(ALMOST) AUTOMATING CAUSAL INFERENCE WITH TRACTABLE PROBABILISTIC CIRCUITS

Anonymous authors
Paper under double-blind review

ABSTRACT

Approximate inference can lead to wrong conclusions when working with causal effects. For exact causal inference, the community has started using Probabilistic Circuits (PCs). However, these PCs have only been shown to work on discrete domains and only for backdoor adjustment, and as we show, existing previous methods fail to represent the dependencies present in the data under general vtree structures. We therefore develop a general-purpose causal inference engine for PCs that cover continuous and mixed domains, general vtree structures, and identifiable interventional queries. To this end, we pinpoint where existing circuits fall short: their conditioning operation is not exactly evaluable for a broad class of estimands, and their construction loses the capacity to model dependencies on general vtree structures. We develop the tools that overcome both: a diversity-preserving circuit construction, and disjoint continuous leaves that extend exact inference to mixed domains. We introduce T-ID, a tractability-aware extension of the ID algorithm, whose soundness and completeness we prove. T-ID takes as input a query, a causal graph, and a circuit, and outputs identifiability, tractability relative to the weak marginal determinisms available in the circuit, and a compositional query with a stated complexity bound. Reproducible experiments on synthetic continuous and mixed structural causal models corroborate our theoretical predictions.

1 INTRODUCTION

Causal inference is concerned with predicting the behavior of a system under interventions, the simplest such prediction being the causal effect of an intervention: any contrast between interventional distributions of the same outcome under different interventions (Pearl, 2009). In practice, such effects are mostly computed approximately. Neural causal models (Xia et al., 2021), deep structural causal models (Pawlowski et al., 2020), and their amortized and implicit variants (Yang et al., 2021; Kocaoglu et al., 2018) specify the causal mechanisms as neural networks and sample from the interventional distribution by ancestral sampling. Such approximations can lead to wrong conclusions about causes and effects (Pearl, 2009), which is dangerous precisely in the high-stakes domains causal inference targets. Computing effects exactly requires an exact model of the interventional distribution. Causal normalizing flows (Khemakhem et al., 2021) are one exact candidate, but they are restricted to continuous variables, invertible mechanisms, and a known causal ordering, and are not connected to identification theory.

Probabilistic circuits (PCs) (Poon & Domingos, 2011; Choi et al., 2020) are tractable probabilistic models that answer a growing catalog of queries — marginals, conditionals, products, and their compositions — exactly and in polynomial time. Tractability is ensured by construction: the circuit must satisfy structural properties dictated by the query, so that, for example, marginal queries that are #P-hard on probabilistic graphical models (Cooper, 1990; Roth, 1996), become linear-time under decomposability and smoothness. Extensions give circuits interventional semantics — predicting parameters from the intervention (Zečević et al., 2021), composing counterfactual worlds (Busch et al., 2025), or using characteristic-function leaves (Poonia et al., 2024) — but presuppose interventional or counterfactual data, which is generally not available.

In what serves as the foundation of our work, Wang & Kwiatkowska (2023) model interventional distributions exactly using only observational data by representing the identified estimand of a causal

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

query in closed form. They achieve this by expressing the estimand as a compositional query over basic probabilistic operations — marginalization, multiplication, and reciprocals — and deriving, via their MD-calculus, the tractability conditions under which such compositions can be evaluated exactly; guided by the composition, a circuit is synthesized that models the interventional distribution exactly in closed form. However, the approach remains limited to discrete domains, is only shown to work on backdoor adjustment, and, as we show, only works with the specific vtree structure used in the paper.

In this work, we develop probabilistic circuits into general-purpose engines for exact causal inference over continuous and mixed domains.[1]

First, we show that the approach of Wang & Kwiatkowska (2023) cannot be extended beyond its original setting. Their conditioning operation, used but never formalized, is not exactly evaluable on their own circuits for a broad class of estimands, on any vtree. This class includes frontdoor adjustment and we characterize it exactly. Furthermore, their construction algorithm fails representationally on general vtrees, where even backdoor adjustment is out of reach. Second, we develop the tractability theory required to overcome both limitations. We formalize conditioning as a circuit operation along with the circuit property that makes it tractable, weak marginal determinism. On this basis we introduce T-ID, a tractability-aware extension of the ID algorithm, and prove its soundness and completeness: given a query, a causal graph, and a circuit, T-ID decides identifiability and tractability relative to the available weak marginal determinisms, and returns a compositional query with a stated complexity bound. Third, we resolve the representational failure through the notion of diversity. We prove that lack of diversity causes precisely this failure and give a construction algorithm that yields high-diversity circuits. Fourth, we extend the framework to continuous and mixed domains by designing disjoint continuous leaves that are expressive and satisfy the required leaf invariants: positivity, normalization, disjointness, and tractable CDFs and moments.

2 PRELIMINARIES

Notation. Uppercase letters (X), bold uppercase ($\mathbf{X}$), and lowercase (x, $\mathbf{x}$) denote variables, sets of variables, and their assignments; $\mathrm{Val}(\mathbf{X})$ is the set of assignments to $\mathbf{X}$. We write $P(\mathbf{V})$ for a distribution and $p(\mathbf{v})$ for its density over base measures μ_X (counting or Lebesgue; integrals reduce to sums), with marginals and conditionals defined by $p(\mathbf{w}) = \int p(\mathbf{v})\, d\mu(\mathbf{v} \setminus \mathbf{w})$ and $p(\mathbf{y} \mid \mathbf{x}) = p(\mathbf{y}, \mathbf{x})/p(\mathbf{x})$.

Definition 1. *A* probabilistic circuit *(PC) $\mathcal{C}$ over $\mathbf{V}$ is a rooted directed acyclic graph of* leaf*,* sum*, and* product *nodes. Each node α encodes a non-negative function p_α over its* scope *$\phi(\alpha) \subseteq \mathbf{V}$, the variables in the leaves below α; leaves encode tractable densities; an internal node computes a weighted sum, with weights $w_i \geq 0$, or a product over its children $\mathrm{ch}(\alpha)$. The circuit encodes the normalized density of its root. A PC is* smooth *if the children of every sum node share its scope and* decomposable *if the children of every product node have pairwise disjoint scopes; together these make marginals computable exactly in time linear in the circuit size (Poon & Domingos, 2011; Choi et al., 2020).*

For $\mathbf{Q} \subseteq \mathbf{V}$, the *restricted scope* of a node is $\phi_{\mathbf{Q}}(\alpha) := \phi(\alpha) \cap \mathbf{Q}$, its *support* is $\mathrm{supp}(\alpha) = \{\mathbf{x} \in \mathrm{Val}(\phi(\alpha)) : p_\alpha(\mathbf{x}) > 0\}$, and its *marginal support* over $\mathbf{Q}$ is $\mathrm{supp}_{\mathbf{Q}}(\alpha) = \{\mathbf{q} \in \mathrm{Val}(\mathbf{Q}) : p_\alpha(\mathbf{q}) > 0\}$, with $p_\alpha(\mathbf{q})$ the marginal density.

We analyze circuits in layered form, organized by a regular md-vtree (defined below) in an extended Einsum Network convention (Peharz et al., 2020): each internal vtree node m with children m_l, m_r induces a *sum layer* $S^{(m)}$ of γ_m *groups* of ν_m *nodes* $S^{(m)}_{ij}$; nodes at the same position j share a support, and each sum node multiplies nodes of the child layers $S^{(m_l)}$ and $S^{(m_r)}$, with the wiring and weights encoded in a non-negative tensor $w^{(m)}$ whose support defines the layer's product nodes. A *vtree* v is a rooted binary tree whose leaves carry subsets partitioning $\mathbf{V}$; a PC *respects* v if it is smooth and decomposable and every leaf and non-trivial sum node has the scope of a vtree node (*structured decomposability*) (Pipatsrisawat & Darwiche, 2008; Wang & Kwiatkowska, 2023), and

[1] Reference implementation:
`https://anonymous.4open.science/r/AlmostAutomatedCausalCircuits-93F4`

two vtrees are *compatible* if isomorphic on their common variables. The explicit contraction and full formalism are given in Appendix B.

Definition 2 (md-vtrees)**.** *An* md-vtree $w = (v, \psi)$ *(Wang & Kwiatkowska, 2023) augments a vtree with a label* $\psi(m) \in \mathcal{P}(\phi(m)) \cup \{\mathcal{U}\}$ *per node* m*, declaring the variables over which the sum nodes at position* m *must be deterministic: a sum node* α *with* $\phi(\alpha) = \phi(m)$ *is required to be* $\psi(m)$*-deterministic, so that fixing the variables in* $\psi(m)$ *selects at most one of its children;* $\mathcal{U}$ *marks the absence of any constraint. A PC* respects w *if it respects* v *and satisfies the constraint at every position;* w *is* regular *if every non-leaf node* m *satisfies* $\psi(m) \in \{\psi(m_l), \psi(m_r), \psi(m_l) \cup \psi(m_r)\}$.

The labels of a regular md-vtree induce a *type* for each layer: *synthesizing* if $\psi(m) = \psi(m_l) \cup \psi(m_r)$ and *left-mixing* if $\psi(m) = \psi(m_l)$ (Wang & Kwiatkowska, 2023); we additionally distinguish *universal* layers ($\psi(m) = \mathcal{U}$) and *pseudo-left-mixing* layers ($\psi(m) = \psi(m_l) \neq \mathcal{U}$ with $\psi(m_r) \neq \mathcal{U}$), reserving left-mixing proper for $\psi(m_r) = \mathcal{U}$.[2] The layer type fixes the determinism requirements on the children of its sum nodes, hence the tractability of operations applied to it.

A sum node is *marginally deterministic* with respect to $\mathbf{Q}$, or $\mathbf{Q}$*-deterministic*, if the marginal supports of its children over $\mathbf{Q}$ are pairwise disjoint (Choi et al., 2020; Wang & Kwiatkowska, 2023). A PC is $\mathbf{Q}$-deterministic if every sum node α with $\phi_{\mathbf{Q}}(\alpha) \neq \emptyset$ is.

Compositional queries. Following Vergari et al. (2021); Wang & Kwiatkowska (2023); Wang et al. (2024) we view queries as compositions of circuit operations — marginalization MARG, product PROD, conditional COND, and reciprocal POW; the MD-calculus (Wang & Kwiatkowska, 2023) derives, applied backward through a composition, sufficient requirements on the inputs. The backdoor estimand $P(Y \mid do(X)) = \int_Z P(Y \mid X, Z)\, P(Z)\, dZ$, for example, is $\mathrm{MARG}(\mathrm{PROD}(\mathrm{COND}(\mathcal{C}; \{X, Z\}), \mathrm{MARG}(\mathcal{C}; \{Z\})); \{Z\})$.

Definition 3 (Structural causal models)**.** *An SCM* $\mathcal{M} = \langle \mathbf{U}, \mathbf{V}, \mathcal{F}, P(\mathbf{U}) \rangle$ *comprises endogenous variables* $\mathbf{V}$*, exogenous noise* $\mathbf{U} \sim P(\mathbf{U})$*, and structural equations* $V_i = f_i(\mathrm{pa}_i, \mathbf{U}_i)$ *with* $\mathrm{pa}_i \subseteq \mathbf{V} \setminus \{V_i\}$*, inducing an acyclic dependency structure. It entails an observational distribution* $P_{\mathcal{M}}(\mathbf{V})$ *and is* Markovian *if the* $\mathbf{U}_i$ *are mutually independent,* semi-Markovian *otherwise.*

Each SCM induces a *causal diagram* $\mathcal{G}$, an acyclic directed mixed graph with $X \to Y$ if $X \in \mathrm{pa}_Y$ and $X \leftrightarrow Y$ if $\mathbf{U}_X$ and $\mathbf{U}_Y$ are dependent; an intervention $\mathrm{do}(\mathbf{x})$ replaces the equations of $\mathbf{X}$ by the constants $\mathbf{x}$, yielding the *interventional distribution* $P(\mathbf{y} \mid \mathrm{do}(\mathbf{x}))$. A query is *identifiable* if its value is uniquely determined by $P(\mathbf{V})$ and $\mathcal{G}$: any two SCMs sharing the diagram and the observational distribution agree on the query. Pearl's do-calculus (Pearl, 2009), sound and complete for identification (Shpitser & Pearl, 2006; Huang & Valtorta, 2006), manipulates interventional distributions, and the *ID algorithm* (Shpitser & Pearl, 2006) renders this constructive, returning an expression for $P(\mathbf{y} \mid \mathrm{do}(\mathbf{x}))$ as a functional of $P(\mathbf{V})$ or reporting non-identifiability.

2.1 Related Work

Approximate effect estimation. The dominant approach learns the mechanisms and approximates the interventional distribution: neural causal models (Xia et al., 2021) and deep structural causal models (Pawlowski et al., 2020) parametrize the equations with neural networks, causal effect VAEs (Louizos et al., 2017; Yang et al., 2021) amortize inference over latent noise, and causal GANs (Kocaoglu et al., 2018) and transformer estimators (Melnychuk et al., 2022) define the effect only implicitly or via Monte Carlo. These methods scale but answer approximately; causal normalizing flows (Khemakhem et al., 2021) answer exactly but assume continuous variables, invertible mechanisms, and a known causal ordering, with no link to identification.

Tractable probabilistic models. Sum-product networks (Poon & Domingos, 2011), their einsum reformulation (Peharz et al., 2020), and probabilistic circuits generally (Choi et al., 2020) answer queries exactly by construction, provided the circuit satisfies the structural properties the query dictates. Interventional extensions predict circuit parameters from the intervention (Zečević et al., 2021), compose counterfactual worlds (Busch et al., 2025), or encode mechanisms in characteristic-function leaves (Poonia et al., 2024); all require interventional or counterfactual training data, which is rarely available, and none addresses identification.

[2] By symmetry we only treat the (pseudo-)left-mixing case.

Causal inference by composition. Closest to our setting, Vergari et al. (2021) frame inference as compositions of tractable circuit operations, and Wang & Kwiatkowska (2023) add the MD-calculus, deriving the determinism requirements under which an estimand's compositional representation evaluates exactly; their construction, however, is discrete-only, demonstrated only for backdoor adjustment, and tied to a single vtree structure, as we detail in Sections 3 and 4.

3 Conditional As a Circuit Operation Via Weak Determinism

Conditional queries $P(\mathbf{N} \mid \mathbf{D})$ can be evaluated on any smooth and decomposable circuit by two feedforward evaluations — on $\mathbf{N} \cup \mathbf{D}$ and on $\mathbf{D}$ alone — divided. This suffices when the conditional is the final object of interest, but not when it is a sub-expression of a larger query such as backdoor adjustment $P(Y \mid do(X)) = \int_Z P(Y \mid X, Z)\, P(Z)\, dZ$, which requires the kernel $P(Y \mid X, Z)$ as a circuit over $Y \cup X \cup Z$ for the product with $P(Z)$ and the marginalization to be exact and tractable, hence the need for a conditional circuit operation. The standard construction is compositional: model $P(\mathbf{N} \mid \mathbf{D})$ as $P(\mathbf{N}, \mathbf{D}) \cdot P(\mathbf{D})^{-1}$,

$$\mathrm{COND}(\mathcal{C}; \mathbf{N} \mid \mathbf{D}) \;:=\; \mathrm{PROD}\big(\mathrm{MARG}(\mathcal{C}; \mathbf{N} \cup \mathbf{D}),\; \mathrm{POW}(\mathrm{MARG}(\mathcal{C}; \mathbf{D}); -1)\big). \tag{1}$$

As Wang & Kwiatkowska (2023) point out, the base circuit must then be $\mathbf{D}$-deterministic in order for the reciprocal to be tractable. Their reference implementation, however, computes conditionals by a different approach, which we show imposes a softer requirement; we formalize both.

Definition 4 (Weak marginal determinism)**.** *Let $\mathbf{Q} \subseteq \mathbf{V}$. A sum node s is* weakly $\mathbf{Q}$-deterministic *if, for all distinct child nodes β_i, β_j of s, either $\mathrm{supp}_{\mathbf{Q}}(\beta_i) \cap \mathrm{supp}_{\mathbf{Q}}(\beta_j) = \emptyset$, or the two marginal densities coincide and $\beta_i(\mathbf{q}) = \beta_j(\mathbf{q})$ for all $\mathbf{q} \in \mathrm{Val}(\mathbf{Q})$. A sum node is weakly $\mathcal{U}$-deterministic unconditionally.*

x In words, the $\mathbf{Q}$-supports of the children form a partition, children in the same block modelling the same density; weak determinism thus strictly generalizes marginal determinism.

Definition 5 (Weakly $\mathbf{Q}$-deterministic circuit)**.** *A PC is* weakly $\mathbf{Q}$-deterministic *if every sum node s with $\phi_{\mathbf{Q}}(s) \neq \emptyset$ is weakly $\mathbf{Q}$-deterministic.*

As in Wang & Kwiatkowska (2023), md-vtrees specify weak determinism, under a slightly adapted reading of the labels $\psi(m)$: every sum node $S^{(m)}$ must be both weakly $\psi(m)$-deterministic *and* weakly $\psi(m_l) \cup \psi(m_r)$-deterministic, which only tightens pseudo-mixing layers and is vacuous for other layer types.

Proposition 6 (Conditional by weight reassignment)**.** *Let C be a weakly $\mathbf{D}$-deterministic PC over $\mathbf{N} \cup \mathbf{D}$ with sum layers $S^{(m)}$ and weights $w^{(m)}$. Reassigning $\tilde{w}^{(m)}$ by layer type: universal layers and layers with $\phi_{\mathbf{D}}(m) = \emptyset$ unchanged; synthesizing layers $\tilde{w}^{(m)}_{ij,klrt} := 1$ wherever $w^{(m)}_{ij,klrt} > 0$; normalized over the right child for left-mixing layers, $\tilde{w}^{(m)}_{ij,klrt} := w^{(m)}_{ij,klrt} / \sum_{r',t'} w^{(m)}_{ij,klr't'}$; for pseudo-left-mixing as for left-mixing if $D = \psi(m)$ and as for synthesizing if $D = \psi(m_l) \cup \psi(m_r)$. This yields a circuit $\tilde{C}$ that models $P(\mathbf{N} \mid \mathbf{D})$ exactly, with the same size as C, computable from C in time linear in $|C|$. The correctness of the operation and complexity are proven in Appendix C.*

The formalization allows for detailed analysis, which uncovers previously unknown costs. Consider the two canonical estimands. Backdoor adjustment, $P(Y \mid do(X)) = \int_Z P(Y \mid X, Z)\, P(Z)\, dZ$, is certified by the MD-calculus at linear cost only if the circuit is $\{Z\}$-deterministic — so that the product with $P(Z)$ remains exact and cheap — *and* $\{X, Z\}$-deterministic, so that the kernel $P(Y \mid X, Z)$ can be materialized. The frontdoor estimand, with mediator M, imposes the same pattern and requires $\{X\}$- and $\{X, M\}$-determinism. In both cases the required determinism scopes are *nested*: one scope strictly contains the other.

Proposition 7 (Nested determinisms force pseudo-mixing layers)**.** *Let w be a regular md-vtree implying both $\mathbf{Q}$- and $\mathbf{Q}'$-determinism for scopes with $\mathbf{Q} \subsetneq \mathbf{Q}'$. Then w contains a pseudo-mixing node: a node m with children m_l, m_r such that $\psi(m) = \psi(m_l)$ and $\psi(m_r) \neq \mathcal{U}$. In particular, backdoor adjustment in linear time and the frontdoor estimand on* any *vtree give rise to pseudo-mixing layers.*

The construction algorithm of Wang & Kwiatkowska (2023) does not make a distinction between mixing and pseudo-mixing layers: it constrains the labelled child and leaves the other unconstrained,

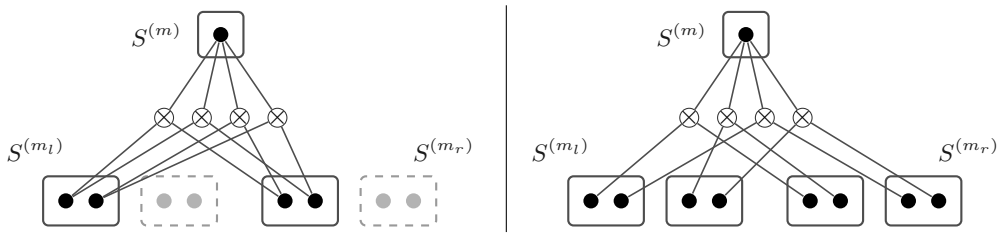


Figure 1: A synthesizing sum node whose child layers each have $\gamma = \nu = 2$ (nodes at the same position share a support; supports within a group are disjoint; dashed groups unused). Left: minimally diverse wiring — one left node recruited per position, so each left support piece is shared across all right states. Right: maximally diverse wiring — each left group is paired by position with a distinct right group, so each support piece is answered by a different copy of its sibling.

which does not fully satisfy weak determinism on pseudo-mixing layers. The informal conditional materializes $P(\mathbf{N} \mid \mathbf{D}) = P(\mathbf{N}, \mathbf{D}) \cdot P(\mathbf{D})^{-1}$ by composing the two marginals with a reciprocal — the COND operator of Section 2 — and computing it exactly and in linear time requires the premise of Proposition 6: weak $\mathbf{D}$-determinism at every $\mathbf{D}$-active layer. This is precisely what the circuits of Wang & Kwiatkowska (2023) lack on pseudo-mixing layers, and Proposition 7 shows that nested determinism requirements force such layers wherever they occur — the frontdoor's $\{X\} \subsetneq \{X, M\}$ among them, on any vtree. Their framework therefore could not have computed the frontdoor estimand, nor any estimand whose required determinisms are nested, which explains their observation that frontdoor adjustment lies within their framework's scope yet is left uninstantiated. Supplying the missing determinism is a matter of circuit construction, which we take up in Section 4.

