INDUCING TRIPLET REGIONS FOR AMORTIZED KIKUCHI LEARNING IN DISCRETE BAYESIAN NETWORK MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning parameters in large discrete Bayesian networks (BNs) from massive incomplete data is challenging due to the high computational cost of repeatedly estimating joint marginals. Standard ELBO-based amortization, such as amortized variational inference (VI), scales well but suffers from restrictive posterior factorization assumptions. To overcome this, we introduce Triplet Region Induction-Constrained Neural Kikuchi Learning (TRNK), a non-ELBO framework grounded in Kikuchi free energy and region graph approximations. TRNK captures higher-order variable interactions for improved accuracy and scalable amortized learning. By leveraging BN structural and factor information, it identifies effective triplet base regions and induces higher-order regions to enhance approximation fidelity and provide orthogonality regularized region interactions. Under these region constraints, TRNK trains an evidence-conditioned deep neural network to produce overlap-consistent region beliefs for each partially observed instance, amortizing inference overhead across the entire dataset. Empirical results show that TRNK significantly outperforms amortized VI and approximate EM baselines in accuracy and efficiency. Furthermore, our region induction algorithm surpasses Bethe and other well-established region-graph alternatives on most benchmarks.

1 INTRODUCTION

Learning parameters from incomplete data for directed acyclic graphs (DAGs), such as Bayesian networks (BNs), becomes challenging when model complexity and data volume increase. Major computational bottlenecks arise when performing exact inference for maximum likelihood estimation of the conditional probability distributions (CPDs) (Koller & Friedman, 2009). Well-known learning methods fall into evidence lower bound (ELBO)- based and non-ELBO-based. Variational inference (Blei et al., 2017) and its variants (Winn et al., 2005; Luttinen, 2016) are classic ELBO-based methods that iteratively optimize the ELBO via per-instance inference, which can be inefficient. Amortized VI, such as the mean-field Variational Autoencoder (VAE) (Kingma & Welling, 2013; Mnih & Gregor, 2014), is effective for large datasets because it uses global parameters to amortize local variational parameters and optimizes the expected ELBO through sampling and reparameterization. However, despite its efficiency, amortized VI often limits the posterior distributions over the latent variables to simple factorizations (Blei et al., 2017). This restriction, along with the amortization gap, can lead to insufficient learning accuracy and makes improvement inconvenient.

Non-ELBO-based methods focus on local factorizations rather than global ones, enhancing the potential for improved learning quality. Representative techniques include Bethe (Yuille, 2002) and region graph methods (Yedidia et al., 2005), which decompose the global joint distribution into local coupled beliefs. Expectation Propagation (EP) (Minka, 2001) approximates the posterior by breaking down the likelihood into local exponential family factors and performing moment matching to minimize local forward KL. The amortized version of these methods mainly focused on Bethe free energy approxiamtion on (Wiseman & Kim, 2019). Although local factorizations offer flexibility, optimizing amortization remains a significant issue. The quality of approximation varied among cluster decomposition variants, and the inherent inconsistencies in amortization and reparameterization defined by their respective free energies often led to unstable training dynamics.

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.

We propose *Triplet Region Induction-Constrained Neural Kikuchi Learning* (TRNK), a region graph-based, non-ELBO approach for discrete Bayesian networks that leverages flexible local factorizations to achieve higher accuracy and robust amortized learning.

To guarantee bounded space complexity, TRNK constructs regions from bottom up via a triplet region induction (TRI) process. It converts the BN into a binary factorized (BF) (Neil et al., 2011) counterpart to maintain manageable region sizes. TRI leverages structural ordering and factor information from the BN to identify sufficient candidate triplets (Lin et al., 2020), then inductively merges them to form higher-order regions; we prove the orthogonality of their interactions for the complete DAG construction. The TRI region graph defines a Kikuchi free energy (KFE) objective that is directly reparameterized into the parameters of a deep neural network, making it naturally compatible with GPU processing for massive datasets. To strictly enforce the constraints of the running-intersection regions, TRNK represents region beliefs using coefficients based on Helmert contrast basis functions (Lancaster, 1965). These coefficients are shared across regions, ensuring that the constraints apply to every value of the network parameters. This differs from other amortized methods, which relied on an ad hoc soft consistency penalty for enforcement (Wiseman & Kim, 2019). During learning, TRNK alternates between updating the neural inference network and CPDs on the evidence-clamped KFE, effectively amortizing inference overhead across the entire dataset. The contributions of the paper are:

