the one that represents the $\hat{\gamma}$ -semantics, for $\gamma \in \{0^+\} \cup (0, \min_a \frac{\beta_a}{\alpha_a})$, can be found in

$$O\left(|\mathcal{VC} + \mathcal{VA}|^4 \log\left(\frac{|\mathcal{VC} + \mathcal{VA}| \beta^{\max}}{\epsilon \beta^{\min}} + \log \frac{1}{p^{\min}}\right)\right) \subseteq \tilde{O}_{\text{BP}}\left(|\mathcal{VC} + \mathcal{VA}|^4 \log \frac{1}{\epsilon}\right)$$

arithmetic operations, each of which can be done in $O(k \log k)$ time.

If β, \mathbb{P} , and γ are all binary numbers specified in k bits, then $\log_2 \frac{\beta^{\max}}{\beta^{\min}} \leq 2k$ and $\log \log \frac{1}{p^{\min}} \leq \log k + \log(2)$. Thus, under these assumptions, such a tree marginal can be found in

$$O\left(|\mathcal{VC} + \mathcal{VA}|^4 \left(\log \frac{|\mathcal{VC} + \mathcal{VA}|}{\epsilon} + k + \log k\right) k \log k\right) \subseteq \tilde{O}\left(|\mathcal{VC} + \mathcal{VA}|^4 \log\left(\frac{1}{\epsilon}\right) k\right)$$

time. Finally, we prove part (b). The ∞ -norm is smaller than the ℓ_2 norm, so if $\|\boldsymbol{\mu} - \boldsymbol{\mu}^*\|_2 < 2^{-(k+1)}$, then any change to $\boldsymbol{\mu}$ of size 2^{-k} or larger will cause it to be further from $\boldsymbol{\mu}^*$. Thus, selecting $\epsilon = 2^{-(k+1)}$ produces the tree marginal of k -bit numbers that is closest to $\boldsymbol{\mu}^*$. Plugging in this value of ϵ , we find that finding it takes $\tilde{O}(|\mathcal{VC} + \mathcal{VA}|^4 k^2)$ time. \square

Lemma 8.A.7. *Let $k \geq 1$ be a fixed integer, and $\Phi, K_0, K_1, \dots, K_k$ be parameters. Given a procedure that produces ϵ -approximate unconditional probabilities in $O(\Phi \cdot (K_0 + \sum_{i=1}^k K_i \log^i \frac{1}{\epsilon}))$ time, we can approximate conditional probabilities $\Pr(B|A)$ to within ϵ in $O(\Phi \cdot (K_0 \log \log \frac{1}{\Pr(A)} + \sum_{i=1}^k K_i \log^i \frac{1}{\epsilon \Pr(A)}))$ time.*

Proof. Let f be our algorithm for approximating unconditional probabilities. If A is an event and $\epsilon > 0$, we write $f(A; \epsilon)$ for the corresponding approximation to $\Pr(A)$, which by definition satisfies

$$\Pr(A) - \epsilon \leq f(A; \delta) \leq \Pr(A) + \epsilon.$$

Now suppose that A and B are both events, and we want to find the conditional probability $\Pr(B|A)$. To do so, we can run the following algorithm.

```

1:  $\delta \leftarrow \epsilon;$ 
2: loop
3:   let  $a \leftarrow f(A; \delta);$ 
4:   if  $a > 2\delta$  then
5:     let  $\delta^* \leftarrow \epsilon(a - \delta)/3;$ 
6:     let  $p \leftarrow f(A; \delta^*)$  and  $q \leftarrow f(A \cap B; \delta^*);$ 
7:     return  $q / (p + \delta^*).$ 
8:   else
9:      $\delta \leftarrow \delta^2;$ 

```

Proof of correctness. We claim that the final output of the algorithm is within ϵ of the true conditional probability $\Pr(B|A)$. In the first iteration in which $a > 2\delta$ (line 4), we know that $\delta \leq a - \delta \leq \Pr(A)$.

By assumption,

$$\Pr(A) - \delta^* \leq p \leq \Pr(A) + \delta^* \quad \text{and} \quad \Pr(A \cap B) - \delta^* \leq q \leq \Pr(A \cap B) + \delta^*,$$

from which it follows that

$$\frac{\Pr(A \cap B) - \delta^*}{\Pr(A) + 2\delta^*} \leq \frac{q}{p + \delta^*} \leq \frac{\Pr(A \cap B) + \delta^*}{\Pr(A)}. \quad (8.22)$$

We now extend the bounds on $q/(p + \delta^*)$ in both directions, starting with the upper bound. Because $a - \delta \leq \Pr(A)$, the RHS of (8.22) is at most

$$\frac{\Pr(A \cap B) + \delta^*}{\Pr(A)} = \Pr(B|A) + \frac{\delta^*}{\Pr(A)} = \Pr(B|A) + \frac{\epsilon(a - \delta)}{3\Pr(A)} \leq \Pr(B|A) + \frac{\epsilon\Pr(A)}{3\Pr(A)} < \Pr(B|A) +$$

The analysis of the lower bound (the LHS of (8.22)) is slightly more complicated, but we still find that

$$\begin{aligned} \frac{\Pr(A \cap B) - \delta^*}{\Pr(A) + 2\delta^*} &= \Pr(B|A) - \frac{\mu^*(x, y)}{\Pr(A)} + \frac{\Pr(A \cap B) - \delta^*}{\Pr(A) + 2\delta^*} \\ &= \Pr(B|A) + \frac{-\Pr(A)\Pr(A \cap B) - 2\delta^*\Pr(A \cap B) + \Pr(A)\Pr(A \cap B) - \delta^*\Pr(A)}{\Pr(A)(\Pr(A) + 2\delta^*)} \\ &= \Pr(B|A) + \frac{-2\delta^*\Pr(B|A) - \delta^*}{\Pr(A) + 2\delta^*} \\ &= \Pr(B|A) - \delta^*\left(\frac{2\Pr(B|A) + 1}{\Pr(A) + 2\delta^*}\right) \\ &\geq \Pr(B|A) - \delta^*\frac{3}{\Pr(A) + \delta^*} \quad \left[\text{since } \Pr(B|A) \leq 1, \text{ and thus } -2\Pr(B|A) \geq -2 \right] \\ &\geq \Pr(B|A) - \delta^*\frac{3}{\Pr(A)} \quad \left[\text{as eliminating } \delta^* \text{ makes this more negative} \right] \\ &= \Pr(B|A) - \frac{\epsilon(a - \delta)}{3} \frac{3}{\Pr(A)} \quad \left[\text{by definition of } \delta^* \right] \\ &\geq \Pr(B|A) - \frac{\epsilon\Pr(A)}{\Pr(A)} \quad \left[\text{since } -(a - \delta) \geq -\Pr(A) \right] \\ &= \Pr(B|A) - \epsilon. \end{aligned}$$

These two arguments extend the bounds of (8.22) in both directions. Chaining all of these inequalities together, we have shown that our procedure returns a number output satisfying

$$\Pr(B|A) - \epsilon \leq \text{output} \leq \Pr(B|A) + \epsilon,$$

and hence calculates the desired conditional probability to within ϵ .

Analysis of Runtime. Let m denote the total number of iterations of the algorithm. We deal with the simple case of $m = 1$ separately. If $m = 1$, then already in the first iteration $a > 2\delta = 2\epsilon$, so by definition $\delta^* > \frac{1}{3}\epsilon^3$. Line 6 is just two calls to the procedure, and takes

$$O\left(\Phi\left(K_0 + \sum_{i=1}^k K_i \log^i \frac{1}{\delta^*}\right)\right) = O\left(\Phi\left(K_0 + \sum_{i=1}^k K_i \log^i \frac{3}{\epsilon^3}\right)\right) \subseteq O\left(\Phi\left(K_0 + \sum_{i=1}^k K_i \log^i \frac{1}{\epsilon}\right)\right) \text{ time} \quad (8.23)$$

Now consider the case where $m > 1$. Observe that, in the final iteration, $\delta = \epsilon^{2^{m-1}}$. The procedure halts when $a > 2\delta$, and the smallest possible value of a that our approximation can return is $\Pr(A) - \delta$. Thus, the procedure must halt by the time $\Pr(A) > 3\delta = 3\epsilon^{2^{m-1}}$. On the other hand, since $m - 1$ iterations are not enough to ensure termination, it must be that $\Pr(A) - \delta' \leq 2\delta'$, where $\delta' := \epsilon^{2^{m-2}}$ is the value of δ in the penultimate iteration. Together, these two facts give us a relationship between m and $\Pr(A)$:

$$\begin{aligned} 3\epsilon^{2^{m-2}} &\geq \Pr(A) && > 3\epsilon^{2^{m-1}} \\ \iff -\log_2 3 - 2^{m-2} \log_2 \epsilon &\leq -\log_2 \Pr(A) && < -\log_2 3 - 2^{m-1} \log_2 \epsilon \\ \iff 2^{m-2} &\leq \left(\log_2 \frac{3}{\Pr(A)} \right) / \log_2(1/\epsilon) && < 2^{m-1}. \end{aligned} \quad (8.24)$$

In particular, the first inequality tells us that the number of required iterations is at most

$$m \leq 2 + \log_2 \log_2 \frac{3}{\Pr(A)} - \log_2 \log_2 \frac{1}{\epsilon} = 2 + \log_2 \log_\epsilon \frac{\Pr(A)}{3}.$$

Across all iterations, the total cost of line 3 is on the order of

$$m\Phi K - \Phi \sum_{i=1}^k K_i \sum_{j=1}^m \log^i(\epsilon^{2^{j-1}})$$

$$\begin{aligned}
&= m\Phi K - \Phi \sum_{i=1}^k K_i \log^i(\epsilon) \sum_{j=0}^{m-1} 2^{kj} \\
&= m\Phi K + \Phi \sum_{i=1}^k K_i \log^i \frac{1}{\epsilon} \frac{2^{im} - 1}{2^i - 1} \\
&< \left(\log \log \frac{3}{\Pr(A)} - \log \log \frac{1}{\epsilon} \right) \Phi K_0 + \Phi \sum_{i=1}^k K_i \log^i \left(\frac{1}{\epsilon} \right) \cdot \left[4^i \left(\log^i \frac{3}{\Pr(A)} \right) / \log^i \left(\frac{1}{\epsilon} \right) \right] / (2^i - 1) \\
&\leq \Phi K_0 \log \log \frac{3}{\Pr(A)} + \Phi \sum_{i=1}^k K_i \frac{4^i}{2^i - 1} \log^i \frac{3}{\Pr(A)} \\
&\subseteq O\left(\Phi \cdot \left(K_0 \log \log \frac{1}{\Pr(A)} + \sum_{i=1}^k K_i \log^i \frac{1}{\Pr(A)} \right) \right). \tag{8.25}
\end{aligned}$$

Line 6 is the last part of the procedure that incurs a nontrivial cost. The procedure executes it one time, in the final iteration. Because $a > 2\delta$ at this point, we know that

$$\delta^* = \frac{\epsilon}{3}(a - \delta) > \frac{\epsilon}{3}\delta = \frac{\epsilon}{3}\epsilon^{2^{m-1}} = \frac{\epsilon}{3}\frac{9}{9}\epsilon^{2(2^{m-2})} = \frac{\epsilon}{27}\left(3\epsilon^{2^{m-2}}\right)^2 \geq \frac{\epsilon}{27}\Pr(A)^2.$$

Thus line 6 requires time

$$O\left(\Phi \cdot \left(K_0 + \sum_{i=1}^k K_i \log^i \frac{27}{\Pr(A)^2 \epsilon} \right) \right) \subseteq O\left(\Phi K_0 + \Phi \sum_{i=1}^k K_i \left(\log \frac{1}{\Pr(A)} + \log \frac{1}{\epsilon} \right)^i \right). \tag{8.26}$$

Summarizing, the total running time is (at most) the sum of (8.23), (8.25), and (8.26), or explicitly,

$$O\left(\Phi \cdot \left(K \log \log \frac{1}{\Pr(A)} + \sum_{i=1}^k K_i \log^i \frac{1}{\epsilon \Pr(A)} \right) \right).$$

□

Theorem 9.2.1. *

Proof. Theorem 8.4.5 gives us an approximation to a calibrated tree marginal that represents the distribution of interest, and Lemma 8.A.7 allows us to approximate conditional probabilities once we can approximate unconditional ones. The final ingredient is to approximate unconditional probabilities using an approximate tree marginal.

Concretely, suppose that we are looking to find $\mu^*(X=x)$, where $\mu^* \in [\![\mathcal{M}]\!]_\gamma$. Once we have a calibrated tree marginal μ that represents μ^* , calculating a marginal $\mu^*(X=x)$ (exactly) from μ can be done with standard methods [56, §10.3.3]. In the worst case, it requires taking a marginal of every cluster, which can be done in $O(|\mathcal{VC}|) \subseteq O(NV^{T+1})$ arithmetic operations.

The wrinkle is that μ only *approximately* represents μ^* , in the sense that there is some μ^* that does represent μ^* such that the L2 norm of $\mu^* - \mu$ is small. As usual, we write μ_C for the components of μ that are associated with cluster C . For each $C \in \mathcal{C}$, let E_C denote the event that $(X \cap C) = x|_C$. That is, the variables of X that lie in cluster C take the values prescribed by x . Then

$$|\Pr_{\mu}(X=x) - \Pr_{\mu^*}(X=x)| \leq \sum_{C \in \mathcal{C}} |\Pr_{\mu_C^*}(E_C) - \Pr_{\mu_C}(E_C)| \leq \sum_{C \in \mathcal{C}} \|\mu_C^* - \mu_C\|_1 = \|\mu^* - \mu\|_1.$$

Applying the L2-L1 norm inequality to the vector $\mu - \mu^*$, we find

$$\|\mu - \mu^*\|_1 \leq \|\mu - \mu^*\|_2 \sqrt{|\mathcal{VC}|} \leq \sqrt{NV^{T+1}} \|\mu - \mu^*\|_2.$$

Thus, to answer unconditional queries about X within (absolute) precision ϵ , it suffices to find a tree marginal within $\epsilon/\sqrt{NV^{T+1}}$ of μ^* by L2 norm.

From the proof of [Theorem 8.4.5](#), we know that we can find such a μ in

$$\begin{aligned} O\left((N+A)^4 V^{4(T+1)} \log\left(\frac{(N+A)^4 V^{4(T+1)} \cdot N^{\frac{1}{2}} V^{\frac{T+1}{2}} \beta^{\max}}{\epsilon} + \log \frac{1}{p^{\min}}\right)\right) \\ \subseteq \tilde{O}\left((N+A)^4 V^{4(T+1)} \left(\log \frac{1}{\epsilon} + \log \frac{\beta^{\max}}{\beta^{\min}}\right)\right) \\ \subseteq \tilde{O}_{\text{BP}}\left(|\mathcal{VA} + \mathcal{VC}|^4 \log \frac{1}{\epsilon}\right) \end{aligned}$$

arithmetic operations, which dominates the number of operations required to then find the marginal probability $\Pr_{\mu}(X=x)$ given the tree marginal μ . Thus, the complexity of finding unconditional probabilities is the same. The arithmetic operations need to be done to precision at most $k \in O(\log 1/\epsilon)$, and can be done in time $O(k \log k)$. Thus, unconditional inference can be done in

$$\begin{aligned} O\left((N+A)^{4.5} V^{4.5(T+1)} \log\left(\frac{(N+A)^4 V^{4(T+1)} \cdot N^{\frac{1}{2}} V^{\frac{T+1}{2}} \beta^{\max}}{\epsilon} + \log \frac{1}{p^{\min}}\right) \log \frac{1}{\epsilon} \log \log \frac{1}{\epsilon}\right) \\ \subseteq \tilde{O}\left((N+A)^4 V^{4(T+1)} \left(\log \frac{1}{\epsilon} + \log \frac{\beta^{\max}}{\beta^{\min}}\right) \log \frac{1}{\epsilon}\right) \text{ time.} \end{aligned}$$

Now that we have characterized the cost of unconditional inference, we can apply [Lemma 8.A.7](#) with $\Phi := (N+A)^4 V^{4(T+1)}$, $k = 2$, $K_0 = 0$, $K_1 := \log \Phi + \log \frac{\beta^{\max}}{\beta^{\min}} + \log \log \frac{1}{p^{\min}}$, and $K_2 = 1$ to find that conditional probabilities can be found in

$$\tilde{O}\left((N+A)^4 V^{4(T+1)} \log \frac{1}{\epsilon \mu^*(x)} \left(\log \frac{\beta^{\max}}{\beta^{\min}} + \log \frac{1}{\epsilon \mu^*(x)}\right)\right) \text{ time,}$$

where $\mu^*(x)$ is shorthand for $\mu^*(X=x)$. □

8.B The Convex-Concave Procedure, and Implementation Details

Optimization problems (8.7) and (8.14) can be extended to apply slightly more broadly. There are some cases where there is a unique optimal distribution but γ is large enough that $\beta \not\geq \gamma\alpha$. In these cases, our convex program will fail to satisfy the dcp requirements, and so we cannot compile it to an exponential conic program—but it turns out to still be a useful building block. We now describe how we can still do inference in some of these cases with the convex-concave procedure, or CCCP [97]. This will give us a local minimum of the PDG scoring function $\llbracket m \rrbracket_\gamma$, without requiring us to write this scoring function in a way that proves its convexity, (as is necessary in order to specify a disciplined convex program). At this point, if we happen to know that the problem is convex (or even just pseudo-convex) for other reasons, then finding this distribution suffices for inference. We now describe how this can be done in more detail.

Suppose $\beta_a < \gamma\alpha_a$ some $a \in \mathcal{A}$. In this case $\llbracket m \rrbracket_\gamma$ may not be convex, in general.⁴ However, we do know how to decompose $\llbracket m \rrbracket_\gamma$ into a sum of a convex function $f(\mu)$ and a concave one $g(\mu)$. Concretely: each term on the second line of (8.6) is either convex or concave, depending on the sign of the quantity $\gamma\alpha_a - \beta_a$. Once we sort the terms into convex terms $f(\mu)$ and strictly concave terms $g(\mu)$, the CCCP tells us to repeatedly solve f plus a linear approximation to g . In more detail, the algorithm proceeds as follows. First, choose an initial guess μ_0 , and

⁴Consider the PDG $(\rightarrow X, Y \leftarrow)$ for instance, which has arcs to X and Y , both with $\alpha = 1$ and $\beta = 0$. The minimizers of $\llbracket \rightarrow X, Y \leftarrow \rrbracket_\gamma$ are the distributions that make X and Y independent. It is easily seen that this set is not convex: X and Y are independent if either variable is deterministic, and every distribution is a convex combination of deterministic distributions.

iteratively use the convex solver as in the main paper to compute

$$\mu_{t+1} := \arg \min_{\mu} f(\mu) + (\mu - \mu_t)^T \nabla g(\mu_t).$$

This can be slow because each iteration of the solver is expensive. Still, it is guaranteed to make progress, since

$$\begin{aligned} f(\mu_{t+1}) + g(\mu_{t+1}) &< f(\mu_{t+1}) + (\mu_{t+1} - \mu_t)^T \nabla g(\mu_t) + g(\mu_t) \\ &\leq f(\mu_t) + (\mu_t - \mu_t)^T \nabla g(\mu_t) + g(\mu_t) \\ &= f(\mu_t) + g(\mu_t). \end{aligned}$$

Furthermore, because in our case g is bounded, the process eventually converges to a local minimum of $\llbracket m \rrbracket_\gamma$. This alone, however, is not sufficient for inference, because we may not be able to use this local minimum to answer queries in a way that is true of *all* minimizing distributions. But, if it happens there is a unique local minimum, then the CCCP will find it, leading to an inference procedure.

Notice that if $\beta \geq \gamma \alpha$, then the concave part g is identically zero, and CCCP converges after making just one call to the convex solver. Therefore, in the cases we could already handle, this extension reduces to the algorithm we described before. For this reason, all of our code that handles problems (8.7) and (8.14) is augmented with the CCCP.

Compared to the black-box optimization baselines (Adam and LBFGS), which also only find one minimum, the CCCP still has some advantages. One can see in [Figure 8.C.4](#), for example, that when $\gamma = 2 > 1 = \max_a (\beta_a / \alpha_a)$, CCCP performs better than the baselines. In fact, the CCCP-augmented solver could probably even higher accuracy, if were we not limiting it to a maximum of only five iterations.

8.C Details on the Empirical Evaluation

Imagine a very steep V -shaped canyon, and inside a small slow-moving stream at a gentle incline. The end of the river may be very far away, and the whole landscape may be smooth and strongly convex, but the gradient will still almost always point perpendicular to it, and rather towards the center of the river. This intuition may help explain why, even though $\|\mathbf{m}\|_\gamma$ is infinitely differentiable in μ and γ -strongly convex, it can still be challenging to optimize, especially when the β 's are very different, or when γ is small. For example, a solution to (8.11) finds a minimizer of O_{Inc} , but such minimizers may be very far away from $\|\mathbf{m}\|_{0+}^*$, despite sharing an objective value.

We now see how this is true even when working with very small PDGs and joint distributions.

8.C.1 Synthetic Experiment: Comparison with Black-Box Optimizers, on Joint Distributions.

Here is a more precise description of our first synthetic experiment, on joint distributions, which contrasts the convex optimization approaches of Section 8.3 with black-box optimizers.

- generate 300 PDGs, each of which has the following quantities, to each of which we choose the following natural numbers uniformly at random:
 - $N \in \{5, \dots, 9\}$ of variables (so that $\mathcal{X} := \{1, \dots, N\}$),
 - $V_X \in \{2, 3\}$ values per variable (so that $|\mathcal{V}X| = V_X$ for each $X \in \mathcal{X}$)

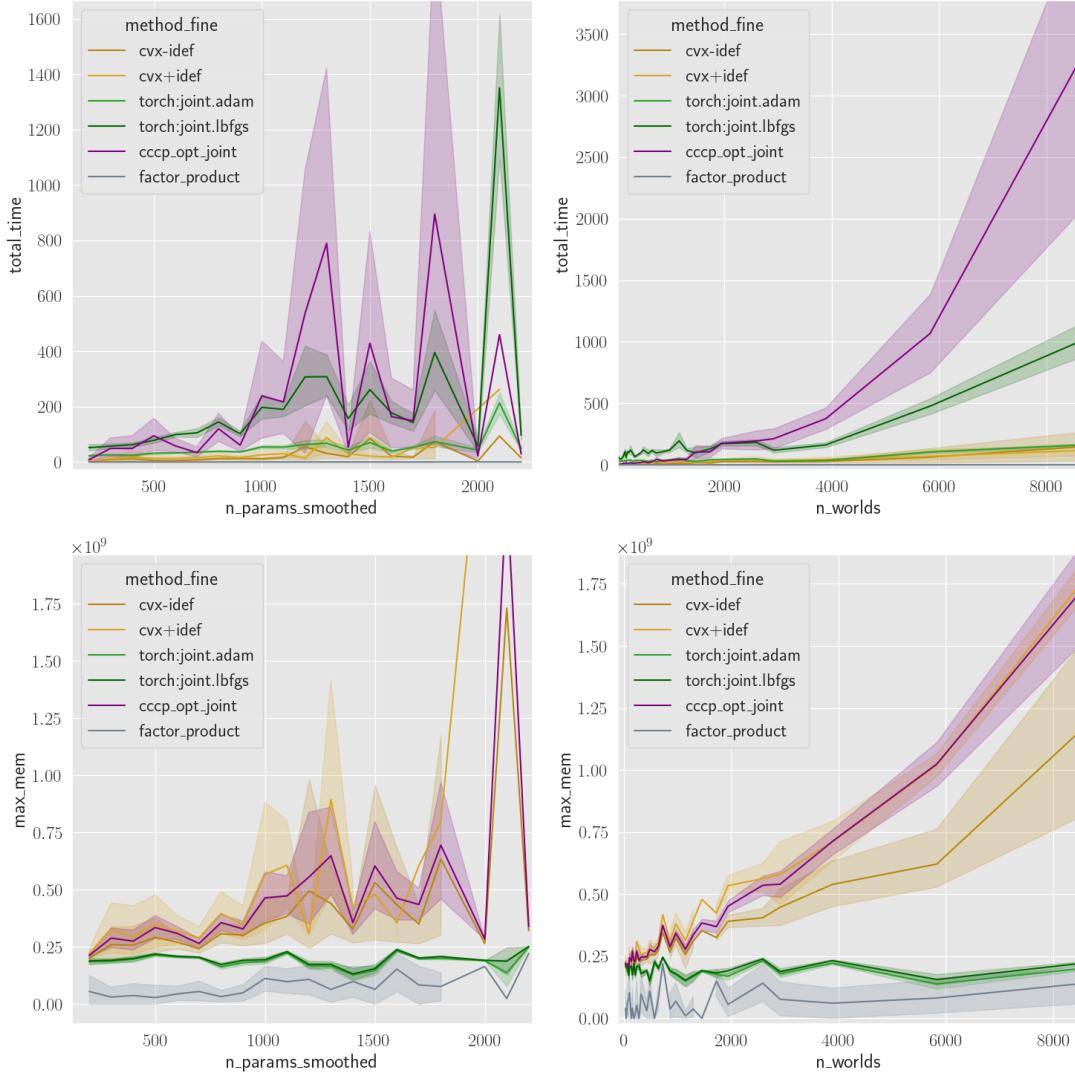


Figure 8.C.1: Resource costs for the joint-distribution optimization setting of Section 8.3. We measure computation time (total_time, top) and maximum memory usage (max_mem, bottom) for the various optimization methods (by color), as the size of the PDG increases, as measured by the number of parameters in the PDG ($n_{\text{params}} = \mathcal{V}\mathcal{A}$, left), and the size of a joint distribution over its variables ($n_{\text{worlds}} = \mathcal{V}\mathcal{X}$, right). Note that the convex solvers for the 0 and 0⁺ semantics are significantly faster than LBFGS, and on par with Adam. However, all three convex-solver based approaches require significantly more memory than the black-box optimizers.