4 Diversity and a Diversity-Preserving Construction

Section 3 located a computational failure in existing methods: their conditioning is not exactly evaluable on their own circuits for a broad class of estimands. We now show that the same construction also gives rise to a second failure, of a representational nature, which diminishes the capacity of the circuit to model dependencies present in the data.

Proposition 8 (Single-group layers)**.** *Every layer of an MDNet is a single group: $\gamma_m = 1$ for all m.*

The proof is given in Appendix B. To understand what this degeneracy means for the dependencies a layer can express, we introduce the *diversity* of a sum layer, using the layered notation of §2.

Definition 9 (Diversity)**.** *Two child nodes are* paired *if some product node has them as its two children. For a set S of sum nodes and a position l, $S|_{h=l}$ restricts S to its nodes at position l; for a set S of right (respectively left) child nodes,* $\mathrm{Sib}(S)$ *denotes the left (respectively right) child nodes paired with some node in S. The* diversity of m_l against m_r at position l *is*

$$\mathrm{D}(l) \;:=\; \Big|\, \mathrm{Sib}\big(\textstyle\bigcup_{p=1}^{\gamma_{m_r}} S_p^{(m_r)}\big)\big|_{h=l} \,\Big|,$$

the number of distinct left nodes at position l recruited by the wiring; the diversity of m_r against m_l is defined symmetrically. A layer is minimally diverse *if $\mathrm{D} = 1$ at every position on both sides.*

If $\mathrm{D}(l) = 1$, a single copy of the l-th left support is shared across all right nodes; larger $\mathrm{D}(l)$ answers different right nodes with decoupled copies, each free to model its scope only where its paired right support lives. The notion is vacuous when all nodes share the full support and becomes decisive precisely when children have disjoint supports, as determinism requires; a single-group layer is the extreme case, where every copy is necessarily shared (Proposition 8).

Remark 10 (Mutual minimal diversity and independence)**.** *If two sibling layers with scopes $\mathbf{L}$ and $\mathbf{R}$ are minimally diverse against each other, the circuit models $\mathbf{L}$ and $\mathbf{R}$ as independent, $P(\mathbf{L}, \mathbf{R}) = P(\mathbf{L})\, P(\mathbf{R})$.*

A single recruited copy on each side is then shared across all product nodes, leaving no wiring through which one side could modulate the other. This representational limit interacts with the vtree: under the MDNet construction, some vtrees cannot model the dependencies a causal query

requires at all. In backdoor adjustment with Z confounding $X \to Y$, the dependence of Y on (X, Z) is representable only if Y is separated from (X, Z) at the root by a mixing layer that keeps the Y subtree diverse — exactly the (forced) vtree of Wang & Kwiatkowska (2023). Its price is paid below the root, where the single-group layers leave the dependencies within (X, Z) unmodelled. Under any other vtree the outcome is worse: separating Z from (X, Y) makes the root synthesizing with single-group children (Proposition 8), minimally diverse against each other, so by remark 10 Z is modelled as independent of (X, Y) and the estimand collapses to the observational distribution, $\widehat{P}(Y \mid do(X)) = \int_Z \widehat{P}(Y \mid X)\, \widehat{P}(Z)\, dZ = \widehat{P}(Y \mid X)$. An experiment in Appendix H verifies this collapse empirically.

To overcome these failures, we propose a construction that preserves diversity and satisfies weak determinisms at every layer, bootstrapped at the leaves and carried up the circuit through the weight tensor structure.

Node budget and layer dimensions. Rather than fixing γ and ν per layer we specify only a total node budget N. The trade-off: nodes refine the partition of the scope — the fine-grained units between which conditioning selects — while groups carry decoupled copies of the whole partition, allowing one side to respond differently to different states of its sibling; nodes buy resolution, groups buy diversity. When both children are constrained, however, the diversity of either against the other is capped by the number of product nodes, so fine partitions on both sides would be wasteful. We therefore allocate N by layer type (Appendix D gives the full procedure): the root receives one node and one group; with a the largest factor of N not exceeding $\sqrt{N}$ and $b = N/a$, pseudo-left-mixing and synthesizing layers take the shape $(\gamma_l, \nu_l, \gamma_r, \nu_r) = (a, b, b, a)$ — oriented toward the less constrained child for synthesizing layers, right on ties — matching each side's group count to the sibling's nodes-per-group, while left-mixing layers get $(1, N, N, 1)$, partitioning the constrained child as finely as possible, and universal layers admit any allocation.

Weight structure. The sparsity pattern of the weight tensor $w^{(m)}$ defines the layer's routing. Besides satisfying the layer's own weak determinism constraints, the routing must keep the supports of the nodes within each group pairwise disjoint — the invariant that the parent layer relies on for its determinism — while exploiting the budget allocation for diversity. In left-mixing layers each left child node is assigned to a single parent sum node from a single group, densely over the right; pseudo-left-mixing layers add that each combination of left and right child supports appears only once; synthesizing layers also use each combination once, but both supports in a product may come from any group, and the node index of each child determines the group of its sibling, pairing every support piece with a different copy. Assignments are drawn randomly (e.g. left child node to parent sum node), with the full procedure and the influence of assignment fairness on learning given in Appendix D.

5 Disjoint Continuous Leaves

Everything developed so far is domain-agnostic: the conditional operation and the construction concern the structure of circuits, and they apply unchanged whether the data is discrete, continuous, or mixed. The only part of the circuit that is domain-specific is the leaf layer, which is also where the determinism of the circuit begins. So the task of extending weakly (marginally) deterministic PCs to continuous (and mixed) domains involves designing continuous leaf layers that satisfy and bootstrap the required determinisms, while being expressive enough to model complex marginal densities.

Consider a leaf group over a single continuous variable X with $\mathrm{Val}(X) = \mathbb{R}$, consisting of $\nu = K$ nodes, each encoding a density p_i over some interval $I_i = (\xi_{i-1}, \xi_i]$ in the real line. The requirements on these nodes are as follows. First, the nodes must have disjoint supports (R1), partitioning $\mathbb{R}$. Second, adjacent nodes must have matching one-sided limits at their shared boundaries (R2), so as to avoid discontinuities in the modelled density that do not exist in the data. Third, each node must be a valid density, positive (R3) and normalized (R4), as any PC leaf must be. Ideally, the nodes should be learnable (D1). And fifth, the nodes should have tractable CDFs (D2) and moments (D3), which are needed for certain queries that we won't cover here.

Observe that (R1), (R3), and (R4) together force each node to contribute equal mass $1/K$, pinning the modelled CDF at ξ_k to the values k/K — precisely its quantiles (Proposition 29, Appendix F).

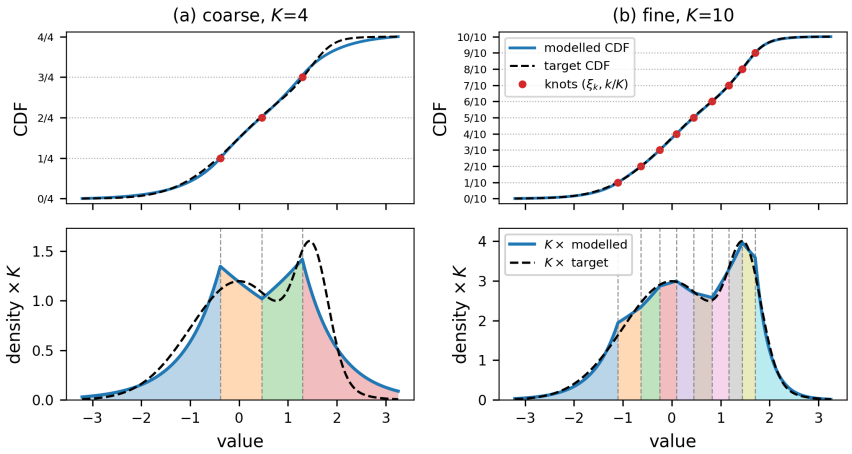


Figure 2: Log-linear spline leaf groups over a continuous variable X, at resolutions $K = 4$ (left) and $K = 10$ (right): K nodes with pairwise disjoint interval supports, unit mass pinning the modelled CDF at $(\xi_k, k/K)$. Top: modelled (solid) vs target (dashed) CDF. Bottom: resulting modelled density $K \cdot p_X$ (solid, per-interval components) vs target $K \cdot p^\star_X$ (dashed). (Color reading recommended.)

The node density p_i is then the derivative of an interpolation of the modelled CDF through the pinned points $(\xi_k, k/K)$, and the problem of leaf design reduces to choosing a monotonically increasing, continuously differentiable interpolation family: the pinning gives (R1) and (R4), differentiability gives (R2), monotonicity gives (R3), and the choice of family settles the remaining desiderata (D1)–(D3).

We instantiate the spline with log-linear segments, the degree-1 logspline family of Kooperberg & Stone (1991): exponential pieces, positive by construction, with mass, CDF, and moments in closed form and integrable tails of the same functional form. A leaf group is parameterized by an anchor ξ_1 and log-heights $\lambda_1, \ldots, \lambda_{K-1} \in \mathbb{R}$. The heights $h_k = e^{\lambda_k}$ are the shared density values at the splits, and unit mass pins each segment's slope and width in closed form. The splits are therefore derived from the heights, and the group has exactly K unconstrained scalars. The two outer components are exponential decays with rates h_1 and h_{K-1}. Appendix F derives the segment parameters from the constraints and verifies (R1)–(R4) and (D1)–(D3), and Figure 2 illustrates the construction.

6 Tractability-Aware Identification

Wang & Kwiatkowska (2023) established that PCs can serve as composable density estimators for causal inference, deriving the backdoor and frontdoor adjustments as explicit compositions of tractable circuit operations — two classical identifiable structures, each obtained by hand. Two instances do not make a universal engine, which raises the question: can the derivation of the compositional form, and the determination of its tractability, themselves be automated for *any* identifiable estimand? This section answers both in the affirmative.

Our basis is a sound and complete identification algorithm, which Shpitser & Pearl (2006) provide, and whose derivations are composed of only three operations on distributions — marginalization, product, and conditioning — all of which we can model given a circuit satisfying the required properties. Universalizing the engine therefore reduces to two requirements: derive the compositional form of any identifiable query automatically, and characterize when the resulting composition is exactly evaluable on a given circuit. The preceding sections supply the conditions: Section 3 established that exact conditioning requires marginal determinism over the conditioning scope and that products remain tractable under support compatibility (Wang et al., 2024, Thm. 3), and Section 4 showed how to construct circuits carrying prescribed determinisms.

We propose *Tractability-Aware Identification* (T-ID), which performs the derivation and the checking in a single pass: it runs the identification recursion, emits each derivation step as the corre-

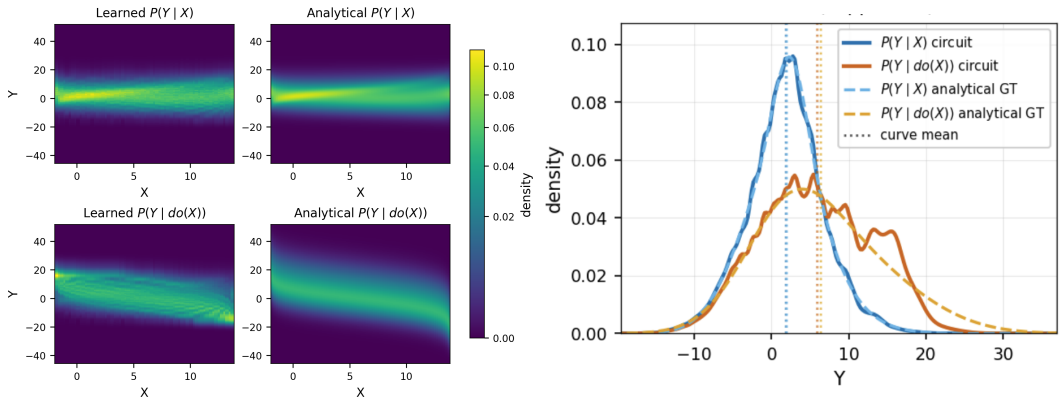


Figure 3: Continuous backdoor setting (continuous confounder Z, treatment X, outcome Y; trained on observational data only, the interventional density is evaluated exactly). $N = 64$, trained on ~ 1 million samples, full protocol in Appendix H. Left: learned (solid) vs analytical (dashed) joint densities of $P(Y \mid X)$ (top) and $P(Y \mid \mathrm{do}(X))$ (bottom). Right: slice at $X = 2.32$ of the circuit's conditionals against the analytical ground truth. (Color reading recommended.)

sponding circuit operation, and admits the step only if the tractability conditions hold — with one extension crucial to soundness: before conditioning, the factorization is fully simplified, guided by d-separations in the causal graph, so that no conditional that can be merged into a simpler one is emitted; e.g., $P(Y|X,Z,W) \cdot P(X|Z)$ simplifies to $P(Y,X|Z,W)$ if $X \perp\!\!\!\perp W \mid Z$, requiring weak $\{Z,W\}$-determinism instead of $\{X,Z,W\}$- and $\{Z\}$-determinism. T-ID terminates in one of three ways: if identification succeeds and every step is admitted, it returns the compositional query, exact relative to the model by construction; if identification fails, the query is unidentifiable; and if identification succeeds but some step is inadmissible, the query is identifiable yet not evaluable on the given circuit, and T-ID reports the missing determinism. Finally, T-ID certifies complexity: the emitted query evaluates in time $\mathcal{O}(|\mathcal{C}|^K)$ in the circuit size, with K also determined by T-ID. Appendix G presents the complete algorithm and establishes its guarantees: soundness, completeness, and a worst-case complexity bound. Inverting the check yields the determinism family a circuit must provide: the same recursion run once in *collection mode* records every conditioning scope the derivation demands instead of verifying it, returning the set $\mathcal{R}$ of required determinisms (REQUIREDDETERMINISMS, Appendix G.3). This is what prescribes the md-labels of our constructions, and it is also where the "almost" in automation shows: the returned family must be anticipated when the vtree and its labels are designed (Appendix D.4).

7 EXPERIMENTS

We evaluate on synthetic data from random SCMs, generated with the CausalProfiler benchmark generator of Panayiotou et al. (2026), whose implementation we adopt: a fixed graph skeleton, mechanisms sampled on top, and paired observational and interventional datasets. We restrict to SCMs whose interventional distribution admits a closed-form expression, so that ground truth is available analytically. All circuits are trained by batched expectation maximization on the observational sample only; interventional data is used exclusively for evaluation.

Figure 3 reports a fully continuous backdoor setting and Figure 4 a mixed collider-door setting compiled by T-ID. In both, the circuits model the interventional distribution — in the mixed case with near-perfect fit to the analytical ground truth — and the emitted query circuit scales with the base circuit size as predicted by T-ID. Table 1 compares our circuits with the MD-Net of Wang & Kwiatkowska (2023) on real binary benchmark (ASIA, CHILD, WIN95PTS, ANDES; protocol in Appendix H). Our method wins decisively on ASIA and ANDES, ties on CHILD, and comes within roughly $2\times$ of MD-Net on WIN95PTS. Further experiments are reported in Appendix H.

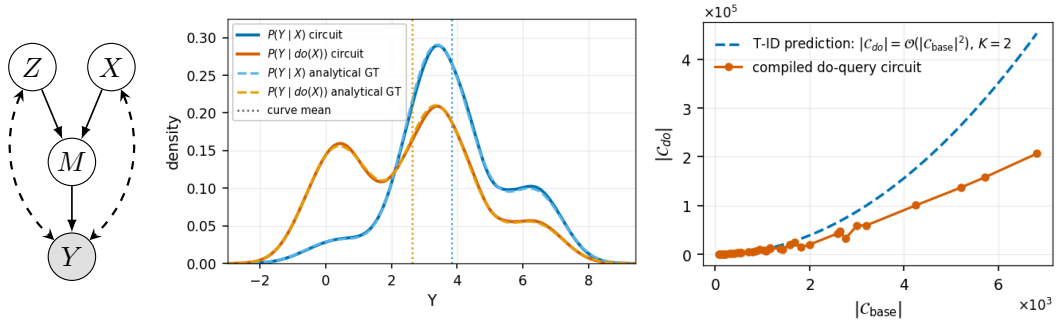


Figure 4: Mixed-domain collider-door setting. $N = 16$, trained on $90,000$ samples. Left: collider-door SCM over binary Z, X, M and conditionally Gaussian outcome Y (dashed bidirected edges: unobserved confounding), from which T-ID compiles the interventional distribution. Middle: slice at $X = 1$ of the learned interventional distribution (solid) vs the analytical ground truth (dashed). Right: size of the emitted query circuit vs the base-circuit size, matching the T-ID prediction $\mathcal{O}(|\mathcal{C}_{\text{base}}|^K)$ with $K = 2$. (Color reading recommended.)

		Ours		MD-Net		
Dataset	Target	MAE	Log-loss	MAE	Log-loss	Naive MAE
ASIA	0.8065	$\mathbf{0.0048} \pm 0.0055$	0.4914 ± 0.0003	0.1477 ± 0.1773	0.5913 ± 0.1439	0.0144
CHILD	0.3164	0.0760 ± 0.0122	0.6394 ± 0.0056	0.0773 ± 0.0085	0.6397 ± 0.0035	0.1345
WIN95PTS	0.0521	0.0071 ± 0.0022	0.2052 ± 0.0003	$\mathbf{0.0037} \pm 0.0023$	0.2048 ± 0.0002	0.0501
ANDES	0.1432	$\mathbf{0.0058} \pm 0.0022$	0.4109 ± 0.0001	0.0205 ± 0.0001	0.4126 ± 0.0000	0.0952

Table 1: Backdoor-adjustment accuracy on the MD-Net benchmark (queries in Table 4; mean $\pm$ std over 5 runs). Each row gives the exact target, our PCs, the released MD-Net, and a naive counting baseline; MAE is measured against the exact target, log-loss is the Bernoulli log-loss of the estimated effect. Best MAE in bold.

8 Limitations and Conclusion

Several limitations remain. First, weight-reassignment conditionals are piecewise in the conditioned variables, and so are all modelled interventional distributions in the intervention variables. Second, T-ID *presumes expressivity*: it checks that each emitted conditional is tractable on the given circuit, not that the circuit can faithfully express it; Appendix E studies when the emitted conditionals are expressible. Third, composition can spike circuit size, as tracked by the exponent K of Section 6, and quickly make an estimand impractical. Finally, more complicated estimands such as the frontdoor adjustment remain difficult to model in continuous domains; model scale, rather than the framework, may be the culprit.

Within these limitations, we have advanced probabilistic circuits toward nearly general-purpose engines for exact causal inference. We formalized conditioning as a circuit operation, identified weak marginal determinism as the tractability condition, and exposed a failure of exactness in the existing compositional approach: its conditioning is exact only under determinism requirements that its own construction violates on nested-scope estimands such as the frontdoor adjustment. The same construction also fails representationally—its layers collapse to minimal diversity—and we gave a diversity-preserving construction satisfying the required weak determinisms at every layer. Disjoint log-linear spline leaves extended the framework to continuous and mixed domains, and T-ID automated identification and composition with proved soundness, completeness, and a complexity bound. Experiments on synthetic continuous and mixed SCMs corroborated these predictions.

Future work includes hardware-aware GPU implementations of the sparse layered operations, structured parameterizations of emitted queryex products such as Monarch matrices (Dao et al., 2022; Zhang et al., 2025) to reduce the estimand exponent, and application to real-world mixed and continuous datasets.

AI usage statement. AI tools were used to implement standard methods, write tests, and edit the manuscript.

REFERENCES

Florian Peter Busch, Moritz Willig, Matej Zečević, Kristian Kersting, and Devendra Singh Dhami. Counterfactual sum-product networks. In *Eighth Workshop on Tractable Probabilistic Modeling*, 2025. URL https://openreview.net/forum?id=wLIsmzqzWJ.

YooJung Choi, Antonio Vergari, and Guy Van den Broeck. Probabilistic circuits: A unifying framework for tractable probabilistic models. Technical report, University of California, Los Angeles, 2020. URL https://starai.cs.ucla.edu/papers/ProbCirc20.pdf.

Gregory F. Cooper. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, 42(2-3):393–405, 1990. doi: 10.1016/0004-3702(90)90060-D. URL https://doi.org/10.1016/0004-3702(90)90060-D.

Tri Dao, Beidi Chen, Nimit S. Sohoni, Arjun Desai, Michael Poli, Jessica Grogan, Alexander Liu, Aniruddh Rao, Atri Rudra, and Christopher Ré. Monarch: Expressive structured matrices for efficient and accurate training. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 4690–4721. PMLR, 2022. URL https://proceedings.mlr.press/v162/dao22a.html.

Yimin Huang and Marco Valtorta. Pearl's calculus of intervention is complete. In *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*, pp. 217–224. AUAI Press, 2006. URL https://www.cse.sc.edu/~mgv/papers/HuangValtortaUAI06.pdf.

Ilyes Khemakhem, Ricardo Monti, Robert Leech, and Aapo Hyvärinen. Causal autoregressive flows. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 3520–3528. PMLR, 2021. URL https://proceedings.mlr.press/v130/khemakhem21a.html.

Murat Kocaoglu, Christopher Snyder, Alexandros G. Dimakis, and Sriram Vishwanath. CausalGAN: Learning causal implicit generative models with adversarial training. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=BJE-4xW0W.

Charles Kooperberg and Charles J. Stone. A study of logspline density estimation. *Computational Statistics & Data Analysis*, 12(3):327–347, 1991. doi: 10.1016/0167-9473(91)90115-I. URL https://doi.org/10.1016/0167-9473(91)90115-I.

Yitao Liang, Jessa Bekker, and Guy Van den Broeck. Learning the structure of probabilistic sentential decision diagrams. In *Proceedings of the 33rd Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 2017. URL https://www.auai.org/uai2017/proceedings/papers/291.pdf.

Christos Louizos, Uri Shalit, Joris M. Mooij, David Sontag, Richard Zemel, and Max Welling. Causal effect inference with deep latent-variable models. In *Advances in Neural Information Processing Systems*, volume 30, pp. 6446–6456. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/hash/94b5bde6de888ddf9cde6748ad2523d1-Abstract.html.