- *Robust bottom-up region induction algorithm.* TRI utilizes small, bounded outer regions to maintain efficiency. We provide a theoretical analysis showing that, in the worst-case complete DAG, TRI preserves edge-relative sufficiency and the orthogonality of region interactions.
- *Region-belief-consistent neural reparameterization.* We develop a coefficient mapping based on Helmert contrast functions, applied consistently across regions, that inherently satisfies the running-intersection constraints. This approach avoids the non-generalizable and ad-hoc penalty weights typically utilized by other amortized methods.
- *Significantly improved accuracy and efficiency.* As a scalable region graph-constrained neural amortized learning method that requires no sampling, TRNK surpasses ELBO-based methods and significantly outperforms related non-ELBO amortized methods in accuracy.

2 Triplet Region Induction

2.1 BNs and Region Graph Based Kikuchi Approximation

A BN is a DAG G representing a joint distribution $p(X)$ with random variables $X = (X_0, \ldots, X_{n-1})$ in a factorized way. A BN encodes Conditional Independence (CI) (Koller & Friedman, 2009) assumptions represented by the factorization of a joint distribution, p as the product of its Conditional Probability Distributions (CPDs). For such CPDs define the probabilistic relationship of a node given its respective parents. In the absence of CI assumptions, we can use the chain rule (Murphy, 2012) to factorize joint distribution, p with an arbitrary n dimensional joint probability $p(X_0, X_1, \ldots, X_{n-1})$ as $p(X_0, X_1, \ldots, X_{n-1}) = p(X_0) \prod_{i=1}^{n-1} p(X_i | X_0, \ldots, X_{i-1})$. Such a factorization corresponds to an n-dimensional *complete DAG*, where every pair of nodes is connected by a directed edge. When exact inference methods (Lauritzen & Spiegelhalter, 1988) become computationally intractable due to high model complexity, approximate inference techniques are required. In particular, region-graph-based Kikuchi approximation (Yedidia et al., 2005) provides an effective alternative.

A *region* r is characterized by a variable scope $\mathcal{X}(r)$ and a set of assigned factors, each having a scope subsumed by $\mathcal{X}(r)$. A *region graph* (RG) $\mathcal{G}$ connects regions via directed containment edges $r \to u$. A valid RG can be constructed using the multi-level cluster variation method (CVM) (Yedidia et al., 2005), where top-level regions are referred to as *outer regions*, and regions at subsequent levels, generated recursively from intersections of higher-level regions are termed *inner regions*. We denote the sets of outer and inner regions by $\mathcal{R}_0$ and $\mathcal{I}$, respectively. In a CVM RG, the *counting numbers* are defined as $c_r = 1 - \sum_{r' \supsetneq r} c_{r'}$, where the summation runs over all ancestors of r.

We can define the region graph-based Kikuchi free energy (KFE) objective as:

$$F_{\mathrm{KFE}}(b) = U(b) - H(b) = \sum_{r \in \mathcal{R}} c_r \sum_{\mathbf{x}_r} b_r(\mathbf{x}_r) \big[E_r(\mathbf{x}_r) + \log b_r(\mathbf{x}_r) \big] \tag{1}$$

where $E_r = -\log \varphi_r$, b_r is a *region belief*, and $H(b) = \sum_{r \in \mathcal{R}} c_r H(b_r)$ is the entropy approximation. For a region carrying several factors, φ_r denotes their product. Beliefs must satisfy normalized and the running-intersection constraints $b_r^{\downarrow u} = b_u$. A Lagrangian term adds constraints L_G to F_{KFE}.