- $A \in \{7, \dots, 14\}$ hyperarcs, each $a \in \{1, \dots, A\} =: \mathcal{A}$ of which has
 - $N_a^S \in \{0, 1, 2, 3\}$ sources, and
 - $N_a^T \in \{1, 2\}$ targets.
- For each arc $a \in \mathcal{A}$, N_a^S of the N variables are chosen without replacement to be sources $S_a \subseteq N$, and N_a^T of remaining variables are chosen to be targets. Finally, to each value of S_a and T_a , a number $p_{a,s,t} \in [0, 1]$ is chosen uniformly at random, and the cpd

$$\mathbb{P}_a(T_a=t \mid S_a=s) = \frac{p_{a,s,t}}{\sum_{t' \in \mathcal{V}(T)} p_{a,s,t'}} \quad \text{is given by normalizing appropriately.}$$

This defines a PDG $\mathcal{M} = (\mathcal{X}, \mathcal{A}, \mathbb{P}, \mathbf{1}, \mathbf{1})$, that has $\alpha = \beta = 1$, which will allow us to compare against belief propagation and other graphical models at $\gamma = 1$. The complexity of this PDG is summarized by two numbers:

- $n_params := \mathcal{V}\mathcal{A}$, the total number of parameters in all cpds of \mathcal{M} , and
- $n_worlds := \mathcal{V}\mathcal{X}$, the dimension of joint distributions over \mathcal{M} 's variables.

- Run MOSEK on (8.5) to find a distribution that minimizes $OInc$; we refer to this method as `cvx-idef`
- Use the result to run MOSEK on (8.10) to find the special distribution $[\mathcal{M}]_{0+}^*$; we refer to this method as `cvx+idef`. These names are due to the fact that $SInc$ is called *IDef* in previous work [82, 81]; thus, this refers to using the convex solver to compute minimizers of $OInc$ with and without considering *IDef*.
- Run the `pytorch` baselines. Let $\theta = [\theta_x]_{x \in \mathcal{V}\mathcal{X}} \in \mathbb{R}^{\mathcal{V}\mathcal{X}}$ be a vector of optimization variables, and choose a representation of the joint distribution, either

by

$$\left(\begin{array}{c} \text{renormalized} \\ \text{simplex} \end{array} \right) \quad \mu_\theta(\mathbf{x}) = \frac{\max\{\theta_{\mathbf{x}}, 0\}}{\sum_{\mathbf{y} \in \mathcal{VX}} \max\{\theta_{\mathbf{y}}, 0\}} \quad \text{or} \quad \mu_\theta(\mathbf{x}) = \frac{\exp(\theta_{\mathbf{x}})}{\sum_{\mathbf{y} \in \mathcal{VX}} \exp(\theta_{\mathbf{y}})} \quad (\text{Gibbs})$$

- For each value of the trade-off parameter $\gamma \in \{0, 10^{-8}, 10^{-4}, 10^{-2}, 1\}$, and each learning rate $lr \in 1E - 3, 1E - 2, 1E - 1, 1E0$, and each optimizer $opt \in \{\text{adam}, \text{L-BFGS}\}$, run opt over the parameters θ to minimize $\llbracket m \rrbracket_\gamma(\mu_\theta)$ until convergence (or a maximum of 1500 iterations)
- We collect the following data about the resulting distribution and the process of computing it:
 - the total time taken to arrive at μ ;
 - the maximum memory taken by the process computing μ ;
 - the objective and its component values:

$$\text{inc} := SInc_m(\mu), \quad \text{idef} := SInc_m(\mu), \quad \text{obj} := OInc_m(\mu) + \gamma SInc_m(\mu) = \llbracket m \rrbracket_\gamma(\mu)$$

The numbers can then be recreated by running our experimental script as follows:

```
python random_expts.py -N 300 -n 5 9 -e 7 14 -v 2 3
--ozrs lbfsg adam
--learning-rates 1E0 1E-1 1E-2 1E-3
--gammas 0 1E-8 1E-4 1E-2 1E0
--num-cores 20
--data-dir random-joint-data
```

which creates a folder called `random-joint-data`, and fills it with `.mpt` files corresponding to each distribution and the method / parameters that gave rise to it.

Analyzing the Results. Look at Figure 8.C.1. Our theoretical analysis, and in particular the proof of Lemma 8.A.4, suggest that the magnitudes of $\mathcal{V}\mathcal{X}$ and $\mathcal{V}\mathcal{A}$ play similar roles in the asymptotic complexity of PDG inference. Our experiments reveal that, at least for random PDGs, the number of worlds is the far more important of the two; observe how much more variation there is on the left side of the figure than the right—and now note that the left side has been smoothed, while the right side has not. The black-box py-torch based approaches clearly have an edge in that they can handle larger models, as evidenced by the cut-offs on the right-hand side of Figure 8.C.8, when with 5GB memory.

Note that the exponential-cone-based methods for the observational limit (gold) are actually faster than L-BFGS (the black-box optimizer with the lowest gap), and also seem to be growing at a slower rate. However, they use significantly more memory, and cannot handle large models. In addition to being faster, our techniques also seem to be more precise; they achieve objective values that are consistently much better than the black-box methods.

Now look at Figure 8.C.3, which contains a break-down of the information in Figure 1. The bottom half of the figure is just the same information, but with each value of γ separated out, so that the special cases of the factor product and 0^+ inference become clear, while the top half shows why it's more important to look

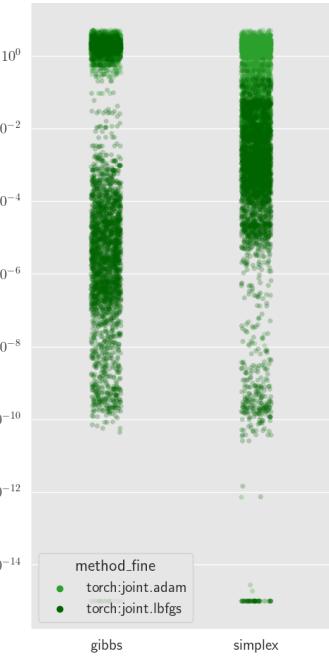


Figure 8.C.2: differences in performance between the Gibbs and simplex parameterizations of probabilities.

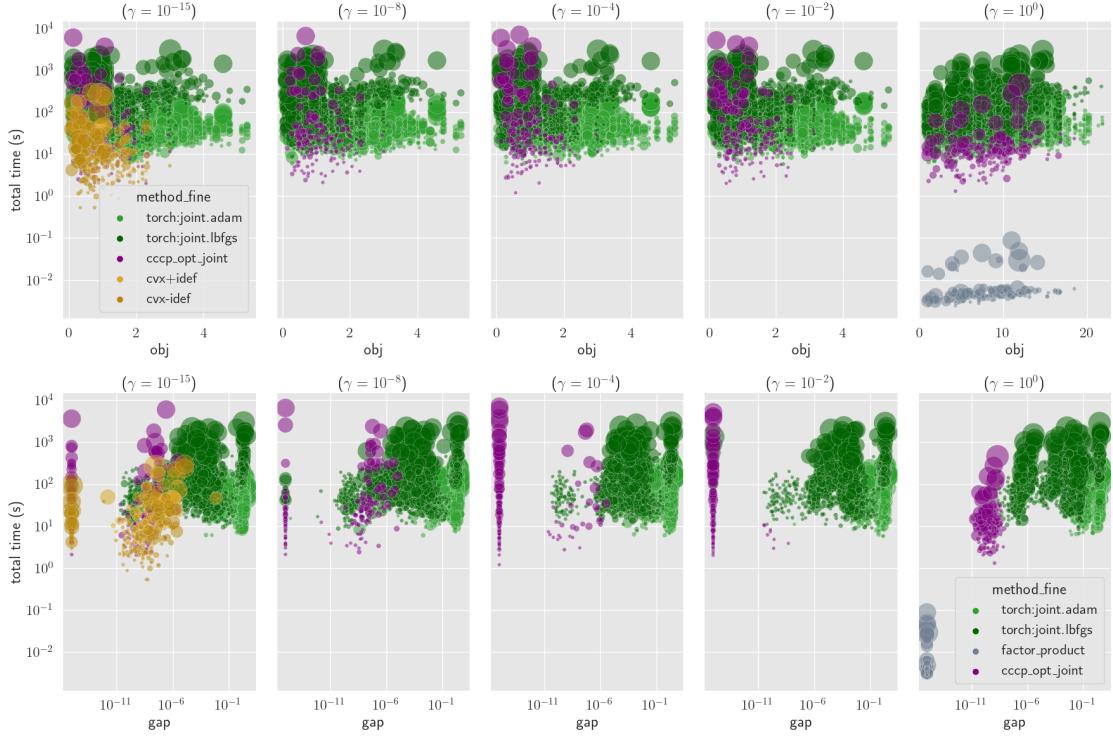


Figure 8.C.3: An un-compressed version of the information in [Figure 1](#), that groups by the value of γ , and also gives the absolute values of the objectives (top row) in addition to the relative gaps (bottom row).

at the gap than the actual objective value for these random PDGs. [Figure 8.C.3](#) also makes it clearer how larger problems take longer, and especially so for `cccp` (violet), which solves the most complex version of the problem (8.7).



8.C.2 Synthetic Experiment: Comparing with Black-Box Optimizers, on Tree Marginals

1. Choose a number of variables $N \in \{8, \dots, 32\}$, and a treewidth $k \in \{1, \dots, 4\}$ uniformly at random. Then draw a random k -tree and corresponding tree of clusters $(\mathcal{C}, \mathcal{T})$, as follows:

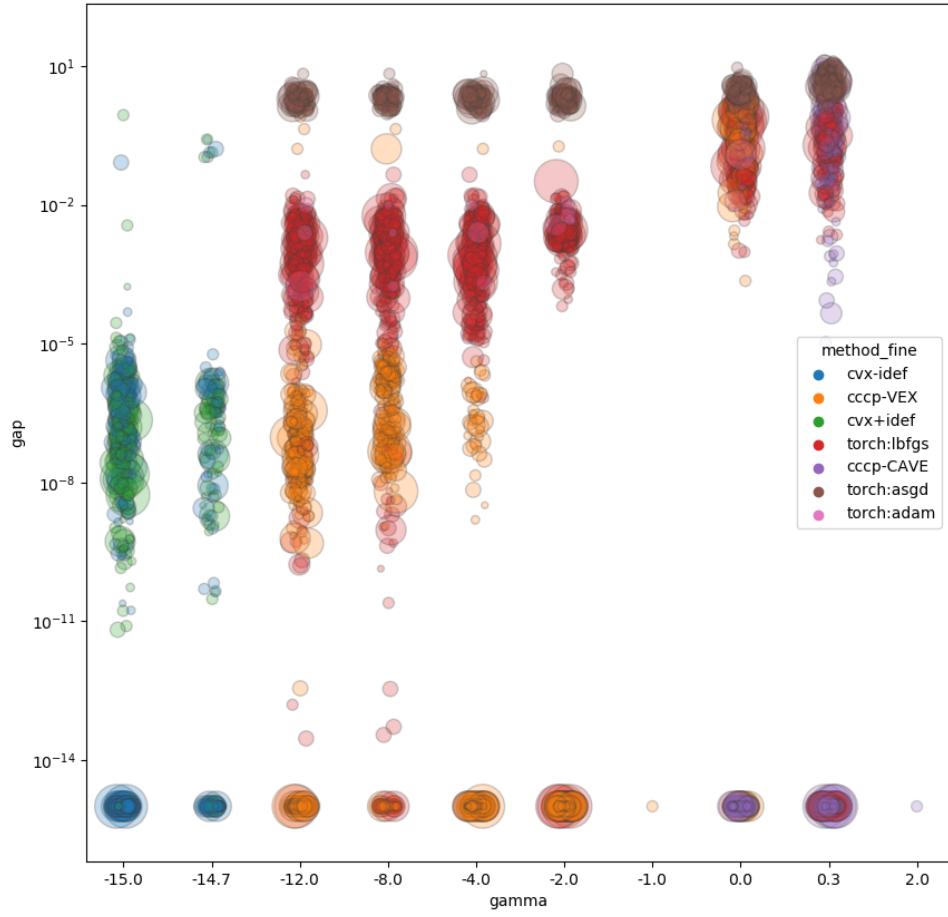


Figure 8.C.4: A graph of the gap (the difference between the attained objective value, and the best objective value obtained across all methods for that value of γ), as γ varies. The x-axis is $\log_{10}(\gamma + 10^{-15})$. As before, colors indicate the optimization method, and the size of the circle illustrates the number of optimization variables (i.e., the number of possible worlds). `cvx-idef` corresponds to just solving (8.5), and `cvx+idef` corresponds to then solving problem (8.10) afterwards. The CCCP runs are split into regimes where the entire problem is convex ($\gamma \leq 1$, labeled `cccp-VEX`), and the entire problem is concave ($\gamma > 1$, labeled `cccp-CAVE`). The optimization approaches `opt_dist` are split into three different optimizers: LBFGS, Adam, and also a third one that performs relatively poorly: accelerated gradient descent. Note that for small γ , the exponential-cone based methods significantly outperform the gradient-based ones.

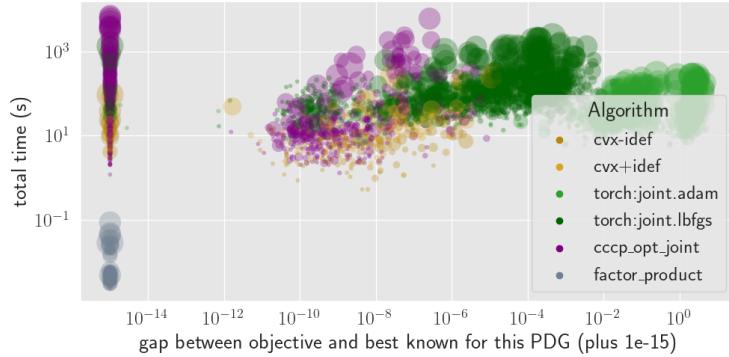


Figure 8.C.5: An analogue of [Figure 1](#), for the cluster setting. Note that there is even more separation between the exponential-cone based approaches, and the black-box optimization based ones. The new grey points on the bottom correspond to belief propagation, which is both faster and typically the most accurate.

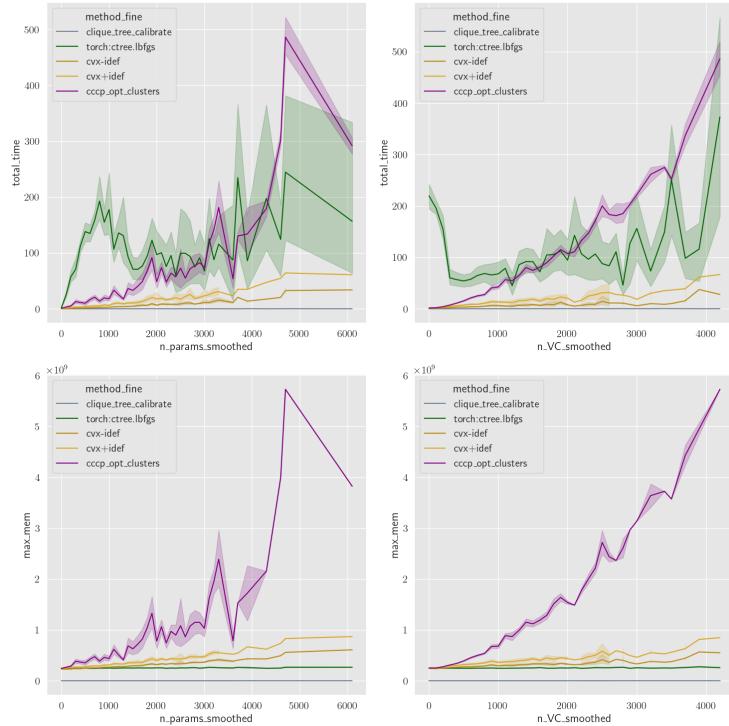


Figure 8.C.6: Resource costs for the cluster setting. Once again, the *O*_{Inc}-optimizing exponential cone methods are in gold, the small-gamma and CCCP is in violet, and the baselines are in green. The bottom line is belief propagation, which is significantly faster and requires very little memory, but also only gives the correct answer under very specific circumstances.

- (a) Initialize $G \leftarrow K_{k+1}$ to a complete graph on $k + 1$ vertices, and $\mathcal{C} \leftarrow \{K_{k+1}\}$ to be set containing a single cluster, and $\mathcal{T} \leftarrow \emptyset$.
- (b) Until there are N vertices: add a new vertex v to G , then randomly select a size k -clique (fully-connected subgraph) $U \subset G$, and add edges between v and every vertex $u \in U$. Add $U \cup \{v\}$ to \mathcal{C} , and add edges to every other cluster $C \in \mathcal{C}$ such that $U \subset C$.
2. Draw the same parameters $V_X \in \{2, 3\}$, $A \in \{8, \dots, 120\}$, $N_a^S \in \{0, 1, 2, 3\}$, and $N_a^T \in \{1, 2\}$ as in [Section 8.C.1](#) uniformly at random. While $N_a^S + N_a^T > k + 1$, for any a , resample N_a^S and N_a^T .
 3. Form a PDG whose structure \mathcal{A} can be decomposed by $(\mathcal{C}, \mathcal{T})$, as follows: for each edge $a \in \mathcal{A}$, sample a cluster $C \in \mathcal{C}$ uniformly at random; then select N_a^S nodes from that cluster without replacement as sources, and N_a^T nodes as targets; this is possible because each cluster has $k + 1$ nodes, and $N_a^S + N_a^T \leq k + 1$ by construction.
 4. Fill in the probabilities by drawing uniform random numbers and re-normalizing, just as before, to form a PDG \mathcal{M}
 5. The black-box optimization baselines work in much the same way also, although now the optimization variables include not one distribution μ but a collection μ_θ of them; this time, we use only the simplex representation of μ_θ . More importantly, we want these clusters to share appropriate marginals; to encourage this, we add a terms to the loss function, so overall, it is

$$\ell(\theta) := [\mathcal{M}]_\gamma(\mu_\theta) + \sum_{C=D \in \mathcal{T}} \exp \left(\sum_{w \in \mathcal{V}(C \cap D)} (\mu_C(C \cap D=w) - \mu_D(C \cap D=w))^2 \right) - 1.$$

This is admittedly pretty ad-hoc; the point is just that it is zero and does not contribute to the gradient if μ_θ is calibrated, and otherwise quickly becomes overwhelmingly important.

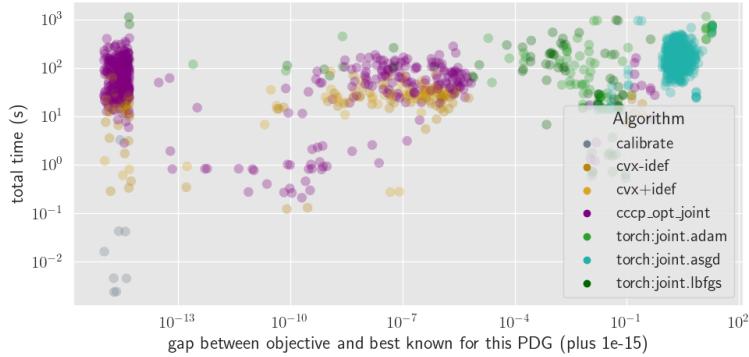


Figure 8.C.7: Gap vs inference time for the small PDGs in the `bnlearn` repository

Analyzing the Results. Observe in [Figure 2](#) that the separation between the tree marginal convex solver and the black-box algorithms is even more distinct. This is because, in this case, the penalty for violating constraints was too small, and the optimization effort was largely wiped out by the calibration before evalution.

This illustrates another general advantage that the convex solver has over black-box optimizers: it is much less brittle and reliant and exactly tuning parameters correctly. Note that even in this minimal example, there were many hyper-parameters that require tuning: the regularization strengths that enforce soft constraints (tree marginal calibration, normalization), as well as learning rate, not to mention various other structural choices: the optimizer, the representation of the distribution, and the maximum number of iterations, none of which are clear-cut choices, but rather require first being tuned to the data. While the convex solver does have internal parameters (tolerances and such) these do not need to be tuned to the problem under normal circumstances.

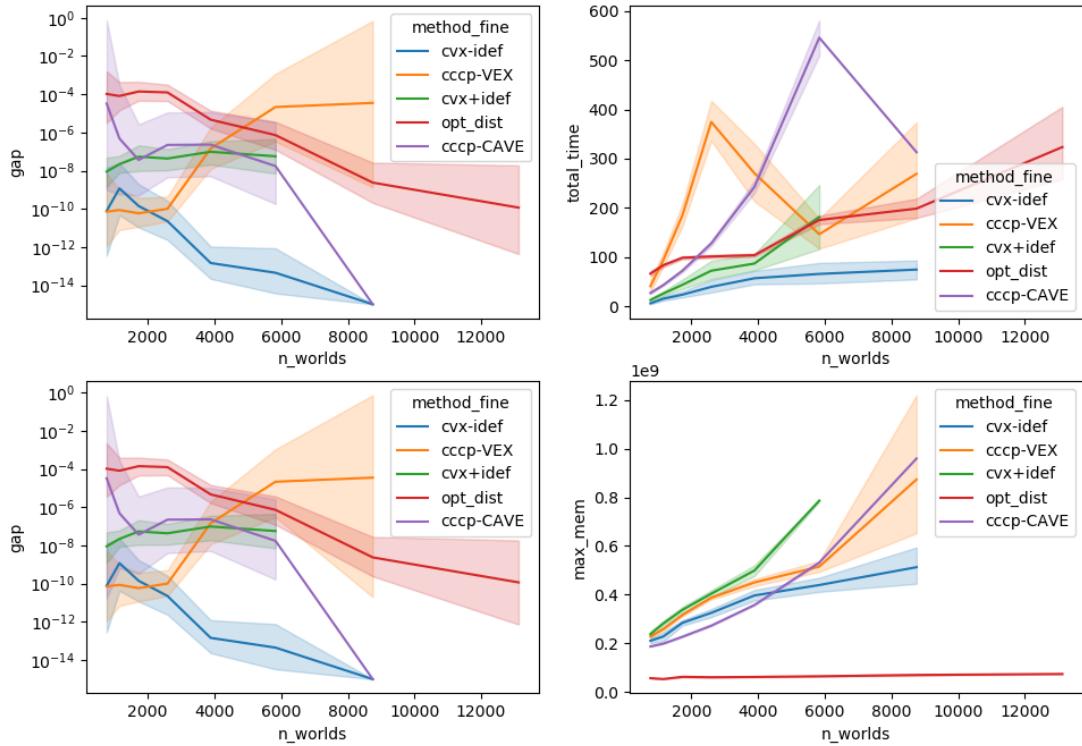


Figure 8.C.8: A variant of [Figure 8.C.1](#), with gap (accuracy) information on the left, and slightly different parameter settings.