Valentyn Melnychuk, Dennis Frauen, and Stefan Feuerriegel. Causal Transformer for estimating counterfactual outcomes. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 15293–15329. PMLR, 2022. URL https://proceedings.mlr.press/v162/melnychuk22a.html.

Panayiotis Panayiotou, Audrey Poinsot, Alessandro Leite, Nicolas Chesneau, Marc Schoenauer, and Özgür Şimşek. CausalProfiler: Generating synthetic benchmarks for rigorous and transparent evaluation of causal machine learning. In *International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=0wCl7EifsY.

Nick Pawlowski, Daniel Coelho de Castro, and Ben Glocker. Deep structural causal models for tractable counterfactual inference. In *Advances in Neural Information Processing Systems*, volume 33, pp. 857–869. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/0987b8b338d6c90bbedd8631bc499221-Abstract.html.

Judea Pearl. *Causality: Models, reasoning, and inference*. Cambridge University Press, second edition, 2009. doi: 10.1017/CBO9780511803161. URL https://doi.org/10.1017/CBO9780511803161.

Robert Peharz, Steven Lang, Antonio Vergari, Karl Stelzner, Alejandro Molina, Martin Trapp, Guy Van den Broeck, Kristian Kersting, and Zoubin Ghahramani. Einsum networks: Fast and scalable learning of tractable probabilistic circuits. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7563–7574. PMLR, 2020. URL https://proceedings.mlr.press/v119/peharz20a.html.

Knot Pipatsrisawat and Adnan Darwiche. New compilation languages based on structured decomposability. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, pp. 517–522. AAAI Press, 2008. URL https://cdn.aaai.org/AAAI/2008/AAAI08-082.pdf.

Hoifung Poon and Pedro Domingos. Sum-product networks: A new deep architecture. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, pp. 337–346. AUAI Press, 2011. URL https://homes.cs.washington.edu/~pedrod/papers/uai11a.pdf.

Harsh Poonia, Moritz Willig, Zhongjie Yu, Matej Zečević, Kristian Kersting, and Devendra Singh Dhami. χSPN: Characteristic interventional sum-product networks for causal inference in hybrid domains. In *Proceedings of the Fortieth Conference on Uncertainty in Artificial Intelligence*, volume 244 of *Proceedings of Machine Learning Research*, pp. 3004–3020. PMLR, 2024. URL https://proceedings.mlr.press/v244/poonia24a.html.

Dan Roth. On the hardness of approximate reasoning. *Artificial Intelligence*, 82(1-2):273–302, 1996. doi: 10.1016/0004-3702(94)00092-1. URL https://doi.org/10.1016/0004-3702(94)00092-1.

Ilya Shpitser and Judea Pearl. Identification of conditional interventional distributions. In *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*, pp. 437–444. AUAI Press, 2006. URL https://ftp.cs.ucla.edu/pub/stat_ser/r329-uai.pdf.

Antonio Vergari, YooJung Choi, Anji Liu, Stefano Teso, and Guy Van den Broeck. A compositional atlas of tractable circuit operations for probabilistic inference. In *Advances in Neural Information Processing Systems*, volume 34, pp. 13189–13201. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/hash/6e01383fd96a17ae51cc3e15447e7533-Abstract.html.

Benjie Wang and Marta Kwiatkowska. Compositional probabilistic and causal inference using tractable circuit models. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 9488–9498. PMLR, 2023. URL https://proceedings.mlr.press/v206/wang23o.html.

Benjie Wang, Denis Deratani Mauá, Guy Van den Broeck, and YooJung Choi. A compositional atlas for algebraic circuits. In *Advances in Neural Information Processing Systems*, volume 37, pp. 141318–141355. Curran Associates, Inc., 2024. doi: 10.52202/079017-4488. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/ff9c70659c39cdd801dd5f5a1201c29e-Abstract-Conference.html.

Kevin Xia, Kai-Zhan Lee, Yoshua Bengio, and Elias Bareinboim. The causal-neural connection: Expressiveness, learnability, and inference. In *Advances in Neural Information Processing Systems*, volume 34, pp. 10823–10836. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/hash/5989add1703e4b0480f75e2390739f34-Abstract.html.

Mengyue Yang, Furui Liu, Zhitang Chen, Xinwei Shen, Jianye Hao, and Jun Wang. CausalVAE: Disentangled representation learning via neural structural causal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9588–9597. IEEE, 2021. doi: 10.1109/CVPR46437.2021.00947. URL https://doi.org/10.1109/CVPR46437.2021.00947.

Matej Zečević, Devendra Singh Dhami, Athresh Karanam, Sriraam Natarajan, and Kristian Kersting. Interventional sum-product networks: Causal inference with tractable probabilistic models. In *Advances in Neural Information Processing Systems*, volume 34, pp. 15019–15031. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/hash/7eb7eabbe9bd03c2fc99881d04da9cbd-Abstract.html.

Honghua Zhang, Meihua Dang, Benjie Wang, Stefano Ermon, Nanyun Peng, and Guy Van den Broeck. Scaling probabilistic circuits via Monarch matrices. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 74804–74818. PMLR, 2025. URL https://proceedings.mlr.press/v267/zhang25q.html.

Contents

A Notation **14**

B Formalization of the MDNet Construction **15**

B.1 Setup and notation 15

B.2 The construction 15

B.3 Group counts and the collapse 16

C Conditional Operation Proofs **17**

C.1 Weight reassignment: Proposition 6 17

D The Diversity-Preserving Construction **18**

D.1 Setup and notation 18

D.2 Node budget allocation 19

D.3 Routing 19

D.4 Pseudo-mixing layers: requirements, cost, and placement 21

E A Study on Expressivity **22**

F Properties of Log-Linear Spline Leaves **24**

G T-ID: The Complete Algorithm and Its Guarantees **27**

G.1 The T-ID recursion 28

G.2 Helper procedures 29

G.3 The required determinisms of a query 30

G.4 Guarantees 30

H Additional Experiments **32**

H.1 Backdoor 32

H.2 Mixed frontdoor 38

H.3 Comparison with MD-Net on real binary data 39

I Experimental Setup **40**

A NOTATION

Table 2 collects the notation used throughout the paper.

Symbol	Meaning
Variables and distributions	
$X, Y, Z, U, V, W, Q, D, N$	single random variables (uppercase)
$\mathbf{X}, \mathbf{V}, \mathbf{U}, \mathbf{Y}, \mathbf{Q}, \mathbf{D}, \mathbf{N}, \mathbf{W}$	sets of variables (bold uppercase)
$x, \mathbf{x}, \mathbf{v}, \mathbf{y}, \mathbf{q}$	assignments to variables / sets (lowercase)
$\mathrm{Val}(\mathbf{X})$	set of assignments to $\mathbf{X}$
μ_X	base measure of X (counting or Lebesgue)
$P(\mathbf{V}),\ p(\mathbf{v})$	a distribution and its density
$p(\mathbf{w}) = \int p(\mathbf{v})\, d\mu$	marginal density over $\mathbf{w} \subseteq \mathbf{v}$
$p(\mathbf{y} \mid \mathbf{x}) = p(\mathbf{y}, \mathbf{x})/p(\mathbf{x})$	conditional density
Probabilistic circuits	
$\mathcal{C}$	a probabilistic circuit (PC)
α, β	circuit nodes
$\phi(\alpha)$	scope of node α: variables in the leaves below α
$\phi_{\mathbf{Q}}(\alpha) = \phi(\alpha) \cap \mathbf{Q}$	restricted scope with respect to $\mathbf{Q}$
$\mathrm{supp}(\alpha)$	support of node α: assignments where $p_\alpha > 0$
$\mathrm{supp}_{\mathbf{Q}}(\alpha)$	marginal support of α over $\mathbf{Q}$
$\mathrm{ch}(\alpha)$	children of node α
$w_i \geq 0$	weight of the i-th child of a sum node
$\lvert\mathcal{C}\rvert$	circuit size (number of edges)
v; m, m_l, m_r	a vtree; a vtree node and its left/right children
$w = (v, \psi)$	an md-vtree: vtree v with md-labels ψ
$\psi(m) \in \mathcal{P}(\phi(m)) \cup \{\mathcal{U}\}$	md-label of m; $\mathcal{U}$ = no determinism constraint
universal, mixing, pseudo-mixing, synthesizing	circuit layer types induced by the labels
$S^{(m)}$	sum layer at vtree node m
γ_m; ν_m	number of groups; nodes per group in $S^{(m)}$
$S^{(m)}_{ij}$	the j-th node of the i-th group of $S^{(m)}$ (i = group, j = position)
support piece	support shared by all nodes at one position (across groups)
$w^{(m)}$	weight tensor of layer m
$w^{(m)}_{ij,klrt} > 0$	product node pairing child nodes $S^{(m_l)}_{kl}$, $S^{(m_r)}_{rt}$ into $S^{(m)}_{ij}$
Determinism, diversity, conditioning	
$\mathbf{Q}$-deterministic	marginal supports of the children pairwise disjoint
weakly $\mathbf{Q}$-deterministic	children have disjoint or identical $\mathbf{Q}$-marginal densities
$\mathrm{MARG}, \mathrm{PROD}, \mathrm{COND}, \mathrm{POW}$	marginalization, product, conditional, reciprocal circuit operations
$\mathrm{D}(l)$	diversity of m_l against m_r at position l
$\mathrm{Sib}(S)$	child nodes on one side paired with nodes in S
Circuit construction (Section 4, Appendix D)	
N	node budget
a; $b = N/a$	largest factor of N not exceeding $\sqrt{N}$; its complement
π	(filled, balanced) assignment of positions/combinations to sum nodes
ρ_l	map selecting the node realizing each left position
σ_l, σ_r	sibling group maps, $\sigma_r(l) = 1 + ((l-1) \bmod \gamma_r)$
$\mathrm{Assign}(M, B; \tau)$	balanced assignment of M items to B bins, fairness temperature τ
σ	temperature of the weight sampling
Continuous leaves (Section 5)	

continued on next page

Symbol	Meaning
K	number of leaf components / spline pieces
$I_i = (\xi_{i-1}, \xi_i]$	interval support of the i-th component
ξ_k	split points (knot locations)
λ_k; $h_k = e^{\lambda_k}$	log-heights; density heights at the splits
Causal models and identification	
$\mathcal{M} = \langle \mathbf{U}, \mathbf{V}, \mathcal{F}, P(\mathbf{U}) \rangle$	structural causal model (SCM)
$V_i = f_i(\mathrm{pa}_i, \mathbf{U}_i)$	structural equation; pa_i = parents of V_i
$\mathcal{G}$	causal diagram (acyclic directed mixed graph)
$\mathrm{do}(\mathbf{x})$	intervention replacing the equations of $\mathbf{X}$ by constants $\mathbf{x}$
$P(\mathbf{y} \mid \mathrm{do}(\mathbf{x}))$	interventional distribution
$\mathrm{An}(\mathbf{Y})_{\mathcal{G}}$	ancestors of $\mathbf{Y}$ in $\mathcal{G}$
$\mathcal{G}_{\bar{\mathbf{X}}}$; $\mathcal{G}_{\underline{\mathbf{X}}}$	graphs with edges into (out of) $\mathbf{X}$ deleted
c-component	maximal bidirected-connected set of $\mathcal{G}$
T-ID and experiments	
T-ID	tractability-aware extension of the ID algorithm
K	complexity exponent of the emitted query
$\mathcal{C}_{\mathrm{base}}$	base circuit before query compilation
x^*, y^*	queried intervention value and outcome value

B Formalization of the MDNet Construction

This appendix formalizes the circuit construction of Wang & Kwiatkowska (2023), restates the layer notation of Section 2, and proves Proposition 8 of Section 4. The source does not define the construction algorithmically; the formalization below is derived from the construction constraints stated in prose in their Section 4 and verified against the reference implementation, which instantiates every layer with a single group. We quote the relevant passages where each constraint is introduced.

B.1 Setup and notation

We use the computational representation of Section 2: for an internal vtree node m with children m_l, m_r, the associated sum layer $S^{(m)}$ consists of γ_m groups of ν_m nodes each, with weight tensor $w^{(m)} \in \mathbb{R}_{\geq 0}^{\gamma_m \times \nu_m \times \gamma_{m_l} \times \nu_{m_l} \times \gamma_{m_r} \times \nu_{m_r}}$; $w^{(m)}_{ij,klrt} > 0$ represents a product node pairing left child node $S^{(m_l)}_{kl}$ with right child node $S^{(m_r)}_{rt}$ and feeding sum node $S^{(m)}_{ij}$ with that weight. Evaluation of the layer is the fused tensor contraction

$$\mathbf{s}^{(m)}(x) \;=\; \mathrm{einsum}\big(\, w^{(m)},\, \mathbf{s}^{(m_l)}(x_l),\, \mathbf{s}^{(m_r)}(x_r)\,;\; ijklrt,\, kl,\, rt \to ij \,\big), \tag{2}$$

that is, $s^{(m)}_{ij}(x) \;=\; \sum_{k,l,r,t} w^{(m)}_{ij,klrt}\, s^{(m_l)}_{kl}(x_l)\, s^{(m_r)}_{rt}(x_r)$, where x_l, x_r are the restrictions of x to $\phi(m_l)$ and $\phi(m_r)$. Decomposability is automatic (each layer multiplies only across the bipartition $(\phi(m_l), \phi(m_r))$) and smoothness is guaranteed by the layer–vtree alignment. The circuit output is $s^{(\mathrm{root})}_{11}(x)$, the root layer consisting of a single group with a single node. Recall that groups are counted only if used: γ_m is the number of groups of $S^{(m)}$ containing at least one node with an outgoing edge. A regular md-vtree labels each internal node m by $\psi(m) \in \{\psi(m_l), \psi(m_r), \psi(m_l) \cup \psi(m_r)\}$ (Wang & Kwiatkowska, 2023).

B.2 The construction

Definition 11 (MDNet construction, after Wang & Kwiatkowska (2023))**.** *Let (v, ψ) be a regular md-vtree over $\mathbf{V}$. The associated MDNet is built bottom-up. Each leaf m becomes an input layer of tractable densities over $\mathrm{Assign}(m)$. Each internal node m becomes a sum layer $S^{(m)}$ according to its label:*

- Mixing *($\psi(m) = \psi(m_l)$; symmetric for $\psi(m_r)$): there is a map $\sigma_l : \{1..\gamma_m\} \to \{1..\gamma_{m_l}\}$ such that $w^{(m)}_{ij,klrt} > 0$ implies $k = \sigma_l(i)$, and the left children within each sum node are distinct: $w^{(m)}_{ij,klrt} > 0 \wedge w^{(m)}_{ij,kl'rt} > 0 \Rightarrow l = l'$. The right children are unconstrained.*
- Synthesizing *($\psi(m) = \psi(m_l) \cup \psi(m_r)$): there are maps $\sigma_l : \{1..\gamma_m\} \to \{1..\gamma_{m_l}\}$ and $\sigma_r : \{1..\gamma_m\} \to \{1..\gamma_{m_r}\}$ such that $w^{(m)}_{ij,klrt} > 0$ implies $k = \sigma_l(i)$ and $r = \sigma_r(i)$, and each combination of children is used at most once within a group: $w^{(m)}_{ij,klrt} > 0 \wedge w^{(m)}_{ij',klrt} > 0 \Rightarrow j = j'$.*

The circuit output is the single node $S^{(\mathrm{root})}_{11}$, the root layer consisting of a single group with a single node, $\gamma_{\mathrm{root}} = \nu_{\mathrm{root}} = 1$.

Relation to the source. The mixing case formalizes the requirement that "the left children of the product nodes $\cup_k P_{i,k}$ must all be distinct, and all come from a single group" (Wang & Kwiatkowska, 2023, Sec. 4): the map σ_l formalizes "from a single group" (one child group per parent group i, quantified over all product nodes of the group, hence stated as a function of i), and the distinctness clause is stated as is; the right children are left unconstrained, matching the source. The synthesizing case formalizes "both the left children and right children of the product nodes $\cup_k P_{i,k}$ are nodes coming from a single group from their respective layers, and that each product node has a unique combination of children from these groups" (Wang & Kwiatkowska, 2023, Sec. 4): the maps σ_l, σ_r fix the two groups, and the unique-combination clause is stated per group. Under the source's convention that each product node is assigned to a single sum node, uniqueness of combinations is automatic within a sum node, and the clause has content across the sum nodes of a group, as formalized above. Consistently, the reference implementation instantiates all layers with $\gamma_m = 1$.

B.3 Group counts and the collapse

Call the edge from m to a child m_c *constrained* if m's label fixes the child's group map: both edges at a synthesizing layer, and the labelled edge at a mixing layer.

Lemma 12 (Group counts decrease along constrained edges)**.** *If layer m constrains its left child via σ_l, then σ_l is surjective and $\gamma_{m_l} \leq \gamma_m$. Symmetrically for the right child of a synthesizing layer.*

Proof. Every group $S^{(m_l)}_k$ of $S^{(m_l)}$ contains a node with an outgoing edge, by the convention that groups are counted only if used. Hence for each $k \leq \gamma_{m_l}$ there is a nonzero $w^{(m)}_{ij,klrt}$, and the constraint gives $\sigma_l(i) = k$. Thus σ_l is surjective onto $\{1..\gamma_{m_l}\}$, which requires $\gamma_{m_l} \leq \gamma_m$. □

Proposition 13 (Single-group collapse)**.** *Let C be an MDNet as in Definition 11. Then $\gamma_{m'} = 1$ for every layer m' reachable from the root via constrained edges. If the root layer is synthesizing, both subtrees are reachable via constrained edges (every internal layer constrains at least one side by regularity, and the root constrains both), so $\gamma_m = 1$ for all layers m.*

Proof. Induction on the length of the constrained path from the root. Base: $\gamma_{\mathrm{root}} = 1$ by construction. Step: if $\gamma_m = 1$ and $m \to m_c$ is constrained, Lemma 12 gives $\gamma_{m_c} \leq \gamma_m = 1$, and $\gamma_{m_c} \geq 1$ since m_c feeds m. Hence $\gamma_{m_c} = 1$. For the second claim: if the root is synthesizing, both child edges are constrained; inductively, every layer in either subtree is reached by a path whose every edge is constrained, since at each internal layer at least one of the two outgoing edges is constrained and the unconstrained side, when taken, is entered from a constrained edge at its parent. □

Remark 14 (Scope of the collapse)**.** *Proposition 13 is stated as finely as the construction permits: layers* not *reachable via constrained edges — which can exist under a non-synthesizing root, on the unconstrained side of mixing layers — retain a group structure. The reference implementation sets $\gamma_m = 1$ unconditionally, so for the circuits actually produced by the construction the blanket statement of Proposition 8 in the body holds.*

C Conditional Operation Proofs

This appendix proves Proposition 6 of Section 3. We work in the layered representation of Appendix B: each internal vtree node m induces a sum layer $S^{(m)}$ with weight tensor $w^{(m)}$ and layer evaluation $s_{ij}^{(m)}(x) = \sum_{klrt} w_{ij,klrt}^{(m)}\, s_{kl}^{(m_l)}(x_l)\, s_{rt}^{(m_r)}(x_r)$. Throughout, $\mathbf{D}$ is the conditioning scope and $\mathbf{N}$ the head.

Blocks and normalizers. Fix an assignment x over $\mathbf{V}$. Each node α of the circuit encodes a non-negative function over its scope, and we write $s_\alpha(x)$ for its value; in particular $s_{ij}^{(m)}(x)$ is the value of node j in group i of layer m. Integrating a node's function over its non-$\mathbf{D}$ variables yields its $\mathbf{D}$-marginal, a function of the conditioning assignment d alone:

$$z_\alpha(d) = \int s_\alpha(x)\; d\mu\big(x_{\phi(\alpha)\setminus\mathbf{D}}\big) \quad \text{with } x_{\mathbf{D}} = d,$$

so $z_\alpha(d)$ is the mass the subtree below α assigns to $\mathbf{D} = d$. For a sum node α with children β, weak $\mathbf{D}$-determinism (Definition 4) partitions the children into *blocks* according to their behaviour on $\mathbf{D}' = \mathbf{D} \cap \phi(\alpha)$: children in the same block model the same $\mathbf{D}'$-marginal density, and distinct blocks have disjoint $\mathbf{D}'$-supports. We write $m_B(d)$ for the shared $\mathbf{D}'$-marginal of block B and $W_B = \sum_{\beta\in B} w_{\alpha\beta}$ for its total weight. Two consequences drive everything below. First, the $\mathbf{D}'$-marginal of the node is $z_\alpha(d) = \sum_B W_B\, m_B(d)$, and since distinct blocks have disjoint $\mathbf{D}'$-supports, $z_\alpha(d) = W_{B^*} m_{B^*}(d)$ for $d \in \mathrm{supp}_{\mathbf{D}}(B^*)$: every other block's marginal vanishes at d. Second, a child value vanishes off its block's domain: $\beta(x) = 0$ whenever $x_{\mathbf{D}} \notin \mathrm{supp}_{\mathbf{D}}(\beta)$. We assume the standard convexity property of trained circuits, that each sum node's weights sum to one, $\sum_\beta w_{\alpha\beta} = 1$.

C.1 Weight reassignment: Proposition 6

Proof idea. We modify the weights so that every node computes the conditional of its own subtree, that is, the ratio of its value to its $\mathbf{D}$-marginal (its normalizer z_α). This is the right target: at the root the ratio is $s_{\mathrm{root}}(x)/z_{\mathrm{root}}(d) = P(\mathbf{N} \mid \mathbf{D})$. Leaves already satisfy it, products inherit it for free because the marginal factorizes over a decomposable product, and the only real work is at sum nodes, where the weights must be renormalized within each block of the weak-determinism partition. The layer-type recipes of Proposition 6 are shown below to be exactly that one rule in the layer structures where they apply.