Region-graph approximations yield efficiency gains when the induced maximum JT cluster size exceeds the BN's maximum factor size. To optimize the approximation starting from minimal space complexity, we construct regions bottom-up, beginning with triplet regions. Specifically, we employ CPD-level Binary Factorization (BF) (Neil et al., 2011) to transform the BN G into G', restricting each node to have at most two parents. This bounds the maximum factor size to three, enabling Kikuchi regions to be progressively scaled up from basic triplets.

A BN G with a child node W that depends on three parent nodes X, Y, Z can be binary factorized into its BF model G' by introducing an intermediate variable E, with its CPD having the form:

$$p(E = e_{ij} \mid X = x_a, Y = y_b) = \begin{cases} 1 & \text{if } a = i \text{ and } b = j \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where X, Y are parents of E, x_a ($a = 1, \ldots, d_X$) and y_b ($b = 1, \ldots, d_Y$) are their states, and e_{ij} denotes the state of E indexed by the pair (i, j), with $i = 1, \ldots, d_X$ and $j = 1, \ldots, d_Y$. Thus E has $d_X d_Y$ states. The remaining CPD therefore involves only three variables $p(W|E, Z)$, which can be obtained similarly to (2). For binary BNs with the parent grouping used here, introducing intermediate variables in G' does not increase the maximum CPD table size (Appendix B.1).

Methodologies for systematically constructing effective region graphs remain scarce; practitioner guidelines are predominantly limited to a few necessary conditions (Yedidia et al., 2005), which do not adequately define a well-functioning region graph. Furthermore, adding regions arbitrarily can even distort the approximation. Consequently, existing construction strategies remain largely heuristic and somewhat ad hoc (Welling, 2004).

2.2 Triplet Region Induction Algorithm

We propose introducing regions that capture variable interactions through an edge-relative sufficient candidate pool; orthogonality is established below for the complete-DAG construction. This is achieved by initializing the outer regions using an established Triplet Region Construction (TRC) algorithm (Lin et al., 2020) on the BF model G'. TRC fully exploits the structural inductive biases of a BN: since a BN adheres to a strict topological ordering, accurately approximating the pairwise joint distribution associated to each moral edge directly enhances the approximation quality of the child variable. Specifically, TRC constructs outer regions via sets of *primary triplets* and *interaction triplets* to better capture the pairwise joint distributions across moral edges.

A *primary triplet region* $\mathcal{P}_v$ encompasses the *family* of X_v (X_v and its parents $\mathrm{pa}(X_v)$), along with its corresponding CPD. The number of primary triplets thus equals the number of triplet CPDs in G'.

An *interaction triplet* $\mathcal{U}$ introduces a three-variable scope endowed with a uniform potential, facilitating consistent pairwise marginal interaction among primary triplets.

TRC identifies interaction triplets using a *coupled Markov blanket* (CMB). Let $\mathcal{E}_m$ denote the set of moral edges introduced between non-adjacent co-parents $\{X_t, E_k\}$ of a common child when constructing the moral graph $M[G']$. For each moral edge $e = (X_t, E_k)$, TRC introduces candidate interaction triplets comprising the pair $\{X_t, E_k\}$ and a node within the CMB $\partial(X_t, E_k)$. This setting guarantees an edge-relative sufficient candidate pool, as variables outside $\partial(X_t, E_k)$ are conditionally d-separated by the CMB. In particular, an interaction triplet whose three constituent pairs all correspond to edges in $M[G']$ is retained, as it directly introduces active pairwise interaction for the node pair $\{X_t, E_k\}$. For symmetric interaction triplets in $M[G']$, TRC prunes one candidate by evaluating the local conditional entropy, thereby avoiding competing candidates for the same moral edge. Any candidate triplet containing a non-edge pair in $M[G']$ can be safely discarded. After identifying all the outer regions, the CVM algorithm is used to obtain the inner regions. The details of TRC are provided in Appendix B.1. TRC provides edge-relative sufficiency through CMB-guided proposals. We also demonstrate that for the worst-case complete DAG it also satisfies the following *orthogonality* property.