8.C.3 Comparing to Belief Propagation, on Tree Marginals.

Since PDGs generalize other graphical models, one might wonder how our method stacks up against algorithms tailored to the more traditional models. In brief: our algorithm is much slower, and only handle much smaller networks. Concretely, our methods can handle all of the “small” networks, and some of the “medium” ones, from the `bnlearn` repository. In these cases, we have verified that the two methods yield the same results. [Figure 8.C.7](#) contains the analogue of [Figures 1](#) and [2](#) for the Bayesian Nets. This graph looks qualitatively quite similar to the other graphs we’ve seen, suggesting that the results in our synthetic experiments hold more broadly for small real-world models as well.

CHAPTER 9
 LOWER BOUNDS, AND THE DEEP CONNECTION BETWEEN
 INCONSISTENCY AND INFERENCE

9.1 A Semantic Connection

9.2 Approximation

While [Theorem 8.4.5](#) gives us a way of doing inference to machine precision in polynomial time, which is the typical use case of an exact algorithm, it is not technically an exact inference algorithm. Indeed, if we require binary representations of numbers, exact inference for PDGs is technically not possible in finite time: in a PDG, the exact answer to an inference query may be an irrational number (even if all components of \mathbb{P}, α , and β are rational). This leads us to formulate approximate inference more precisely.

Definition 9.2.1 (approximate PDG inference). An instance of problem **APPROX-PDG-INFER** is a tuple (m, γ, Q, ϵ) , where m is a PDG with variables \mathcal{X} , $\gamma \in \{0^+\} \cup [0, \infty]$ is the relative importance of structural information, Q is a conditional probability query of the form “ $\Pr(Y=y|X=x) = ?$ ”, where $X, Y \subseteq \mathcal{X}$ and $(x, y) \in \mathcal{V}(X, Y)$, and $\epsilon > 0$ is the precision desired for the answer. A solution to this problem instance is a pair of numbers (r^-, r^+) such that

$$r^- \leq \inf_{\mu \in [m]_\gamma^*} \mu(Y=y|X=x) \leq r^- + \epsilon$$

and $r^+ \geq \sup_{\mu \in [m]_\gamma^*} \mu(Y=y|X=x) \geq r^+ - \epsilon.$ \square

The problem we solved in [Sections 8.3](#) and [8.4](#) is the special case in which m

is assumed to be proper and $\gamma \in \{0^+\} \cup (0, \min_a \frac{\beta_a}{\alpha_a})$. This is enough to ensure there is a unique optimal distribution $\mu^* \in [\![m]\!]_\gamma^*$, with respect to which we must answer all queries. In this case, the definition above essentially amounts to providing a single number p such that $p - \epsilon \leq \mu^*(Y=y|X=x) \leq p + \epsilon$. We call this easier subproblem APPROX-INFER-CVX. We will also be interested in the unconditional variants of both inference problems, in which no additional evidence is supplied (i.e., $X = \emptyset$). We now define the analogous problem of approximately calculating a PDG's degree of inconsistency.

Definition 9.2.2 (approximate inconsistency calculation). An instance of problem APPROX-CALC-INC is a triple (m, γ, ϵ) , where m is a PDG, $\gamma \geq 0$, and $\epsilon > 0$ is the desired precision. A solution to this problem instance is a number r such that $|\langle\langle m \rangle\rangle_\gamma - r| < \epsilon$. □

The interior point method behind [Theorem 8.4.5](#) solves APPROX-CALC-INC in the process of finding a tree marginal for inference. But, technically, it does not solve APPROX-PDG-INFER. A solution to APPROX-PDG-INFER is a conditional probability, not a calibrated tree marginal. While a calibrated tree marginal does allow us to compute conditional probabilities, an ϵ -close tree marginal does not give us ϵ -close answers to probabilistic queries, especially those conditioned on improbable events (i.e., finding $\Pr(Y=y|X=x)$ when $\Pr(X=x) \approx 0$). Nevertheless, because precision is so cheap, the interior point method behind [Theorem 8.4.5](#) can still be used as a subroutine to solve APPROX-INFER-CVX.

Theorem 9.2.1. APPROX-INFER-CVX can be solved in

[link to
proof]

$$\tilde{O}\left((N+A)^4 V^{4(T+1)} \log \frac{1}{\epsilon \mu^*(x)} \left[\log \frac{\beta^{\max}}{\beta^{\min}} + \log \frac{1}{\epsilon \mu^*(x)} \right] \right)$$

time,¹ where $\mu^*(x)$ is the probability of the event $X=x$ in the optimal distribution μ^* ,
 $\beta^{\max} := \max_{a \in \mathcal{A}} \beta_a$ is the largest observational confidence,

$$\text{and } \beta^{\min} := \begin{cases} \min_{a \in \mathcal{A}} \{\beta_a : \beta_a > 0\} & \text{if } \gamma = 0^+ \\ \gamma & \text{if } \gamma > 0 \end{cases} .$$

The factor of $\log(1/\mu^*(x))$ is unusual, but even exact inference algorithms typically must write down $\mu^*(X=x)$ on the way to calculating $\mu^*(Y=x|X=x)$, which implicitly incurs a cost of at least $\log(1/\mu^*(x))$. A Bayesian network with N variables in which cpds are articulated to precision k can have nonzero marginal probabilities as small as 2^{-Nk} , in which case the additional worst case overhead for small probabilities is linear. We conjecture that it is not possible to form smaller marginal probabilities with PDGs, although the question remains open. Algorithmically speaking, [Theorem 9.2.1](#) extends [Theorem 8.4.5](#) in three key ways.

1. We must request additional precision to ensure that the marginal probabilities deviate at most ϵ from the true ones. This sense of approximation effectively bounds the ℓ_1 norm of $\mu^* - \mu$, while [Theorem 8.4.5](#) (a) bounds its ℓ_2 norm and (b) bounds its ℓ_∞ norm.
2. We must introduce a loop to refine precision until we have a suitably precise estimate of $\Pr(X=x)$.
3. Rather than directly dividing our estimate of $\Pr(Y=y, X=x)$ by our estimate of $\Pr(X=x)$, we calculate something slightly more stable.

¹At the cost of substantial overhead and engineering effort, the exponent 4 can be reduced to 2.872, by appeal to Skajaa and Ye [88] and the current best matrix multiplication algorithm [25, $O(n^{2.372})$] to invert $n \times n$ linear systems.

See the proof for details. One immediate corrolary of [Theorem 9.2.1](#) is that $\text{APPROX-INFER-CVX} \subseteq \text{EXP}$, without the assumption of bounded treewidth.

It may be worth noting that there is at least one instance in the literature where *approximate* Bayesian Network (BN) inference is tractable for a subclass of models other than those of bounded treewidth: [? \]](#) give a randomized algorithm for the special class of Bayesian Networks that do not have extreme conditional probabilities. Specifically they show that, assuming a network with N nodes, the inference problem is in $\text{RP}(N, 1/\epsilon)$. In addition to restricting to a different class of models (bounded conditional probabilities, but not bounded treewidth), their approximation algorithm in another significant respect: it is polynomial in $1/\epsilon$, rather than in $\log(1/\epsilon)$. Thus, the time it requires is exponential with respect to the number of requested digits, while our algorithm takes linear time.

Because PDGs generalize BNs, approximate inference for PDGs is at least as hard as it is for BNs.

Proposition 9.2.2 ([\[? \]](#)). *APPROX-PDG-INFER is #P-hard.*

Thus, the exponential time of [Theorem 9.2.1](#) is the best we could have hoped for, in the general case. The argument is due to [? \]](#), although we have altered it somewhat.

9.3 An Algorithmic Connection

Our approach to $\hat{\gamma}$ -inference computes $\langle\!\langle m \rangle\!\rangle_\gamma$ as a side effect. But suppose that we were interested in calculating only this inconsistency. Might there be a more

direct, asymptotically easier way to do so? In general, the answer is no.

Theorem 9.3.1. (a) Determining whether there is a distribution that satisfies all cpds of a PDG is NP-hard.

[link to proof]

(b) Calculating a PDG's degree of inconsistency (exactly) is #P hard.

(c) APPROX-CALC-INC is #P hard, even for fixed $\gamma \geq 0$ and $\epsilon > 0$.

Richardson and Halpern [82]'s original approach to inferring the probability of Y in a PDG m was to minimize their combined inconsistency. The idea is to add a hypothesis distribution $h(Y)$ to m , and adjust h to minimize the overall inconsistency $\langle\!\langle m + h \rangle\!\rangle_\gamma$. Parts (b) and (c) of Theorem 9.3.1 significantly undermine this approach, because even just calculating $\langle\!\langle m + h \rangle\!\rangle_\gamma$ is intractable. Typically minimizing a function is more difficult than evaluating it, so one might imagine the intractability of $\langle\!\langle m + h \rangle\!\rangle_\gamma$ to be merely the first of many difficulties—yet it turns out to be the only one. There is a strong sense in which being able to calculate inconsistency is enough to perform inference efficiently. Specifically, with oracle access to the inconsistency $\langle\!\langle m + h \rangle\!\rangle_\gamma$, Richardson and Halpern's approach gives right answer with the best possible asymptotic time complexity. Thus, while it may not be a practical inference algorithm, it is a powerful reduction from inference to inconsistency calculation.

Theorem 9.3.2. (a) There is an $O(\log^{1/\epsilon})$ -time reduction from unconditional APPROX-INFER-CVX to the problem of determining which of two PDGs is more inconsistent, using $O(\log^{1/\epsilon})$ subroutine calls.

[link to proof]

(b) There is an $O\left(\log \frac{\langle\!\langle m \rangle\!\rangle_\gamma}{\gamma \epsilon \mu^*(x)} \cdot \log \frac{1}{\epsilon \mu^*(x)}\right)$ time reduction from APPROX-INFER-CVX to APPROX-CALC-INC using $O(\log(1/\epsilon) \log \log^{1/\mu^*(x)})$ calls to the inconsistency subroutine.

(c) *There is also an $O(|\mathcal{VC}|)$ reduction from APPROX-CALC-INC to APPROX-INFER-CVX. With the additional assumption of bounded treewidth, this is linear in the number of variables in the PDG.*

Recall that the runtime of $O(\log(1/\epsilon))$ achieved by part (a) is optimal, because it is the complexity of writing down an answer, which in general requires $\log(1/\epsilon)$ bits. While it is a clean result, part (a) is unsatisfying as a complexity result because it relies heavily on being able to compare the two inconsistencies in constant time. Part (b) fleshes out the algorithm of part (a) more precisely by reducing to inconsistency approximation (which we now know is computable), and also extends the procedure to handle to conditional probability queries. This leads to a significantly more complex analysis, and a more expensive reuction, although it is possible that much of the difference in the costs is due to loose bounds in our analysis. Part (c) is a straightforward observation in light of the results in [Section 8.4](#).

To summarize: in the range of γ 's in which we have an (approximate) inference algorithm for PDGs, (approximately) calculating a PDG's degree of inconsistency is at least as difficult. For PDGs of bounded treewidth, the two problems are equivalent, and can be solved in polynomial time.

9.4 The Reductions

We now turn to [Theorem 9.3.1](#). We begin by proving parts (a) and (b) directly by reduction to SAT and #SAT, respectively.

Theorem 9.3.1.

- (a) Determining whether there is a distribution that satisfies all cpds of a PDG is NP-hard.
- (b) Calculating a PDG's degree of inconsistency (exactly) is #P hard.
- (c) APPROX-CALC-INC is #P hard, even for fixed $\gamma \geq 0$ and $\epsilon > 0$.

Proof. (a). We can directly encode SAT problems in PDGs. Choose any CNF formula

$$\varphi = \bigwedge_{j \in \mathcal{J}} \bigvee_{i \in \mathcal{I}(j)} (X_{j,i})$$

over binary variables $\mathbf{X} := \bigcup_{j,i} X_{j,i}$, and let $n := |\mathbf{X}|$ denote the total number of variables in φ . Let \mathcal{M}_φ be the PDG containing every variable $X \in \mathbf{X}$ and a binary variable C_j (taking the value 0 or 1) for each clause $j \in \mathcal{J}$, as well as the following edges, for each $j \in \mathcal{J}$:

- a hyperedge $\{X_{j,i} : i \in \mathcal{I}(j)\} \rightarrowtail C_j$, together with a degenerate cpd encoding the boolean OR function (i.e., the truth of C_j given $\{X_{j,i}\}$);
- an edge $\mathbb{1} \rightarrowtail C_j$, together with a cpd asserting C_j be equal to 1.

First, note that the number of nodes, edges, and non-zero entries in the cpds are polynomial in the $|\mathcal{J}|, |\mathbf{X}|$, and the total number of parameters in a simple matrix representation of the cpds is also polynomial if \mathcal{I} is bounded (e.g., if φ is a 3-CNF formula). A satisfying assignment $\mathbf{x} \models \varphi$ of the variables \mathbf{X} can be regarded as a degenerate joint distribution $\delta_{\mathbf{x}=\mathbf{x}}$ on \mathbf{X} , and extends uniquely to a full joint distribution $\mu_{\mathbf{x}} \in \Delta \mathcal{V}(\mathcal{M}_\varphi)$ consistent with all of the edges, by

$$\mu_{\mathbf{x}} = \delta_{\mathbf{x}} \otimes \delta_{\{C_j = \vee_i x_{j,i}\}}$$

Conversely, if μ is a joint distribution consistent with the edges above, then any point \mathbf{x} in the support of $\mu(\mathbf{X})$ must be a satisfying assignment, since the two classes of edges respectively ensure that $1 = \mu(C_j=1 \mid \mathbf{X}=\mathbf{x}) = \bigvee_{i \in \mathcal{I}(j)} \mathbf{x}_{j,i}$ for all $j \in \mathcal{J}$, and so $\mathbf{x} \models \varphi$.

Thus, $\{\langle\!\langle \mathbf{m}_\varphi \rangle\!\rangle\} \neq \emptyset$ if and only if φ is satisfiable, so an algorithm for determining if a PDG is consistent can also be adapted (in polynomial space and time) for use as a SAT solver, and so the problem of determining if a PDG consistent is NP-hard.

(b) Hardness of exact computation. We prove this by reduction to #SAT. Again, let φ be some CNF formula over \mathbf{X} , and construct \mathbf{m}_φ as in [the proof](#) of [Theorem 9.3.1](#). Furthermore, let $\llbracket \varphi \rrbracket := \{\mathbf{x} : \mathbf{x} \models \varphi\}$ be the set of assignments to \mathbf{X} satisfying φ , and $\#\varphi := |\llbracket \varphi \rrbracket|$ denote the number such assignments. We now claim that

$$\#\varphi = \exp \left[-\frac{1}{\gamma} \langle\!\langle \mathbf{m}_\varphi \rangle\!\rangle_\gamma \right]. \quad (9.1)$$

Once we do so, we will have reduced the #P-hard problem of computing $\#\varphi$ to the problem of computing $\langle\!\langle \mathbf{m} \rangle\!\rangle_\gamma$ (exactly).

We now prove (9.1). By definition, we have

$$\langle\!\langle \mathbf{m}_\varphi \rangle\!\rangle_\gamma = \inf_{\mu} \left[OInc_{\mathbf{m}_\varphi}(\mu) + \gamma SInc_{\mathbf{m}_\varphi}(\mu) \right].$$

We start with a claim about first term.

Claim 9.4.0.1. $OInc_{\mathbf{m}_\varphi}(\mu) = \begin{cases} 0 & \text{if } \text{Supp } \mu \subseteq \llbracket \varphi \rrbracket \times \{\mathbf{1}\} \\ \infty & \text{otherwise.} \end{cases}$

Proof. Writing out the definition explicitly, the first can be written as

$$OInc_{\mathbf{m}_\varphi}(\mu) = \sum_j \left[D\left(\mu(C_j) \parallel \delta_1\right) + \mathbb{E}_{\mathbf{x} \sim \mu(\mathbf{X}_j)} D\left(\mu(C_j \mid \mathbf{X}_j = \mathbf{x}) \parallel \delta_{\vee_i \mathbf{x}_{j,i}}\right) \right], \quad (9.2)$$

where $\mathbf{X}_j = \{X_{ij} : j \in \mathcal{I}(j)\}$ is the set of variables that appear in clause j , and $\delta_{(-)}$ is the probability distribution placing all mass on the point indicated by its subscript. As a reminder, the relative entropy is given by

$$D(\mu(\Omega) \parallel \nu(\Omega)) := \mathbb{E}_{\omega \sim \mu} \log \frac{\mu(\omega)}{\nu(\omega)}, \quad \begin{array}{l} \text{and in particular,} \\ \text{if } \Omega \text{ is binary,} \end{array} \quad D(\mu(\Omega) \parallel \delta_\omega) = \begin{cases} 0 & \text{if } \mu(\omega) = 1; \\ \infty & \text{otherwise.} \end{cases}$$

Applying this to (9.2), we find that either:

1. Every term of (9.2) is finite (and zero) so $OInc_{m_\varphi}(\mu) = 0$, which happens when $\mu(C_j = 1) = 1$ and $\mu(C_j = \vee_i x_{j,i}) = 1$ for all j . In this case, $\mathbf{c} = \mathbf{1} = \{\vee_i x_{j,i}\}_j$ so $\mathbf{x} \models \varphi$ for every $(\mathbf{c}, x) \in \text{Supp } \mu$;
2. Some term of (9.2) is infinite, so that $OInc_{m_\varphi}(\mu) = \infty$, which happens if some j , either
 - (a) $\mu(C_j \neq 1) > 0$ — in which case there is some $(\mathbf{x}, c) \in \text{Supp } \mu$ with $\mathbf{c} \neq \mathbf{1}$, or
 - (b) $\text{Supp } \mu(\mathbf{C}) = \{\mathbf{1}\}$, but $\mu(C_j \neq \vee_i x_{j,i}) > 0$ — in which case there is some $(\mathbf{x}, 1) \in \text{Supp } \mu$ for which $1 = c_j \neq \vee_i x_{j,i}$, and so $\mathbf{x} \not\models \varphi$.

Condensing and rearranging slightly, we have shown that

$$OInc_{m_\varphi}(\mu) = \begin{cases} 0 & \text{if } \mathbf{x} \models \varphi \text{ and } \mathbf{c} = \mathbf{1} \text{ for all } (\mathbf{x}, \mathbf{c}) \in \text{Supp } \mu \\ \infty & \text{otherwise} \end{cases}.$$

□

Because $SInc$ is bounded, it follows immediately that $\langle\!\langle m_\varphi \rangle\!\rangle_\gamma$ is finite if and only if there is some distribution $\mu \in \Delta\mathcal{V}(\mathbf{X}, \mathbf{C})$ for which $OInc_{m_\varphi}(\mu)$ is finite,

or equivalently, by ?? 9.4.0.1, iff there exists some $\mu(\mathbf{X}) \in \Delta\mathcal{V}(\mathbf{X})$ for which $\text{Supp } \mu(\mathbf{X}) \subseteq \llbracket \varphi \rrbracket$, which in turn is true if and only if φ is satisfiable.

In particular, if φ is not satisfiable (i.e., $\#\varphi = 0$), then $\langle\!\langle m_\varphi \rangle\!\rangle_\gamma = +\infty$, and

$$\exp\left[-\frac{1}{\gamma}\langle\!\langle m_\varphi \rangle\!\rangle_\gamma\right] = \exp[-\infty] = 0 = \#\varphi,$$

so in this case (9.1) holds as promised. On the other hand, if φ is satisfiable, then, again by ?? 9.4.0.1, every μ minimizing $\llbracket m_\varphi \rrbracket_\gamma$, (i.e., every $\mu \in \llbracket m_\varphi \rrbracket_\gamma^*$) must be supported entirely on $\llbracket \varphi \rrbracket$ and have $OInc_{m_\varphi}(\mu) = 0$. As a result, we have

$$\langle\!\langle m_\varphi \rangle\!\rangle_\gamma = \inf_{\mu \in \Delta[\llbracket \varphi \rrbracket \times \{\mathbf{1}\}]} \gamma SInc_{m_\varphi}(\mu).$$

A priori, by the definition of $SInc_{m_\varphi}$, we have

$$SInc_{m_\varphi}(\mu) = -H(\mu) + \sum_j \left[\alpha_{j,1} H_\mu(C_j \mid \mathbf{X}_j) + \alpha_{j,0} H_\mu(C_j) \right],$$

where $\alpha_{j,0}$ and $\alpha_{j,1}$ are values of α for the edges of m_φ , which we have not specified because they are rendered irrelevant by the fact that their corresponding cpds are deterministic. We now show how this plays out in the present case. Any $\mu \in \Delta[\llbracket \varphi \rrbracket \times \{\mathbf{1}\}]$ we consider has a degenerate marginal on \mathbf{C} . Specifically, for every j , we have $\mu(C_j) = \delta_1$, and since entropy is non-negative and never increased by conditioning,

$$0 \leq H_\mu(C_j \mid \mathbf{X}_j) \leq H_\mu(C_j) = 0.$$

Therefore, $SInc_{m_\varphi}(\mu)$ reduces to the negative entropy of μ . Finally, making use of the fact that the maximum entropy distribution μ^* supported on a finite set S

is the uniform distribution on S , and has $H(\mu^*) = \log |S|$, we have

$$\begin{aligned}\langle\!\langle m_\varphi \rangle\!\rangle_\gamma &= \inf_{\mu \in \Delta([\varphi] \times \{\mathbf{1}\})} \gamma SInc_{m_\varphi}(\mu) \\ &= \inf_{\mu \in \Delta([\varphi] \times \{\mathbf{1}\})} -\gamma H(\mu) \\ &= -\gamma \sup_{\mu \in \Delta([\varphi] \times \{\mathbf{1}\})} H(\mu) \\ &= -\gamma \log(\#\varphi),\end{aligned}$$

giving us

$$\#\varphi = \exp\left[-\frac{1}{\gamma} \langle\!\langle m_\varphi \rangle\!\rangle_\gamma\right],$$

as desired. We have now reduced #SAT to computing $\langle\!\langle m \rangle\!\rangle_\gamma$, for $\gamma > 0$ and an arbitrary PDG m , which is therefore #P-hard.