Proof. We prove by bottom-up induction over the circuit that every modified node $\tilde{\alpha}$ computes the conditional of its subtree,

$$\tilde{s}_\alpha(x) = \frac{s_\alpha(x)}{z_\alpha(x_{\mathbf{D}})}, \qquad \text{with } \tilde{s}_\alpha = s_\alpha \text{ when } \phi_{\mathbf{D}}(\alpha) = \emptyset, \tag{3}$$

which at the root is the claim, since $z_{\mathrm{root}}(d) = P(\mathbf{D} = d)$.

Leaves over $\mathbf{D}$-variables are inputs to both numerator and denominator; leaves over non-$\mathbf{D}$ variables are normalized densities with $z = 1$ and are left unchanged, satisfying the invariant trivially.

Product nodes need no modification: by decomposability the children β, γ scope-partition the node, the integrals factorize, $z_\alpha(d) = z_\beta(d_{\mathbf{D}\cap\phi(\beta)})\, z_\gamma(d_{\mathbf{D}\cap\phi(\gamma)})$, and hence $s_\alpha(x)/z_\alpha(d) = \tilde{s}_\beta(x) \cdot \tilde{s}_\gamma(x)$ by the induction hypothesis.

Sum nodes. Write $s_\alpha(x) = \sum_B \sum_{\beta\in B} w_{\alpha\beta}\, s_\beta(x)$ with children already satisfying the invariant, and suppose α is reassigned weights $\tilde{w}_{\alpha\beta}$. The modified value is

$$\tilde{s}_\alpha(x) = \sum_B \frac{1}{m_B(d)} \sum_{\beta\in B} \tilde{w}_{\alpha\beta}\, s_\beta(x),$$

while the target, on the $\mathbf{D}$-domain of a single block B^*, block-localizes by the two vanishing consequences above:

$$\frac{s_\alpha(x)}{z_\alpha(d)} = \frac{\sum_{\beta\in B^*} w_{\alpha\beta}\, s_\beta(x)}{W_{B^*}\, m_{B^*}(d)} \qquad \text{for } d \in \mathrm{supp}_{\mathbf{D}}(B^*).$$

Matching coefficients of the block's child functions $\{s_\beta\}_{\beta \in B^*}$, for all x with $x_{\mathbf{D}} = d$ and all d, forces the unique reassignment: normalization within each block,

$$\tilde{w}_{\alpha\beta} = \frac{w_{\alpha\beta}}{W_{B(\beta)}}, \tag{4}$$

independently of d. (If two children of a block are identical functions, only the sum of their reassigned weights is determined; the recipes below preserve within-block weight ratios, so the verification is unaffected.)

It remains to check that the per-layer-type recipes of Proposition 6 instantiate equation 4 under the structure of md-layers.

Universal layers and layers with $\phi_{\mathbf{D}}(m) = \emptyset$ (unchanged): for $\phi_{\mathbf{D}}(m) = \emptyset$ there is a single trivial block with $m_B \equiv 1$, and convexity gives $W_B = 1$, so equation 4 is the identity. For universal layers, unchanged weights coincide with equation 4 exactly when the node's partition is a single block, in which case convexity again gives $W_B = 1$.

Left-mixing layers ($\psi(m) = \psi(m_l)$, $\mathbf{D} = \psi(m) \subseteq \phi(m_l)$), normalized over the right child: children of $S_{ij}^{(m)}$ pair a fixed left group $k = \sigma_l(i)$ with right nodes (r, t), so the $\mathbf{D}$-marginal of a child depends only on its left subnode l and the blocks are exactly the l-fibers. The recipe $\tilde{w}_{\alpha\beta} = w_{\alpha\beta} / \sum_{r't'} w^{(m)}_{ij,klr't'}$ is precisely equation 4, since $\sum_{r't'} w^{(m)}_{ij,klr't'} = W_B$ for the block of β. No further condition is needed.

Synthesizing layers ($\psi(m) = \psi(m_l) \cup \psi(m_r)$), $\tilde{w} := \mathbf{1}[w > 0]$: equation 4 requires $\tilde{w}_{\alpha\beta} = w_{\alpha\beta}/W_B$, which the indicator rule realizes exactly when every block is a singleton, i.e. when the children of each sum node are pairwise $\mathbf{D}$-disjoint. This is the case at synthesizing layers and at pseudo-left-mixing layers with $\mathbf{D} = \psi(m_l) \cup \psi(m_r)$, where the children of each sum node are pairwise $\mathbf{D}$-disjoint: each block B is a singleton with $W_B = w_{\alpha\beta}$, so equation 4 evaluates to 1 wherever $w_{\alpha\beta} > 0$, matching the indicator rule.

Pseudo-left-mixing layers with $\mathbf{D} = \psi(m)$ behave as left-mixing, covered above.

Every reassignment is a local modification of a weight tensor and no additional nodes are created, so $|\tilde{C}| = |C|$ and so the circuit size, and thus the complexity of its evaluation, is preserved. □

D The Diversity-Preserving Construction

This appendix formalizes the diversity-preserving construction of Section 4: the node budget allocation of each layer, the routing — the sparsity pattern of the weight tensor $w^{(m)}$, which determines the product nodes of each sum node — for every layer type, and the weight initialization. The formalization states the construction as it is required to be; the one place where the reference implementation deviates from it — the routing of pseudo-mixing layers — is flagged in Section D.4. We then isolate the two structural invariants that the routing must maintain at every layer and analyze the pseudo-mixing layers, at which the determinism requirement applies twice and whose expressivity cost motivates the placement rule of thumb for vtrees. Which conditionals the resulting circuits can express is the subject of Section E.

D.1 Setup and notation

We employ the layered computational representation recalled in Appendix B: an internal vtree node m with children m_l, m_r induces a sum layer $S^{(m)}$ of γ_m groups of ν_m nodes $S_{ij}^{(m)}$, whose wiring and weights are encoded in $w^{(m)} \in \mathbb{R}_{\geq 0}^{\gamma_m \times \nu_m \times \gamma_{m_l} \times \nu_{m_l} \times \gamma_{m_r} \times \nu_{m_r}}$; $w^{(m)}_{ij,klrt} > 0$ pairs the child nodes $S_{kl}^{(m_l)}$ and $S_{rt}^{(m_r)}$ in a product feeding $S_{ij}^{(m)}$. Nodes at the same position share a common support, and groups carry decoupled copies of the same partition of the scope (Section 4). The layer type of m is determined by the md-labels (Section 2). The construction takes as input a regular md-vtree (v, ψ), a node budget $N \in \mathbb{N}$, and an input distribution for every leaf variable, and proceeds bottom-up, fixing the dimensions of a layer's children before the children themselves are built.

Definition 15 (Balanced assignment)**.** $\mathrm{Assign}(M, B; \tau)$ *assigns M items to B bins one item at a time: after counts $c_1, \ldots, c_B$ with target load $\bar{c} = M/B$, bin b scores $s_b = -\tau\, c_b/\bar{c} + \xi_b$ with*

independent jitter $\xi_b \sim \mathcal{N}(0,1)$, and the item is placed in a bin sampled from $\mathrm{softmax}(s)$. *$\tau = 0$ recovers uniform random assignment; larger τ concentrates the distribution on balanced loads, while the jitter leaves unbalanced assignments possible. The* filled *variant* Assign^* *retries until every bin is non-empty (which requires $B \leq M$), falling back to a deterministic round-robin assignment.*

Definition 16 (Weight sampling)**.** *Each admissible entry of $w^{(m)}$ is initialized as* $\exp(\sigma\varepsilon)$ *with $\varepsilon \sim \mathcal{N}(0,1)$ and temperature $\sigma > 0$, and the tensor is normalized within each sum node over its children. The circuit stores* $\log w^{(m)}$ *as the trainable parameter. Universal layers are stored densely; all other layers are stored sparsely, with the mask $\{w^{(m)} > 0\}$ fixed once by the routing.*

D.2 Node budget allocation

Let a denote the largest factor of N not exceeding $\sqrt{N}$, and let $b = N/a$.

Definition 17 (Budget allocation)**.** *The root layer receives a single group with a single node, $\gamma_{\mathrm{root}} = \nu_{\mathrm{root}} = 1$, regardless of N. Every other layer receives dimensions from the budget N according to its type:*

- Universal*: $(\gamma_l, \nu_l, \gamma_r, \nu_r) = (1, N, 1, N)$; the allocation is arbitrary, since universal layers carry no disjointness constraint and the dense weights render the group structure irrelevant.*
- Left-mixing*: $(\gamma_l, \nu_l, \gamma_r, \nu_r) = (1, N, N, 1)$: the constrained labelled child is partitioned as finely as possible, into a single group of N nodes, and the unconstrained sibling answers with N groups of one node, thereby maximizing diversity.*
- Pseudo-left-mixing*: $(\gamma_l, \nu_l, \gamma_r, \nu_r) = (a, b, b, a)$: the labelled child carries a groups of b nodes and the constrained sibling b groups of a nodes, so that the sibling's group count matches the labelled child's nodes-per-group and each support piece can be answered by a distinct copy.*
- Synthesizing*: the shape of pseudo-left-mixing, oriented toward the less constrained child, with ties broken to the right: the oriented child receives $(\gamma, \nu) = (b, a)$ and its sibling (a, b).*

Right-mixing and pseudo-right-mixing layers are symmetric.

The rationale is that of Section 4: in a constrained layer, nodes refine the partition of the scope while groups buy diversity, but when both children are constrained, the diversity of either against the other is limited by the number of product nodes, i.e. by $\nu_l \cdot \nu_r$, so that a fine partition on both sides would be wasted. The allocation therefore concentrates nodes where they refine a partition and groups where they answer a sibling. For synthesizing layers, a child that is itself synthesizing signals that it is the more constrained of the two; it receives the finer partition, (a groups of b nodes). The orientation follows a rule of thumb that ranks layer types by constrainedness: unconstrained leaf $<$ universal $<$ mixing $<$ synthesizing $<$ pseudo-mixing $<$ constrained leaf.

D.3 Routing

The routing defines the support of $w^{(m)}$ and hence the product nodes of the layer. Recall that nodes at the same position share a support; we refer to the common support of the nodes at position j as the j-th *support piece* of the layer, so that each group replicates one and the same partition of the scope into pieces. We keep the two notions separate throughout: a *node* is a specific computational unit $S_{ij}^{(m)}$ of a group, whereas a support piece is shared by all nodes at one position across the groups; the routing maps assign nodes to sum nodes, but the determinism and disjointness constraints are phrased in terms of support pieces. The routing must maintain two invariants, stated here once and verified at the end of this subsection, after the leaf layers and the caps have been introduced:

(I1) *own weak determinism*: every constrained sum layer is weakly $\psi(m)$-deterministic, and weakly $\psi(m_l) \cup \psi(m_r)$-deterministic at pseudo-mixing layers, per the adapted reading of md-labels of Section 3 (the combined label is vacuous at all other layer types); universal layers carry no constraint;

(I2) *parent invariant*: within each group of a constrained layer, the $\psi(m)$-marginal supports of the sum nodes are pairwise disjoint, so that the parent layer can be weakly deterministic.

Definition 18 (Routing)**.** *Let m be an internal vtree node of layer type per Section 2, with parent dimensions (γ, ν) and child dimensions (γ_l, ν_l), (γ_r, ν_r) per Definition 17 (after the caps below). The support of $w^{(m)}$ is defined as follows.*

- Universal*: $w^{(m)} > 0$ everywhere.*
- Left-mixing*: there is a filled balanced assignment $\pi : \{1..\nu_l\} \to \{1..\nu\}$ of left positions to sum nodes and a map $\rho_l : \{1..\nu_l\} \to \{1..\gamma_l\}$ selecting, for each position, the node $(\rho_l(l), l)$ that realizes it, such that $w^{(m)}_{ij,klrt} > 0$ iff $j = \pi(l)$ and $k = \rho_l(l)$; the support is dense over the right indices (r, t). Each left support piece is recruited, through its selected node, by exactly one sum node.*
- Pseudo-left-mixing*: as left-mixing, except that the right child is constrained and each left support piece is answered by a single sibling group: there is a filled balanced assignment $\pi : \{1..\nu_l\} \to \{1..\nu\}$ of left positions to sum nodes, a map $\rho_l : \{1..\nu_l\} \to \{1..\gamma_l\}$ selecting the node that realizes each left position, and a map $\sigma_r(l) = 1 + ((l-1) \bmod \gamma_r)$, such that $w^{(m)}_{ij,klrt} > 0$ iff $j = \pi(l)$, $k = \rho_l(l)$, and $r = \sigma_r(l)$; the support is dense over the right positions t within the selected group. Each left support piece is recruited, through its selected node, by exactly one sum node, so the sum nodes of the layer have pairwise disjoint supports over the left MD set $\psi(m) = \psi(m_l)$, as the parent invariant (I2) demands.*
- Synthesizing*: the group of* each *child is determined by the node index of its sibling, $\sigma_l(t) = 1 + ((t-1) \bmod \gamma_l)$ and $\sigma_r(l) = 1 + ((l-1) \bmod \gamma_r)$, and there is a filled balanced assignment $\pi : \{1..\nu_l\} \times \{1..\nu_r\} \to \{1..\nu\}$ of combinations of positions to sum nodes — every combination is assigned to exactly one sum node, and every sum node receives at least one — such that $w^{(m)}_{ij,klrt} > 0$ iff $j = \pi(l, t)$, $k = \sigma_l(t)$, and $r = \sigma_r(l)$.*

Right-mixing and pseudo-right-mixing layers are symmetric.

Remark 19 (Sibling maps at equal dimensions)**.** *The budget allocation of Definition 17 sets $\nu_l = \gamma_r$ at pseudo-left-mixing and synthesizing layers (both equal b in (a, b, b, a)); symmetrically $\gamma_l = \nu_r$, both a), so σ_r — and σ_l at synthesizing layers — reduces to the identity: the node index of each child selects the sibling group of the same index. The modular form is stated for completeness, because the caps below can alter the child dimensions after allocation: a discrete leaf, for instance, can be partitioned into at most $|\mathrm{Cat}(X)|$ support pieces, so a leaf layer may end up with fewer nodes than its sibling has groups, and the pairing then wraps around.*

Leaf layers. The leaf layers establish the invariants at the bottom of the circuit. A constrained leaf ($\psi(m) \neq \mathcal{U}$ over variable X) becomes a leaf layer whose support pieces partition $\mathrm{Val}(X)$, each of the ν nodes in a group realizing one piece: the categories of a categorical variable, the quantile splits of a spline leaf (Section 5, initialized at the empirical $1/\nu$-quantiles when data is available), or an equipartition of the support for the remaining distribution families. An unconstrained leaf ($\psi(m) = \mathcal{U}$) instead lets all of its nodes share the single full-support piece and is wrapped in a dense mixture layer — a normalized mixture over all of its group–node components with learnable weights — so that its support is the full $\mathrm{Val}(X)$ and no unit is born without mass (see below).

Dead units and the caps. A unit born without mass — a leaf node with empty support, or a sum node receiving no child combination — is a *dead unit*: it contributes nothing to the circuit, and gradient-based training cannot revive it. The construction enforces three dimensional invariants so that no dead unit is created:

1. *Leaf cap*: a constrained categorical leaf has at most one node per category, $\nu \leq |\mathrm{Cat}(X)|$; partitioning fewer categories into more units would create empty-support units.
2. *Parent cap*: a constrained internal layer has at most as many nodes per group as disjoint children available to each unit, $\nu \leq \nu_l$ at left-mixing and pseudo-left-mixing, $\nu \leq \nu_r$ at right-mixing and pseudo-right-mixing, and $\nu \leq \nu_l\, \nu_r$ at synthesizing layers. Only ν is ever capped, never γ: groups are unconstrained replicas, and capping $\gamma\nu$ jointly would collapse the layer to a single unit and factorize the joint.
3. *Filled assignment*: every sum node receives at least one child combination, by the filled balanced assignment of Definition 15; this requires ν not to exceed the available slots, which the parent cap guarantees dimensionally.

Fairness. The balanced assignment controls how unevenly child combinations are spread over the sum nodes of a layer. With temperature $\tau = 0$ the assignment is uniform random, and some sum nodes receive many more combinations than others; biasing the assignment toward balanced loads ($\tau > 0$) markedly stabilizes training in our experiments, while the jitter term keeps unbalanced — though no longer degenerate — assignments possible.

It remains to verify (I1) and (I2). Both follow by a bottom-up induction over the vtree, with the leaf partitions as the base case.

Theorem 20 (Parent invariant)**.** *Within each group of every constrained layer of the constructed circuit, distinct sum nodes have pairwise disjoint $\psi(m)$-marginal supports.*

Proof. Induction over the height of the vtree node. *Base*: a constrained leaf partitions $\mathrm{Val}(X)$ into its support pieces, so distinct positions are disjoint; an unconstrained leaf carries no constraint. *Step*: let m be constrained and assume the claim for the children. *Left-mixing*: the $\psi(m)$-marginal support of a sum node is the union of the left support pieces it recruits (the right child is unconstrained, so marginalizing the right scope leaves the left factor); distinct sum nodes recruit disjoint sets of left support pieces by the routing, and the pieces themselves are pairwise disjoint by the induction hypothesis, so the unions are disjoint. *Pseudo-left-mixing*: the same argument, using that each left support piece is recruited by exactly one sum node. *Synthesizing*: the $\psi(m)$-support of a sum node is the union of rectangles $\text{piece}_l \times \text{piece}_t$ over its combinations; distinct sum nodes receive disjoint sets of combinations (π is functional), and rectangles over distinct position pairs are disjoint by the induction hypothesis, so the unions are disjoint. □

Theorem 21 (Weak determinism)**.** *Every constrained sum layer of the constructed circuit is weakly $\psi(m)$-deterministic; pseudo-mixing layers are additionally weakly $\psi(m_l) \cup \psi(m_r)$-deterministic.*

Proof. Induction as in Theorem 20; by that theorem, distinct positions within each group of the child layers have pairwise disjoint supports. *Left-mixing*: two children of one sum node either share the left node, in which case their $\psi(m)$-marginal densities coincide (both are the left factor renormalized by the mass of the right child), which weak determinism permits (Definition 4), or they use distinct left nodes at distinct positions, whose supports are disjoint. *Pseudo-left-mixing*: over $\psi(m)$ the same argument applies; over the combined label, two distinct children $(l, t) \neq (l', t')$ either share the left node, in which case their right partners are the nodes at positions $t \neq t'$ of the single group $\sigma_r(l)$, whose support pieces are disjoint, or they differ in the left node and are disjoint on the left. *Synthesizing*: if the two combinations share the left node, their right partners are nodes of the single group $\sigma_r(l)$ at distinct positions; if they share the right node, the symmetric argument applies through σ_l; if they share neither, they differ in position on at least one side. In all cases the children have disjoint supports over $\psi(m_l) \cup \psi(m_r) = \psi(m)$. *Universal*: vacuous. □

Theorem 22 (No dead units)**.** *At initialization, every unit of the constructed circuit has positive mass.*

Proof. Bottom-up induction. *Leaves*: a constrained categorical leaf has one node per category by the leaf cap, and the remaining families split the support into ν nonempty pieces, so every leaf node has nonempty support; an unconstrained leaf mixes full-support components with strictly positive weights. *Internal layers*: the parent cap bounds ν by the number of available child slots, so the filled balanced assignment exists and every sum node receives at least one child combination; all initialized weights are positive and remain so under normalization, and every child unit has positive mass by the induction hypothesis, hence every sum node has positive mass. □

D.4 Pseudo-mixing layers: requirements, cost, and placement

Pseudo-mixing layers are forced by nested determinism scopes (Proposition 7), and they are the one layer type at which the determinism requirement applies twice: a pseudo-left-mixing layer must be weakly deterministic with respect to *both* the labelled scope $\psi(m) = \psi(m_l)$ *and* the combined label $\psi(m_l) \cup \psi(m_r)$, whereas at every other layer type the combined requirement is vacuous (Section 3). The two invariants of Section D.3 are not equally amenable to relaxation. Weak determinism (I1) is what makes conditioning exact (Proposition 6), so it admits no routing workaround; its doubled strength at pseudo-mixing layers is what throttles their wiring — each sum node recruits only its own

set of left support pieces — and, as Section E shows, can cost the circuit precisely the conditionals a causal estimand requires. The parent invariant (I2), by contrast, is needed only so that the *parent* layer can condition on the labelled child exactly, and it can be discharged in two ways.

Discharge at the root. At the root the layer has a single sum node and no parent, so (I2) is trivially satisfied: no disjointness structure is required, and the single sum node may recruit every combination of child positions. Placement at the root is therefore the unconditional discharge, and it is the default in our construction. If several pseudo-mixing layers are required, as nested determinism scopes multiply, the layer placed at the root is chosen according to which conditionals the target queries need to express. This rule is baked into the vtree construction, and it is what precludes a fully automatic pipeline: the placement must be anticipated when the md-labels are designed.

Discharge below the root: route as synthesizing. Below the root, (I2) can instead be discharged on the query side. The routing of Definition 18 assigns *left support pieces* to sum nodes, so the sum nodes are disjoint over the left MD set $\psi(m) = \psi(m_l)$ and the parent can condition on the labelled child exactly. If the target queries will never condition on the labelled child alone, the layer may instead be routed *as synthesizing*: combinations of child positions — $\pi : \{1..\nu_l\} \times \{1..\nu_r\} \to \{1..\nu\}$ — are assigned to sum nodes, which are then disjoint only over the combined label $\psi(m_l) \cup \psi(m_r)$. The sum nodes still have disjoint support for any joint assignment of both children, so conditioning on the two children jointly is unaffected; what is lost is exactness of conditioning solely on the labelled child, since the weight reassignment of Proposition 6 presupposes that each left support piece is recruited by a single sum node. As Section E shows, this relaxation can also buy back expressivity that the left-piece routing surrenders.