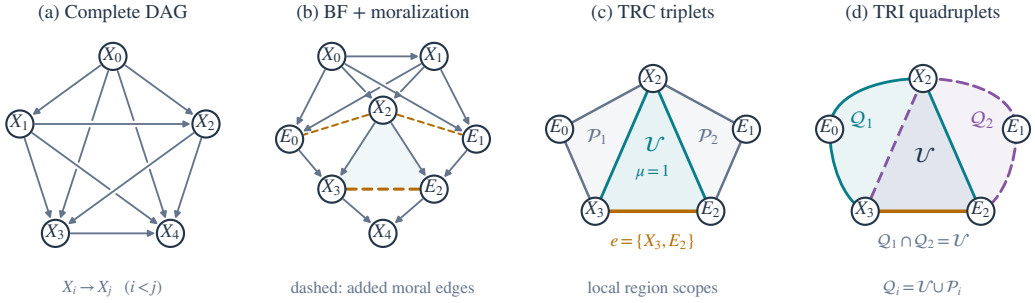


Figure 1: (a) A 5-dimensional complete DAG G; (b) its BF G', with dashed moral edges; (c) The TRC interaction triplet is associated with the node pair $\{X_t, E_k\}$ and two neighboring primary triplets; (d) TRI merges interaction triplets to primary triplets, resulting in the green and purple quadruplets.

Definition 1 (Orthogonality of region interactions)**.** *Newly induced interactions are inner regions that contain a moral-edge pair, are not primary triplet scopes, and are contained in at most one primary triplet. Orthogonality requires each such region to have counting number* -1.

The **Triplet Region Induction** (TRI) process generalizes TRC regions to higher-order compositions. It merges the appropriately chosen TRC interaction triplets with the primary triplet regions. For the complete-DAG construction analyzed below, the resulting regions have the defined orthogonality property: each newly introduced inner region has exactly two outer-region parents and counting number -1.

TRI raises size four regions by merging a TRC interaction triplet with the primary triplet region to introduce triplet interactions associated with the node pair $\{X_t, E_k\}$.

Definition 2 (Admissible triplet pair)**.** *For* $e = (X_t, E_k) \in \mathcal{E}_m$*, let* $\mathcal{U}_e = \langle \{X_s, X_t, E_k\}, \mu \rangle$ *be the selected TRC interaction triplet. A triplet pair* $(\mathcal{U}_e, \mathcal{P}_v)$ *is admissible when* $X_v \in \{X_t, E_k\}$*,* $\mathcal{P}_v$ *is a primary triplet region and* $|\mathcal{X}(\mathcal{U}_e) \cup \mathcal{X}(\mathcal{P}_v)| = 4$.

Definition 3 (TRI size four outer regions)**.** *Enumerate all admissible pairs from the TRC outer regions. For each pair, form the scope* $\mathcal{X}(\mathcal{Q}) = \mathcal{X}(\mathcal{U}_e) \cup \mathcal{X}(\mathcal{P}_v)$*, retaining each distinct scope once. The outer regions* $\mathcal{R}_0$ *consists of these quadruplets and the old regions not contained in a larger region. Reassign each original CPD to exactly one containing outer region.*

The TRI algorithm enumerates all admissible triplet pairs, ensuring that each distinct quadruplet scope is retained only once. In the worst case, G is a complete DAG and G' is its BF model; this yields at most $2|\mathcal{E}_m|$ admissible pairs and up to $2|\mathcal{E}_m|$ quadruplets prior to the removal of duplicates.

TRI retains edge-relative sufficiency for the selected candidate pool. Since the CMB provides a sufficient candidate pool, the merged regions preserve candidate coverage for the node pair $\{X_t, E_k\}$; the size-four construction includes only the admissible merged scopes described above.

Proposition 1. *For the BF model of a complete DAG constructed according to Eq.* (16)*, with distinct intermediate variables for different CPDs, TRI with size-four outer regions satisfies the orthogonality of region interactions.*

The proof is provided in Appendix B.2.