To show the same for $\gamma = 0$, it suffices to add an additional hyperedge pointing to all variables, and associate it with a joint uniform distribution, and confidence 1, resulting in a new PDG m'_φ . Because this new edge's contribution to $OInc_m$ equals $D(\mu \parallel \text{Unif}(\mathcal{X})) = \log |\mathcal{V}\mathcal{X}| - H(\mu)$, we have

$$[\![m'_\varphi]\!]_0(\mu) = OInc_{m'_\varphi}(\mu) = [\![m_\varphi]\!](\mu) + \log |\mathcal{V}\mathcal{X}| - H(\mu) = [\![m_\varphi]\!]_1(\mu) + \log |\mathcal{V}\mathcal{X}|.$$

Since this is true for all μ , we can take the of this equation over μ , and so conclude that

$$\begin{aligned}\langle\!\langle m'_\varphi \rangle\!\rangle_0 &= \langle\!\langle m_\varphi \rangle\!\rangle_1 + \log |\mathcal{V}\mathcal{X}| = \log(|\mathcal{V}\mathcal{X}|/\#\varphi) \\ \implies \#\varphi &= |\mathcal{V}\mathcal{X}| \exp(-\langle\!\langle m'_\varphi \rangle\!\rangle_0)\end{aligned}$$

Thus, the number of satisfying assignments can be found through via an oracle for $\langle\!\langle - \rangle\!\rangle_0$, as well. This shows that calculating this purely observational inconsistency is #P-hard as well.

(c) Hardness of approximation. To calculate $\#\varphi$ exactly, it turns out that we do not need to know $\langle\!\langle m_\varphi \rangle\!\rangle_\gamma$ exactly. Instead, we claim it suffices to approximate it to within $\epsilon < \gamma \log(1 + 2^{-(n+1)})$.

Suppose that $|r - \langle\!\langle m_\varphi \rangle\!\rangle_\gamma| < \epsilon$. Then

$$\begin{aligned} \exp\left(-\frac{r}{\gamma}\right) &\in \exp\left[-\frac{1}{\gamma}\left(\langle\!\langle m_\varphi \rangle\!\rangle_\gamma \pm \epsilon\right)\right] \\ &= \exp\left[-\frac{1}{\gamma}\langle\!\langle m_\varphi \rangle\!\rangle_\gamma\right] \cdot \exp(\pm\epsilon/\gamma) \\ &= \#\varphi \cdot \exp(\pm\epsilon/\gamma) \\ &= [\#\varphi \exp(-\epsilon/\gamma), \#\varphi \exp(+\epsilon/\gamma)]. \end{aligned}$$

Since $\#\varphi$ is a natural number and at most 2^n , If we can get a relative approximation of it to within a factor of $2^{-(n+1)}$, then rounding that approximate value to the nearest whole number gives the exact value of $\#\varphi$. Thus, it suffices to choose ϵ small enough that

$$\exp(-\epsilon/\gamma) > 1 - 2^{-(n+1)} \quad \text{and} \quad \exp(+\epsilon/\gamma) < 1 + 2^{-(n+1)};$$

this is satisfied any choice of $\epsilon < \gamma \log(1 + 2^{-(n+1)})$. Thus, being able to approximate $\langle\!\langle m_\varphi \rangle\!\rangle_\gamma$ sufficiently closely will tell us whether or not $\varphi \in \text{SAT}$. Note that for large n , the maximum value of ϵ for which this is true is on the order of $\epsilon_{\max} \in \Theta(\gamma 2^{-n})$. It follows that $\log(1/\epsilon_{\max}) \in O(n)$, and so values of ϵ small enough to determine the satisfiability of a formula φ with n variables can be specified in time $O(n)$. Thus, the problem APPROX-CALC-INC is #P hard (in the size of its input). \square

Inference via Inconsistency Minimization. We now address [Theorem 9.3.2](#), which is closely related to Richardson and Halpern [82]'s original idea for an

inference algorithm. While that idea does not yield an efficient inference algorithm, it does yield an efficient reduction from inconsistency minimization to inference. In order to prove this, we first need another construction with PDGs. A probability over a (set of) variables can be viewed as a vector whose elements sum to one. It turns out that it is possible to use the machinery of PDGs to, effectively, give only one value of such a probability vector. That is, for any $p \in [0, 1]$, we can construct a PDG that represents the belief that $\Pr(Y=y) = p$, but say nothing about how the probability splits between other values of y . We now describe that construction.

We first introduce an auxiliary binary variable Y_y , with $\mathcal{V}(Y_y) = \{y, \neg y\}$, and takes the value y if $Y = y$, and $\neg y$ if $Y \neq y$. Note that this variable is a function of the value of variable Y (although we will need to enforce this with an additional arc), and therefore there is a unique way to extend a distribution over variables including Y to also include the variable Y_y .

With this definition, there is now an obvious way to add a hyperarc with no source and target Y_y , together with a asserting that $\Pr(Y=y) = p$. This cpd is written as a vector \hat{p} on the right of the figure below. The PDG we have just constructed is illustrated on the left of the figure below. In addition to \hat{p} and the new variable, this PDG includes the structural constraint s needed to define the variables Y_y in terms of Y ; it is a deterministic function, drawn with a double-headed gray arrow.

$$\begin{array}{ccc}
\hat{p} \rightarrow & \boxed{\begin{array}{c} Y_y \\ \bullet \\ y \quad \neg y \end{array}} & s(Y_y|Y) := \begin{cases} y & \text{if } Y = y \\ \neg y & \text{if } Y \neq y \end{cases} \\
& \uparrow s & \\
& \boxed{Y} & \hat{p}(Y_y) := \begin{bmatrix} y & \neg y \\ p & 1-p \end{bmatrix}
\end{array}$$

So, when we add $\Pr(Y = y) = p$ to a PDG m , what we really mean is: first convert construct a widget as above, and add that structure (i.e., the new variable Y_y , its definition s , and the cpd \hat{p}) to m . In what sense does this “work”? The first order of business is to prove that it behaves as we should expect, semantically, in the case we’re interested in.

Lemma 9.4.1. *Suppose that m is a PDG with variables \mathcal{X} and $\beta \geq 0$. Then, for all $Y \subseteq \mathcal{X}$, $y \in \mathcal{V}Y$, $p \in [0, 1]$ and $\gamma \geq 0$, we have that:*

$$\langle\!\langle m + \Pr(Y=y) = p \rangle\!\rangle_\gamma \geq \langle\!\langle m \rangle\!\rangle_\gamma,$$

with equality if and only if there exists $\mu \in \llbracket m \rrbracket_\gamma^*$ such that $\mu(Y=y) = p$.

Proof. The inequality is immediate; it is an instance of monotonicity of inconsistency [81, Lemma 1]. Intuitively: believing more cannot make you any less inconsistent. We now prove that equality holds iff there is a minimizer with the appropriate conditional probability.

(\Leftarrow). Suppose that there is some $\mu \in \llbracket m \rrbracket_\gamma^*$ with $\mu(Y=y) = p$. Because $\mu \in \llbracket m \rrbracket_\gamma^*$, we know that $\llbracket m \rrbracket_\gamma(\mu) = \langle\!\langle m \rangle\!\rangle$. Let $\hat{\mu}$ be the extension of μ to the new

variable “ Y_y ”, whose value is a function of Y according to s . Then

$$\begin{aligned}
\langle\langle \mathbf{m} + \Pr(Y=y) = p \rangle\rangle_\gamma &\leq [\mathbf{m} + \Pr(Y=y) = p]_\gamma(\hat{\mu}) \\
&= [\mathbf{m}]_\gamma(\mu) + \mathbb{E}_\mu \left[\log \frac{\hat{\mu}(Y_y)}{\hat{p}(Y_y)} \right] \\
&= [\mathbf{m}]_\gamma(\mu) + \mu(Y=y) \log \frac{\mu(Y=y)}{p} + \mu(Y \neq y) \log \frac{\mu(Y \neq y)}{1-p} \\
&= [\mathbf{m}]_\gamma(\mu) + \mu(Y=y) \log(1) + \mu(Y \neq y) \log(1) \\
&= [\mathbf{m}]_\gamma(\mu) = \langle\langle \mathbf{m} \rangle\rangle_\gamma.
\end{aligned}$$

To complete this direction of the proof, it suffices to observe that we already knew the inequality held in the opposite direction (by monotonicity), so the two terms are equal.

(\implies). Suppose that the two inconsistencies are equal, i.e.,
 $\langle\langle \mathbf{m} + \Pr(Y=y) = p \rangle\rangle_\gamma = \langle\langle \mathbf{m} \rangle\rangle_\gamma$.

This time, choose $\hat{\mu} \in [\mathbf{m} + \Pr(Y=y) = p]^*_\gamma$, and define μ to be its marginal on the variables of \mathbf{m} (which contains the same information as $\hat{\mu}$ itself). Let $q := \mu(Y=y)$. Then

$$\begin{aligned}
\langle\langle \mathbf{m} \rangle\rangle_\gamma &= \langle\langle \mathbf{m} + \Pr(Y=y) = p \rangle\rangle_\gamma \\
&= [\mathbf{m} + \Pr(Y=y) = p]_\gamma(\hat{\mu}) \\
&= [\mathbf{m}]_\gamma(\mu) + \mu(Y=y) \log \frac{\mu(Y=y)}{p} + \mu(Y \neq y) \log \frac{\mu(Y \neq y)}{1-p} \\
&= [\mathbf{m}]_\gamma(\mu) + \left[q \log \frac{q}{p} + (1-q) \log \frac{1-q}{1-p} \right] \\
&= [\mathbf{m}]_\gamma(\mu) + D(q \parallel p) \\
&\geq \langle\langle \mathbf{m} \rangle\rangle_\gamma + D(q \parallel p)
\end{aligned}$$

Therefore $0 \geq D(q \parallel p)$. But relative entropy is non-negative (Gibbs inequality; see any introductory text on information theory, such as MacKay [63]), so we

actually know that $D(q \parallel p) = 0$, and thus $p = \mu(Y=y)$. In addition, the algebra above shows that $\mu \in [\![\mathbf{m}]\!]_\gamma^*$, as its score is $\langle\!\langle \mathbf{m} \rangle\!\rangle_\gamma$. Thus, we have found $\mu \in [\![\mathbf{m}]\!]_\gamma^*$ such that $\mu(Y=y) = p$, completing the proof. \square

We next show that the overall inconsistency is strictly convex in the parameter $p \in [0, 1]$. It is (notationally) simpler to state (and equally easy to prove) this result in the general case.

Lemma 9.4.2. Fix $Y \subseteq \mathcal{X}$, and $\gamma \in (0, \min_a \frac{\beta_a}{\alpha_a})$. As $h = h(Y)$ ranges over $\Delta \mathcal{V}Y$, the function $h \mapsto \langle\!\langle \mathbf{m} + h \rangle\!\rangle_\gamma$ is strictly convex.

[link to
proof]

Proof. We start by expanding the definitions. If h is a cpd on Y given X , then

$$\begin{aligned}\langle\!\langle \mathbf{m} + h \rangle\!\rangle_\gamma &= \inf_{\mu} [\![\mathbf{m} + h]\!]_\gamma(\mu) \\ &= \inf_{\mu} \left[[\![\mathbf{m}]\!]_\gamma(\mu) + D\left(\mu(Y) \parallel h(Y)\right) \right].\end{aligned}$$

Fix $\gamma \leq \min_a \frac{\beta_a}{\alpha_a}$. Then we know that $[\![\mathbf{m}]\!]_\gamma(\mu)$ is a γ -strongly convex (so, in particular, strictly convex) function of μ , and hence there is a unique joint distribution which minimizes it. We now show that the overall inconsistency is strictly convex in h .

Suppose that $h_1(Y)$ and $h_2(Y)$ are two distributions over Y . Let μ_1, μ_2 and μ_λ be the joint distributions that minimize $[\![\mathbf{m} + h_1]\!]_\gamma$ and $[\![\mathbf{m} + h_2]\!]_\gamma$, respectively. For every $\lambda \in [0, 1]$, define $h_\lambda := (1 - \lambda)h_1 + \lambda h_2$, $\mu_\lambda := (1 - \lambda)\mu_1 + \lambda\mu_2$, and μ_λ^* to be a minimizer of $[\![\mathbf{m} + h_\lambda]\!]_\gamma$. The following is a simple consequence of these

definitions:

$$\begin{aligned}
\langle\langle \mathbf{m} + h_\lambda \rangle\rangle_\gamma &= [\![\mathbf{m} + h_\lambda]\!]_\gamma(\mu_\lambda^*) \\
&\leq [\![\mathbf{m} + h_\lambda]\!]_\gamma(\mu_\lambda) \\
&= [\![\mathbf{m}]\!]_\gamma(\mu_\lambda) + \mathbf{D}\left(\mu_\lambda(Y) \parallel h_\lambda(Y)\right).
\end{aligned}$$

By the convexity of $[\![\mathbf{m}]\!]_\gamma$ and \mathbf{D} , we have

$$[\![\mathbf{m}]\!]_\gamma(\mu_\lambda) \leq (1 - \lambda)[![\mathbf{m}]\!]_\gamma(\mu_1) + \lambda[\![\mathbf{m}]\!]_\gamma(\mu_2) \quad (9.3)$$

$$\begin{aligned}
\text{and } \mathbf{D}\left(\mu_\lambda(Y) \parallel h_\lambda(Y)\right) &\leq (1 - \lambda)\mathbf{D}\left(\mu_1(Y) \parallel h_1(Y)\right) \\
&\quad + \lambda \mathbf{D}\left(\mu_2(Y) \parallel h_2(Y)\right). \quad (9.4)
\end{aligned}$$

If $\mu_1 \neq \mu_2$ then since $[\![\mathbf{m}]\!]$ is strictly convex, (9.3) must be a strict inequality. On the other hand, if $\mu_1 = \mu_2$, then since $\mu_\lambda = \mu_1 = \mu_2$ and \mathbf{D} is strictly convex in its second argument when its first argument is fixed, (9.4) must be a strict inequality. In either case, the sum of the two inequalities must be strict. Combining this with the first inequality, we get

$$\begin{aligned}
\langle\langle \mathbf{m} + h_\lambda \rangle\rangle_\gamma &\leq [\![\mathbf{m}]\!]_\gamma(\mu_\lambda) + \mathbf{D}\left(\mu_\lambda(Y) \parallel h_\lambda(Y)\right) \\
&< (\lambda - 1) \left[[\![\mathbf{m}]\!]_\gamma(\mu_1) + \mathbf{D}\left(\mu_1(Y) \parallel h_1(Y)\right) \right] \\
&\quad + \lambda \left[[\![\mathbf{m}]\!]_\gamma(\mu_2) + \mathbf{D}\left(\mu_2(Y) \parallel h_2(Y)\right) \right] \\
&= (\lambda - 1)\langle\langle \mathbf{m} + h_1 \rangle\rangle + \lambda \langle\langle \mathbf{m} + h_2 \rangle\rangle,
\end{aligned}$$

which shows that $\langle\langle \mathbf{m} + h \rangle\rangle$ is *strictly* convex in h , as desired. \square

Let \mathbf{m} be a PDG with $\beta \geq \mathbf{0}$ and variables \mathcal{X} , and fix $Y \subseteq \mathcal{X}$, $y \in \mathcal{V}Y$, and $\gamma \in (0, \min_a \frac{\beta_a}{\alpha_a})$. For $p \in [0, 1]$, define

$$f(p) := \langle\langle \mathbf{m} + \Pr(Y=y) \rangle\rangle_\gamma. \quad (9.5)$$

The next several results (?? 9.4.2.1 and Lemmas 9.4.3, 9.4.4 and 9.4.6 to 9.4.8) are properties of this function $f(p)$.

Corollary 9.4.2.1. *The function f defined in (9.5) is strictly convex.*

Proof. Simply take h to be the cpd \hat{p} , absorb the other components of (the PDG representation of) $\Pr(Y=y) = p$ into \mathbf{m} , and then apply Lemma 9.4.2. \square

The results from this point until the proof of Theorem 14 are all technical results that support the more precise analysis of part (b). We recommend returning to these results as needed, after first reading the proof of part (a).

Lemma 9.4.3. *For $p \in (0, 1]$, let μ_p be the unique element of $[\![\mathbf{m} + \Pr(Y=y) = p]\!]_\gamma^*$. Then $f'(p) = \frac{p - \mu_p^*(Y=y)}{p(1-p)}$.*

Proof. First, suppose μ_p^* is in the interior of the simplex. Since it minimizes the differentiable function

$$[\![\mathbf{m} + \Pr(Y=y) = p]\!]_\gamma = \mu \mapsto [\![\mathbf{m}]\!]_\gamma + D(\mu(Y=y) \parallel p),$$

the gradient of that function at μ_p^* must be zero:

$$\nabla [\![\mathbf{m} + \Pr(Y=y) = p]\!]_\gamma(\mu_p^*) = \nabla_\mu \left[D(\mu(Y=y) \parallel p) \right]_{\mu=\mu_p^*} + \nabla [\![\mathbf{m}]\!](\mu_p^*) = 0. \quad (9.6)$$

What is the derivative of f ? Observe that f is the sum of two compositions of differentiable maps:

$$\begin{aligned} f_m := & \quad p \mapsto \mu_p^* \mapsto [\![\mathbf{m}]\!]_\gamma(\mu_p^*) \\ \text{and} \quad f_{\Pr} := & \quad p \mapsto (p, \mu_p^*) \mapsto D(\mu_p^*(Y=y) \parallel p). \end{aligned}$$

Thus, we can use the multivariate chain rule. for any differentiable functions $h : \mathbb{R}^m \rightarrow \mathbb{R}^k$, and $g : \mathbb{R}^k \rightarrow \mathbb{R}^n$, their composition $g \circ h : \mathbb{R}^m \rightarrow \mathbb{R}^n$ is also a differentiable map whose Jacobian is $\mathbf{J}_{g \circ h}(x) = \mathbf{J}_g(h(x))\mathbf{J}_h(x)$. In our case, g will be a scalar map ($n = 1$), so $\mathbf{J}_g(h(x)) = [\dots, \frac{\partial g}{\partial x_j}, \dots](h(x)) = \nabla g(h(x))^\top$. Let $\mathbf{J}_{\mu_p^*}(p)$ be the Jacobian of the map $p \mapsto \mu_p^*$. Then

$$\begin{aligned} f'(p) &= f'_M(p) + f'_{Pr}(p) \\ &= (\nabla \llbracket \mathbf{m} \rrbracket(\mu_p^*))^\top \mathbf{J}_{\mu_p^*} + \left(\nabla_\mu \left[\mu(x) D(\mu(y|x) \parallel p) \right]_{\mu=\mu_p^*} \right)^\top \mathbf{J}_{\mu_p^*} + \frac{\partial}{\partial p} \left[D(\mu_p^*(Y=y) \parallel p) \right] \end{aligned}$$

(Alternatively, the line above can be derived from the law of total derivative.²)

$$\begin{aligned} &= \cancel{\left(\nabla_\mu \left[D(\mu(Y=y) \parallel p) \right]_{\mu=\mu_p^*} \right)} + \cancel{\nabla \llbracket \mathbf{m} \rrbracket(\mu_p^*)^\top} \mathbf{J}_{\mu_p^*}(p) + \frac{\partial}{\partial p} D(\mu_p^*(Y=y) \parallel p) \\ &= \frac{\partial}{\partial p} D(\mu_p^*(Y=y) \parallel p) \quad \text{by (9.6)} \end{aligned}$$

Finally,

$$\begin{aligned} \frac{d}{dp} \left[D(q \parallel p) \right] &= \frac{d}{dp} \left[q \log \frac{q}{p} + (1-q) \log \frac{1-q}{1-p} \right] \\ &= q \left(\frac{p}{q} \right) \frac{d}{dp} \left[\frac{q}{p} \right] + (1-q) \left(\frac{1-p}{1-q} \right) \frac{d}{dp} \left[\frac{1-q}{1-p} \right] \\ &= pq \left(\frac{-1}{p^2} \right) + (1-p)(1-q) \left(\frac{-1}{(1-p)^2} \right) (-1) \\ &= -\frac{q}{p} + \frac{1-q}{1-p} \\ &= \frac{-(1-p)q + p(1-q)}{p(1-p)} \\ &= \frac{pq - q + p - pq}{p(1-p)} = \frac{p - q}{p(1-p)}. \end{aligned}$$

Thus, we find

$$f'(p) = \frac{p - \mu_p^*(Y=y)}{p(1-p)}, \quad \text{as promised.} \quad \square$$

²Law of total derivative: $\frac{df}{dp} = \sum_{w \in \mathcal{VX}} \frac{\partial D(\mu(y|x) \parallel p)}{\partial \mu_p^*(w)}(\mu_p^*, p) \frac{\partial \mu_p^*(w)}{\partial p} + \frac{\partial}{\partial p} D(\mu_p^*, p) + \sum_{w \in \mathcal{VX}} \frac{\partial \llbracket \mathbf{m} \rrbracket_\gamma}{\partial \mu_p^*(w)}(\mu_p^*) \frac{\partial \mu_p^*(w)}{\partial p}.$

Lemma 9.4.4. *If $0 < p_1 < p_2 < 1$ and μ_1^*, μ_2^* are respective minimizing distributions, then*

$$f(p_2) \geq f(p_1) + f'(p_1)(p_2 - p_1) + \frac{1}{2}\gamma\|\mu_1^* - \mu_2^*\|_1^2.$$

Proof. The general approach is to adapt and strengthen the proof of Lemma 9.4.2, to show something like strong convexity, in this special case. Define

$$\mathbf{m}_1 := \mathbf{m} + \Pr(Y=y) = p_1 \quad \text{and} \quad \mathbf{m}_2 := \mathbf{m} + \Pr(Y=y) = p_2.$$

Choose $\mu_1 \in [\![\mathbf{m}_1]\!]_\gamma^*$ and $\mu_2 \in [\![\mathbf{m}_2]\!]_\gamma^*$. As before, let $m_1 := \mu_1(Y=y)$ and $m_2 := \mu_2(Y=y)$. Then

$$f(p_1) = \langle\langle \mathbf{m}_1 \rangle\rangle_\gamma = [\![\mathbf{m}_1]\!]_\gamma(\mu_1) = [\![\mathbf{m}]\!]_\gamma(\mu_1) + \mathbf{D}(m_1 \parallel p_1)$$

and $f(p_2) = \langle\langle \mathbf{m}_2 \rangle\rangle_\gamma = [\![\mathbf{m}_2]\!]_\gamma(\mu_2) = [\![\mathbf{m}]\!]_\gamma(\mu_1) + \mathbf{D}(m_2 \parallel p_2),$

where, as before, $\mathbf{D}(m \parallel p) = m \log \frac{m}{p} + (1-m) \log \frac{1-m}{1-p}$ is the relative entropy between Bernouli distributions with respective parameters m and p .