Our reference implementation instantiates pseudo-mixing layers with the combination-based routing throughout — a workaround that sidesteps the tighter parent cap and disjointness requirement of Definition 18. In all our experiments this is benign: pseudo-mixing layers occur only at the root, where the layer has a single sum node and the two routings coincide, so the discrepancy has no observable effect. Below the root, the synthesizing routing is correct exactly when no target query conditions on the labelled child alone; otherwise Definition 18 is the correct construction. Pseudo-mixing layers are thus weakly deterministic on both labels by routing, with (I2) discharged at the root or, where the queries permit, below it; the next section studies the expressivity cost that remains.

E A Study on Expressivity

As discussed in Section 8, T-ID presumes expressivity: it establishes that each conditional in the derivation is tractable on the given circuit, but not that the circuit can faithfully express it. This appendix studies when the conditionals that T-ID emits are expressible by circuits carrying the required weak determinisms. In Section 4 we showed that the construction of Wang & Kwiatkowska (2023) allowed the tractable computation of $P(Y|do(X))$ for the backdoor adjustment estimand, and that even in that setting, it could only *express* the estimand given a specific vtree structure. Our construction mitigated the computational failure as well as the representational failure caused by minimal diversity. When sufficiently constrained by determinisms, however, our circuits too can fail to express certain conditionals. We attribute this to an inherent loss of expressivity that accompanies the tractability trade-off and is intrinsic to probabilistic circuits.

Expressivity is lost exclusively in pseudo-mixing layers, and the losses all stem from one wiring limitation, which we illustrate on the collider-door SCM. For the collider-door SCM, the interventional distribution $P(Y|do(X))$ is identified by the estimand $\int_{Z,M} P(M|X,Z) \int_{X'} P(X',Z)P(Y|X',Z,M)$, so a circuit computing it exactly must carry weak $\{X,Z\}$- and $\{X,Z,M\}$-determinism. The circuit realizing this requirement is not unique, however: it may adhere to different vtrees, and some vtrees support the computation of this estimand better than others (Figure 5). Under the left-piece routing of Definition 18, every sum node of a pseudo-left-mixing layer contains, for each of its assigned left support pieces, every right support piece exactly once; it therefore ranges over *all* support pieces of the right child, and only the left child's pieces modulate which sum nodes activate — and hence which groups of the parent layer answer. A dependence that must manifest as variation *in the right child* has no wiring through the layer and is lost.

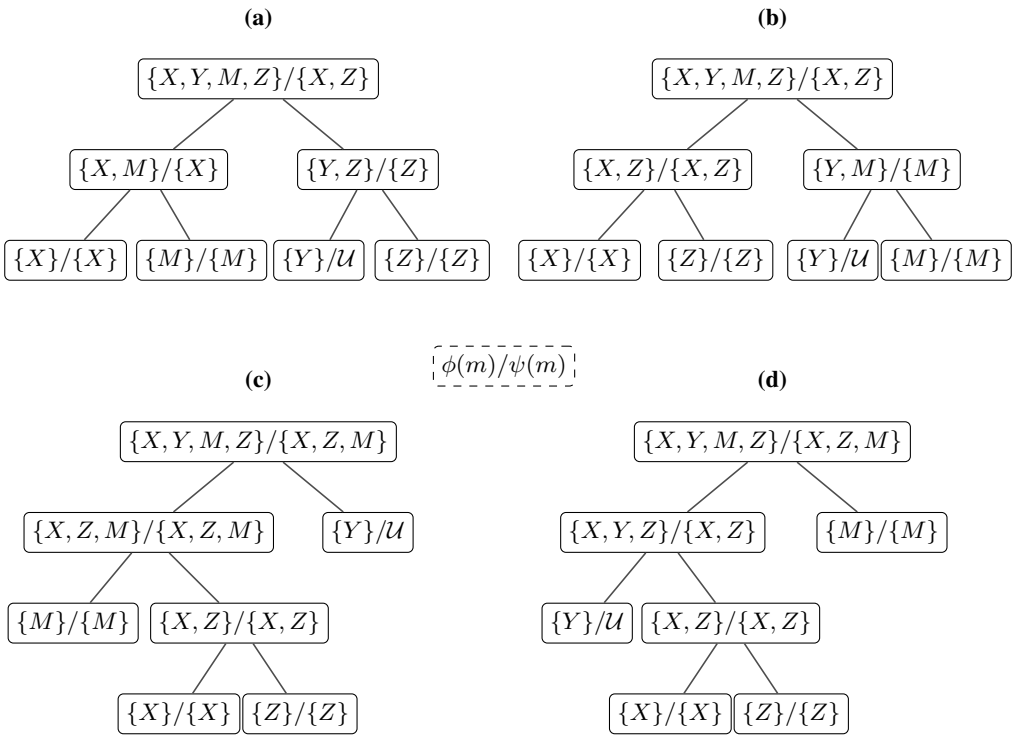


Figure 5: Vtrees of the collider-door setting, each node annotated as $\phi(m)/\psi(m)$: $\phi(m)$ is the node scope and $\psi(m)$ its MD label. (a), (d): the root separates (X, M) from (Y, Z); (b), (c): the root separates (X, Y, Z) from M.

Consider the vtree of Figure 5(a). In the left-pseudo-mixing layer at the $\{X, M\}$ node, distinct support pieces of X activate distinct nodes of the layer (no two nodes share the same X-support), each answered by a — potentially distinct — group of the $\{Y, Z\}$ layer realizing a complementary density, so the dependence of the circuit on X is carried by the group structure of the parent. But all sum nodes of the $\{X, M\}$ layer range over the same M support pieces, so the activation of the layer does not vary with M, and the circuit becomes independent of M as well. Consequently $P(Y|X, Z, M)$ is expressed as $P(Y|X, Z)$, and the identified estimand reduces to the marginal density of Y:

$$\int_{Z,M} P(M|X,Z) \int_{X'} P(X',Z)P(Y|X',Z,M) = \int_{Z,M} P(M|X,Z) \int_{X'} P(X',Z)P(Y|X',Z)$$
$$= \int_{Z} P(Y,Z) \int_{M} P(M|X,Z) = P(Y).$$

In the vtree of Figure 5(d), the root is itself pseudo-left-mixing, separating $\{X, Y, Z\}$ from $\{M\}$, and a pseudo-left-mixing layer is minimally diverse against its right child: all left support pieces of a sum node come from a single group, of which there is only one at the root, so the scope of the left child is modelled as independent of that of the right child (Remark 10). Here this means in particular that the effect of M on Y is not captured, and $P(Y|X, Z, M)$ is modelled as $P(Y|X, Z)$, leading to the same failure. The vtree of Figure 5(c) fails as well: the effect of M on Y is not expressed, for the same reason as in (a).

The collapse in (a) is a property of the routing, not of the vtree. If the $\{X, M\}$ layer is routed *as synthesizing* — combinations of X- and M-support pieces assigned to sum nodes, as permitted below the root whenever no query conditions on the labelled child alone (Section D.4) — then distinct M support pieces activate distinct sum nodes of the $\{X, M\}$ layer, each answered by a distinct group of the $\{Y, Z\}$ layer, and the effect of M on Y is expressed after all. The price is the parent invariant: the root can no longer condition on the $\{X, M\}$ child exactly. Whether that price is payable is a property of the target queries, and discharging the parent invariant — at the root, or

below it where the queries permit — is what makes the mixed frontdoor and collider-door settings fittable in the first place.

Consider finally the vtree of Figure 5(b). It loses the effect of M on X and on Z, while still expressing the effect of X on M and the joint effect of X, Z, and M on Y, which suffices to express the identified estimand. It is exactly the vtree used in our experiments, whose results are reported in Section 7. These observations motivate the following rule of thumb for vtree construction: in the presence of a single nested determinism (as in the collider-door SCM, the frontdoor setting, or linear-time backdoor (Wang et al., 2024)), the only pseudo-mixing layer should be placed at the root, separating the variables of the smaller determinism from the remaining variables; if it cannot be placed at the root, it should be routed as synthesizing whenever the target queries never condition on its labelled child alone. This rule also governed the vtree construction of the mixed frontdoor experiment reported in Section H.2.

Multiple nested determinisms. The rule of thumb presumes a single nested determinism. When several determinism scopes are nested, each nested pair forces a pseudo-mixing layer (Proposition 7), and the construction becomes genuinely difficult to automate: each pseudo-mixing layer must be individually classified against the target queries — placed at the root, routed as synthesizing, or built with the left-piece routing of Definition 18 if some query does condition on its labelled child alone — and the choices interact, since the conditionals one layer is asked to express depend on where the other layers sit. The vtree, the placement, and the per-layer routing must then be co-designed by hand.

Regularity is not expressivity-optimal. The labelling algorithm of Wang & Kwiatkowska (2023) was claimed to always return a regular md-vtree (their Proposition 4), and regularity was believed to be without loss: every md-vtree admits a regular relabelling that implies the same determinisms while accommodating a weakly larger class of circuits (their Theorem 2). Both claims fail once a concrete determinism requirement is fixed.

Consider the vtree of Figure 6 and the determinism requirements $S = \{\{X, M\}, \{X, Z\}, \{X\}\}$, as arise for instance when a query must condition on (X, M), on (X, Z), and on X alone. Any md-labelling implying S (their Definition 12) must assign the $\{M, Z\}$ node a label contained in $\{X, M\} \cap \{X, Z\} = \emptyset$, and the $\{Y, M, Z\}$ node likewise, rendering both layers trivial. Regularity at the $\{M, Z\}$ node then forces $\psi(\{M\}) = \emptyset$ or $\psi(\{Z\}) = \emptyset$: every regular md-labelling of this vtree implying S collapses the M or Z leaf layer to a single component, modelling that variable as a context-free factor, independent of the remaining variables. No regular md-labelling of this vtree can both imply S and express dependencies through M and Z.

What is more, the algorithm itself — which intersects at each node the scopes of the requirements meeting that node with the node's own scope — assigns the labels $\psi(\{X, Y, M, Z\}) = \{X\}$; $\psi(\{X\}) = \{X\}$; $\psi(\{Y, M, Z\}) = \emptyset$; $\psi(\{Y\}) = \mathcal{U}$; $\psi(\{M, Z\}) = \emptyset$; $\psi(\{M\}) = \{M\}$; $\psi(\{Z\}) = \{Z\}$. Regularity (Definition 2) requires the label of every internal node m with children m_l, m_r to satisfy $\psi(m) = \psi(m_l)$, $\psi(m) = \psi(m_r)$, or $\psi(m) = \psi(m_l) \cup \psi(m_r)$. This fails at the $\{M, Z\}$ node: the node is assigned the label $\emptyset$ while its children carry the labels $\{M\}$ and $\{Z\}$, and $\emptyset$ equals neither $\{M\}$, nor $\{Z\}$, nor $\{M\} \cup \{Z\} = \{M, Z\}$. Proposition 4 of Wang & Kwiatkowska (2023), which claims the output of the algorithm to be regular, is therefore false. What this implies for the construction of such deterministic circuits is left as an open question.

F Properties of Log-Linear Spline Leaves

We collect the properties of the log-linear spline leaves of Definition 23, establishing requirements (R1)–(R4) and desiderata (D1)–(D3), together with the resulting closed-form quantities and implementation remarks. We begin by restating the construction, introduced in Section 5, for convenience.

Definition 23 (Log-linear spline leaf; restated from Section 5)**.** *A* log-linear spline leaf group *over X with K components is parameterized by an anchor ξ_1 and log-heights $\lambda_1, \ldots, \lambda_{K-1} \in \mathbb{R}$ setting the shared boundary heights $h_k := e^{\lambda_k} > 0$ of (R2), with the convention $h_0 = h_K := 0$: the log-slope is $\theta_i := h_i - h_{i-1}$, and per-component normalization fixes the interior widths*

$$\Delta_i := \frac{\lambda_i - \lambda_{i-1}}{e^{\lambda_i} - e^{\lambda_{i-1}}} = \frac{1}{L(h_{i-1}, h_i)}, \qquad \text{where } L(a, b) := \frac{b - a}{\log b - \log a}$$

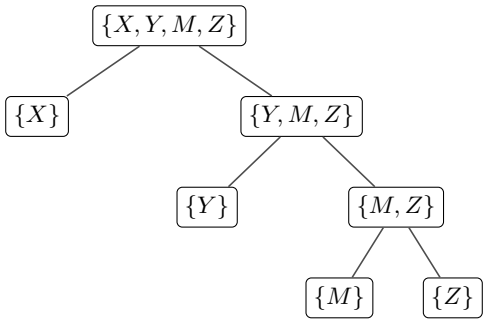


Figure 6: The counterexample vtree, drawn with scopes only. For the requirements $S = \{\{X, M\}, \{X, Z\}, \{X\}\}$, no regular md-labelling implying S avoids trivializing a leaf layer, and the labelling algorithm of Wang & Kwiatkowska (2023) assigns the $\{M, Z\}$ node the label $\emptyset$, violating regularity.

is the logarithmic mean*, extended by continuity to $L(h, h) = h$, i.e. $\Delta_i = 1/h_{i-1}$ when $h_i = h_{i-1}$; the splits are $\xi_i := \xi_1 + \sum_{j=2}^{i} \Delta_j$. With $I_i := (\xi_{i-1}, \xi_i]$, $\xi_0 := -\infty$, $\xi_K := +\infty$, the components are the log-linear (piecewise exponential) densities*

$$p_i(x) = h_{i-1}\, e^{\theta_i (x - \xi_{i-1})}\, \mathbf{1}\{x \in I_i\}, \qquad i = 2, \ldots, K-1, \tag{5}$$

while the two outer components are the exponential tails forced by integrability and normalization,

$$p_1(x) = h_1\, e^{h_1 (x - \xi_1)}\, \mathbf{1}\{x \le \xi_1\}, \qquad p_K(x) = h_{K-1}\, e^{-h_{K-1}(x - \xi_{K-1})}\, \mathbf{1}\{x > \xi_{K-1}\}. \tag{6}$$

Proposition 24 (Validity)**.** *For every $(\xi_1, \lambda_1, \ldots, \lambda_{K-1}) \in \mathbb{R}^K$, the components of Definition 23 are proper densities satisfying* (R1)–(R4)*, and the patched marginal p_X is a continuous, normalized density on $\mathbb{R}$ with exponential tails of rates h_1 (left) and h_{K-1} (right). Moreover, the splits and the densities equation 5 are smooth functions of the parameters, so* (D1) *holds.*

Proof. Each p_i is strictly positive on I_i and vanishes outside it, since $h_{i-1} > 0$ and the exponential is positive; the $I_i = (\xi_{i-1}, \xi_i]$ are pairwise disjoint because $\xi_1 < \cdots < \xi_{K-1}$ and cover $\mathbb{R}$ with $\xi_0 = -\infty$, $\xi_K = +\infty$, so (R1) holds. Each p_i is normalized: for an interior piece, the defining identity of the widths reads $\theta_i \Delta_i = \log h_i - \log h_{i-1}$, hence $e^{\theta_i \Delta_i} = h_i / h_{i-1}$ and

$$\int_{\xi_{i-1}}^{\xi_i} p_i(x)\, dx \;=\; h_{i-1}\, \frac{e^{\theta_i \Delta_i} - 1}{\theta_i} \;=\; \frac{h_i - h_{i-1}}{\theta_i} \;=\; 1;$$

for the tails, $\int_{-\infty}^{\xi_1} h_1 e^{h_1 (x - \xi_1)}\, dx = 1$ and $\int_{\xi_{K-1}}^{\infty} h_{K-1} e^{-h_{K-1}(x - \xi_{K-1})}\, dx = 1$, which gives (R4). For (R2), observe that at each interior split the one-sided limits coincide with the shared boundary height: $p_i(\xi_i) = h_{i-1} e^{\theta_i \Delta_i} = h_i = \lim_{x \downarrow \xi_i} p_{i+1}(x)$, where the last equality uses that piece $i + 1$ starts at its initial height h_i at its left endpoint ξ_i. Hence the patched marginal is continuous at every knot and smooth elsewhere. For (D1), the logarithmic mean in the width identity extends smoothly to the diagonal, $L(h, h) = h$, and is smooth and strictly positive on $(0, \infty)^2$; hence each Δ_i, each ξ_i, and each p_i is a smooth function of $(\xi_1, \lambda_1, \ldots, \lambda_{K-1})$, and (R1)–(R4) hold by construction for every parameter value. □

Proposition 25 (Tractable integrals and moments)**.** *The components of Definition 23 satisfy* (D2) *and* (D3)*. Writing $F_i(x) := \int_{\xi_{i-1}}^{x} p_i(t)\, dt \in [0, 1]$ for the cumulative mass of component i up to x:*

$$F_i(x) \;=\; \begin{cases} e^{h_1 (x - \xi_1)}, & i = 1, \\ h_{i-1} \left(e^{\theta_i (x - \xi_{i-1})} - 1 \right) / \theta_i, & 2 \le i \le K-1, \\ 1 - e^{-h_{K-1}(x - \xi_{K-1})}, & i = K, \end{cases} \tag{7}$$

$$F_i^{-1}(v) \;=\; \begin{cases} \xi_1 + \log(v)/h_1, & i = 1, \\ \xi_{i-1} + \log\!\left(1 + \theta_i v / h_{i-1}\right) / \theta_i, & 2 \le i \le K-1, \\ \xi_{K-1} - \log(1 - v)/h_{K-1}, & i = K, \end{cases} \tag{8}$$

with the understanding $F_i(x) = h_{i-1}(x - \xi_{i-1})$ and $F_i^{-1}(v) = \xi_{i-1} + v/h_{i-1}$ when $\theta_i = 0$. The marginal CDF is $F(x) = \frac{1}{K}\big(i - 1 + F_i(x)\big)$ for $x \in I_i$. As for moments, the tails are exponential: $\mathbb{E}_{p_1}[X] = \xi_1 - 1/h_1$, $\mathrm{Var}_{p_1}[X] = 1/h_1^2$, and $\mathbb{E}_{p_K}[X] = \xi_{K-1} + 1/h_{K-1}$, $\mathrm{Var}_{p_K}[X] = 1/h_{K-1}^2$. For an interior piece, with $T := X - \xi_{i-1}$,

$$\mathbb{E}[T] \;=\; \frac{\Delta_i h_i - 1}{\theta_i}, \qquad \mathbb{E}[T^2] \;=\; \frac{\Delta_i^2 h_i - 2\,\mathbb{E}[T]}{\theta_i}, \tag{9}$$

and $\mathbb{E}[X] = \xi_{i-1} + \mathbb{E}[T]$, $\mathrm{Var}[X] = \mathbb{E}[T^2] - \mathbb{E}[T]^2$; as $h_i \to h_{i-1}$ these reduce to the moments of the uniform distribution on I_i, namely $\mathbb{E}[T] \to \Delta_i/2$ and $\mathrm{Var}[T] \to \Delta_i^2/12$. Moments of the marginal follow as $\mathbb{E}_{p_X}[X^r] = \frac{1}{K}\sum_i \mathbb{E}_{p_i}[X^r]$.

Proof. All piece CDFs follow by integrating the exponentials equation 5–equation 6, using $e^{\theta_i \Delta_i} = h_i/h_{i-1}$ to see $F_i(\xi_i) = 1$; inverting is elementary. For the interior moments, integrate by parts: with $\theta = \theta_i$, $\Delta = \Delta_i$, $\mathbb{E}[T] = \int_0^\Delta t\, h_{i-1} e^{\theta t} dt = \big[t h_{i-1} e^{\theta t}/\theta\big]_0^\Delta - \frac{1}{\theta}\int_0^\Delta h_{i-1} e^{\theta t} dt = (\Delta h_i - 1)/\theta$, since the integral is the total mass 1; the same step applied once more gives $\mathbb{E}[T^2] = \big[t^2 h_{i-1} e^{\theta t}/\theta\big]_0^\Delta - \frac{2}{\theta}\mathbb{E}[T] = (\Delta^2 h_i - 2\,\mathbb{E}[T])/\theta$. The tail moments are those of an exponential distribution, and the uniform limits follow from the smooth extension at $\theta_i = 0$. □

Remark 26 (Exact sampling)**.** *Since the marginal over X is the uniform combination of the components, sampling from p_X reduces to uniform component selection plus inversion: draw $u \sim \mathrm{U}(0,1)$, set $i = \lceil Ku \rceil$, and return $x = F_i^{-1}(Ku - i + 1)$ via equation 8. Each leaf evaluation is likewise in closed form, $\log p_X(x) = \lambda_{i-1} + \theta_i(x - \xi_{i-1}) - \log K$ for $x \in I_i$.*

Remark 27 (Parameter tying across groups)**.** *In our implementation the leaf parameters are tied across the γ_m groups of the layer (§2), so nodes at the same position share the same component and support in every group, as required by the computational representation. More generally, one may tie only the split locations across groups; the width identity of Definition 23 then constrains each group's height profile to the widths, and a single boundary height of a group determines all remaining ones recursively, since $\Delta_i L(h_{i-1}, \cdot) = 1$ has a unique solution ($L(h_{i-1}, \cdot)$ is strictly increasing onto $(0, \infty)$).*

Remark 28 (Numerical stability)**.** *All quantities are evaluated in the log domain. The width Δ_i is computed from the relative height difference $z_i := (h_i - h_{i-1})/h_{i-1}$ as $\Delta_i = \log 1\mathrm{p}(z_i)/(h_{i-1} z_i)$, switching to the Taylor branch $1 - z_i/2 + z_i^2/3$ when $|z_i|$ is small, and the same continuous extension is used for F_i, F_i^{-1} and equation 9 near $\theta_i = 0$, so that gradients remain well defined throughout training.*

Additional proofs.

Proposition 29. *Let $p_1, \ldots, p_K$ satisfy (R1) and (R4), and let F_X be the CDF of p_X. Then every component contributes mass exactly $1/K$ to the marginal, and the CDF is pinned at the boundaries:*

$$F_X(\xi_k) \;=\; \frac{k}{K}, \qquad k = 0, \ldots, K, \tag{10}$$

that is, ξ_k is the (k/K)-quantile of the modelled marginal.