In Figure 1, a worst-case example, a 5-dimensional complete DAG G shown in (a) undergoes a BF process to obtain its BF model G' shown in (b). Two admissible triplet pairs are shown in (c). The local configuration is given by:

$$\mathcal{X}(\mathcal{U}) = \{X_2, X_3, E_2\}, \quad \mathcal{X}(\mathcal{P}_1) = \{X_2, X_3, E_0\}, \quad \mathcal{X}(\mathcal{P}_2) = \{X_2, E_1, E_2\} \tag{3}$$

where $\mathcal{U}$ is a selected interaction triplet. It gives $\mathcal{Q}_1 = \mathcal{U} \cup \mathcal{P}_1$ and $\mathcal{Q}_2 = \mathcal{U} \cup \mathcal{P}_2$. Figure 1(d) shows the two merged TRI quadruplets $\{E_0, X_2, X_3, E_2\}$ and $\{E_1, X_2, X_3, E_2\}$. We provide the corresponding TRC and TRI region graph construction details in Appendix A.1.

For the five distinct variables in (3), the TRI merge does not increase the local KL relative to TRC. The general statement and proof are given in Appendix B.3.

$$D_{\mathrm{KL}}(p\|q_3) - D_{\mathrm{KL}}(p\|q_4) = I_p(E_0; E_2 \mid X_2, X_3) + I_p(E_1; X_3 \mid X_2, E_2) \geq 0. \tag{4}$$

where I_p is the conditional mutual information, q_3 and q_4 are local distributions derived from TRC and TRI, respectively, and p denotes the true target distribution being approximated.

A general BN can be directly mapped to its associated complete DAG and factorized using the BF. This allows the TRC outer regions to be obtained directly from the model without constructing it (see Appendix B.1).

Beyond the size-four construction formalized here, TRI could be extended to construct size-five and larger outer regions by recursively merging admissible lower-order outer regions.

Applying the CVM to TRI outer regions can produce a region graph $\mathcal{G}_1$ with multiple levels, introducing containment paths and cycles that cause numerical instability and increase marginalization costs. To resolve this, we construct an equivalent two-level representation $\mathcal{G}_2$ that eliminates redundant paths while preserving region counting numbers (details in Appendix A.1).

3 TRI-Constrained Neural Kikuchi Learning

We introduce TRI-constrained neural Kikuchi learning (TRNK), which uses a neural network to parameterize outer region beliefs and shares their coefficients across overlapping regions so that all region beliefs satisfy the running-intersection constraints (Section 3.1). With fixed CPDs, optimizing the KFE yields approximate marginals. With incomplete data, these beliefs provide expected family statistics for CPD learning (Section 3.2).

3.1 Consistent Region Beliefs with Shared Coefficients

We ensure region consistency as follows. The network initially proposes a belief for each outer region; while these proposed beliefs are normalized, they may conflict with overlapping regions over shared variables. To resolve this, we construct a joint table (rather than a distribution) using a contrast basis to represent and calibrate region beliefs over shared variables. This is achieved by expressing each proposed belief as a uniform distribution plus a set of correction terms, one for each subset of its variables and is parameterized by shared coefficients. Overlapping regions subsequently utilize these shared coefficients for each common term, ensuring that all tables are normalized and consistent across the inner regions. Finally, the table is updated by applying minimal necessary modifications and eliminating negative entries to yield a valid, normalized, and nonnegative distribution.

Outer-region proposals. Each variable X_i of G' has a learnable embedding e_i. We apply L self-attention blocks (Vaswani et al., 2017),

$$[h_1, \dots, h_n] = \mathrm{Attn}^{(L)} \circ \cdots \circ \mathrm{Attn}^{(1)}([e_1, \dots, e_n]), \tag{5}$$

and a shared multilayer perceptron f maps the representations of the variables in an outer region $r \in \mathcal{R}_0$ to a proposal $\widetilde{b}_r$ over the joint state space Ω_r of r:

$$\widetilde{b}_r = \mathrm{softmax}\big(f([h_i]_{i\in\mathcal{X}(r)}; \boldsymbol{\omega}) + w \log \varphi_r\big), \tag{6}$$

where $\boldsymbol{\omega}$ denotes the network parameters, φ_r is the product of the factors assigned to r (Section 2.1), and w is a fixed constant; the second term biases each proposal toward its own factors and receives no gradient. The proposals are normalized, but in general they disagree on the variables they share.