For each $\lambda \in [0, 1]$, define

$$p_\lambda := (1-\lambda)p_1 + \lambda p_2, \quad \mathbf{m}_\lambda := \mathbf{m} + \Pr(Y=y) = p_\lambda, \quad \text{and} \quad \mu_\lambda := (1-\lambda)\mu_1 + \lambda\mu_2.$$

We now provide stronger analogues of (9.3) and (9.4). Since $[\![\mathbf{m}]\!]_\gamma$ is not just convex but also γ -strongly convex, with respect to the 1-norm (Lemma 9.4.5, below), we can strengthen (9.3) to

$$[\![\mathbf{m}]\!]_\gamma(\mu_\lambda) \leq (1-\lambda)[\![\mathbf{m}]\!]_\gamma(\mu_1) + \lambda[\![\mathbf{m}]\!]_\gamma(\mu_2) - \frac{\gamma}{2}(1-\lambda)\lambda\|\mu_1 - \mu_2\|_1^2.$$

Adding this inequality to the analogous one describing joint convexity of \mathbf{D} in

its two arguments (9.4), we find that

$$\begin{aligned}
& \llbracket \mathbf{m} \rrbracket_{\gamma}(\mu_{\lambda}) + \mathbf{D}(m_{\lambda} \parallel p_{\lambda}) \\
& \leq (1 - \lambda) \left(\llbracket \mathbf{m} \rrbracket_{\gamma}(\mu_1) + \mathbf{D}(m_1 \parallel p_1) \right) + \lambda \left(\llbracket \mathbf{m} \rrbracket_{\gamma}(\mu_2) + \mathbf{D}(m_2 \parallel p_2) \right) - \frac{\gamma}{2} (1 - \lambda) \lambda \|\mu_1 - \mu_2\|_1^2 \\
& = (1 - \lambda) f(p_1) + \lambda f(p_2) - \frac{\gamma}{2} (1 - \lambda) \lambda \|\mu_1 - \mu_2\|_1^2.
\end{aligned} \tag{9.7}$$

Putting it all together, we find that

$$\begin{aligned}
f(p_{\lambda}) &= \langle \langle \mathbf{m}_{\lambda} \rangle \rangle_{\gamma} \\
&\leq \llbracket \mathbf{m}_{\lambda} \rrbracket_{\gamma}(\mu_{\lambda}) \\
&= \llbracket \mathbf{m} \rrbracket_{\gamma}(\mu_{\lambda}) + \mathbf{D}(m_{\lambda} \parallel p_{\lambda}) \\
&\leq (1 - \lambda) f(p_1) + \lambda f(p_2) - \frac{\gamma}{2} (1 - \lambda) \lambda \|\mu_1 - \mu_2\|_1^2
\end{aligned} \tag{9.7}.$$

Since this is true for all $\lambda \in [0, 1]$, we can divide by λ , rearrange, and take the limit as $\lambda \rightarrow 0$, to find:

$$\begin{aligned}
\frac{\gamma}{2} (1 - \lambda) \|\mu_1 - \mu_2\|_1^2 &\leq \frac{(1 - \lambda) f(p_1) - f(p_1 + \lambda(p_2 - p_1))}{\lambda} + f(p_2) \\
&= \frac{f(p_1) - f(p_1 + \lambda(p_2 - p_1))}{\lambda} + f(p_2) - f(p_1) \\
\implies f(p_2) - f(p_1) &\geq \lim_{\lambda \rightarrow 0} f(p_1) + \frac{f(p_1 + \lambda(p_2 - p_1)) - f(p_1)}{\lambda} \frac{\gamma}{2} (1 - \lambda) \|\mu_1 - \mu_2\|_1^2 \\
&= f'(p_1)(p_2 - p_1) + \frac{\gamma}{2} \|\mu_1 - \mu_2\|_1^2,
\end{aligned}$$

as desired. \square

Lemma 9.4.5. *Negative entropy is 1-strongly convex with respect with respect to the L1 norm, i.e., $f(\mathbf{p}) = \sum_i p_i \log p_i$ satisfies*

$$f(\mathbf{q}) \geq f(\mathbf{p}) + \nabla f(\mathbf{p})^T (\mathbf{q} - \mathbf{p}) + \frac{1}{2} \|\mathbf{p} - \mathbf{q}\|_1^2.$$

Proof. First, $\nabla f(\mathbf{p}) = \log(\mathbf{p}) + 1$. Thus,

$$\begin{aligned}
& f(\mathbf{q}) - f(\mathbf{p}) - \nabla f(\mathbf{p})^\top (\mathbf{q} - \mathbf{p}) \\
&= \sum_i [q_i \log q_i - p_i \log p_i - (\log p_i + 1)(q_i - p_i)] \\
&= \sum_i [q_i \log q_i - \overline{\log p_i} - q_i \log p_i + \overline{\log p_i}] \\
&= \sum_i q_i \log \frac{q_i}{p_i} \\
&= D(q \| p) \\
&\geq 2\delta(\mathbf{p}, \mathbf{q})^2 \quad [\text{by Pinsker's inequality [?]}] \\
&= 2\left(\frac{1}{2}\|\mathbf{p} - \mathbf{q}\|_1\right)^2 \\
&= \frac{1}{2}\|\mathbf{p} - \mathbf{q}\|_1^2
\end{aligned}$$

where $\delta(\mathbf{p}, \mathbf{q})$ is the total variation distance between \mathbf{p} and \mathbf{q} as measures, and $\|\mathbf{p} - \mathbf{q}\|_1 = \sum_i |p_i - q_i|$ is the L1 norm of their difference, as points on a simplex. \square

[Lemma 9.4.4](#) guarantees that if the optimal distributions corresponding to adding p_1 and p_2 to the PDG (μ_1 and μ_2 , respectively) are far apart, then so are $f(p_1)$ and $f(p_2)$. But what if these optimal distributions are close together? It turns out that if $\mu_1 \approx \mu_2$ then it's still the case that $f(p_1)$ and $f(p_2)$ are far apart, provided that p_1 and p_2 are. However, showing this requires an entirely approach, which we pursue in [Lemma 9.4.7](#). But first, we need two intermediate technical results.

Lemma 9.4.6. *For all $p_1, p_2 \in [0, 1]$,*

$$[\![\mathbf{m}]\!]_\gamma(\mu_1^*) - [\![\mathbf{m}]\!]_\gamma(\mu_2^*) \geq (m_2 - m_1) \log \frac{m_2}{p_2} \frac{1 - p_2}{1 - m_2},$$

where $m_1 = \mu_1^*(Y=y)$ is the marginal of $\mu_1^* \in [\![\mathbf{m} + \Pr(Y=y) = p_1]\!]_\gamma^*$, and $m_2 = \mu_2^*(Y=y)$ is defined symmetrically.

Proof. For $p, q \in [0, 1]$, $D(p \parallel q) := p \log \frac{p}{q} + (1-p) \log \frac{1-p}{1-q}$ is the relative entropy between the Bernouli distributions described by their parameters. First, we calculate

$$\begin{aligned}\frac{d}{dp} D(p \parallel q) &= \log \frac{p}{q} + \frac{p}{q} q \frac{d}{dp} \left[\frac{p}{q} \right] + (-1) \log \frac{1-p}{1-q} + (1-p) \frac{d}{dp} \left[\frac{1-p}{1-q} \right] \\ &= \log \frac{p}{q} + 1 - \log \frac{1-p}{1-q} - 1 \\ &= \log \left(\frac{p}{q} \frac{1-q}{1-p} \right).\end{aligned}$$

Thus,

$$\begin{aligned}\nabla_{\mu} [D(\mu(Y=y) \parallel q)] &= \nabla_{\mu} [\mu(Y=y)] \log \left(\frac{\mu(Y=y)}{q} \frac{1-q}{1-\mu(Y=y)} \right) \\ &= \mathbb{1}[Y=y] \log \left(\frac{\mu(Y=y)}{q} \frac{1-q}{1-\mu(Y=y)} \right).\end{aligned}$$

Recall the stationary conditions, which state that, since $\mu_2^* \in [\mathbf{m} + \Pr(Y=y) = p_2]$, we have

$$\nabla_{\mu} [D(\mu(Y=y) \parallel p_2)]_{\mu=\mu_2^*} = -\nabla_{\mu} [\mathbf{m}]_{\gamma}(\mu_2^*).$$

Now use the above to compute the directional derivative of interest:

$$\begin{aligned}\nabla_{\mu} [\mathbf{m}]_{\gamma}(\mu_2^*)^T (\mu_1^* - \mu_2^*) \\ &= -(\mu_1^* - \mu_2^*)^T \nabla_{\mu} [D(\mu(Y=y) \parallel p_2)]_{\mu=\mu_2^*} \\ &= (\mu_2^*(Y=y) - \mu_1^*(Y=y)) \log \left(\frac{\mu_2^*(Y=y)}{p_2} \frac{1-p_2}{1-\mu_2^*(Y=y)} \right) \\ &= (m_2 - m_1) \log \frac{m_2}{p_2} \frac{1-p_2}{1-m_2}.\end{aligned}$$

Finally, since $[\mathbf{m}]_{\gamma}$ is convex, we have

$$\begin{aligned}[\mathbf{m}]_{\gamma}(\mu_1^*) - [\mathbf{m}]_{\gamma}(\mu_2^*) &\geq \nabla_{\mu} [\mathbf{m}]_{\gamma}(\mu_2^*)^T (\mu_1^* - \mu_2^*) \\ &= s_1(m_2 - m_1) \log \frac{m_2}{p_2} \frac{1-p_2}{1-m_2}\end{aligned}$$

as promised. □

Lemma 9.4.7. Suppose that $0 < b < z < p^* < 1$, and let

$$\mu_z^* \in [\![\mathbf{m} + \Pr(Y=y) = z]\!]_\gamma^* \quad \text{and} \quad \mu_b^* \in [\![\mathbf{m} + \Pr(Y=y) = b]\!]_\gamma^*$$

be the respective optimal distributions for their corresponding PDGs. If $\|\mu_b - \mu_z\|_1 \leq \delta$, then

$$f(b) - f(z) \geq (z - b)^2 - \left(\frac{2}{b} + \log \frac{1-b}{1-z} + \log \frac{1}{b} \right) \delta.$$

Proof. Because we have assumed $b < z < p^*$ and f is strictly convex (Lemma 9.4.2), it follows from Lemma 9.4.3 that $m_z := \mu_z^*(Y=y) \geq z$ and $m_b := \mu_b^*(Y=y) \geq b$. Now

$$\begin{aligned} f(b) - f(z) &= ([\![\mathbf{m}]\!]_\gamma(\mu_b^*) + D(m_b \| b)) - ([\![\mathbf{m}]\!]_\gamma(\mu_z^*) + D(m_z \| z)) \\ &= ([\![\mathbf{m}]\!]_\gamma(\mu_b^*) - [\![\mathbf{m}]\!]_\gamma(\mu_z^*)) + D(m_b \| b) - D(m_z \| z). \end{aligned}$$

Lemma 9.4.6 will give us a lower bound for the first half; we now investigate the second half. The first step is some algebraic manipulation.

$$\begin{aligned} D(m_b \| b) - D(m_z \| z) &= -m_z \log \frac{m_z}{z} - (1-m_z) \log \frac{1-m_z}{1-z} + m_b \log \frac{m_b}{b} + (1-m_b) \log \frac{1-m_b}{1-b} \\ &= m_z \log \frac{z}{m_z} + (1-m_z) \log \frac{1-z}{1-m_z} + (m_b + m_z - m_z) \log \frac{m_b}{b} \\ &\quad + ((1-m_b) + (1-m_z) - (1-m_z)) \log \frac{1-m_b}{1-b} \tag{add z-marginal terms} \\ &= m_z \left(\log \frac{z}{m_z} + \log \frac{m_b}{b} \right) + (1-m_z) \left(\log \frac{1-z}{1-m_z} + \log \frac{1-m_b}{1-b} \right) \\ &\quad + (m_b - m_z) \log \frac{m_b}{b} + ((1-m_b) - (1-m_z)) \log \frac{1-m_b}{1-b} \tag{collect z-marginal terms} \\ &= m_z \log \frac{z}{b} \frac{\textcolor{orange}{m_b}}{\textcolor{orange}{m_z}} + (1-m_z) \log \frac{1-z}{1-b} \frac{\textcolor{orange}{1-m_b}}{\textcolor{orange}{1-m_z}} \\ &\quad + (m_b - m_z) \log \frac{m_b}{b} \frac{1-b}{1-m_b} \\ &=: \blacksquare_0 + \blacksquare_1 + \blacksquare_2, \end{aligned}$$

where, to be explicit, we have defined

$$\begin{aligned}\blacksquare_0 &= m_z \log \frac{z}{b} + (1 - m_z) \log \frac{1 - z}{1 - b} \\ \blacksquare_1 &= m_z \log \frac{m_b}{m_z} + (1 - m_z) \log \frac{1 - m_b}{1 - m_z} = -D(m_z \parallel m_b) \\ \blacksquare_2 &= (m_b - m_z) \log \frac{m_b}{b} \frac{1 - b}{1 - m_b}.\end{aligned}$$

There is one final quantity that will play a similar role. Let

$$\blacksquare_4 := (m_z - m_b) \log \frac{m_z}{z} \frac{1 - z}{1 - m_z}$$

be the lower bound on $\llbracket m \rrbracket_\gamma(\mu_b^*) - \llbracket m \rrbracket_\gamma(\mu_z^*)$ obtained by applying Lemma 9.4.6 with $p_1 = b$ and $p_2 = z$. With these definitions, we have $f(b) - f(z) \geq \blacksquare_0 + \blacksquare_1 + \blacksquare_2 + \blacksquare_4$.

Observe that if m_z were equal to z , then \blacksquare_0 would equal $D(z \parallel b)$. But in fact we know that $m_z > z$. It is easy to see that that \blacksquare_0 is linear in m_z with positive slope, since $z > b$ and $1 - b > 1 - z$. It follows that $\blacksquare_0 > D(z \parallel b)$.

Let's step back for a moment. Lemma 9.4.4 shows that $f(b)$ and $f(z)$ cannot be too close, provided that μ_z^* and μ_b^* are far apart. In the equations above, we can see the beginnings of a complementary argument: if μ_z^* and μ_b^* are close together, then $m_b \approx m_b$, and so all terms apart from \blacksquare_0 (i.e., $\blacksquare_1 + \blacksquare_2 + \blacksquare_4$) go to zero. And yet, because of \blacksquare_0 , $f(b)$ and $f(z)$ remain far apart. To make this argument precise, we now merge \blacksquare_1 , \blacksquare_2 , and \blacksquare_4 back together, calculating

$$\begin{aligned}\blacksquare_2 + \blacksquare_4 + \blacksquare_1 &= (m_b - m_z) \log \frac{m_b}{b} \frac{1 - b}{1 - m_b} \frac{z}{m_z} \frac{1 - m_z}{1 - z} - D(m_z \parallel m_b) \\ &= (m_b - m_z) \log \frac{z}{b} \frac{1 - b}{1 - z} + (m_b - m_z) \log \frac{m_b}{m_z} \frac{1 - m_z}{1 - m_b} + m_z \log \frac{m_b}{m_z} + (1 - m_z) \log \frac{1 - m_b}{1 - m_z} \\ &= (m_b - m_z) \log \frac{z}{b} \frac{1 - b}{1 - z} + (m_b - m_z + m_z) \log \frac{m_b}{m_z} + (1 - m_z + m_z - m_b) \frac{1 - m_b}{1 - m_z} \\ &= (m_b - m_z) \log \frac{z}{b} \frac{1 - b}{1 - z} + D(m_b \parallel m_z) \\ &\geq (m_b - m_z) \log \frac{z}{b} \frac{1 - b}{1 - z}.\end{aligned}$$

Suppose that $\|\mu_z^* - \mu_b^*\|_1 \leq \delta$. Because the total variation distance $\text{TV}(p, q)$ is half the L1-norm $\|p - q\|_1$ for discrete distributions,

$$\delta \geq \|\mu_z^* - \mu_b^*\|_1 = 2 \text{TV}(\mu_z^*, \mu_b^*) \geq 2 |\mu_z^*(Y=y) - \mu_b^*(Y=y)| = 2|m_z - m_b|.$$

Thus,

$$\blacksquare_1 + \blacksquare_2 + \blacksquare_4 \geq -\frac{\delta}{2} \log \frac{z}{b} \frac{1-b}{1-z}.$$

All that remains is \blacksquare_0 . To put things in a convenient form, we apply Pinsker's inequality. The total variation distance between two Bernoulli distributions (i.e., binary distributions) with respective positive probabilities p and q is just $|p - q|$. Pinsker's inequality [?] in this case says: $\frac{1}{2}D(p \parallel q) \geq (p - q)^2$. Thus, $\blacksquare_0 > D(z \parallel b) \geq 2(z - b)^2$, and so we have

$$f(b) - f(z) \geq 2(z - b)^2 - \left(\frac{1}{2} \log \frac{z}{b} \frac{1-b}{1-z} \right) \delta \quad \text{as promised. } \square$$

Lemma 9.4.8. Suppose that $b < z < p^*$. Furthermore, suppose that $|\log \frac{z}{b} \frac{1-b}{1-z}| \leq k$. Not only is it the case that $f(b) > f(z)$, but also

$$f(b) - f(z) \geq \frac{k^2}{32\gamma} \log^2 \left(1 + 16 \frac{\gamma}{k^2} (z - b)^2 \right).$$

Proof. We now have two bounds that work in different regimes. If $\delta = \|\mu_b^* - \mu_z^*\|_2$ is large, then the argument of Lemma 9.4.4 is effective, as it shows that a separation between $f(b)$ and $f(z)$ that scales with δ^2 . On the other hand, if δ is small, we saw in Lemma 9.4.7 a very different approach that still gets us a separation of $2(z - b)^2$ even if $\delta = 0$. We now combine the two cases to eliminate the (unknown) parameter δ from our complexity analysis. (Our algorithm is no different in the two cases; all that differs is the analysis.)

Taken together, we know that we attain the maximum of the two lower bounds, which is weakest when they coincide. We can then solve for the worst-case value

of δ , which leads to the smallest possible separation between $f(z)$ and $f(b)$.

Setting the two bounds equal to one another:

$$\begin{aligned} \frac{1}{2}\gamma\delta_{\text{worst}}^2 &= 2(z-b)^2 - \left(\frac{1}{2}\log\frac{z}{b}\frac{1-b}{1-z}\right)\delta \\ \iff \quad \frac{\gamma}{2}\delta_{\text{worst}}^2 + \left(\frac{1}{2}\log\frac{z}{b}\frac{1-b}{1-z}\right)\delta_{\text{worst}} - 2(z-b)^2 &= 0. \end{aligned}$$

The quadratic equation then tells us that

$$\delta_{\text{worst}} = \frac{1}{\gamma} \left(-B + \sqrt{B^2 + 4\gamma(z-b)^2} \right), \quad \text{where} \quad B := \frac{1}{2} \log \frac{z}{b} \frac{1-b}{1-z}.$$

It is easily verified that this expression for δ_{worst} is decreasing in B . Therefore, we get a lower bound on it by plugging in our upper bound $\frac{k}{2}$ for B . Thus $\delta_{\text{worst}} \geq \frac{1}{\gamma} \left(-k + \sqrt{k^2 + 4\gamma(z-b)^2} \right)$.

Square roots are not easy to manipulate in general, and this expression in particular has involves a nested subtraction that makes it hard to characterize. To make things clearer, we now begin to loosen this bound to get a quantity that is easier to think about. The first observation is that, for any numbers $A, B > 0$,

$$-B + \sqrt{B^2 + A} = B \left(-1 + \sqrt{1 + \frac{A}{B^2}} \right).$$

This manipulation puts the square root in the standard form $\sqrt{1+x}$. Here is the second observation: for all x , $-1 + \sqrt{1+x} \geq \frac{1}{2}\log(1+x)$ (verified in [Lemma 9.4.9](#) below). Although it gives a looser bound, the logarithm is easier to manipulate and no longer involves subtraction. Applying these two transformations in our

case, we find:

$$\begin{aligned}
\delta_{\text{worst}} &= \frac{1}{\gamma} \left(-B + \sqrt{B^2 + 4\gamma(z-b)^2} \right) \\
&\geq \frac{1}{\gamma} \left(-\frac{k}{2} + \sqrt{\frac{k^2}{4} + 4\gamma(z-b)^2} \right) \\
&= \frac{k}{2\gamma} \left(-1 + \sqrt{1 + \frac{16}{k^2}\gamma(z-b)^2} \right) \\
&\geq \frac{k}{4\gamma} \log \left(1 + \frac{16}{k^2}\gamma(z-b)^2 \right).
\end{aligned}$$

Finally, since $f(b) - f(z) \geq \frac{\gamma}{2}\delta_{\text{worst}}^2$, to get lower bound for $f(b) - f(z)$, we simply need to square this lower bound for δ_{worst} and multiply by $\gamma/2$. As a result,

$$\begin{aligned}
f(b) - f(z) &\geq \frac{\gamma}{2} \frac{k^2}{16\gamma^2} \log^2 \left(1 + \frac{16\gamma}{k^2}(z-b)^2 \right) \\
&= \frac{k^2}{32\gamma} \log^2 \left(1 + \frac{16\gamma}{k^2}(z-b)^2 \right),
\end{aligned}$$

where $\log^2(x)$ means $(\log(x))^2$. □

Lemma 9.4.9. $-1 + \sqrt{1+x} \geq \frac{1}{2} \log(1+x)$.

Proof. Apply the well-known inequality $e^y \geq 1 + y$ with $y = -1 + \sqrt{1+x}$, to get

$$\begin{aligned}
\exp(-1 + \sqrt{1+x}) &\geq \sqrt{1+x}, \\
\text{which implies } -1 + \sqrt{1+x} &\geq \log \sqrt{1+x} = \frac{1}{2} \log(1+x). \quad \square
\end{aligned}$$

We are now ready to tackle the theorem itself.

Theorem 9.3.2.

- (a) *There is an $O(\log^{1/\epsilon})$ -time reduction from unconditional APPROX-INFER-CVX to the problem of determining which of two PDGs is more inconsistent, using $O(\log^{1/\epsilon})$ subroutine calls.*

- (b) There is an $O\left(\log \frac{\langle\!\langle m \rangle\!\rangle_\gamma}{\gamma \epsilon \mu^*(x)} \cdot \log \frac{1}{\epsilon \mu^*(x)}\right)$ time reduction from APPROX-INFERENCE-CVX to APPROX-CALC-INC using $O(\log(1/\epsilon) \log \log 1/\mu^*(x))$ calls to the inconsistency subroutine.
- (c) There is also an $O(|VC|)$ reduction from APPROX-CALC-INC to APPROX-INFERENCE-CVX. With the additional assumption of bounded treewidth, this is linear in the number of variables in the PDG.

Proof. (a,b). Suppose that we have access to a procedure that can calculate a PDG's degree of inconsistency. The idea behind the reductions of both (a) and (b) is to perform $\hat{\gamma}$ -inference on a given PDG m , using this procedure as a subroutine. The complexity of the reduction depends on the specification of the inconsistency calculation procedure. We will perform two analyses, the second building on the first.

1. First, we assume the procedure simply tells us which of two PDGs is more inconsistent. With this assumption, we get an algorithm that can answer unconditional probability queries with the optimal complexity of part (a) of the theorem.
2. We then provide a refinement of that algorithm that still works if the inconsistency calculation procedure produces only finite-precision binary approximations to inconsistency values—thus reducing the problem of approximate inference that of approximately calculating PDG inconsistency. This considerably more difficult analysis gives us part (b). In addition, we extend the algorithm, using Lemma 8.A.7, so that it can also answer conditional queries.

We begin by describing our algorithm, which uses the first variant of the inconsistency procedure (the one that tells us which of two PDGs is more inconsistent) to produce a sequence of approximations (p_1, p_2, \dots) that converges exponentially to

$$p^* := \llbracket m \rrbracket_{\gamma}^*(Y=y) \stackrel{\text{(Lemma 9.4.1)}}{=} \arg \min_p \langle\!\langle m + (\Pr(Y=y) = p) \rangle\!\rangle_{\gamma}$$

through a variant of binary search. More precisely, the algorithm we present below is an extremely close relative of an algorithm known to the competitive programming community as *trinary search* [?]. In some ways it is slightly more efficient, but its analysis is more complex.

The state of the algorithm consists of three points in an interval $a, b, c \in [0, 1]$, where $a \leq b \leq c$. Intuitively, b is our current best guess at p^* , while a is a lower bound, and c is an upper bound. Once again (as in (9.5) and in preceding lemmas), let f be the function

$$\begin{aligned} f : [0, 1] &\rightarrow \bar{\mathbb{R}} \\ p &\mapsto \langle\!\langle m + (\Pr(Y=y) = p) \rangle\!\rangle_{\gamma}. \end{aligned}$$

Both variants of the inconsistency calculation procedure will be employed for the sole purpose of determining whether or not $f(p) > f(p')$, given $p, p' \in [0, 1]$. We start with the simpler variant, which can directly determine which of two PDGs has greater inconsistency. In this case, the \gg and \ll in the algorithm below should be interpreted simply as $>$ as $<$. This will enable us to approximate the minimizer p^* of f arbitrarily closely, with the following algorithm.