Proof of Proposition 29. Since the supports are pairwise disjoint, at most one term of p_X is non-zero at any point, hence $\int_{I_i} p_X(x)\, d\mu_X(x) = \frac{1}{K}\int_{I_i} p_i(x)\, d\mu_X(x) = \frac{1}{K}$ by (R4), and $\bigcup_{i \le k} I_i = (-\infty, \xi_k]$ by the convention $I_i = (\xi_{i-1}, \xi_i]$. Summing over $i \le k$ gives $F_X(\xi_k) = k/K$. □

Remark 30 (Equal mass is with respect to the modelled marginal)**.** *Proposition 29 is exact, but the quantiles are those of the* modelled *marginal p_X, and this cannot be strengthened: since every component is constrained to unit mass, the knots cannot additionally be pinned at the quantiles of the true marginal — the two requirements are compatible only in the perfect-model limit. With limited capacity (say, too few knots K), likelihood maximization relocates the knots to where they serve the fit, and under the true distribution the components then contribute non-uniform mass; insisting on true quantiles while letting the boundary heights float would instead force mismatched one-sided limits at the splits and break (R2). Neither tension arises in our parameterization, because the heights are the free parameters and the splits are derived from them: continuity of p_X rests on*

the shared boundary heights, which hold for every parameter value by construction, not on any quantile-matching property. Initializing the splits approximately at the empirical $1/K$-quantiles is accordingly just a sensible starting point, not a constraint the fit must preserve.

Proposition 31. *Let $\widetilde{F}$ be strictly increasing and C^1 on $\mathbb{R}$ with $\lim_{x\to-\infty}\widetilde{F}(x) = 0$, $\lim_{x\to\infty}\widetilde{F}(x) = 1$, and $\widetilde{F}(\xi_k) = k/K$ at knots $\xi_0 := -\infty < \xi_1 < \cdots < \xi_{K-1} < \xi_K := +\infty$. Define, with $I_i := (\xi_{i-1}, \xi_i]$,*

$$p_i(x) \;:=\; K\,\widetilde{F}'(x)\,\mathbf{1}\{x \in I_i\}, \qquad i = 1, \ldots, K. \tag{11}$$

Then $p_1, \ldots, p_K$ satisfy (R1), (R2) and (R4). If, furthermore, $\widetilde{F}$ and the knots are smooth functions of an unconstrained parameter vector, with strict monotonicity and the pinned values holding for every parameter value, then (D1) holds as well. Conversely, any $p_1, \ldots, p_K$ satisfying (R1)–(R4) arise in this way from their marginal CDF F_X.

Proof of Proposition 31. (R1) Strict monotonicity gives $\widetilde{F}' > 0$ on $\mathbb{R}$, so p_i is strictly positive on I_i and vanishes outside it; the $I_i = (\xi_{i-1}, \xi_i]$ are pairwise disjoint and cover $\mathbb{R}$. *(R4)* Integrating over I_i and using the pinned values, $\int_{I_i} p_i = K\big(\widetilde{F}(\xi_i) - \widetilde{F}(\xi_{i-1})\big) = K\big(\frac{i}{K} - \frac{i-1}{K}\big) = 1$ (the integral is unaffected by the boundary convention). *(R2)* Since $\widetilde{F}$ is C^1, $\widetilde{F}'$ is continuous, so the one-sided limits at each shared boundary agree: $\lim_{x\uparrow\xi_i} p_i(x) = K\widetilde{F}'(\xi_i) = \lim_{x\downarrow\xi_i} p_{i+1}(x)$. *(D1)* Smoothness of $\widetilde{F}$, $\widetilde{F}'$, and the knots in the parameters carries over to the p_i, and (R1)–(R4) hold for every parameter value by assumption. For the converse, given $p_1, \ldots, p_K$ satisfying (R1)–(R4), the CDF F_X of $p_X := \frac{1}{K}\sum_i p_i$ is strictly increasing, C^1 by (R2), and satisfies $F_X(\xi_k) = k/K$ by Proposition 29; and $KF_X'(x)\mathbf{1}\{x \in I_i\} = p_i(x)$. □

Remark 32 (Widths and the logarithmic mean)**.** *As the width identity in Definition 23 shows, $\Delta_i = 1/L(h_{i-1}, h_i)$; since $L(a, b) > 0$ for all $a, b > 0$, the widths are positive for all parameter values and the split locations are always strictly ordered.*

G T-ID: The Complete Algorithm and Its Guarantees

This appendix presents T-ID in full. The algorithm runs the identification recursion of Shpitser & Pearl (2006) line by line, but every derivation step is *emitted* as the corresponding circuit operation and *admitted* only if its tractability requirements are met. The state carried through the recursion is an *uncompiled circuit* $\mathfrak{P} = (\mathcal{C}, \mathcal{D}, K)$ consisting of an *emitted composition* $\mathcal{C}$ — a symbolic expression in the circuit operators (marginalization, conditioning, product) applied to a *base circuit* $\mathcal{C}_0$ representing the observational distribution $P(\mathbf{V})$ — a *determinism family* $\mathcal{D} \subseteq 2^{\mathbf{V}}$ recording the scopes on which the represented function is weakly marginally deterministic (Definition 5), and a *complexity exponent* $K \geq 1$. We extend the scope notation to uncompiled circuits in the natural way: $\phi(\mathfrak{P})$ is the set of variables over which the estimand represented by $\mathfrak{P}$ is defined. Evaluating a returned composition on a base circuit $\mathcal{C}_0$ that provides the determinisms asked for along the way yields the estimand exactly, in time $\mathcal{O}(|\mathcal{C}_0|^K)$.

Atomic factors. The currency of the simplification procedure is the *atomic factor* $F = P(\mathbf{N} \mid \mathbf{D})$, a conditional with *head* $\mathbf{N}$ and *conditioning set* $\mathbf{D}$, two disjoint scopes. Whenever a recursive call returns a plain conditional of the observational distribution $P(\mathbf{V})$ — as opposed to a composite estimand such as an interventional marginal — it is tagged with its factor, together with the *context* variables it was derived under, so that call sites can simplify without inspecting the emitted composition. Only conditionals of the observational distribution are tagged. A marginal of a rebased estimand is an *interventional* marginal (e.g. the frontdoor $\sum_{x'} P(x')P(y \mid x', m)$) and must not be re-expressed as a single factor of $P(\mathbf{V})$.

Cost model. We use the size bounds for circuit composition of Wang et al. (2024). Two smooth and decomposable circuits are *compatible* if their product nodes partition the shared scope in the same way (Wang et al., 2024, Def. 7). The product of compatible circuits of sizes n_1, n_2 is exact and computable in $\mathcal{O}(n_1 n_2)$ (Wang et al., 2024, Thm. 2). They are *support-compatible* if the parts of the two circuits overlapping the shared scope are isomorphic with pairwise disjoint child

supports on the shared scope (Wang et al., 2024, Def. 8), and then the product is exact and computable in $\mathcal{O}(\max(n_1, n_2))$ (Wang et al., 2024, Thm. 3). Marginalization is exact and linear-time on smooth and decomposable circuits (Wang et al., 2024, Thm. 1), and conditioning is exact and linear-time on weakly deterministic circuits by Proposition 6. Every uncompiled circuit handled by T-ID is derived from the same base circuit $\mathcal{C}_0$ and respects its vtree, so any two operands are compatible and every product is exact. Support-compatibility is inherited from the base circuit along the pipeline of operations, as in the causal case studies of Wang et al. (2024, App. B.2): the isomorphism between the shared parts survives marginalization outside the shared scope, conditioning, and products. An operand with exponent K_i therefore has size $\mathcal{O}(|\mathcal{C}_0|^{K_i})$, and the two product bounds read $\mathcal{O}(|\mathcal{C}_0|^{K_1+K_2})$ and $\mathcal{O}(|\mathcal{C}_0|^{\max(K_1,K_2)})$, respectively. These are worst-case costs of the standard constructions of Wang et al. (2024) — linear marginalization, linear support-compatible product, quadratic compatible product — which T-ID adopts as its implementations of the operators.

Determinism bookkeeping. Write $\mathrm{IsDet}(\mathcal{D}, \mathbf{Q})$ for the membership test $\mathbf{Q} \in \mathcal{D}$. Exact membership is required, subset coverage is not sufficient. The three operations update the uncompiled circuit as follows. *Marginalization* $\mathrm{MARG}(\mathfrak{P}; \mathbf{X})$ returns the composition with $\mathbf{X}$ marginalized out. It is exact and linear-time, and *wipes* every determinism that intersects the removed set, $\mathcal{D} \leftarrow \{\mathbf{Q} \in \mathcal{D} : \mathbf{Q} \cap \mathbf{X} = \emptyset\}$. K is unchanged. *Conditioning* $\mathrm{COND}(\mathfrak{P}; \mathbf{N} \mid \mathbf{D})$ marginalizes away $\phi(\mathfrak{P}) \setminus (\mathbf{N} \cup \mathbf{D})$ and then requires the conditioning set to survive as an exact member of the family. By Proposition 6 the operation is then exact and linear-time, so K is unchanged. If the requirement fails, T-ID raises a *tractability error*: the query is identifiable, but not evaluable on the given circuit, and the offending scope is reported. When T-ID is run in *collection mode* (Section G.3 below), the membership check is instead skipped and the conditioning set is recorded into an output set. Collection mode never raises TractabilityError. *Products* $\mathrm{PROD}(\mathfrak{P}_1, \mathfrak{P}_2)$ of circuits derived from the same base circuit are exact, since the operands are compatible. The product inherits determinisms by the single-pair rule

$$\mathrm{ProdDet}(\mathbf{Q}_1, \mathbf{Q}_2; \mathbf{A}) = \begin{cases} \mathbf{Q}_1, & \mathbf{Q}_1 = \mathbf{Q}_2 \subseteq \mathbf{A}, \\ \mathbf{Q}_1 \cup \mathbf{Q}_2, & \mathbf{Q}_1 \supseteq \mathbf{A} \text{ and } \mathbf{Q}_2 \supseteq \mathbf{A}, \\ \text{none}, & \text{otherwise}, \end{cases} \tag{12}$$

where $\mathbf{A} = \phi(\mathfrak{P}_1) \cap \phi(\mathfrak{P}_2)$ is the shared scope. This is the MD-calculus product rule (Wang & Kwiatkowska, 2023, Thm. 3): identical determinisms inside the shared region are kept, and a union is claimed only when both operands are deterministic over sets covering all of $\mathbf{A}$. T-ID uses this rule as the certification test for support-compatibility: if a determinism over the shared scope is certified, the product is charged the linear bound, $K = \max(K_1, K_2)$. Otherwise the product is charged the quadratic bound, $K = K_1 + K_2$.

G.1 The T-ID recursion

Algorithm 1 gives T-ID. It takes a query $\mathbf{y}$, an intervention $\mathbf{x}$, an uncompiled circuit $\mathfrak{P}$ over the base circuit, and the graph $\mathcal{G}$, and returns the uncompiled circuit for $P(\mathbf{y} \mid \mathrm{do}(\mathbf{x}))$. We write $\mathcal{G} \setminus \mathbf{x}$ for the graph with the nodes $\mathbf{x}$ removed, $\mathcal{G}[\mathbf{Z}]$ for the subgraph induced by $\mathbf{Z}$, and $\mathrm{C}(\mathcal{G})$ for the c-components (maximal bidirected-connected districts) of $\mathcal{G}$.

The recursion follows the same case analysis as the ID algorithm of Shpitser & Pearl (2006), so identification behaviour is unchanged. What T-ID adds is the emitted composition, the determinism check inside COND, and the exponent bookkeeping inside MULTIPLYFACTORS. The c-component factorization step is the only place where factors are *simplified* before conditioning: atomic factors are merged pairwise into coarser factors with the same semantics whenever the causal graph licenses it, which both reduces the number of conditioning operations and weakens the determinisms they require — e.g. $P(Y \mid X, Z, W) \cdot P(X \mid Z)$ simplifies to $P(Y, X \mid Z, W)$ when $X \perp\!\!\!\perp W \mid Z$, requiring weak $\{Z, W\}$-determinism instead of $\{X, Z, W\}$- and $\{Z\}$-determinism.

Minimal factors: BUILDCONDSET. Given a target set $\mathbf{T}$, order the nodes of $\mathcal{G}$ topologically and let $\mathrm{preds}(v)$ denote the predecessors of v. For each $v \in \mathbf{T}$ compute a minimal subset $\tilde{\pi}(v) \subseteq \mathrm{preds}(v)$ with $v \perp\!\!\!\perp \mathrm{preds}(v) \setminus \tilde{\pi}(v) \mid \tilde{\pi}(v)$ — the minimal d-separating predecessor sets of the ID algorithm's BUILD_COND_SET step — form the atomic factors $P(v \mid \tilde{\pi}(v))$, simplify them with SIMPLIFYFACTORS, materialize each surviving factor by COND, and return their MULTIPLYFACTORS product.

Algorithm 1 T-ID($\mathbf{y}$, $\mathbf{x}$, $\mathfrak{P}$, $\mathcal{G}$). Returns the uncompiled circuit $(\mathcal{C}, \mathcal{D}, K)$ for $P(\mathbf{y} \mid \mathrm{do}(\mathbf{x}))$.

Require: query $\mathbf{y}$, intervention $\mathbf{x}$, uncompiled circuit $\mathfrak{P}$, graph $\mathcal{G}$
if $\mathbf{x} = \emptyset$ **then return** $\mathrm{MARG}(\mathfrak{P}; \mathbf{V} \setminus \mathbf{y})$, tagged $P(\mathbf{y} \mid \emptyset)$ if $\mathfrak{P}$ represents the observational distribution
end if
if $\mathbf{V} \neq \mathrm{An}(\mathbf{y})_{\mathcal{G}}$ **then**
  $\mathfrak{P}' \leftarrow \mathrm{MARG}(\mathfrak{P}; \mathbf{V} \setminus \mathrm{An}(\mathbf{y})_{\mathcal{G}})$ ▷ only graph nodes are marginalized
  return T-ID($\mathbf{y}$, $\mathbf{x} \cap \mathrm{An}(\mathbf{y})_{\mathcal{G}}$, $\mathfrak{P}'$, $\mathcal{G}[\mathrm{An}(\mathbf{y})_{\mathcal{G}}]$)
end if
$\mathbf{W} \leftarrow (\mathbf{V} \setminus \mathbf{x}) \setminus \mathrm{An}(\mathbf{y})_{\mathcal{G}_{\overline{\mathbf{x}}}}$
if $\mathbf{W} \neq \emptyset$ **then return** T-ID($\mathbf{y}$, $\mathbf{x} \cup \mathbf{W}$, $\mathfrak{P}$, $\mathcal{G}$)
end if
$\{\mathbf{R}_1, \ldots, \mathbf{R}_k\} \leftarrow \mathrm{C}(\mathcal{G} \setminus \mathbf{x})$
if $k > 1$ **then**
  for $i = 1, \ldots, k$ **do**
    $\mathfrak{P}_i \leftarrow$ T-ID($\mathbf{R}_i$, $\mathbf{V} \setminus \mathbf{R}_i$, $\mathfrak{P}$, $\mathcal{G}$)
  end for
  let $\{F_i\}$ be the atomic factors and $\{\mathfrak{P}_j\}$ the non-atomic circuits among $\{\mathfrak{P}_i\}$
  $\{F'_r\} \leftarrow$ SIMPLIFYFACTORS($\{F_i\}, \mathcal{G}$) ▷ merge factors via d-separations
  return $\mathrm{MARG}\big(\text{MULTIPLYFACTORS}\big(\{\mathrm{COND}(\mathfrak{P}; F'_r)\}_r \cup \{\mathfrak{P}_j\}_j\big); \mathbf{V} \setminus (\mathbf{y} \cup \mathbf{x})\big)$
end if
let $\mathbf{R}$ be the unique element of $\mathrm{C}(\mathcal{G} \setminus \mathbf{x})$; $\{C_1, \ldots, C_m\} \leftarrow \mathrm{C}(\mathcal{G})$
if $m = 1$ **and** $C_1 = \mathbf{V}$ **then throw** IdentificationError ▷ a hedge witnesses non-identifiability
end if
if $\mathbf{R} \in \mathrm{C}(\mathcal{G})$ **then return** $\mathrm{MARG}(\text{BUILDCONDSET}(\mathfrak{P}, \mathbf{R}, \mathcal{G}); \mathbf{R} \setminus \mathbf{y})$
end if
if $\exists\, C \in \mathrm{C}(\mathcal{G})$ s.t. $\mathbf{R} \subset C$ **then**
  $\mathfrak{P}_C \leftarrow$ BUILDCONDSET($\mathfrak{P}, C, \mathcal{G}$)
  return T-ID($\mathbf{y}$, $\mathbf{x} \cap C$, $\mathfrak{P}_C$, $\mathcal{G}[C]$)
end if
throw IdentificationError

G.2 HELPER PROCEDURES

Algorithm 2 materializes a conditional factor from the uncompiled circuit, MULTIPLYFACTORS (Algorithm 3) forms exact products while certifying support-compatibility and accumulating the complexity exponent, and SIMPLIFYFACTORS (Algorithm 4) merges atomic factors wherever the graph licenses it.

Algorithm 2 Cond($\mathfrak{P}$, $F = P(\mathbf{N} \mid \mathbf{D})$). Materializes the conditional F from $\mathfrak{P}$.

Require: uncompiled circuit $\mathfrak{P}$ with $\mathbf{N} \cup \mathbf{D} \subseteq \phi(\mathfrak{P})$, factor F
$\mathbf{M} \leftarrow \phi(\mathfrak{P}) \setminus (\mathbf{N} \cup \mathbf{D})$
$\mathfrak{P}' \leftarrow \mathrm{MARG}(\mathfrak{P}; \mathbf{M})$ ▷ $\mathcal{D}$ loses every scope meeting $\mathbf{M}$
if $\mathbf{D} = \emptyset$ **then return** $\mathfrak{P}'$ tagged with F ▷ a marginal conditional needs no determinism
end if
if $\mathbf{D} \notin \mathcal{D}(\mathfrak{P}')$ **then throw** TractabilityError($\mathbf{D}$) ▷ identifiable, but not evaluable on this circuit
end if
return the circuit for $P(\mathbf{N} \mid \mathbf{D})$ over $\mathbf{N} \cup \mathbf{D}$, tagged with F, determinisms $\mathcal{D}(\mathfrak{P}')$, exponent $K(\mathfrak{P}')$

Algorithm 3 MultiplyFactors($\mathfrak{P}_1, \ldots, \mathfrak{P}_m$). Exact product of uncompiled circuits with determinism and exponent tracking.

Require: uncompiled circuits $\mathfrak{P}_1, \ldots, \mathfrak{P}_m$
$\mathcal{C} \leftarrow \mathrm{PROD}(\mathcal{C}_1, \ldots, \mathcal{C}_m)$
$\mathcal{D} \leftarrow \bigcup$ pairwise $\mathrm{ProdDet}(\mathbf{Q}_1, \mathbf{Q}_2; \mathbf{A})$ over $\mathbf{Q}_1 \in \mathcal{D}_1$, $\mathbf{Q}_2 \in \mathcal{D}_2$, accumulating the shared scope $\mathbf{A}$ along the product ▷ rule equation 12
$K \leftarrow$ fold K_i: $\max(K, K_i)$ if rule equation 12 certifies the shared scope, $K + K_i$ otherwise
return $(\mathcal{C}, \mathcal{D}, K)$, untagged

Algorithm 4 SimplifyFactors($\{F_i\}$, $\mathcal{G}$). Merge atomic factors pairwise into coarser factors with the same semantics.

Require: factors $F_i = P(\mathbf{N}_i \mid \mathbf{D}_i)$ with $\mathbf{N}_i \cap \mathbf{D}_i = \emptyset$, graph $\mathcal{G}$
function MERGE(F_i, F_j) ▷ attempt to absorb F_i into F_j
  $\mathbf{D}'_i \leftarrow \mathbf{D}_i$; $\mathbf{D}'_j \leftarrow \mathbf{D}_j$
  if $\mathbf{N}_j \perp\!\!\!\perp \mathbf{N}_i \setminus \mathbf{D}_j \mid \mathbf{D}_j$ **then** $\mathbf{D}'_j \leftarrow \mathbf{D}'_j \cup \mathbf{N}_i$ ▷ F_j's conditioning set absorbs $\mathbf{N}_i$
  end if
  if $\mathbf{N}_j \perp\!\!\!\perp \mathbf{D}_i \setminus \mathbf{D}_j \mid \mathbf{D}_j$ **then** $\mathbf{D}'_j \leftarrow \mathbf{D}'_j \cup \mathbf{D}_i$
  end if
  if $\mathbf{N}_i \perp\!\!\!\perp (\mathbf{D}_j \setminus \mathbf{N}_i) \setminus \mathbf{D}_i \mid \mathbf{D}_i$ **then** $\mathbf{D}'_i \leftarrow \mathbf{D}'_i \cup (\mathbf{D}_j \setminus \mathbf{N}_i)$ ▷ F_i's conditioning set absorbs $\mathbf{D}_j$
  end if
  if $\mathbf{N}_j \cap \mathbf{D}'_i = \emptyset$ **and** $\mathbf{D}'_j = \mathbf{N}_i \cup \mathbf{D}'_i$ **then return** $P(\mathbf{N}_i \cup \mathbf{N}_j \mid \mathbf{D}'_i)$ ▷ conditioning on $\mathbf{N}_i$ is absorbed into the head
  end if
  return none
end function
repeat
  for each ordered pair (F_i, F_j), $i \neq j$ **do**
    if MERGE(F_i, F_j) succeeds with F **then**
      replace (F_i, F_j) by F and restart the scan
    end if
  end for
until no merge succeeds
return the remaining factors

G.3 THE REQUIRED DETERMINISMS OF A QUERY

T-ID checks tractability against the determinisms actually present in a given circuit. To instead derive the determinism family a base circuit *must* provide for a query to be evaluable at all, the same recursion is run once in *collection mode*: every COND records its conditioning set into an output set in place of the membership check, and no composition needs to be emitted. Because the control flow is literally that of Algorithm 1, collection mode raises *IdentificationError* under exactly the same circumstances as T-ID.