Shared contrast coefficients. For each variable X_i with d_i states, let $\chi_{i,1}, \dots, \chi_{i,d_i-1}$ be zero-mean contrasts that are orthonormal under the uniform distribution on their states. We use scaled Helmert contrasts as shown in Lancaster (1965). For a nonempty $S \subseteq \mathcal{X}(r)$ and indices $\boldsymbol{\ell}_S = (\ell_i)_{i\in S}$ with $\ell_i \in \{1, \dots, d_i - 1\}$, let $\Phi_{S,\boldsymbol{\ell}_S}(\mathbf{x}_S) = \prod_{i\in S} \chi_{i,\ell_i}(x_i)$. Together with the constant function, these products form an orthonormal basis under the uniform distribution on Ω_r, so every distribution q on Ω_r can be written as

$$q(\mathbf{x}_r) = \frac{1}{|\Omega_r|}\Big[1 + \sum_{\varnothing\neq S\subseteq\mathcal{X}(r)} \sum_{\boldsymbol{\ell}_S} \mathbb{E}_q\big[\Phi_{S,\boldsymbol{\ell}_S}\big]\, \Phi_{S,\boldsymbol{\ell}_S}(\mathbf{x}_S)\Big]. \tag{7}$$

When a region table is marginalized over a variable to obtain a subregion marginal, every term in Eq. (7) that includes this variable becomes zero because the contrasts have a zero mean. Other terms

retain their original coefficients. The marginal on $\mathcal{X}(u)$ is determined by the coefficients $\mathbb{E}_q[\Phi_{S,\boldsymbol{\ell}_S}]$, for $S \subseteq \mathcal{X}(u)$. Two regions agree on $\mathcal{X}(u)$ when they use identical values for these coefficients.

The coefficients are shared along the edges of a region graph. For each edge $r \to u$ and for each subset $S \subseteq \mathcal{X}(u)$, the coefficients for $(S, \boldsymbol{\ell_S})$ must be equal in both r and u. Therefore, we group them into the same class. Classes that share a coefficient are merged together. Each class κ is assigned a single value a_κ, which represents a weighted average of the coefficients $\mathbb{E}_{\widetilde{b}_{r'}}[\Phi_{S,\boldsymbol{\ell_S}}]$. This average is computed from proposals in the outer regions of the class, plus an additional offset z_κ.

$$a_\kappa = \sum_{(r',S,\boldsymbol{\ell}_S)\in\kappa,\ r'\in\mathcal{R}_0} \gamma_{\kappa,r'}\, \mathbb{E}_{\widetilde{b}_{r'}}\big[\Phi_{S,\boldsymbol{\ell}_S}\big] + z_\kappa, \qquad \gamma_{\kappa,r'} \geq 0, \quad \textstyle\sum_{r'} \gamma_{\kappa,r'} = 1. \tag{8}$$

Each outer or inner region constructs its table using class values, where $\kappa(r, S, \boldsymbol{\ell}_S)$ denotes the coefficient class for $(S, \boldsymbol{\ell}_S)$ in region r:

$$\hat{b}_r(\mathbf{x}_r) = \frac{1}{|\Omega_r|}\Big[1 + \sum_{\varnothing\neq S\subseteq\mathcal{X}(r)} \sum_{\boldsymbol{\ell}_S} a_{\kappa(r,S,\boldsymbol{\ell}_S)}\, \Phi_{S,\boldsymbol{\ell}_S}(\mathbf{x}_S)\Big]. \tag{9}$$

The zero-mean contrasts ensure normalization ($\sum_{\mathbf{x}_r} \hat{b}_r = 1$), while coefficient sharing guarantees edge consistency ($\hat{b}_r^{\downarrow u} = \hat{b}_u$) across the region graph. Furthermore, because a_κ averages coefficients from different proposals, $\hat{b}_r$ may occasionally contain negative entries.