Initialize $(a, b, c) \leftarrow (0, \frac{1}{2}, 1)$;

while $|c - a| > \epsilon$ **do**

if $b - a \geq c - b$ **then**

 Let $z := \frac{b+a}{2}$;

if $f(b) \gg f(z)$ **then**

$(a, b, c) \leftarrow (a, z, b)$;

else

$(a, b, c) \leftarrow (z, b, c)$;

else if $c - b > b - a$ **then**

 Let $z := \frac{b+c}{2}$;

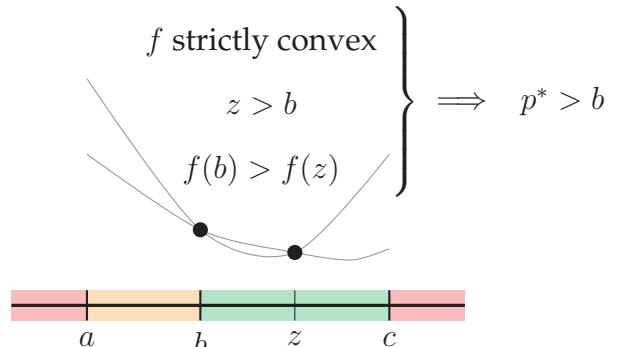
if $f(z) \ll f(b)$ **then**

$(a, b, c) \leftarrow (b, z, c)$;

else

$(a, b, c) \leftarrow (a, b, z)$;

return b ;



We begin by proving that this algorithm does indeed output a point within ϵ of p^* . Because f is convex, this algorithm satisfies an important invariant:

Claim 9.4.9.1. *Both b and p^* always lie in the interval $[a, c]$.*

Proof. We proceed by induction on i . At the beginning, it is obviously true that b and p^* lie in $[a, b] = [0, 1]$, which contains the entire domain of f . Now, suppose inductively that $p^* \in [a, b]$ at some iteration of the algorithm i .

(case 1) If $b - a \geq c - b$, then $z \in [a, b]$.

- Suppose $f(z) < f(b)$. Then for all $y > b$, it must be the case that $f(y) > f(b)$ by convexity of f . (For if $f(y) < f(b)$, then segment between $(z, f(z))$ and $(y, f(y))$ would lie entirely below $(b, f(b))$, which contradicts convexity). Thus, we can rule out all such y as possible minimizers of f , so we can restrict our attention to $[a, b]$, which contains p^* (and x).
- On the other hand, if $f(z) > f(b)$, then it must be the case that no $y < z$ can be a minimizer of f by convexity, with the same reasoning as above. (Namely, if $f(y) < f(z)$ then the segment between $(y, f(y))$ and $(b, f(b))$ lies below $(z, f(z))$, contradicting convexity). Thus the true minimizer p^* lies in $[z, c]$, an interval which contains b .

(case 2) The other case is symmetric; we include it for completeness. Suppose

$$c - b > b - a, \text{ and so } z = \frac{b+c}{2}.$$

- Suppose $f(z) < f(b)$. Then $f(y) > f(b)$ for all $y < b$ (because if $f(y) < f(b)$, then segment between $(y, f(y))$ and $(x, f(x))$ would lie below $(b, f(b))$). So $p^*, z \in [b, c]$.
- On the other hand, if $f(z) > f(b)$, then $f(y) > f(z)$ for all $y > z$ (because, if $f(y) < f(z)$ then the segment between $(y, f(y))$ and $(b, f(b))$ lies below $(z, f(z))$, contradicting convexity). So $p^*, b \in [a, z]$. \square

In every case, p^* is still in what becomes the interval $[a, c]$ in the next iteration ($i + 1$). So, by induction, $p^* \in [a, b]$ at every iteration of the algorithm, proving ?? 9.4.9.1. \square

We have shown that both b and p^* lie within $[a, c]$, and we know that, at termination, $|c - a| < \epsilon$. Therefore, the final value of b (i.e., the output of the

algorithm) must be within ϵ of p^* .

Next, we analyze the complexity of this algorithm, modulo the complexity of comparing the numbers $f(z)$ and $f(b)$, which we will later bound precisely. Each iteration reduces the size of the interval $[a, c]$ by a factor of at least $3/4$. This is because in each case we focus on the larger half of the interval, and ultimately discard either half or all of it—so we reduce the size of the interval by at least one quarter. It follows that, after n iterations, the size of the interval is at most $(3/4)^n$, and thus the total number of iterations is at most $\lceil \log(1/\epsilon)/(\log 4/3) \rceil$. Apart from the time needed to compare $f(b)$ and $f(z)$, it is easy to see that each iteration of the algorithm takes constant time. So overall, it requires $\log(1/\epsilon)$ (enough to track the numbers $\{a, b, c\}$, plus a reference to the PDG M), and time $O(\log 1/\epsilon)$, which is linear in the number of bits returned. This completes the proof of [Theorem 9.3.2 \(a\)](#).

Although it is common to assume that numbers can be compared in $O(1)$ time, and this is an assumption well suited to modern computer architecture, it is arguably not appropriate in this context. How do we know a `float64` has enough precision to do the comparison? The obvious approach to implementing the inconsistency calculation subroutine would be to repeatedly request more and more precise estimates of inconsistency, until one is larger than the other—but this procedure does not terminate if the two PDGs have the same inconsistency. So, a priori, it's not even clear that the decision $f(b) > f(z)$ is computable. It is not hard to show that it is in fact computable. Because f is strictly convex, it must be the case that $f(b) > f(z)$, $f(z) > f(b)$, or $f(b) > f(\frac{b+z}{2})$. In the last case, we can act as if $f(b) > f(z)$, and the algorithm will be correct, because the argument supporting ?? 9.4.9.1 still applies. Thus, by running the subroutine on all three questions until one of them returns true, and then aborting the other two

calculations, we can see that the problem of interest is decidable. But how long does it take? We now provide a deeper analysis of the comparison between $f(z)$ and $f(b)$ when the inconsistency calculation procedure can give us only finite approximations to the true value.

Part (b): reduction to approximate inconsistency calculation. Instead of assuming that we have direct access to the numbers $f(b)$ and $f(z)$ and can compare them in one step, we now adopt a weaker assumption, that we only have access to finite approximations to them. With this model of computation, it is not obvious that we can determine which of $f(b)$ and $f(z)$ is bigger—but fortunately, we do not need to. This is because, when $f(z)$ and $f(b)$ are close, p^* lies between z and b , and so both branches of the algorithm maintain the invariant $p^* \in [a, c]$. To simplify our analysis, we will default to keeping the “left” branch (with the smaller numbers), if the queried approximations to $f(z)$ and $f(b)$ are too close to determine whether one is larger than the other.

More precisely, the test “ $f(z) \ll f(b)$ ” is now shorthand for the following procedure:

- Run the inconsistency calculation procedure to obtain approximations to $f(z)$ and $f(b)$ that are correct to within

$$\epsilon' := \frac{1}{16\gamma} \log^2 \left(1 + 8\gamma(z-b)^2 \right). \quad (\text{This number comes from Lemma 9.4.8.}) \quad (9.8)$$

Call these approximations $\tilde{f}(z)$ and $\tilde{f}(b)$. By definition, they satisfy $|f(z) - \tilde{f}(z)| \leq \epsilon'$ and $|f(b) - \tilde{f}(b)| \leq \epsilon'$. If $|\tilde{f}(z) - \tilde{f}(b)| > \epsilon'$ (so that we know for sure which of $f(z)$ and $f(b)$ is larger based on these approximations), then return TRUE If $\tilde{f}(z) < \tilde{f}(b)$, and FALSE otherwise.

- On the other hand, if $|\tilde{f}(z) - \tilde{f}(b)| \leq \epsilon'$, return TRUE if $z < b$ and FALSE if $b > z$.

The remainder of the proof of correctness demonstrates that this level of precision is enough to never mistakenly eliminate the branch containing p^* .

We begin by proving a series of three additional invariants about the values (a, b, z, c) in each iteration, which are required for our analysis. The first property is that b and z are not too close to the boundary or each other.

Claim 9.4.9.2. *At the beginning of each iteration, $b \in [\frac{\epsilon}{2}, 1 - \frac{\epsilon}{2}]$, $z \in [\frac{\epsilon}{4}, 1 - \frac{\epsilon}{4}]$, and $|b - z| \geq \frac{\epsilon}{4}$.*

We prove this by contradiction. Initially, $b = \frac{1}{2}$ so it's neither the case that $b < \frac{\epsilon}{2}$ nor that $b > 1 - \frac{\epsilon}{2}$ for any $\epsilon < 1$. (The procedure terminates immediately if $\epsilon \geq 1$.) In search of a contradiction, suppose that either $b < \frac{\epsilon}{2}$ or $b > 1 - \frac{\epsilon}{2}$ later on. Specifically, suppose it first occurs in the $(t + 1)^{\text{st}}$ iteration, and let $(a_{t+1}, b_{t+1}, c_{t+1})$ to refer to the values of (a, b, c) in that iteration. Let (a_t, b_t, z_t, c_t) denote the values of the variables in the previous iteration. We know that $b_{t+1} \notin [\frac{\epsilon}{2}, 1 - \frac{\epsilon}{2}]$ and $b_t \in [\frac{\epsilon}{2}, 1 - \frac{\epsilon}{2}]$. In particular, $b_t \neq b_{t+1}$, which means the procedure cannot have taken the second or fourth branches in the t^{th} iteration. There are two remaining cases, corresponding to the first and third branches.

- **(branch 1)** In this case, $b_t - a_t \geq c_t - b_t$ and $z_t = (a_t + b_t)/2$. Furthermore, as a result of the assignment in this branch, we have $c_{t+1} = b_t$ and $b_{t+1} = z_t = (a_t + b_t)/2$.

– If $b_{t+1} < \frac{\epsilon}{2}$, this means $a_t + b_t < \epsilon$. As $a_t \geq 0$, this implies $b_t = c_{t+1} < \epsilon$.

But then $|c_{t+1} - a_{t+1}| \leq c_{t+1} < \epsilon$, so the algorithm must have already

terminated! This is a contradiction.

- If $b_{t+1} > 1 - \frac{\epsilon}{2}$, then $1 - \frac{\epsilon}{2} < b_{t+1} = z_t = (a_t + b_t)/2 < b_t$, contradicting our assumption that $b_t \leq 1 - \frac{\epsilon}{2}$.
- **(branch 3)** In this case, $c_t - b_t > b_t - a_t$ and $z_t = (b_t + c_t)/2$. The assignment at the end of this branch ensures that $a_{t+1} = b_t$ and $b_{t+1} = z_t$.
 - If $b_{t+1} < \frac{\epsilon}{2}$, then $b_t = a_{t+1} < b_{t+1} < \frac{\epsilon}{2}$. which is a contradiction.
 - If $b_{t+1} = (b_t + c_t)/2 > 1 - \frac{\epsilon}{2}$, then, since $c_t \leq 1$, we know $b_t + 1 > 2 - \epsilon$, so $b_t = a_{t+1} > 1 - \epsilon$. But now $|c_{t+1} - a_{t+1}| \leq 1 - a_{t+1} < 1 - (1 - \epsilon) = \epsilon$. So the algorithm must have already terminated.

Thus, it cannot be the case that $b < \frac{\epsilon}{2}$ or $b > 1 - \frac{\epsilon}{2}$ in any iteration of the algorithm. The fact that $z \in [\frac{\epsilon}{4}, 1 - \frac{\epsilon}{4}]$ follows immediately from the definition of z in either branch. Finally, $|z - b| = \frac{1}{2} \max\{c - b, b - a\} \geq \frac{1}{2}(\frac{c-a}{2}) > \frac{\epsilon}{4}$. This completes the proof of ?? 9.4.9.2. \square

Claim 9.4.9.3. *It is always the case that $b \in [\frac{2a+c}{3}, \frac{a+2c}{3}]$.*

We prove this by induction. It is clearly true at initialization; suppose it is also true at time t , i.e., $\frac{2a_t+c_t}{3} \leq b_t \leq \frac{a_t+2c_t}{3}$. We now show the same is true at time $t + 1$ in each of the four cases of the algorithm.

- **(branch 1)** At the end, we assign $a_{t+1} = a_t$, $b_{t+1} = z = \frac{a_t+b_t}{2}$, and $c_{t+1} = b_t$. So,

$$\frac{2a_{t+1}+c_{t+1}}{3} = \frac{2a_t+b_t}{3} < \underbrace{\frac{a_t+b_t}{2}}_{=b_{t+1}} < \frac{a_t+2b_t}{3} = \frac{a_{t+1}+2c_{t+1}}{3}.$$

- **(branch 2)** In this case, we must make use of the fact that, in the first two branches $b - a \geq c - b$, meaning $a_t + c_t \leq 2b_t$. As in the first branch, we have $z = \frac{a_t + b_t}{2}$. This time, however, $a_{t+1} = z = \frac{a_t + b_t}{2}$, $b_{t+1} = b_t$, and $c_{t+1} = c_t$. Thus, we find

$$\begin{aligned}\frac{2a_{t+1} + c_{t+1}}{3} &= \frac{a_t + b_t + c_t}{3} \leq \frac{(2b_t) + b_t}{3} = b_t \\ &= b_{t+1} \leq \frac{a_t + 2c_t}{3} < \frac{\frac{a_t + b_t}{2} + 2c_t}{3} = \frac{a_{t+1} + 2c_{t+1}}{3}.\end{aligned}$$

- **(branch 3)** Symmetric with branch 1. Concretely, $a_{t+1} = b_t$, $b_{t+1} = z = \frac{b_t + c_t}{2}$, and $c_{t+1} = c_t$. Thus,

$$\frac{2a_{t+1} + c_{t+1}}{3} = \frac{2b_t + c_t}{3} < \underbrace{\frac{b_t + c_t}{2}}_{= b_{t+1}} < \frac{b_t + 2c_t}{3} = \frac{a_{t+1} + 2c_{t+1}}{3}.$$

- **(branch 4)** Symmetric with branch 2. Concretely, $a_{t+1} = a_t$, $b_{t+1} = b_t$, $c_{t+1} = z = \frac{b_t + c_t}{2}$, and we know $2b_t < a_t + c_t$. Thus,

$$\begin{aligned}\frac{2a_{t+1} + c_{t+1}}{3} &= \frac{2a_t + \frac{c_t + b_t}{2}}{3} < \frac{2a_t + c_t}{3} \leq b_t \\ &= b_{t+1} = \frac{2b_t + b_t}{3} < \frac{(a_t + c_t) + b_t}{3} = \frac{a_{t+1} + 2c_{t+1}}{3}.\end{aligned}$$

The final result we need is a bound for Lemma 9.4.8.

Claim 9.4.9.4. $|\log \frac{z}{b} \frac{1-b}{1-z}| \leq \log 4$ ($< \sqrt{2}$).

Proof. Let $\bar{a} := 1-a$, $\bar{b} := 1-b$, and $\bar{c} := 1-c$. ?? 9.4.9.3 tells us that $b \geq \frac{2a+c}{3} \geq \frac{c}{3}$, and also that $b \leq \frac{a+2c}{3}$, which gives us a symmetric fact: $\bar{b} = 1-b \geq \frac{3}{3} - \frac{a}{3} - \frac{2c}{3} = \frac{\bar{a}+2\bar{c}}{3} \geq \frac{\bar{a}}{3}$. Consider two cases, corresponding to the two definitions of z . Either $z = (a+b)/2$ or $z = (b+c)/2$.

Case 1. $\frac{a+b}{2} = z < b$. Thus,

$$\begin{aligned} \left| \log \frac{z}{b} \frac{1-b}{1-z} \right| &= \log \frac{b}{z} \frac{1-z}{1-b} \\ &= \log \frac{2b}{a+b} \frac{1-\frac{a+b}{2}}{1-b} \\ &= \log \frac{b}{a+b} \frac{2-a-b}{1-b} \\ &= \log \frac{b}{a+b} + \log \frac{\bar{a}+\bar{b}}{\bar{b}} \\ &\leq \log \frac{\bar{a}+\bar{b}}{\bar{b}} \quad [\text{as first term is negative}] \\ &= \log \left(1 + \frac{\bar{a}}{\bar{b}} \right) \\ &\leq \log \left(1 + \frac{\bar{a}}{(\bar{a}/3)} \right) = \log 4. \end{aligned}$$

□

Case 2. $\frac{b+c}{2} = z > b$. Thus,

$$\begin{aligned} \left| \log \frac{z}{b} \frac{1-b}{1-z} \right| &= \log \frac{b}{z} \frac{1-b}{1-z} \\ &= \log \frac{b+c}{2b} \frac{1-b}{1-\frac{b+c}{2}} \\ &= \log \frac{b+c}{b} \frac{1-b}{2-b-c} \\ &= \log \frac{b+c}{b} + \log \frac{\bar{b}}{\bar{b}+\bar{c}} \\ &\leq \log \frac{b+c}{b} \quad [\text{as second term is nega-}] \\ &= \log \left(1 + \frac{c}{b} \right) \\ &\leq \log \left(1 + \frac{c}{(c/3)} \right) = \log 4. \end{aligned}$$

We are now in a position to prove that we never mistakenly eliminate p^* when comparing truncated representations. Without loss of generality, suppose that $z > b$, as the two cases are symmetric. Since we choose the left branch in the event of a tie, we have made a mistake if we instead needed to have chosen the right branch: $p^* > z$. In search of a contradiction, suppose that indeed this is the case. Under these conditions (i.e., $b < z < p^*$), and in light of ?? 9.4.9.4, we can apply Lemma 9.4.8 with $k = \sqrt{2} > \ln 4$, which tells us that

$$f(b) - f(z) > \frac{1}{16\gamma} \log^2 \left(1 + 8\gamma(z-b)^2 \right).$$

The definition of ϵ' in (9.8) is constructed precisely to make sure that this is never true. Therefore, the algorithm cannot choose the wrong branch.

Complexity Analysis. We now provide a more careful analysis of the runtime of the algorithm. We already have a bound on the number of iterations required; what is missing is a bound on how long it takes to compute \ll , i.e., to compare the approximations $\tilde{f}(z)$ and $\tilde{f}(b)$. Assuming these numbers are in binary format,

$\tilde{f}(z)$ is of the form $A.B$, and $\tilde{f}(b)$ is of the form $A'.B'$, where $\{A, A', B, B'\}$ are binary sequences.

Without loss of generality, assume that $|A| \leq |A'|$. (Otherwise, swap their labels.) The complexity of comparing the two numbers $\tilde{f}(z)$ and $\tilde{f}(b)$ does not depend on $|A'|$, the longer of the two sequences to the left of the radix point. This is because once we see the radix point in one number but not the other, we can immediately conclude the former is smaller. In the first iteration, $|A|$ is at most

$$\begin{aligned} |A| &\leq \max \left\{ 0, \log_2 \langle\!\langle \mathbf{m} + \Pr(Y=y)=\frac{1}{2} \rangle\!\rangle_\gamma \right\} \leq \max \left\{ 0, \log \left(\langle\!\langle \mathbf{m} \rangle\!\rangle_\gamma + \mathbf{D}(p^* \parallel .5) \right) \right\} \\ &\leq \log_2 \left(\max \{0, \langle\!\langle \mathbf{m} \rangle\!\rangle_\gamma\} + 1 \right) \quad [\text{since } \mathbf{D}(p \parallel .5) \leq \\ &\in O(\log \langle\!\langle \mathbf{m} \rangle\!\rangle_\gamma). \end{aligned}$$

Furthermore, $|A|$ cannot increase as the algorithm progresses, because whichever of $\{z, b\}$ leads to a smaller value of f becomes the new value of b in the following iteration.

Next, we derive an upper bound on the number of bits of B and B' that we must compare. Taking the (base-2) logarithm of (9.8), we find that we need to examine at most

$$\begin{aligned} |B| &\leq 4 + \log_2(\gamma) - 2 \log_2 \log(1 + 8\gamma(z - b)^2) \\ &\leq 4 + \log_2(\gamma) - 2 \log_2 \log \left(1 + \frac{1}{2}\gamma\epsilon^2 \right) \quad [\text{since } (z - b) \geq \frac{\epsilon}{4}] \end{aligned}$$

bits to the right of the radix point in order to eliminate the possibility that $f(b) < f(z)$. This expression is still not very friendly; we now loosen it even further to provide a bound of a simpler, more recognizable form. When $x \geq 0$, we know that $\log(1 + x) \geq 1 - \frac{1}{1+x} = \frac{x}{x+1}$; it follows that $-\log_2 \log(1 + x) \leq$

$-\log_2\left(\frac{x}{x+1}\right) = \log_2(1 + \frac{1}{x})$. Thus,

$$\begin{aligned} |B| + |A| &\leq 4 + \log_2(\gamma) + 2\log_2\left(1 + \frac{2}{\gamma\epsilon^2}\right) + \log_2(\langle\!\langle m \rangle\!\rangle_\gamma + 1) \\ &\in O\left(\log\frac{1}{\epsilon} + |\log\gamma| + \log\langle\!\langle m \rangle\!\rangle_\gamma\right). \end{aligned}$$

Recall that the process takes at most $O(\log \frac{1}{\epsilon})$ iterations—but in the process, produces the same number of bits of output, since $\log(\frac{1}{\epsilon})$ is the number of bits need to encode the final approximation to p^* . So, accounting for the time needed to compare $f(z)$ and $f(b)$, the algorithm runs in time

$$O\left(\log\frac{1}{\epsilon} \cdot \left(\log\frac{1}{\gamma} + \log\frac{1}{\epsilon} + \log\langle\!\langle m \rangle\!\rangle_\gamma\right)\right).$$

At this point, we have shown reduced unconditional inference to inconsistency calculation. To extend the reduction to conditional queries, we can apply Lemma 8.A.7 with $k = 2$, $K_0 = 0$, $K_1 = \log\frac{1}{\gamma} + \log(1 + \langle\!\langle m \rangle\!\rangle_\gamma)$, $K_2 = 1$, and $\Phi = 1$, to get an algorithm that runs in

$$O\left(\left(\log\frac{1}{\gamma} + \log\langle\!\langle m \rangle\!\rangle_\gamma\right) \cdot \log\frac{1}{\epsilon\mu^*(X=x)} + \log^2\frac{1}{\epsilon\mu^*(X=x)}\right) \text{ time.}$$

It uses $O(\log\log\frac{1}{\mu^*(X=x)} \cdot \log\frac{1}{\epsilon})$ calls to the inconsistency calculation procedure.

Finally, we remark that, often we are only interested in doing inference up to the precision that is tracked by a typical computer. In this case, by selecting $\epsilon \leq 10^{-78}$, the procedure above runs in constant time, making at most 1555 inconsistency procedure calls before outputting the 64-bit float that is closest to p^* .

The other direction: reducing inconsistency calculation to inference. This reduction is much simpler, shares more techniques with the primary thrust of the paper. First find a tree decomposition $(\mathcal{C}, \mathcal{T})$ of the PDG's structure, and then

query the marginals of each clique. Because of the work we've already done, we know this information is enough information to simply evaluate the scoring function, including the joint entropy, by (8.13).

CHAPTER 10

REASONING WITH PDGS

10.1 Quantitative Monotonicity and Equivalence

10.2 Qualitative Monotonicity and Equivalence

10.2.1 QIM Equivalence

[now that we know this version of Theorem 2 is false, much of this discussion doesn't make sense.] Applying Theorem 5.2.2 with $\mathcal{A}' = \emptyset$ yeilds a quintessential special case: $\mu \models \boxed{X} \rightarrow \boxed{Y}$ iff Y is a function of X according to μ . At first glance, this already seems to capture the essence of Theorem 5.2.2; is it really meaningfully weaker? In fact it is; to illustrate, our next example is another graph that behaves the same way—but not in all contexts.