Algorithm 5 RequiredDeterminisms($\mathbf{y}$, $\mathbf{x}$, $\mathcal{G}$). Returns the determinism family that renders $P(\mathbf{y} \mid \mathrm{do}(\mathbf{x}))$ evaluable.

Require: query $\mathbf{y}$, intervention $\mathbf{x}$, graph $\mathcal{G}$
$\mathcal{R} \leftarrow \emptyset$
run T-ID($\mathbf{y}$, $\mathbf{x}$, $\mathfrak{P}_0$, $\mathcal{G}$) in collection mode: every COND records its conditioning set $\mathbf{D}$ into $\mathcal{R}$ instead of checking $\mathbf{D} \in \mathcal{D}$
return $\mathcal{R}$

Equivalently, REQUIREDDETERMINISMS executes the identification derivation symbolically on the observational distribution with all determinism checks disabled, and returns the set of conditioning scopes the derivation would have demanded. Because determinism over a set implies determinism over each of its subsets, a base circuit whose determinisms cover $\mathcal{R}$ — providing, for each $\mathbf{Q} \in \mathcal{R}$, some $\mathbf{Q}' \supseteq \mathbf{Q}$ — makes every conditioning step in the derivation admissible, so the query is evaluable on any circuit carrying such a family.

G.4 GUARANTEES

Proposition 33 (Identification equivalence)**.** T-ID$(\mathbf{y}, \mathbf{x}, \cdot, \mathcal{G})$ *raises* IdentificationError *if and only if* $P(\mathbf{y} \mid \mathrm{do}(\mathbf{x}))$ *is not identifiable from* $P(\mathbf{V})$ *on* $\mathcal{G}$*, in which case a hedge witnesses non-identifiability.*

Proof. The case analysis of Algorithm 1 reproduces the ID recursion of Shpitser & Pearl (2006) exactly, and the additions — emitting operations, checking determinisms, and updating exponents — neither branch nor prune. The only change to the derived expression is the SIMPLIFYFACTORS merge step, and each merge is licensed by a d-separation in $\mathcal{G}$. Under the Markov assumption —

that $P(\mathbf{V})$ satisfies the conditional independencies encoded by $\mathcal{G}$, as standard in identification — each such d-separation is a conditional independence of $P(\mathbf{V})$, so each merge preserves the encoded distribution, and the returned functional coincides with the one derived by the ID algorithm, which is sound and complete for identification (Shpitser & Pearl, 2006; Huang & Valtorta, 2006). □

Lemma 34 (Minimal separators and irreducible factors)**.** *For each variable v and predecessor set in $\mathcal{G}$ there is a unique minimal subset $\tilde{\pi}(v)$ d-separating v from its remaining predecessors.* SIMPLIFYFACTORS *terminates and returns a uniquely determined set of irreducible factors: no two factors in its output can be validly merged into a single coarser conditional.*

Proof. For a fixed pair (v, w), d-separation satisfies the Intersection axiom: if A and B both d-separate v from w, so does $A \cap B$. The family of separating sets is therefore closed under pairwise intersection and, the predecessor set being finite, contains a unique minimal member m_w. Now let $\mathbf{W}$ be a predecessor set. Any set π with $v \perp\!\!\!\perp \mathbf{W} \mid \pi$ must contain m_w for every $w \in \mathbf{W}$ (Decomposition), and $\pi^* = \bigcup_{w \in \mathbf{W}} m_w$ itself separates v from $\mathbf{W}$ (Composition, monotonicity), so π^* is the unique minimal separator. Applying this to $\mathbf{W} = \mathrm{preds}(v) \setminus \pi$ subject to $\pi \supseteq \mathrm{pa}(v)$, which any valid π must satisfy because a parent of v cannot be d-separated from v by predecessors of v alone, yields the unique minimal $\tilde{\pi}(v)$. For the procedure, each MERGE step strictly decreases the number of factors, so SIMPLIFYFACTORS terminates, and by the loop guard its output admits no further merge. The scan order is fixed, so the output is uniquely determined. □

Theorem 35 (Evaluation soundness)**.** *If* T-ID *returns $(\mathcal{C}, \mathcal{D}, K)$ for $(\mathbf{y}, \mathbf{x}, \mathcal{G})$ on a base circuit $\mathcal{C}_0$ carrying the determinisms $\mathcal{D}$, then evaluating $\mathcal{C}$ on $\mathcal{C}_0$ yields exactly $P(\mathbf{y} \mid \mathrm{do}(\mathbf{x}))$.*

Proof. We prove by structural induction over the nodes of the emitted composition that every node encodes exactly the estimand derived for it by the recursion, and that its circuit is a variant of $\mathcal{C}_0$ respecting its vtree. The leaf $\mathcal{C}_0$ encodes $P(\mathbf{V})$ by definition. A MARG node encodes the exact marginal of its child by smoothness and decomposability (Wang et al., 2024, Thm. 1), and its circuit stays vtree-respecting. A COND node was admitted, so its conditioning set is an exact member of the determinism family of its child, and by Proposition 6 it encodes exactly the required conditional over the required scope, with the structural properties preserved. A PROD node has operands that are, by induction, variants of $\mathcal{C}_0$ respecting its vtree, hence compatible, so by Wang et al. (2024, Thm. 2) it encodes exactly the product of the operands' functions, and its output is again compatible with both operands. The root therefore encodes exactly the functional derived by the recursion, which by Proposition 33 equals $P(\mathbf{y} \mid \mathrm{do}(\mathbf{x}))$. □

Exactness in Theorem 35 is exactness relative to the base circuit's model of the observational distribution: evaluating the composition reproduces the estimand of whatever distribution $\mathcal{C}_0$ encodes. Theorem 35 does not certify that this estimand is close to the true $P(\mathbf{y} \mid \mathrm{do}(\mathbf{x}))$ — that rests entirely on the expressivity and fit of $\mathcal{C}_0$, as in any circuit-based inference. The limiting case makes the boundary clear: a fully factorized base circuit satisfies every determinism, so T-ID admits every identifiable query, but such a circuit expresses only independent distributions and is useless in practice. T-ID certifies that the estimand of the modeled distribution is computed exactly and tractably, not that the model is faithful.

Proposition 36 (Completeness of the tractability check)**.** *If a determinism demanded by* REQUIREDDETERMINISMS *for $(\mathbf{y}, \mathbf{x}, \mathcal{G})$ is not provided by the base circuit, then no composition of the operations* MARG*,* COND*, and* PROD *that materializes the minimal kernel decomposition of the estimand evaluates exactly on $\mathcal{C}_0$.*

Proof. By Lemma 34 the simplified factor set is the unique minimal kernel decomposition of the estimand, so any evaluation of this class must materialize each factor $P(\mathbf{N}_r \mid \mathbf{D}_r)$. Among the three operations, only COND produces a conditional of the observational distribution as a standalone circuit, and its admission precondition — exact membership of the conditioning set in the determinism family — is precisely the requirement under which it is exact (Proposition 6). The missing determinism belongs to some factor's conditioning set $\mathbf{D}_r$, so any composition either materializes that factor without the precondition and is not exact, or does not materialize it and does not compute the estimand. □

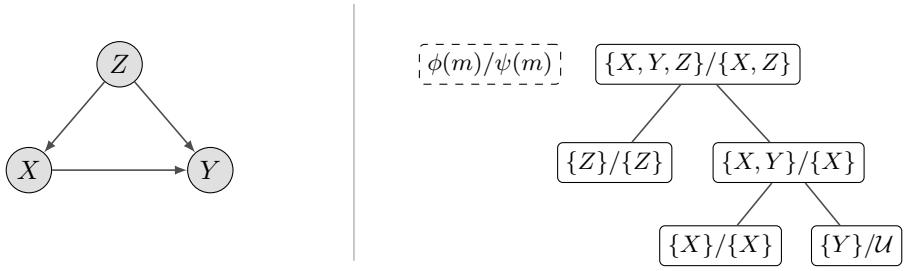


Figure 7: The backdoor experiments. Left: the backdoor SCM ($Z \to X$, $Z \to Y$, $X \to Y$; all variables continuous) whose dependence structure the circuit must accommodate. Right: the md-vtree: the root separates the confounder Z from (X, Y), with the confounder set $\{X, Z\}$ as the only determinism.

Proposition 36 is stated relative to the operator calculus of Section 3: evaluators that do not materialize the kernel decomposition, for example circuits trained directly on interventional data, fall outside what the check certifies.

Theorem 37 (Complexity)**.** *Evaluating the composition returned by* T-ID *on* $\mathcal{C}_0$ *costs* $\mathcal{O}(|\mathcal{C}_0|^K)$.

Proof. By structural induction over the nodes of the emitted composition. The leaf $\mathcal{C}_0$ has size $|\mathcal{C}_0|$ and $K = 1$. A MARG or admitted COND node has linear cost and inherits the child's exponent by the cost model, and K is unchanged at the node. A PROD node with children of exponents K_1, K_2 costs $\mathcal{O}(|\mathcal{C}_0|^{\max(K_1,K_2)})$ when rule equation 12 certifies a determinism over the shared scope and $\mathcal{O}(|\mathcal{C}_0|^{K_1+K_2})$ otherwise (Wang et al., 2024, Thms. 2 and 3), which is exactly how K is updated. Hence the root is evaluable in $\mathcal{O}(|\mathcal{C}_0|^K)$. Each product node is charged exactly once in this accounting. K therefore certifies a worst-case upper bound under the operation costs of the cost model, not a lower bound. □

Since K starts at 1 at the leaves and only the product nodes modify it, K never exceeds the number of atomic operands, so evaluation is polynomial in the size of the base circuit. Together, Proposition 33, Theorem 35, Proposition 36, and Theorem 37 establish the three outcomes stated in Section 6: a returned compositional query with complexity bound $\mathcal{O}(|\mathcal{C}_0|^K)$; *IdentificationError* when the query is not identifiable; and *TractabilityError* — with the missing determinism scope reported — when the query is identifiable but the given circuit does not support one of its conditioning steps. REQUIREDDETERMINISMS is the same recursion run in collection mode, so it raises *IdentificationError* under exactly the same circumstances and otherwise returns the family whose coverage makes every COND in the derivation admissible.

H ADDITIONAL EXPERIMENTS

This appendix collects further experiments complementing Section 7.

H.1 BACKDOOR

Setup. Our main controlled benchmark is a synthetic backdoor SCM over a continuous confounder Z, a continuous treatment X, and a continuous outcome Y (Figure 7), with randomized tractable mechanisms such that the exact analytical ground truth of $P(Y \mid X)$ and $P(Y \mid \mathrm{do}(X))$ is available in closed form. We generate observational datasets of size 25K to 1.2M samples as well as their interventional counterparts (by fixing the exogenous variables and resampling on the intervened SCM). We train one MD circuit per seed and configuration, with the confounder set $\{X, Z\}$ declared as the only determinism. The md-vtree is shown in Figure 7. Unless stated otherwise all models use N=64 components per sum node and are trained on the 400K-sample dataset.

Training protocol. All circuits are trained with minibatch EM for 300 epochs on batches of 8192 samples (4096 for N=128), with an initial step size of 0.1 decayed by a factor 0.98 every 5 epochs.

Config	do NLL	Obs test NLL
N=8	4.0773 ± 0.0169	6.3376 ± 0.0044
N=16	3.9376 ± 0.0255	6.2600 ± 0.0017
N=32	3.8629 ± 0.0171	5.7937 ± 0.0007
N=64	3.7973 ± 0.0276	5.7774 ± 0.0010
N=128	3.7646 ± 0.0314	5.4391 ± 0.0034
N=64 G1	4.2488 ± 0.0035	6.9402 ± 0.0003
N=64 unc.	—	5.1030 ± 0.0019

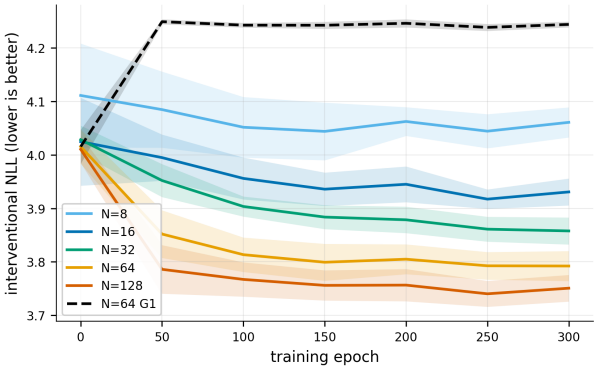


Figure 8: Backdoor scaling study, mean $\pm$ std over 5 seeds. Left: final do NLL of the compiled circuit representing $P(Y \mid \mathrm{do}(X))$ evaluated on the interventional dataset (400K samples), and observed test NLL on the observational test set, for varying circuit size N; "G1" denotes the single-group degenerate variant, "unc." the unconstrained model as a baseline, though incapable of modeling $P(Y|do(X))$. Right: interventional test NLL of the compiled do-query circuits over training epochs; larger N improves the interventional fit monotonically, and the single-group variant plateaus at a much worse level. Comparison with the unconstrained baseline shows the expressivity lost as a result of enforcing weak determinism. (Color reading recommended.)

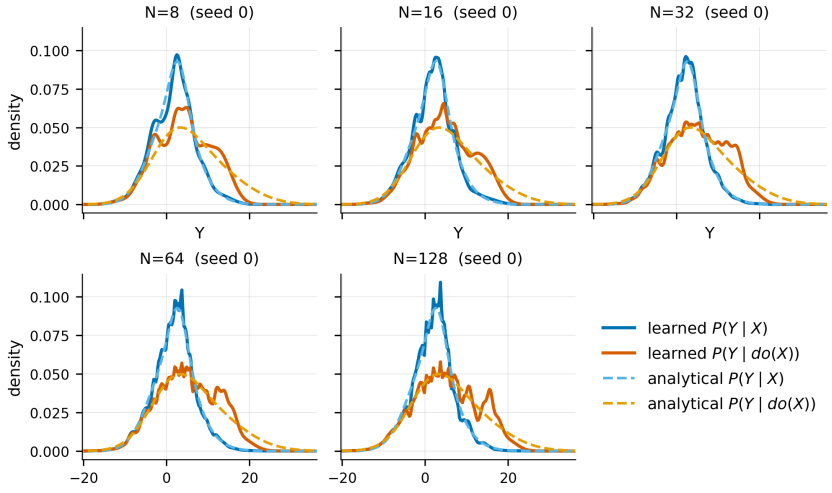


Figure 9: Y-density slices at X=3.5 for increasing N (seed 0). Higher capacity fits the analytical interventional density ever more closely; at N=128 small spurious modes appear, an early sign of fitting observational noise. (Color reading recommended.)

After training, the backdoor adjustment estimand identified by T-ID is compiled into a do-query circuit that shares all parameters with the base circuit, so a single model is trained. $P(Y|X)$ is computed as the ratio of two circuit evaluations $P(Y, X)$ and $P(X)$, $P(Y|do(X))$ via a compositional query in a single pass. We report mean $\pm$ std over 5 seeds throughout, with surface and slice visualizations showing only a single seed (seed 0). Every likelihood and surface metric is evaluated against the exact analytical ground truth (no Monte-Carlo estimation).

Figure 8 traces interventional test NLL over the course of training. Our circuits improve their interventional fit monotonically and rank-order by capacity throughout training, whereas the single-group G1 variant plateaus at a substantially worse level: group sparsity does not merely slow optimization, it removes the structure needed to represent the adjustment estimand, consistent with its poor final do NLL in the table.

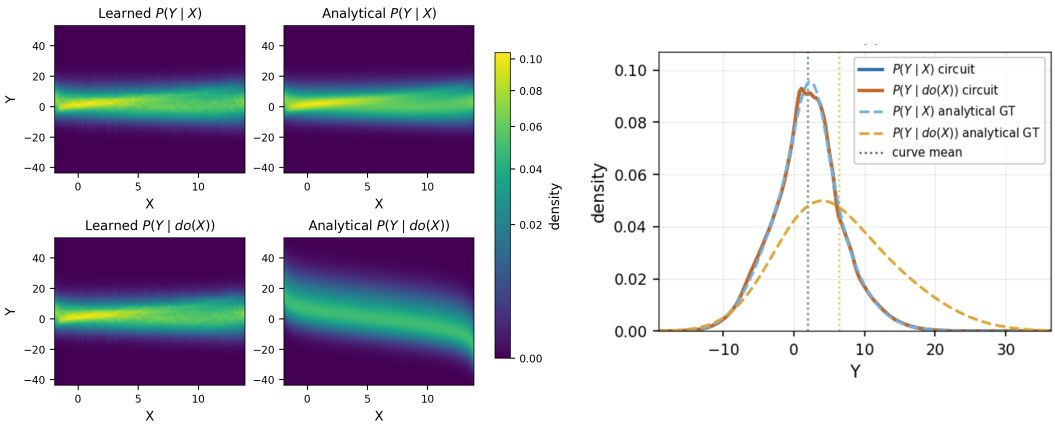


Figure 10: Collapse of the single-group variant (N=64 G1, 400K samples, seed 0; layout as in Figure 16). Left: learned (left) vs. analytical (right) $P(Y \mid X)$ (top) and $P(Y \mid \mathrm{do}(X))$ (bottom): the learned interventional surface is identical to the learned observational one. Right: Y-density slice at X=3.5; the circuit's interventional density (red) coincides with its observational one (blue) and with the analytical observational ground truth, while the analytical interventional density (dashed orange) is shifted and substantially wider. (Color reading recommended.)

Capacity scaling. Increasing N refines the partition of the constrained variables (here X and Z) rather than the factorization of the density, and this is directly visible in the learned surfaces. Figure 11 compares the learned $P(Y \mid X)$ and $P(Y \mid \mathrm{do}(X))$ against the analytical ground truth: at N=8 the interventional surface is a coarse, visibly piecewise-in-X approximation, and each doubling of N resolves finer structure along the treatment axis until, at N=128, the piecewise segments track the ground-truth ridge closely. The observational surface, by contrast, is already well fit at small N — quantitatively, Figure 12 shows the X-averaged $\mathrm{KL}(P_{\mathrm{GT}} \| P_{\mathrm{learned}})$ of $P(Y \mid X)$ is essentially zero for every N, while the do-KL improves from 0.52 nats at N=8 to 0.21 nats at N=128 with clearly diminishing returns.

While increasing capacity gives us a smoother joint $P(Y|do(X))$ in X, and an overall better fit, Figure 9 shows that currently it comes at the cost of slight overfitting. The N=8 model slightly misses the shape of the interventional density, but is significantly more stable. Higher capacities close the gap, but start to pick up noise around the ground truth.

Collapse of the single-group variant. The plateau of the G1 variant in Figure 8 has a precise surface-level signature. Figure 10 shows the learned $P(Y \mid X)$ and $P(Y \mid \mathrm{do}(X))$ of the single-group model: the two coincide. With a single group, the root layer is minimally diverse against its children (Remark 10), so Z is modelled as independent of (X, Y) and the backdoor adjustment collapses to the observational conditional, $\widehat{P}(Y \mid \mathrm{do}(X)) = \int_Z \widehat{P}(Y \mid X)\, \widehat{P}(Z)\, dZ = \widehat{P}(Y \mid X)$. The slice of Figure 10 makes the same point in one dimension: the circuit's interventional density tracks the analytical observational density and misses the shifted, wider analytical interventional one. This is exactly the collapse predicted in Section 4, now verified empirically; the multi-group construction is what avoids it.

Data scaling. Holding capacity fixed at N=64, Figures 14 and 13 show the effect of the training set size. The 25K model's surfaces are visibly speckled and its interventional slice at X=3.5 is highly indicative of overfitting; each increase in data smooths the surfaces and pulls the slice onto the analytical curves, with 1.2M samples very close to ground truth. Figure 12 makes the data–support interaction explicit: the per-X do-KL is minimal where the treatment is well observed and rising steeply in the tails of the X marginal. Increasing the data set enlarges the region of X where the interventional query is accurate while lowering the error floor everywhere.