Positivity. Let the reference $\pi = \{\pi_r\}$ represent region beliefs that are strictly positive, normalized, and consistent. Initially, we define $\pi_r = \frac{1}{|\Omega_r|}$, corresponding to a uniform distribution that satisfies all three conditions. The normalization and running-intersection constraints can be expressed as linear equations involving the entries in the table. As both π and $\hat{b}$ satisfy these constraints, so do the tables

$$b_r = \pi_r + \rho\,(\hat{b}_r - \pi_r), \qquad r \in \mathcal{R}, \tag{10}$$

for every scalar ρ shared by all regions. We use these tables as the region beliefs of the KFE in Eq. (1). We select the largest value of ρ in $[0, 1]$ for which the inequality $b_r \geq \frac{\epsilon}{|\Omega_r|}$ holds across all regions and configurations, with a fixed $0 < \epsilon < 1$. After each update of the parameters, the reference is adjusted to be a convex combination of the current beliefs and the uniform distribution. In the subsequent step, we update the beliefs based on the current values rather than reverting to the uniform distribution, ensuring that a small ρ does not draw the beliefs back towards uniformity.

We continue these updates until the beliefs stabilize. According to Proposition 2 below, every update produces beliefs that are positive, normalized, and consistent, guaranteeing that the final beliefs will share these properties as well. With fixed CPDs, we optimize the KFE over $(\boldsymbol{\omega}, z)$, with b defined by Eq. (10) and the region energy specified in Appendix C.1.

Proposition 2. *For all $\boldsymbol{\omega}$, γ, and z, and every reference π as above with $\pi_r > \epsilon/|\Omega_r|$, the beliefs in Eq.* (10) *are normalized, satisfy $b_r \geq \epsilon/|\Omega_r|$, and satisfy $b_r^{\downarrow u} = b_u$ on every edge of $\mathcal{G}_2$.*

The proof is given in Appendix B.4.

3.2 Parameter Learning from Incomplete Data

Given N training instances where each instance s provides observed values $\mathbf{x}_{O_s}^{(s)}$ for variables O_s while variables H_s remain missing. Our objective is to learn the CPDs θ of the BN G. Because the CPD of every intermediate variable in G' is a deterministic function of its parents' states, the marginal belief over the family of X_v in G' directly determines its marginal b_{P_v} on the family $P_v = \{X_v\} \cup \mathrm{pa}(X_v)$ in G. We therefore optimize exclusively for the CPDs in G, and reconstruct the factors φ_r using the current parameters θ. For each instance, the observed variables are clamped, and the beliefs $b^{(s)}$ are constructed following the method in Section 3.1.

During inference, there is only a single set of beliefs to infer, so the self-attention blocks (Eq. (5)) and the perceptron f (Eq. (6)) take solely variable embeddings as input. In contrast, during learning, each training instance exhibits its own observations and corresponding posterior. Rather than optimizing beliefs separately for each of the N instances, we amortize inference over the N instances in one network, used for all instances in place of Eqs. (5)–(6), which maps the observed values of instance s and the current CPDs to the proposals $\widetilde{b}_r^{(s)}$. The CPDs are an input because they vary during training and determine the posterior.

Objective and updates. We minimize the mean evidence-clamped KFE

$$\mathcal{L}(\theta, \boldsymbol{\omega}, \gamma) = \frac{1}{N} \sum_{s=1}^{N} F_{\mathrm{KFE}}^{\mathrm{clamped}}\big(b^{(s)}, \theta\big),$$
$$F_{\mathrm{KFE}}^{\mathrm{clamped}}(b, \theta) = -\sum_{v} \mathbb{E}_{b_{P_v}}\left[\log p_\theta(X_v \mid \mathbf{X}_{\mathrm{pa}(X_v)})\right] - \sum_{r \in \mathcal{R}} c_r H(b_r). \tag{11}$$

The first term is the energy $U(b)$ of the KFE in Eq. (1). Since the CPDs are normalized, there is no need for a partition function. $\mathcal{L}$ is a Kikuchi approximation of the negative log-likelihood.