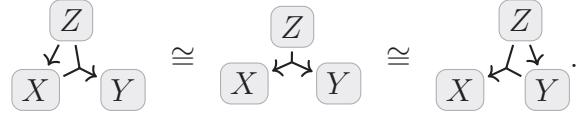
Example 16. In the appendix, we prove $\mu \models \boxed{X} \rightarrow \boxed{Y} \leftarrow$ iff Y is a function of X (according to μ). But, in general, this graph says something distinct from (and stronger than, as we will see in Section 5.B) the example above. After an adding the hyperarc $\emptyset \rightarrow \{X\}$ to both graphs, for example, they behave differently: every distribution μ satisfying $X \twoheadrightarrow Y$ also satisfies $\mu \models \rightarrow \boxed{X} \not\rightarrow \boxed{Y}$, but only when Y is a constant can it be the case that $\mu \models \rightarrow \boxed{X} \rightarrow \boxed{Y} \leftarrow$. \triangle

To distinguish between hypergraphs that are not interchangeable, we clearly need a stronger notion of equivalence.

Given hypergraphs \mathcal{A}_1 and \mathcal{A}_2 , we can form the combined hypergraph $\mathcal{A}_1 + \mathcal{A}_2$

that consists of the disjoint union of the two sets of hyperarcs, and the union of their nodes. We say that \mathcal{A} and \mathcal{A}' are (*structurally*) *equivalent* ($\mathcal{A} \cong \mathcal{A}'$) if for every context \mathcal{A}'' and distribution μ , we have that $\mu \models \mathcal{A} + \mathcal{A}''$ iff $\mu \models \mathcal{A}' + \mathcal{A}''$. By construction, structural equivalence (\cong) is itself invariant to additional context: if $\mathcal{A} \cong \mathcal{A}'$ then $\mathcal{A} + \mathcal{A}'' \cong \mathcal{A}' + \mathcal{A}''$. Our next result is a simple, intuitive, and particularly useful equivalence.

Proposition 10.2.1. *The following hypergraphs are equivalent:*



These three hypergraphs correspond, respectively, to equivalent factorizations of a conditional probability measure

$$P(X|Z)P(Y|X, Z) = P(X, Y|Z) = P(X|Y, Z)P(Y|Z).$$

Proposition 10.2.1 provides a simple and useful way to relate QIM-compatibility of different hypergraphs. If we restrict to acyclic structures, for instance, we find:

Theorem 10.2.2 (Chickering [15]). *Any two qualitative Bayesian Networks that represent the same independencies can be proven equivalent using only instances of Proposition 10.2.1 (in which X, Y, Z may be sets of variables).*

Theorem 10.2.2 is essentially a restatement of main result of Chickering [15], but it is simpler to state in terms of directed hypergraph equivalences. To state the result in its original form, one has to first define an edge $X \rightarrow Y$ to be *covered* in a graph G iff $\text{Pa}_G(Y) = \text{Pa}_G(X) \cup \{X\}$; then, the result states that all equivalent BN structures are related by a chain of reversed covered edges. Observe that this notion of covering is implicit in Theorem 10.2.2. Theorem 10.2.2

is one demonstration of the usefulness of [Proposition 10.2.1](#), but the latter applies far more broadly, to cyclic structures and beyond. It becomes even more useful in tandem with the definition of monotonicity presented in [Section 5.B](#), which is an analogue of implication.

Part IV

Foundations

CHAPTER 11
CONFIDENCE

CHAPTER 12
RELATIVE ENTROPY SOUP

CHAPTER 13

THE CATEGORY THEORY OF PDGS

A Categorical Definition of a PDG

Note that \mathcal{V} is implicit in \mathbb{P} . The two can be expressed as a functor, which is arguably the most compact definition of an (unweighted) PDG. An *unweighted PDG* $\langle \mathcal{V}, \mathcal{P} \rangle$ over a structure $(\mathcal{N}, \mathcal{A})$ is just a functor

$$\mathbb{P} : \mathcal{A}^* \rightarrow \text{Stoch}$$

whose action on objects \mathcal{N} is $X \mapsto \mathcal{V}X$, and whose action on the generating morphisms $X \xrightarrow{a} Y \in \mathcal{A}$ is written $\mathbb{P}_a(Y|X)$. We drop the symbol \mathcal{V} in this context, using only the symbol \mathbb{P} , because \mathcal{V} can be recovered by the action on the identity morphisms of \mathcal{A}^* . Given small category J (such as the free category generated by a graph), a functor $F : J \rightarrow \mathcal{C}$ is often called a *diagram* of \mathcal{C} (of shape J). Therefore, an unweighted PDG is a diagram of the Stoch , of shape generated by its underlying hypergraph. In addition to probabilities, a PDG also contains confidences $\beta = \{\beta_a\}_{a \in \mathcal{A}}$ about the reliability of those probabilities.

We now pursue a clean categorical picture of quantitative PDGs. At a quantitative level, positive structural weights can be captured by negative observational weights. This is because the gradient of $-\hat{\nabla}_\mu H_\mu(Y|X)$, the gradient of the structural loss corresponding to a hyperarc $X \xrightarrow{a} Y$, is the same as $+\hat{\nabla} D(\mu(X, Y) \| \mu(X)\lambda_Y(Y))$, the gradient of the observational loss corresponding to a uniform distribution. Furthermore, the weight β_a may be absorbed into the cpd \mathbb{P}_a by dropping the requirement that measures be normalized. This is because the pair $(p(Y|X), \beta)$ as a can be encoded¹ as a single conditional measure

¹However, there is no way to combine β with p that results in a quantity q (independent of μ)

$(1 - e^{-\beta})p(Y|X)$ losslessly, because $p(Y|X)$ can be reconstituted by renormalizing, and $\beta = -\log(1 - k)$ can be recovered from the normalization constant k . The only exception is when $\beta = 0$, but in this case the cpd does not matter semantically, and so if anything it is a bonus that this representation identifies all cpds supplied with confidence $\beta = 0$.

Furthermore, with this representation, the effect of composition is very compelling. Suppose we compose $p(Y|X)$ with confidence β_1 with $q(Z|Y)$ with confidence β_2 , where both $\beta_1, \beta_2 \in [0, \infty]$. Then the composite

$$\begin{aligned} r(Z|X) &= \int_Y (1 - e^{-\beta_2})p(Z|Y)(1 - e^{-\beta_1})dp(Y|X) \\ &= (1 - e^{-\beta_1} - e^{-\beta_2} + e^{-\beta_1-\beta_2}) (q \circ p)(Z|X) \\ &\approx (1 - \exp(-\min\{\beta_1, \beta_2\})) (q \circ p)(Z|X). \end{aligned}$$

In particular, the composite will be fully trusted iff both components are $\beta_1 = \beta_2 = \infty$, and if either has confidence zero, then the composite will also.² As a result, all data in a PDG $(\mathcal{A}, \mathcal{N}, \alpha, \mathcal{V}, \mathbb{P}, \beta)$ may be specified together with a single functor

$$m : \mathcal{A}^* \rightarrow \mathbb{M}\text{eas}_\Delta. \quad (13.1)$$

In other words, a PDG is a *diagram*, in the usual categorical sense, of conditional (sub)distributions between measurable spaces.

This connection induces a number of category-theory flavored questions about PDGs:

that can be plugged directly into the ordinary expression for KL divergence:

$$\beta \log \frac{\mu}{p} = \log \frac{\mu}{q} \implies q = \mu^{1-\beta} p^\beta.$$

[Zhu & Rower] suggest that this is not the appropriate notion of relative entropy, for unsigned measures; instead, one should use $D(\mu \parallel \nu) := \int \log \frac{d\mu}{d\nu} d\mu + \int d\mu - \int d\nu$, but even with this

²Keep in mind that, even if $p(Y|X)$ and $q(Z|Y)$ are both marginals of a shared distribution $\mu(X, Y, Z)$, and this is known with extreme confidence, their composite will only be correct if the information is somehow “independent”. This is where I think α should enter the picture, ideally.

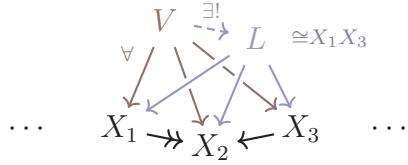
1. A PDG $m : \mathcal{A}^* \rightarrow \text{Meas}_{\Delta}$ is a diagram in the category of subprobability distributions. When does it have a limit? What about a colimit? What do limits and colimits of PDGs mean?
2. If PDGs are functors, what are the natural transformations between them? What do they mean?
3. How does inconsistency arise in this categorical picture?
4. Can we study qualitative PDGs separately in this picture? Why are α combined with β if the former are purely qualitative?
5. PDGs can be given semantics in more than one way, in principle — relative entropy is a natural choice, but, even then, it can be used in either direction. Yet this functorial definition of a PDG does not contain this information. So is there any way it can possibly interact with relative entropy that defines the semantics? If so, what is the categorical picture of the role of relative entropy?
 - For general loss functions (e.g., reverse KL), how does this picture interact with confidence functions?

13.0.1 Limits

Definition 13.0.1. Let m be a PDG, with variables \mathcal{X} . The *local marginal polytope*

$$\mathbb{L}(m) := \left\{ \{\mu_X \in \Delta \mathcal{V}X\}_{X \in \mathcal{X}} \mid \forall S \xrightarrow{a} T \in \mathcal{A}. \mu_T = \mathbb{P}_a \circ \mu_S \right\}. \quad (13.2)$$

consists of all marginals over the variables \mathcal{N} that are locally consistent with all arcs. \square



Example 17. 1. Suppose $\mathcal{A} = \{\rightarrow X\}$, and its unique arc is

△

Our next results are sensitive to the particulars of the PDG encoding as a functor. Let $\mathbf{m}^{+\mathcal{X}} := \mathbf{m} \cup \{\mathbf{X} \rightarrow \mathbf{Y}\}_{\mathbf{Y} \subset \mathbf{Y} \subset \mathcal{X}}$ be the PDG \mathbf{m} augmented with additional structure describing the relationships between all subsets of variables. That is, \mathcal{A}^* , as the free hypergraph over nodes $2^{\mathcal{N}}$, complete with structural coherence maps.³ Let \mathbf{m}^{hold} be the PDG where $p(\mathbf{Y}|\mathbf{X})$ is really attached to a hyperarc $\mathbf{X} \rightarrow \mathbf{X} \cup \mathbf{Y}$, implicitly the identity along $\mathbf{X} \setminus \mathbf{Y}$. Write $\mathbf{m}^{+\mathcal{X}, hold}$ for the PDG with both alterations.

Theorem 13.0.1. Suppose \mathbf{m} is a PDG in which every arc has full confidence. Then,

1. $\text{Cones}(\mathbf{m}, 1) \cong \mathbb{L}(\mathbf{m})$;
2. $\text{Cones}(\mathbf{m}^{+\mathcal{X}, hold}, 1) \cong \{\mathbf{m}\}$.
3. $\lim \mathbf{m}^{+\mathcal{X}, hold} = \text{Ext}\{\mathbf{m}\}$, where $\text{Ext}(S)$ is the set of extreme points of a set S (e.g., the vertices of S , when S is a polytope).
4. $\lim \mathbf{m} = \text{Ext } \mathbb{L}(\mathbf{m})$.

Proof. Part 1 is immediate; it just points out that the local marginal polytope, defined in the graphical models literature, is the limit in this context.

³Note that these coherence maps do not include hyperarcs on these variables to put them back together, e.g., $\{\{X\}, \{Y\}\} \rightarrow \{X, Y\}$. Including such a map is appropriate only if one believes X and Y are independent.

Now for part 2. A cone over $\mathbf{m}^{+\mathcal{X},\text{hold}}$ with vertex 1 is a collection of distributions $\{\mu_X(X)\}_{X \in \mathcal{X}}$ such that, for all $S \xrightarrow{a} T \in \mathcal{A}$, $\mu_T(S, T) = \mathbb{P}_a(T|S)\mu_S(S)$. (This familiar notation is not problematic if $S \cap T = \emptyset$, but otherwise we mean $\mu_T(S, T) = \int_S \mathbb{P}_a(S, T|s')\mu_S(s') ds'$, properly overwriting variables in S according to p). In particular, $\mathbf{m}^{+\mathcal{X},\text{hold}}$ has downprojections, so the cone data must satisfy $\mu_{\mathbf{Y}}(\mathbf{Y}) = \mu_{\mathbf{X}}(\mathbf{Y})$ whenever $\mathbf{Y} \subseteq \mathbf{X} \subseteq \mathcal{X}$. In particular, this means all variables are determined by the particular marginal $\mu_{\mathcal{X}}$, pointing to the joint variable \mathcal{X} , which is present in $\mathbf{m}^{+\mathcal{X}}$ and $\mathbf{m}^{+\mathcal{X},\text{hold}}$. Such a distribution (and its induced marginals) creates a cone over 1 only if it matches the appropriate conditional probability distributions for these other arcs. When $S_a \cap T_a = \emptyset$ for all a , that corresponds precisely to the requirement that μ matches all of the conditional marginals of \mathbb{P} (i.e., $\mu \in \{\mathbf{m}\}$). On the other hand, if $S_a \cap T_a \neq \emptyset$ for some a , e.g., for a self loop $p(X|X)$,

□

13.0.2 Colimits

The colimits are arguably even more interesting: they summarize the most general thing that is known by all variables.

There's always a co-cone with vertex 1, and there's a unique way to f

13.0.3 Natural Transformations

Suppose $\mathbf{m}_1, \mathbf{m}_2 : \mathcal{A}^* \rightarrow \mathbb{M}\mathbf{eas}_{\Delta}$ are two PDGs generated by the same (hyper)graph \mathcal{A} . What is a natural transformation $\eta : \mathbf{m}_1 \Rightarrow \mathbf{m}_2$?

By definition, it is a collection of stochastic maps $\{\eta_X : \mathcal{m}_1(X) \rightarrow \mathcal{m}_2(X)\}_{X \in \mathcal{N}}$,⁴ satisfying the property that, for all $a : S \rightarrow T \in \mathcal{A}$,⁵ the diagram

$$\begin{array}{ccc} \mathcal{m}_1(X) & \xrightarrow{m_1(a)} & \mathcal{m}_1(Y) \\ \downarrow \eta_X & & \downarrow \eta_Y \\ \mathcal{m}_2(X) & \xrightarrow{m_2(a)} & \mathcal{m}_2(Y) \end{array} \quad \left(\begin{array}{l} \text{or, in the} \\ \text{original no-} \\ \text{tation,} \end{array} \quad \begin{array}{ccc} \mathcal{V}_1 X & \xrightarrow{\mathbb{P}_a} & \mathcal{V}_1 Y \\ \eta_X \downarrow & & \downarrow \eta_Y \\ \mathcal{V}_2 X & \xrightarrow{\mathbb{P}_a^2} & \mathcal{V}_2 Y \end{array} \right)$$

commutes. This is a diagram in the category Meas_Δ . I immediately have questions:

1. What does the space of natural transformations from \mathcal{m}_1 to itself look like?
2. What if \mathcal{m}_1 and \mathcal{m}_2 differ only in β ? What is the effect of different encodings?
3. In what situations is there a natural transformation from one PDG into the other?
4. What about for certain special PDGs? What are some special PDGs with structure \mathcal{A} ?

Fix a PDG $\mathcal{m} : \mathcal{A}^* \rightarrow \text{Meas}_\Delta$. For a measurable space W , let Δ_W be the functor assigning each $N \in \text{ob } \mathcal{A}^*$ to the measurable space W , and each arc a to the identity map $\text{id}_W : W \rightarrow W$. Can we characterize the natural transformations from \mathcal{I} to \mathcal{m} ?

⁴Normally, we have been using the notation $\mathcal{V}_1(X)$ and $\mathcal{V}_2(X)$ for this concept, but for now we'll try this more traditional notation, and see if that works better.

⁵We have to verify this for all $a \in \mathcal{A}^*$, technically, but because \mathcal{A}^* is a free category, it suffices to check it only for the generating arcs $a \in \mathcal{A}$.

13.0.4 Additional Structure Preserved by PDGs

Monoidal Structure

If we allow unions of variables at the qualitative level, this means we are working in a different category \mathcal{A}^{**} whose objects $\text{ob } \mathcal{A}^{**}$ are all subsets of \mathcal{N} , and equipped with down-projection morphisms, and joining hyperarcs. Every PDG $m : \mathcal{A}^* \rightarrow \text{Meas}_{\Delta}$ can be naturally lifted to a PDG $m^{+\mathcal{X}} : \mathcal{A}^{**} \rightarrow \text{Meas}_{\Delta}$ in the obvious way: taking the additional joint variables to the appropriate product of measurable spaces, and treating their downprojections ($X \twoheadrightarrow Y$, for $X \supseteq Y$) appropriately. We already saw one consequence of this change: the limits of such a PDG must be internally coherent in a certain way; they represent not the local marginal polytope, but the global marginal polytope.

Variable Union as Monoidal Structure, for $\mathcal{A}^{*,hold}$ Usually people call the a monoidal operation “tensor”, but we will now define a monoidal operation that does not line up with the usual tensor product. On objects, which are sets of variables, \odot behaves like union, and on morphisms, it simply multiplies densities.

$$\begin{aligned} A \odot C &= A \cup C; \\ \odot(p(B|A), q(D|C)) &:= p(B|A) \cdot q(D|C) \\ &= (a_0, y, c_0) \mapsto \left((b_0, z, d_0) \mapsto p(b_0, z|a_0, y)q(d_0, z|c_0, y) \right). \end{aligned}$$

where z gives the values of the common variables $A \cap C$, so that $a = (a_0, z)$ and $c = (c_0, z)$.

If A and C are disjoint, as are B and D , then \odot coincides with the tensor

product \otimes . When they share variables, the resulting operation does something different: instead of having two distinct copies of each input and output, $p \odot q$ takes in only one copy of each shared sources, and produces a density over only one copy of their shared targets. For example $p(X) \odot p(X)$ is the subprobability $p(X)^2$. It is worth noting that a morphism $p(Y|X)$ is idempotent, in the sense that $p(Y|X) \odot p(Y|X)$ iff it is a deterministic function.

Although \odot is well-defined, it is not functorial in all cases,⁶ and hence inadmissible as the basis of a monoidal structure for the full category of stochastic morphisms . However, if we restrict to the subcategory of purely generative morphisms—that is, arcs a satisfying $T_a \supseteq S_a$ —then \odot becomes functorial.

CAREFUL! This messes up the types of composition! If $f : X \rightarrow Y$ is converted to $f' : X \rightarrow XY$ and $g : Y \rightarrow Z$ is converted to $g' : Y \rightarrow YZ$, then $g' \circ f'$ is not defined (because $XY \neq Y$) and so composition cannot proceed, without first including a forget/downprojection map!

There are also many other properties that we must verify in order to get a symmetric monoidal category; we now verify them.

- **Functoriality.** We need to show that

$$\left(\begin{array}{c} X_1 \\ f_1 \downarrow \\ X_2 \\ f_2 \downarrow \\ X_3 \end{array} \right)_p \odot \left(\begin{array}{c} Y_1 \\ g_1 \downarrow \\ Y_2 \\ g_2 \downarrow \\ Y_3 \end{array} \right)_q = \begin{array}{c} X_1 \cup Y_1 \\ \downarrow f_1 \odot g_1 \\ X_2 \cup Y_2 \\ \downarrow f_2 \odot g_2 \\ X_3 \cup Y_3 \end{array}.$$

⁶For example, $p(Y|X) \odot p(Y|X) = p^2(Y|X)$ which is a strict subprobability measure, while $(q \circ p)(Z|X)$ and $(r \circ p)(W|X)$ are both cpds, and $(q \circ p) \odot (r \circ p)$ will also be a cpd on $W \cup Z$, supposing that W and Z are disjoint. So it cannot be the case that $(q \circ p) \odot (q \circ p)$

As mentioned above, this is not true in general. But we have assumed that $X_1 \subseteq X_2 \subseteq X_3$ and $Y_1 \subseteq Y_2 \subseteq X_3$. To simplify notation, let's redefine $X_3 \leftarrow X_3 \setminus X_2$ and $X_2 \leftarrow X_2 \setminus X_1$. In this new notation, our goal becomes proving the commutativity of the following diagram:

$$\left(\begin{array}{c} X_1 \\ f_1 \downarrow \\ X_2 \cup X_1 \\ f_2 \downarrow \\ X_3 \cup X_2 \cup X_1 \end{array} \right) \odot \left(\begin{array}{c} Y_1 \\ g_1 \downarrow \\ Y_2 \cup Y_1 \\ g_2 \downarrow \\ Y_3 \cup Y_2 \cup Y_1 \end{array} \right) = \begin{array}{c} X_1 \cup Y_1 \\ \downarrow f_1 \odot g_1 \\ X_2 \cup X_1 \cup Y_2 \cup Y_1 \\ \downarrow f_2 \odot g_2 \\ X_3 \cup X_2 \cup X_1 \cup Y_3 \cup Y_2 \cup Y_1 \end{array} .$$

We now compute

$$\begin{aligned} (f_2 \circ f_1)(x''_1)(x_1, x_2, x_3) &= \iint_{X_2, X_1} f_1(x'_2, x'_1 | x''_1) f_2(x_3, x_2, x_1 | x'_2, x'_1) dx'_1 dx'_2 \\ &= \iint_{X_2, X_1} f_1(x'_2 | x''_1) f_2(x_3 | x_2) \delta(x_1) dx'_1 dx'_2 \end{aligned}$$

•

(Pre)additivity

13.0.5 The Category of PDGs

13.1 Dependency Graphs for Other Monads

The Big Questions:

1. How does the monadic view of composition (bind / multiply), which describes composition in the underlying Kleisli category, interact with the

“scoring function semantics”?

2. Is there an important shared feature among the monads T for which analogues of PDGs work out? To set up the scoring semantics, it seems we need
 - (a) a way to quantify “degree of functional dependence” along an arc $S \rightarrow T$, in the limit object $\lim \mathcal{M}$, and
 - (b) a way to quantify degree of between $T(X, Y, Z)$ and $X \rightarrow T(Y)$.

If there is an analogue of marginalization, then there is a map $T(X, Y, Z) \rightarrow T(X, Y)$, and there is a

3. Since most monads do not construct continuous geometry as nicely as the probability monad, relations are not continuous, there is no obvious analogue of parallel, symmetric, “mixture composition”. Even when we give a loss function semantics (which we do below) this does not correspond to an obvious computational picture in the same way.

13.1.1 Relational Dependency Graphs

We will use \mathcal{P} do denote the relational monad.

Scoring Function Semantics. Suppose that instead of mapping $X \xrightarrow{a} Y$ to a conditional probability $\mathbb{P}_a(Y|X)$, we instead map it to a relation $R_a(X, Y)$.

The analogue of a scoring function might operate on a universal reation

$U \subseteq \mathcal{V}\mathcal{X}$, and look something like:

$$Inc(U) = \sum_{X \xrightarrow{a} Y \in \mathcal{A}} \beta_a \left\| U(X, Y) - R_a(X, Y) \right\|_1$$

where $\left\| U(X, Y) - R_a(X, Y) \right\|_1 = \#\left\{ (x, y) \in \mathcal{V}(X, Y) \mid R(x, y) \Leftrightarrow \exists \mathbf{z}. U(x, y, \mathbf{z}) \right\}$,

with each $\beta_a \in \mathbb{N}$.