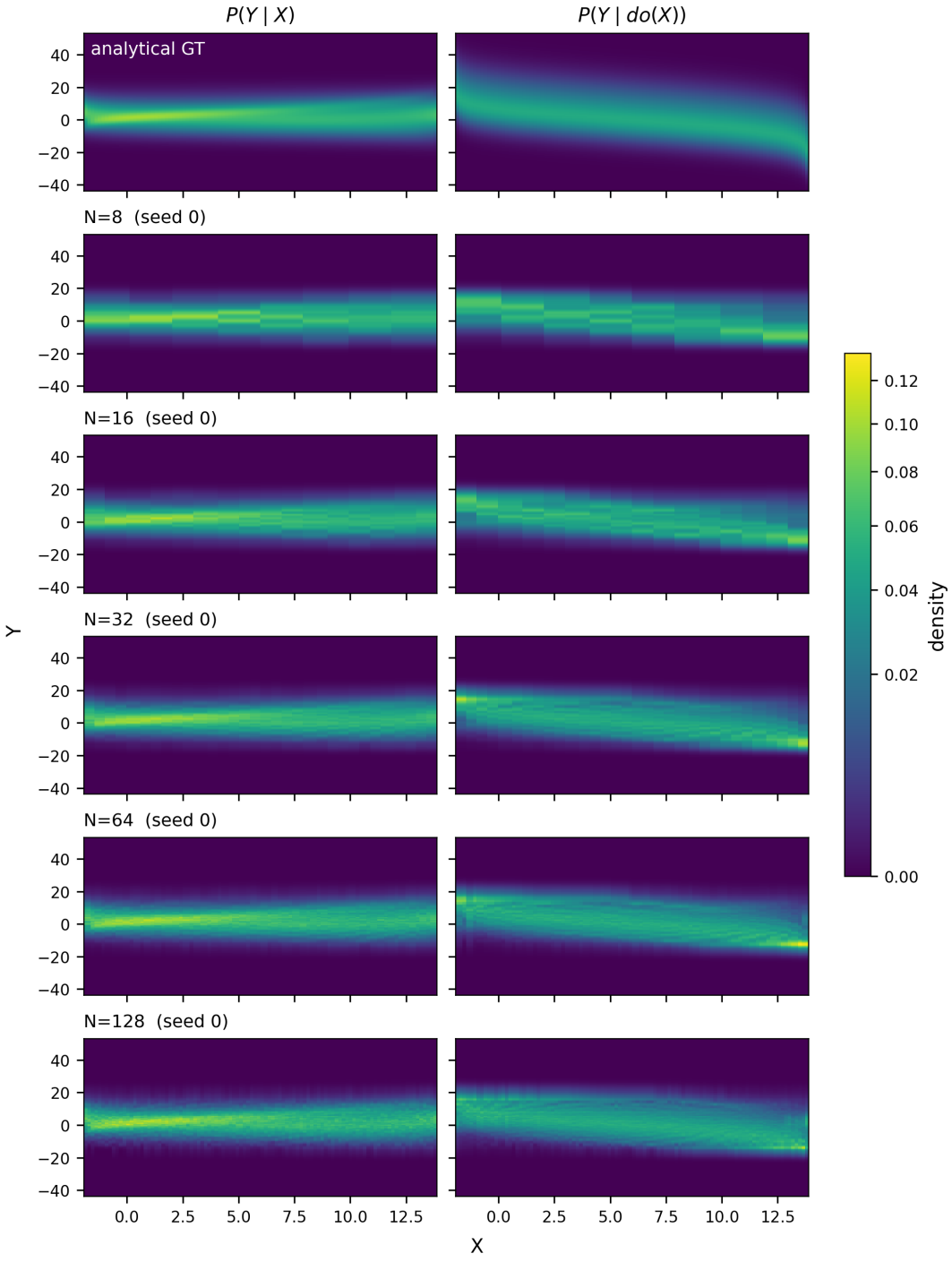


Figure 11: Learned (rows 2–6, seed 0) vs. analytical (top row) $P(Y \mid X)$ (left) and $P(Y \mid \mathrm{do}(X))$ (right) for increasing circuit size N. Higher N resolves progressively finer structure along X; the interventional surface is piecewise in X, with the segment boundaries becoming finer as the MD state space is refined. (Color reading recommended.)

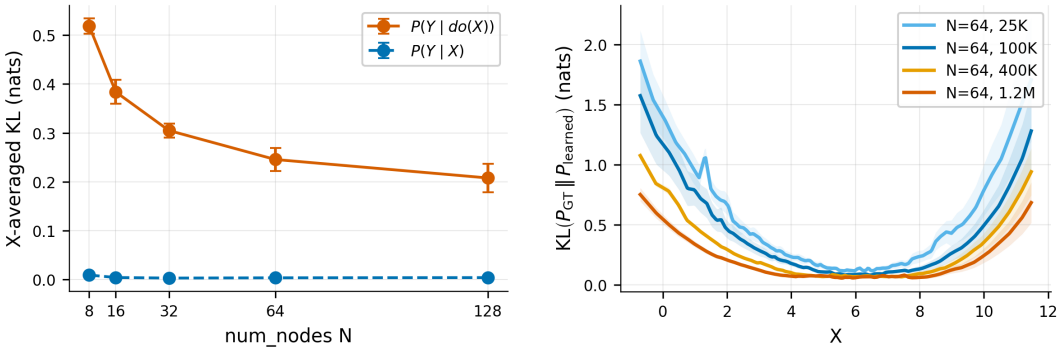


Figure 12: KL diagnostics of the learned surfaces (mean $\pm$ std over 5 seeds). (a) $\mathrm{KL}(P_{\mathrm{GT}} \| P_{\mathrm{learned}})$ averaged over the true marginal X density vs. circuit size N: all capacities fit the observational $P(Y \mid X)$ almost perfectly; interventional accuracy improves with N but with diminishing returns. (b) Per-X $\mathrm{KL}(P_{\mathrm{GT}} \| P_{\mathrm{learned}})$ of $P(Y \mid \mathrm{do}(X))$ over $X \in [\mu_X - 2\sigma_X, \mu_X + 2\sigma_X]$ for increasing training set sizes: estimation error grows in less-observed regions of the treatment space, and more data widens the accurately identified region while lowering the error floor. (Color reading recommended.)

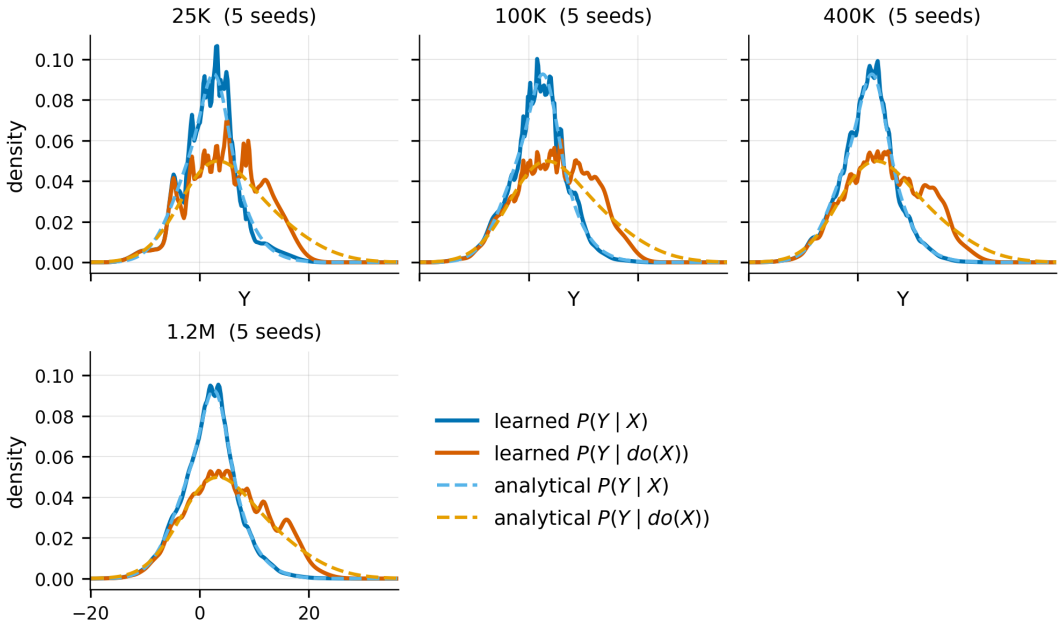


Figure 13: Y-density slices at $X{=}3.5$ for increasing training set sizes (seed 0; the analytical GT is the same curve in every panel). The interventional density is overshot and distorted at 25K and converges to the ground truth with more data. (Color reading recommended.)

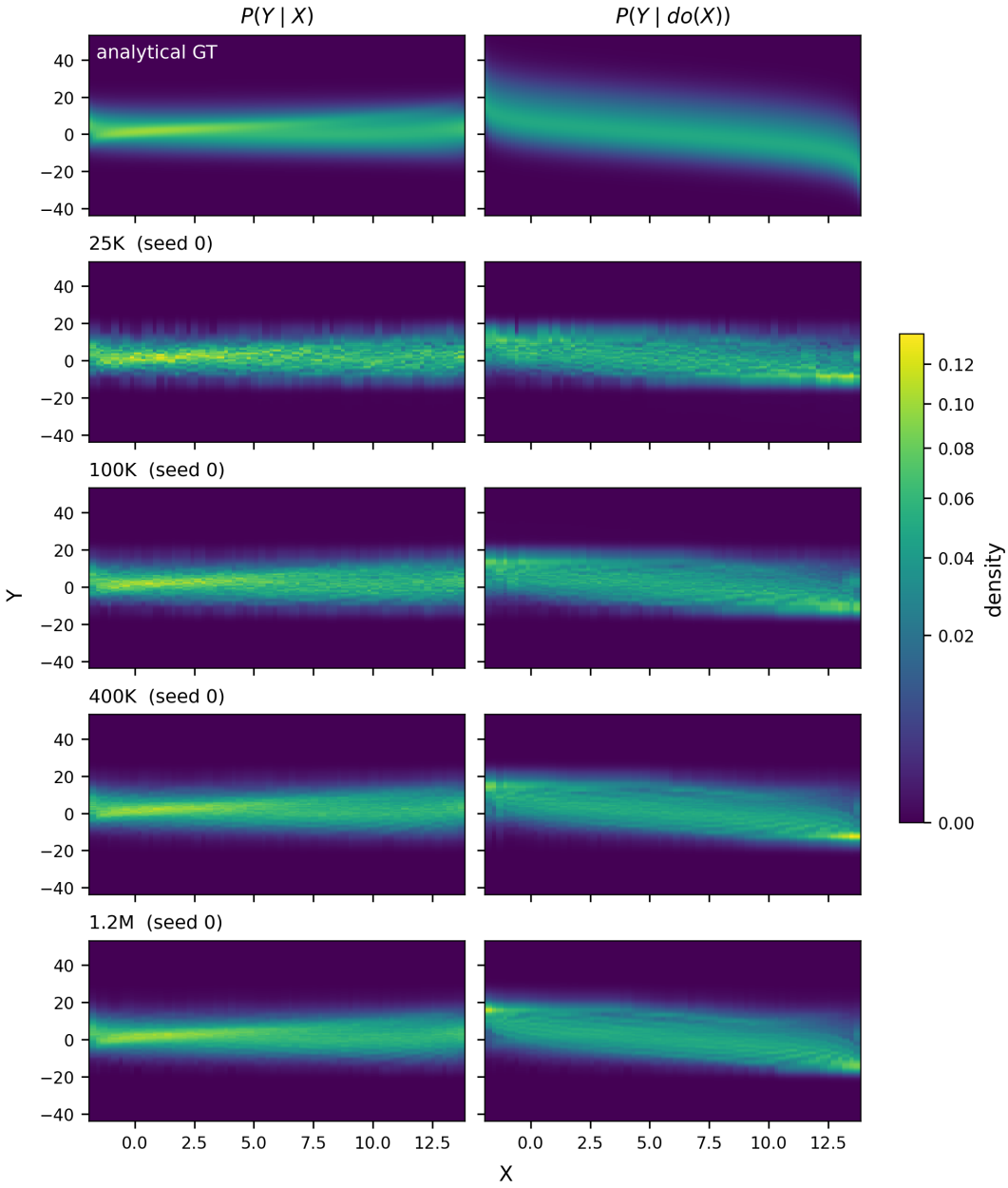


Figure 14: Learned vs. analytical surfaces (layout as in Figure 11) for $N{=}64$ trained on 25K to 1.2M samples. More data progressively denoises the learned surfaces. (Color reading recommended.)

	NLL	KL (nats)
$P(Y \mid X)$ (obs)	1.6191 ± 0.0001	0.0001 ± 0.0000
$P(Y \mid \mathrm{do}(X))$	1.1564 ± 0.0010	0.0302 ± 0.0011

Table 3: Mixed frontdoor results (50K samples, N=16, 5 seeds). KL is computed against the analytical ground truth on a grid over the two treatment values and 200 outcome points, averaged over X under its analytical marginal.

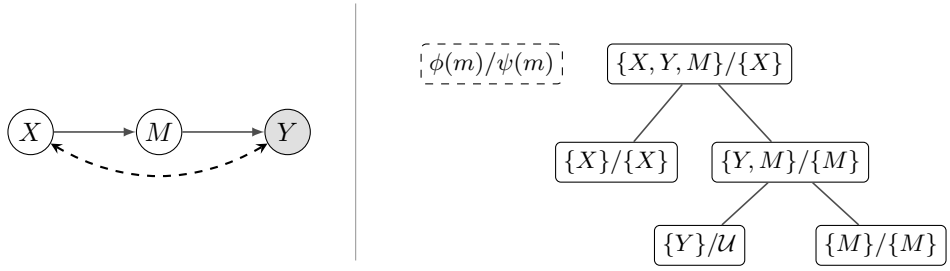


Figure 15: The mixed frontdoor experiments. Left: the frontdoor SCM ($X \to M \to Y$; X binary, M discrete, Y continuous; dashed bidirected edge: latent confounding of X and Y by a hidden variable U), which makes identification through the frontdoor criterion necessary. Right: the md-vtree: the root separates the treatment X from (Y, M), and the MD labels $\{X, M\}$ (root) and $\{M\}$ support the conditionals $P(m \mid x)$ and $P(y \mid m, x')$ of the frontdoor estimand.

H.2 Mixed frontdoor

Setup. The mixed frontdoor benchmark exercises identification through a mediator under latent confounding with heterogeneous variable types. The SCM (Figure 15) has a binary treatment X, a discrete mediator M, and a continuous outcome Y, with a hidden discrete variable U confounding the X–Y pair; the $X \to M \to Y$ chain is unconfounded, so the effect of X on Y is identified only by the frontdoor criterion,

$$P(Y \mid \mathrm{do}(X{=}x)) \;=\; \sum_{m} P(m \mid x) \sum_{x'} P(Y \mid m, x')\, P(x'),$$

which T-ID recovers and the query compiler turns into a do-circuit. We use a positivity-gap variant of the dataset (50K samples, $90/10$ train/test split) in which the observational treatment distribution is imbalanced ($P(X{=}0) \approx 0.19$), so the outer reweighting over x' leans heavily on the rare treatment arm; exact analytical ground truth is available from the randomized tractable mechanisms.

Model and training. The circuit uses N=16 components per sum node over the pinned vtree of Figure 15, with MD labels $\{X, M\}$ at the root and $\{M\}$ at the (Y, M) node; X and M are categorical leaves and Y a log-linear spline leaf. Training follows the same minibatch EM protocol as the backdoor study (300 epochs, batch 4096, initial step size 0.1 decayed by 0.98 every 5 epochs), mean $\pm$ std over 5 seeds.

Results. Table 3 summarizes the fit. The observational surface is essentially exact; the frontdoor-adjusted interventional surface is recovered with do-KL of 0.030 nats despite the positivity gap. Figure 16 (left panel) shows the learned against analytical $P(Y \mid X)$ and $P(Y \mid \mathrm{do}(X))$: the two discrete treatment bands are cleanly separated and the interventional band correctly averages away the confounding visible in the observational one. The right panel isolates the rarer treatment arm, X=0 ($\approx 19\%$ of the data): even here the learned observational and interventional densities track their analytical counterparts, including the secondary mode — the circuit resolves the mixture structure of the frontdoor-adjusted density rather than collapsing to its mean.

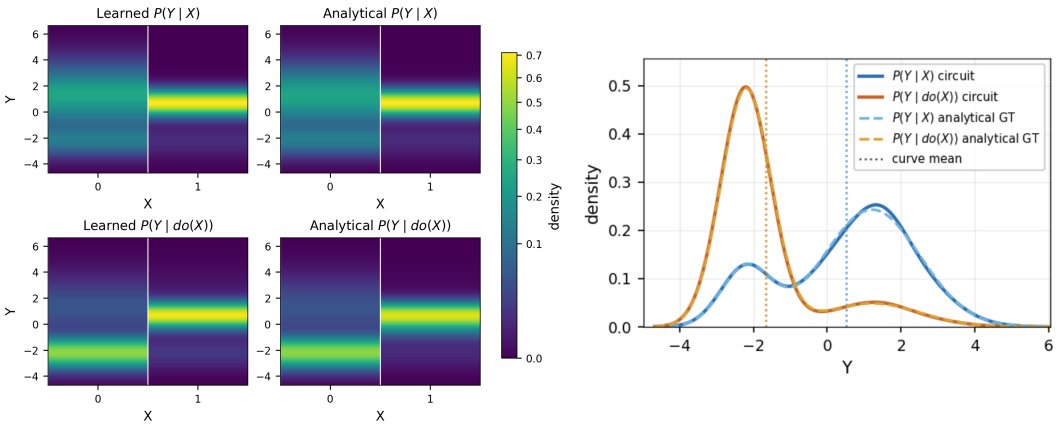


Figure 16: Mixed frontdoor results (50K samples, N=16, seed 0). Left: learned (left) vs. analytical (right) $P(Y \mid X)$ (top) and $P(Y \mid \mathrm{do}(X))$ (bottom) over the two treatment values. Right: Y-density slice at X=0. Both interventional and observational surfaces show a near-perfect fit. (Color reading recommended.)

H.3 Comparison with MD-Net on real binary data

We compare our method against the MD-Net of Wang & Kwiatkowska (2023) on four binary Bayesian networks from their benchmark: ASIA (8 variables), CHILD (20), WIN95PTS (76), and ANDES (223). Each dataset consists of 1,000 binarized sample rows drawn from the original network, taken from the authors' released artifacts. For each dataset we answer one backdoor-adjustment query $P(Y = y^* \mid \mathrm{do}(X = x^*))$ with the adjustment set Z and exact target taken from their released setup; the query details are summarized in Table 4.

Our method. We train an MD circuit whose vtree is learned by bottom-up pairwise-MI induction (Liang et al., 2017) together with the MD labeling of Wang & Kwiatkowska (2023), using the MD set $X \cup Z$ with X and Y separated in the vtree as deep as possible. Each region has 10 sum units and categorical leaves with one selector per category. No structure is hand-crafted — the vtree is learned from the data and is deterministic given it. Training is batched EM on all 1,000 rows (no train/test split) for 50 iterations with batch size 256, step size 1.0 without decay, and leaf learning rate 0.05. The backdoor estimand is compiled compositionally as $\mathrm{MARG}_z\big[\mathrm{PROD}\big(P(y \mid x, z),\, P(z)\big)\big]$, with the remaining variables W marginalized out of the joint before conditioning, following the protocol of Wang & Kwiatkowska (2023). The estimate is the two-point ratio $P(Y{=}y^* \mid X{=}x^*) / \sum_y P(Y{=}y \mid X{=}x^*)$. We report the mean absolute error (MAE) against the exact target and the Bernoulli log-loss $-(t \log p + (1-t)\log(1-p))$ with p clipped at 10^{-10}, which is scale-aware: errors near 0 or 1 cost proportionally more. All numbers are mean $\pm$ std over 5 seeds (0–4), and our runs are bit-reproducible on CPU (numpy, Python-random, and torch all seeded). Total query time (compile + evaluate, CPU) is 0.07 s (ASIA), 0.19 s (CHILD), 0.67 s (WIN95PTS), and 1.27 s (ANDES).

MD-Net baseline. We run the released code of Wang & Kwiatkowska (2023) unmodified (jax 0.4.30) on all four datasets $\times$ 5 runs, with $\mathrm{sum}_h = 10$ and weights drawn from a fixed `PRNGKey(0)`. Their paper protocol is a single count-EM step; this is not a limitation of our reproduction — their released full-EM loop produces NaN gradients from structurally dead units, so no converged-training setting exists in the released code (we verified this directly). Moreover, their vtree scope partitions are shuffled with an unseeded RNG, so individual runs are not bit-reproducible — only the run distribution is. Their estimate is likewise a two-point ratio, so the comparison is on equal footing.

Results. Table 1 (main text) reports mean $\pm$ std over 5 runs, alongside a naive baseline that performs counting backdoor adjustment directly on the raw data. Our pipeline is seeded end-to-end, whereas the MD-Net runs vary due to the unseeded vtree shuffle. For fairness, we note that MD-

Table 4: The backdoor-adjustment queries of the MD-Net real-data benchmark (one per dataset, from the released setup of Wang & Kwiatkowska (2023)): treatment variable X, outcome variable Y, size of the adjustment set Z, queried values (x^*, y^*), and the exact Bayesian-network target $P(Y{=}y^* \mid \mathrm{do}(X{=}x^*))$.

Dataset	X	Y	$\lvert Z\rvert$	(x^*, y^*)	Exact target
ASIA	bronc	dysp	4	$(0, 0)$	0.8065
CHILD	Sick	XrayReport	13	$(0, 0)$	0.3164
WIN95PTS	REPEAT	Problem2	59	$(1, 1)$	0.0521
ANDES	SNode_154	SNode_155	221	$(1, 1)$	0.1432

Net's numbers reflect its paper protocol (a single count-EM step); the authors have released no converged-training variant to compare against.

I Experimental Setup

Compute. All GPU experiments ran on a single DGX node with 16 Tesla V100-SXM3-32GB GPUs. The 20 configurations (16 synthetic, 4 binary BNs) were each repeated over 5 seeds; seeds of a configuration were pinned to individual GPUs and ran concurrently, one seed per GPU — roughly 100 training runs in total. The largest continuous run (N = 128, 1.2M samples) trains in under one hour per seed; the full suite, including per-checkpoint evaluation and grid scoring, completes in a few hours on the node.

Setup (summary). Each experiment trains a single probabilistic circuit on observational data by minibatch EM (300 epochs on the synthetic benchmarks, batch 8192 or 4096 for the largest model; 50 full-EM epochs, batch 256, on the binary BNs; step size 0.1 decayed by 0.98 every 5 epochs, full steps for the BNs). Model checkpoints are saved every 50 epochs on synthetic runs so accuracy can be traced over training. The benchmarks are: synthetic continuous backdoor/frontdoor/colliderdoor SCMs (25K-1.2M samples, exact analytical ground truth) and four real-world binary Bayesian networks from the MDNet study, evaluated against the exact BN posterior. Each trained circuit is scored on held-out observational and interventional likelihood, and on KL/Hellinger distance to the analytical $P(Y|X)$ and $P(Y|do(X))$ surfaces over a 100×200 grid; all ground truth is analytical, no Monte-Carlo estimation. Reported numbers are mean ± std over the 5 seeds.

Code availability. The full pipeline, all training and evaluation code is available at `https://anonymous.4open.science/r/AlmostAutomatedCausalCircuits-93F4`.