The analogue of a qualitative arrow $X \xrightarrow{a} Y$, indicating that one attribute determines another, also requires a scoring function. The fact that the argument to $IDef\mathcal{A}, \alpha$ is a joint distribution may not be critical, so long as we can find a suitable replacement notion of uncertainty along an arc. One possible analogue of conditional entropy might then be

$$\begin{aligned} H_U(Y|X) &= \log(\text{maximum # of possible values of } Y \text{ given } X) \\ &= \max_{x \in \mathcal{V}X} \log \#\{y \in \mathcal{V}Y \mid \exists \omega \in U. S_\omega = s, T_\omega = t\}. \\ &= \max_{x \in \mathcal{V}X} \log \#\{y \in \mathcal{V}Y \mid \exists \mathbf{z}. U(x, y, \mathbf{z})\}. \end{aligned} \tag{13.3}$$

This shares an important property with conditional entropy: it is zero iff the value of Y is determined by X in the relation. It is undefined when U is empty, and otherwise non-negative. The other important property of conditional entropy is monotonicity with respect to weakening.⁷

Altogether, the analogue of $IDef$ is

$$IDef\mathcal{A}^*(U) := \sum_{a \in \mathcal{A}} \max_{x \in \mathcal{V}X} \log \#\{y \in \mathcal{V}Y \mid \exists \mathbf{z}. U(x, y, \mathbf{z})\} - \log \#U$$

Here, as in the probabilistic case, there is an interpretation in terms of storage costs. Suppose U is fixed and known. The first term is the number of bits needed

⁷Clearly an extra target to a hyperarc (i.e., extending Y to $Y' = (Y, Z)$) can only make (13.3) larger. Perhaps less obviously: it also has the property that adding an extra source (i.e., extending X to $X' = (X, Z)$) can only reduce the value of (13.3). This is because if the maximum is over joint pairs (x, z) , then only the maximum number of $\{y : (x, y, z) \in U\}$ contribute, while all such z are amalgamated and make the number larger in the case of an existential quantifier.

to specify separately each target given the source (knowing that the result is in U), while the second is the number of bits needed to specify an element of U directly. The value is undefined iff $|U| = 0$ because, intuitively, it is impossible to specify a joint setting $\omega \in U$ if U is empty.

Now, for some examples.

Example 18. Suppose $\mathcal{A} = \{\rightarrow X, Y \leftarrow\}$. Then $IDef\mathcal{A}^*(U(X, Y)) \geq 0$ with equality iff $U(X, Y) = U_X(X) \bowtie U_Y(Y)$, for some unary relations $U_X(X)$ and $U_Y(Y)$. \triangle

More generally, it can be shown that, for target-partitinal hypergraphs \mathcal{A} without sources, the quantity $IDef\mathcal{A}^*(U)$ measures how far a joint relation U is from decomposing independently along the specified arcs.

Proposition 13.1.1. *Let $\{\mathcal{X}_1, \dots, \mathcal{X}_n\}$ be a partition of \mathcal{X} , and $\mathcal{A} = \{\rightarrow \mathcal{X}_i\}_{i=1}^n$ be the hypergraph consisting of a single hyperarc pointing to each partition, each without any sources. In this case, $IDef\mathcal{A}^*(U) \geq 0$, with equality iff $U = U_1 \bowtie \dots \bowtie U_n$ is the natural join of n subrelations $U_i \subseteq \mathcal{V}(\mathcal{X}_i)$. (In addition, in that case it must be that each $U_i = \prod_{\mathcal{X}_i}(U)$ is the projection of U onto the variables \mathcal{X}_i .)*

Let's turn to some more complicated examples. First, let's start with overlapping targets, and keep everything unconditional. It is easy to see that

$$IDef[\rightarrow X \leftarrow]^*(U(X)) = \log |U|,$$

which is defined when $U \neq \emptyset$, non-negative on this domain, and equal to zero if and only if $U(X)$ is a singleton. Thus, it captures determinism (almost) exactly the same way as in the probabilistic case. Now, for a more complicated example.

Example 19. In the probabilistic setting, all $\mu(X, Y)$ are compatible with the hypergraph $[\rightarrow X \rightarrow Y]$, intuitively because optimal codes for specifying $(x, y) \sim \mu$ directly take the same amount of information as optimal codes to first specify x , and then use a code dependent on x to specify y .

In this relational setting, the two are not always the same. If $U = \{(x, y) : x \in S, y = f(x)\}$, then $H_U(X) = \log |S|$ and $H_U(Y|X) = 0$, since there is always precisely 1 y for which $(x, y) \in U$. Thus $IDef[\rightarrow X \rightarrow Y]^*(U) = 0$. Under the cost-of-storage interpretation: this means when U can be generated by selecting a subset of S and applying a (known) function, then specifying a joint sample (x, y) requires the same number of bits as specifying x .

A generalization of this holds for what might be called “ k -multi-functional” relationships, where $k \geq 1$ is some natural number. Suppose $f : X \rightarrow Y^k$ produces k distinct values of Y for each $x \in X$. If $U = \{(x, y) : x \in S, y \in f(x)\}$, then $H_U(Y|X) = \log k$, but also $|U| = k|S|$. So again $IDef[\rightarrow X \rightarrow Y]^*(U) = 0$. This is because, again, it takes the same number of bits to first specify a value of X , and then use that value to specify a value of Y , as it does to specify (x, y) together. It can be shown that, as a function of non-empty relations U , the value of $IDef[\rightarrow X \rightarrow Y]^*(U)$ is non-negative and zero precisely if U is of the form described above.

This is because, when f may produce a variable number of points depending on x , $IDef[\rightarrow X \rightarrow Y]^*(U)$ will be positive overall. So, when x is such that $|f(x)|$ is maximal, then specifying first x and then the appropriate y , is less efficient than specifying (x, y) together. \triangle

More generally, for directed graphs, we have an analogue of a conditional

independence.

Definition 13.1.1. Suppose $A, B, C \subseteq \mathcal{X}$. In a relation $R(\mathcal{X})$, A and B are said to be conditionally independent given C (symbolically, $R \models A \perp\!\!\!\perp B \mid C$) iff

$$\forall(a, b, c) \in \mathcal{V}(A, B, C). \quad R(a, b, c) \iff (\exists a' \in \mathcal{V}(A). R(a', b, c) \wedge \exists b' \in \mathcal{V}(B). R(a, b', c)).$$

We write $A \perp\!\!\!\perp B$ to abbreviate $A \perp\!\!\!\perp B \mid \emptyset$, i.e., the special case where there are no given variables. \square

Proposition 13.1.2. $R(A) \models A \perp\!\!\!\perp A$

Proposition 13.1.3. Suppose A, B, C are sets of attributes. $R \models A \perp\!\!\!\perp B \mid C$ iff $R \models (A \setminus C) \perp\!\!\!\perp (B \setminus C) \mid C$.

Proof. First, we claim that, for all $a, b, c \in \mathcal{V}(A, B, C)$, we have that $R(a, b, c) \iff R(a[A \setminus C], b[B \setminus C], c)$. If $\{a, b, c\}$ agree on shared values, then $R(a[A \setminus C], b[B \setminus C], c)$ must equal $R(a, b, c)$, by definition. On the other hand, if $\{a, b, c\}$ do not agree on shared values, then $R(a, b, c) = 0$, and there are three possibilities for conflict. If this is because a and c conflict, then it is possible that $R(a[A \setminus C], b, c) = 1$. **contradicting our claim!**

Let $A' := A \setminus C$ and $B' := B \setminus C$.

Suppose $R \models A \perp\!\!\!\perp B \mid C$, meaning that for all a, b, c ,

$$\begin{aligned} R(a, b, c) &\iff \exists a'' \in \mathcal{V}(A). R(a'', b, c) \wedge \exists b'' \in \mathcal{V}(B). R(a, b'', c) \\ &\iff \exists a'' \in \mathcal{V}(A). R(a''[A \setminus C], b[B \setminus C], c) \wedge a''[A \cap C] = c[A \cap C] \\ &\quad \wedge \exists b'' \in \mathcal{V}(B). R(a[A \setminus C], b''[B \setminus C], c) \wedge b''[B \cap C] = c[B \cap C] \\ &\iff \exists a' \in \mathcal{V}(A'). R(a', b[B'], c) \wedge \exists b' \in \mathcal{V}(B'). R(a[A'], b', c). \end{aligned}$$

incomplete; possibly untrue

□

Proposition 13.1.4. $R(\mathcal{X}_1) \bowtie S(\mathcal{X}_2) \models \mathcal{X}_1 \perp\!\!\!\perp \mathcal{X}_2 \mid \mathcal{X}_1 \cap \mathcal{X}_2$.

Proof.

□

Proposition 13.1.5. If G is a directed acyclic graph, then $\text{IDef } \mathcal{A}_G^*(U) \geq 0$, with

Example 20. Now, consider the 2-cycle

△

The relationship between relations and probabilities. There is a map $\text{Supp}_X : \Delta X \rightarrow 2^X$ that takes a probability measure to its support set. In fact, it is a natural transformation

$$\begin{array}{ccc}
 & \Delta & \\
 & \Downarrow \text{Supp} & \\
 \text{FinSet} & \xrightarrow{\quad} & \text{Set} , \\
 & \Downarrow \text{Supp} & \\
 & \Delta X \xrightarrow{\delta f} \Delta Y & \\
 & \downarrow \text{Supp} & \downarrow \text{Supp} \\
 2^X & \xrightarrow{\bar{f}} & 2^Y
 \end{array}$$

since the diagram

commutes for all $f : X \rightarrow Y$,⁸ where $\bar{f}(S) = \{f(x) : x \in S\}$, often simply written as just \bar{f} to indicate the obvious extension of f itself to subsets of X , is the application of the functor $2^{(-)}$ on f .

((Can we use this to say something about how this interacts with IDef ? What about $H_\mu(Y|X)$ vs $H_{\text{Supp } \mu}(Y|X)$?))

⁸Proof: $y \in \text{Supp}(\delta f(\mu)) \iff y \in f^{-1}(\text{Supp}(\mu)) \iff y \in \bar{f}(\text{Supp}(\mu))$.

Part V

Conclusions

BIBLIOGRAPHY

- [1] Akshay Agrawal, Robin Verschueren, Steven Diamond, and Stephen Boyd. A rewriting system for convex optimization problems. *Journal of Control and Decision*, 5(1):42–60, 2018.
- [2] S.M. Aji and R.J. McEliece. The generalized distributive law. *IEEE Transactions on Information Theory*, 46(2):325–343, 2000. doi: 10.1109/18.825794.
- [3] Shun-ichi Amari. *Information geometry and its applications*, volume 194. Springer, 2016.
- [4] MOSEK ApS. *MOSEK Optimizer API for Python* 10.0.25, 2022. URL <https://docs.mosek.com/10.0/pythonapi/index.html>.
- [5] Riley Badenbroek and Joachim Dahl. An algorithm for nonsymmetric conic optimization inspired by mosek. *Optimization Methods and Software*, pages 1–38, 2021.
- [6] Christel Baier, Clemens Dubslaff, Holger Hermanns, and Nikolai Käfer. On the foundations of cycles in bayesian networks. In *Lecture Notes in Computer Science*, pages 343–363. Springer Nature Switzerland, 2022. doi: 10.1007/978-3-031-22337-2_17. URL https://doi.org/10.1007%2F978-3-031-22337-2_17.
- [7] A. Balke and J. Pearl. Probabilistic evaluation of counterfactual queries. In *Proc. Eleventh National Conference on Artificial Intelligence (AAAI '94)*, pages 200–207, 1994.
- [8] Sander Beckers, Joseph Y. Halpern, and Christopher Hitchcock. Causal models with constraints, 2023.

- [9] Umberto Bertele and Francesco Brioschi. *Nonserial dynamic programming*. Academic Press, Inc., 1972.
- [10] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Advanced Information Systems Engineering*, pages 387–402. Springer Berlin Heidelberg, 2013. doi: 10.1007/978-3-642-40994-3_25. URL https://doi.org/10.1007%2F978-3-642-40994-3_25.
- [11] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.
- [12] Hans L Bodlaender. A linear time algorithm for finding tree-decompositions of small treewidth. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pages 226–234, 1993.
- [13] Stephen P Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [14] Venkat Chandrasekaran, Nathan Srebro, and Prahladh Harsha. Complexity of inference in graphical models. *arXiv preprint arXiv:1206.3240*, 2012.
- [15] David Maxwell Chickering. A transformational characterization of equivalent bayesian network structures. *CoRR*, abs/1302.4938, 2013. URL <http://arxiv.org/abs/1302.4938>.
- [16] Christophe Chipot and Andrew Pohorille. Free energy calculations. *Springer Series in Chemical Physics*, 86:159–184, 2007.
- [17] Yoeng-Jin Chu. On the shortest arborescence of a directed graph. *Scientia Sinica*, 14:1396–1400, 1965.

- [18] Andrzej Cichocki and Shun-ichi Amari. Families of alpha beta and gamma divergences: Flexible and robust measures of similarities. *Entropy*, 12(6):1532–1568, 2010.
- [19] Bruno Courcelle. The monadic second-order logic of graphs. i. recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, 1990. ISSN 0890-5401. doi: [https://doi.org/10.1016/0890-5401\(90\)90043-H](https://doi.org/10.1016/0890-5401(90)90043-H). URL <https://www.sciencedirect.com/science/article/pii/089054019090043H>.
- [20] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, New York, 1991.
- [21] Joachim Dahl and Erling D Andersen. A primal-dual interior-point algorithm for nonsymmetric exponential-cone optimization. *Mathematical Programming*, 194(1):341–370, 2022.
- [22] A Philip Dawid. The well-calibrated bayesian. *Journal of the American Statistical Association*, 77(379):605–610, 1982.
- [23] Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.
- [24] Alexander Domahidi, Eric Chu, and Stephen Boyd. Ecos: An socp solver for embedded systems. In *2013 European Control Conference (ECC)*, pages 3071–3076, 2013. doi: 10.23919/ECC.2013.6669541.
- [25] Ran Duan, Hongxun Wu, and Renfei Zhou. Faster matrix multiplication via asymmetric hashing. *arXiv preprint*, 2022. doi: 10.48550/ARXIV.2210.10173. URL <https://arxiv.org/abs/2210.10173>.

- [26] DK Fadeev. Zum begriff der entropie einer endlichen wahrscheinlichkeitsschemas. *Arbeiten zur Informationstheorie I*. Deutscher Verlag der Wissenschaften, pages 85–90, 1957.
- [27] Leon Festinger. Cognitive dissonance. *Scientific American*, 207(4):93–106, 1962.
- [28] Roger Fletcher. *Practical methods of optimization*. John Wiley & Sons, 2013.
- [29] Karl Friston. The free-energy principle: a rough guide to the brain? *Trends in Cognitive Sciences*, 13(7):293–301, 2009.
- [30] Giorgio Gallo, Giustino Longo, Stefano Pallottino, and Sang Nguyen. Directed hypergraphs and applications. *Discrete Applied Mathematics*, 42(2):177–201, 1993. ISSN 0166-218X. doi: [https://doi.org/10.1016/0166-218X\(93\)90045-P](https://doi.org/10.1016/0166-218X(93)90045-P). URL <https://www.sciencedirect.com/science/article/pii/0166218X9390045P>.
- [31] Dan Geiger, Thomas Verma, and Judea Pearl. Identifying independence in bayesian networks. *Networks*, 20(5):507–534, 1990. doi: <https://doi.org/10.1002/net.3230200504>. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/net.3230200504>.
- [32] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.
- [33] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2014.
- [34] Michael Charles Grant. *Disciplined Convex Programming*. PhD thesis, Stanford University, December 2004. URL https://web.stanford.edu/~boyd/papers/pdf/mcg_thesis.pdf.

- [35] Aditya Grover and Stefano Ermon. Lecture notes in deep generative models. deepgenerativemodels.github.io/notes/, 2018.
- [36] J. Y. Halpern and S. Leung. Weighted sets of probabilities and minimax weighted expected regret: new approaches for representing uncertainty and making decisions. *Theory and Decision*, 79(3):415–450, 2015.
- [37] Joseph Y Halpern. Axiomatizing causal reasoning. *Journal of Artificial Intelligence Research*, 12:317–337, 2000.
- [38] Joseph Y Halpern. *Reasoning about uncertainty*. MIT press, 2017.
- [39] Joseph Y Halpern. *Reasoning About Uncertainty*. MIT press, 2017.
- [40] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, second edition, 2009.
- [41] David Heckerman, David Maxwell Chickering, Christopher Meek, Robert Rounthwaite, and Carl Kadie. Dependency networks for inference, collaborative filtering, and data visualization. *Journal of Machine Learning Research*, 1(Oct):49–75, 2000.
- [42] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. Beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.
- [43] Christopher Hitchcock. Causal Models. In Edward N. Zalta and Uri Nodelman, editors, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Summer 2024 edition, 2024.

- [44] James E. Matheson Howard, Ronald A. Influence diagrams. *Readings on the Principles and Applications of Decision Analysis*, pages 719–763, 1983.
- [45] Shruti Jadon. A survey of loss functions for semantic segmentation. In *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pages 1–7. IEEE, 2020.
- [46] Ryan James. (stumbling blocks) on the road to understanding multivariate information theory. Discrete Information Theory package documentation, 2018. URL <https://dit.readthedocs.io/en/latest/stumbling.html>.
- [47] Ryan G. James and James P. Crutchfield. Multivariate dependence beyond shannon information. *Entropy*, 19(10), 2017. ISSN 1099-4300. doi: 10.3390/e19100531. URL <https://www.mdpi.com/1099-4300/19/10/531>.
- [48] Edwin T Jaynes. Information theory and statistical mechanics. *Physical review*, 106(4):620, 1957.
- [49] R. C. Jeffrey. Probable knowledge. In I. Lakatos, editor, *International Colloquium in the Philosophy of Science: The Problem of Inductive Logic*, pages 157–185. North-Holland, Amsterdam, 1968.
- [50] Narendra Karmarkar. A new polynomial-time algorithm for linear programming. In *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing*, pages 302–311, 1984.
- [51] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [52] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

- [53] Varsha Kishore, Xiangyu Chen, Yan Wang, Boyi Li, and Kilian Q Weinberger. Fixed neural network steganography: Train the images, not the network. In *International Conference on Learning Representations*, 2021.
- [54] Ivan Kobyzev, Simon J.D. Prince, and Marcus A. Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):3964–3979, nov 2021. doi: 10.1109/tpami.2020.2992934. URL <https://doi.org/10.1109/tpami.2020.2992934>.
- [55] D. Koller and N. Friedman. *Probabilistic Graphical Models*. MIT Press, Cambridge, MA, 2009.
- [56] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT press, 2009.
- [57] F. R. Kschischang, B. J. Frey, and H. . Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, 2001.
- [58] Frank R Kschischang, Brendan J Frey, and H-A Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on information theory*, 47(2):498–519, 2001.
- [59] S. L. Lauritzen, A. P. Dawid, B. N. Larsen, and H.-G. Leimer. Independence properties of directed markov fields. *Networks*, 20(5):491–505, 1990. doi: <https://doi.org/10.1002/net.3230200503>. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/net.3230200503>.
- [60] Steffen L Lauritzen and David J Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems.

Journal of the Royal Statistical Society: Series B (Methodological), 50(2):157–194, 1988.

- [61] leonbloy (<https://math.stackexchange.com/users/312/leonbloy>). conditioning reduces mutual information. Mathematics Stack Exchange, 2015. URL <https://math.stackexchange.com/q/1219753>. URL:<https://math.stackexchange.com/q/1219753> (version: 2015-04-04).
- [62] Jianzhu Ma, Jian Peng, Sheng Wang, and Jinbo Xu. Estimating the partition function of graphical models using langevin importance sampling. In *Artificial Intelligence and Statistics*, pages 433–441. PMLR, 2013.
- [63] David J. C. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003.
- [64] Marc Mezard and Andrea Montanari. *Information, physics, and computation*. Oxford University Press, 2009.
- [65] Tom Minka. Divergence measures and message passing. Technical Report MSR-TR-2005-173, Microsoft Research, Cambridge, U.K., 2005.
- [66] In Jae Myung. Tutorial on maximum likelihood estimation. *Journal of mathematical Psychology*, 47(1):90–100, 2003.
- [67] Pavel Naumov and Brittany Nicholls. R.e. axiomatization of conditional independence, 2013.
- [68] Radford M Neal and Geoffrey E Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in Graphical Models*, pages 355–368. Springer, 1998.
- [69] Yu E Nesterov, Michael J Todd, and Yinyu Ye. Infeasible-start primal-dual

methods and infeasibility detectors for nonlinear programming problems.
Technical report, 1999.

- [70] Yurii Nesterov and Arkadii Nemirovskii. *Interior-Point Polynomial Algorithms in Convex Programming*. SIAM, 1994.
- [71] Frank Nielsen. Chernoff information of exponential families. *arXiv preprint arXiv:1102.2684*, 2011.
- [72] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019. URL <https://arxiv.org/abs/1912.01703>.
- [73] HP Patil. On the structure of k-trees. *Journal of Combinatorics, Information and System Sciences*, 11(2-4):57–64, 1986.
- [74] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, San Francisco, 1988.
- [75] J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, New York, 2000.
- [76] J Pearl and A Paz. Graphoids: A graphbased logic for reasoning about relevance relations. *advances in artificial intelligence*, vol. ii, 1987.
- [77] Judea Pearl. *Causality*. Cambridge university press, 2009.
- [78] Jason Rennie. On l2-norm regularization and the gaussian prior. 2003.

- [79] Alfr d R nyi. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, pages 547–561. University of California Press, 1961.
- [80] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR, 2014.
- [81] Oliver E Richardson. Loss as the inconsistency of a probabilistic dependency graph: Choose your model, not your loss function. *AISTATS ’22*, 151, 2022.
- [82] Oliver E Richardson and Joseph Y Halpern. Probabilistic dependency graphs. *AAAI ’21*, 2021.
- [83] Oliver E Richardson, Joseph Y Halpern, and Christopher De Sa. Inference in probabilistic dependency graphs. *UAI ’23*, 2023.
- [84] D. B. Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*, 66(5):688–701, 1974.
- [85] Bart Selman, David G Mitchell, and Hector J Levesque. Generating hard satisfiability problems. *Artificial intelligence*, 81(1-2):17–29, 1996.
- [86] Glenn R Shafer and Prakash P Shenoy. Probability propagation. *Annals of mathematics and Artificial Intelligence*, 2:327–351, 1990.
- [87] Michael Sipser. *Introduction to the Theory of Computation* (2nd ed.). Thomson Course Technology., second edition, 2006. ISBN 978-0-534-95097-2.
- [88] Anders Skajaa and Yinyu Ye. A homogeneous interior-point algorithm for

- nonsymmetric convex conic optimization. *Mathematical Programming*, 150(2):391–422, 2015.
- [89] Esteban G Tabak and Eric Vanden-Eijnden. Density estimation by dual ascent of the log-likelihood. *Communications in Mathematical Sciences*, 8(1):217–233, 2010.
- [90] Myron Tribus. Information theory as the basis for thermostatics and thermodynamics. 1961.
- [91] Tim Van Erven and Peter Harremos. Rényi divergence and kullback-leibler divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820, 2014.
- [92] Martin J Wainwright, Tommi S Jaakkola, and Alan S Willsky. Tree-based reparameterization framework for analysis of sum-product and related algorithms. *IEEE Transactions on information theory*, 49(5):1120–1146, 2003.
- [93] Martin J Wainwright, Michael I Jordan, et al. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305, 2008.
- [94] Qi Wang, Yue Ma, Kun Zhao, and Yingjie Tian. A comprehensive survey of loss functions in machine learning. *Annals of Data Science*, pages 1–26, 2020.
- [95] Paul L. Williams and Randall D. Beer. Nonnegative decomposition of multivariate information, 2010.
- [96] Peter M Williams. Bayesian regularization and pruning using a laplace prior. *Neural Computation*, 7(1):117–143, 1995.
- [97] Alan L Yuille and Anand Rangarajan. The concave-convex procedure. *Neural computation*, 15(4):915–936, 2003.