Training alternates between two steps. First, perform one gradient step on $(\boldsymbol{\omega}, \gamma)$ with θ held constant, using all training instances. Then, recompute the beliefs of all instances to obtain the expected counts:

$$N_v(y, k) = \sum_{s=1}^{N} b_{P_v}^{(s)}\big(x_v = k,\, \mathbf{x}_{\mathrm{pa}(X_v)} = y\big) \tag{12}$$

for each child state k and parent configuration y. When the beliefs are fixed, the entropy in Eq. (11) does not depend on θ, meaning that the minimizer over θ corresponds to the row-normalized counts. We use a smoothed and damped form of this minimizer:

$$p_v^{\mathrm{new}}(k \mid y) = (1 - \alpha)\, p_v^{\mathrm{old}}(k \mid y) + \alpha\, \frac{N_v(y, k) + \beta}{\sum_{j=1}^{d_v} N_v(y, j) + d_v \beta}, \tag{13}$$

where $\beta > 0$ is a pseudocount and $\alpha \in (0, 1]$ is the damping factor. Training halts when the validation NLL no longer improves. Appendix C.2 gives the derivation of the objective, the construction of $b^{(s)}$ under evidence, the network architecture, and the hyperparameters.

4 Experiments

The evaluated BNs represent a diverse range of structures: worst-case complete DAGs (Dense) (Neil et al., 2011), high-treewidth Grid BNs (treewidth up to 29) (Van den Broeck et al., 2015), Hidden Markov Models (HMMs) (Barber, 2012), the PASCAL competition BN Promedas (Elidan et al., 2011) (characterized by extreme factors where most CPD entries are 0 or 1), and real-world benchmarks like Barley (Kristensen, 1998), Diabetes (Andreassen et al., 1991), and the Pigs Gene BN (Jensen, 1998). As shown in Table 2, these models span from 9 to 894 variables and treewidths from 5 to 29.

Inference Comparisons. We begin by evaluating TRNK against traditional region graph-based message-passing and newer neural-based inference approaches. This evaluation directly tests the performance of our region graph construction and neural network inference techniques. We considered five relevant inference methods: LBP (Ihler et al., 2005), IJGP-MP (Merlin) (Marinescu, 2019), and DAMP-BP (Pretti, 2005) (LBP with a damping coefficient of 0.5) are message-passing methods. In contrast, Bethe-neural (Wiseman & Kim, 2019) and IJGP-neural are based on neural network parameterizations. Bethe-neural is an earlier amortized algorithm that optimizes Bethe free energy. It employs an ad hoc soft consistency penalty weight to ensure consistency across per-model region beliefs. Both IJGP-neural and TRNK utilize the same neural network inference solver but differ in their region graph constructions. IJGP variants use model-specific i-bounds to ensure both IJGP and TRNK remain within the same maximum region table size, which is defined as the product of the cardinalities of all variables involved in a region, thus making the comparison fair. All neural-based methods utilize the same GPU processing platform, while message-passing runs on the CPU.

Figure 2 (a) and (b) present singleton marginal inference results across 8 BNs. TRNK maintains low space complexity by bounding the region size at four, while IJGP variants adjust their i-bound accordingly. LBP and DAMP-BP outperformed the Bethe-neural on 7 out of 8 BNs, indicating that the Bethe-neural associated neural parameterization is not effectively regularizing the belief constraints across different models. In contrast, the IJGP-neural achieved results very close to those of IJGP-MP, suggesting that our proposed neural parameterization effectively captures the region graph behavior defined by the IJGP. On average, both LBP and Bethe-neural were less accurate than the IJGP variants, confirming that high-order region graph construction can achieve more accurate

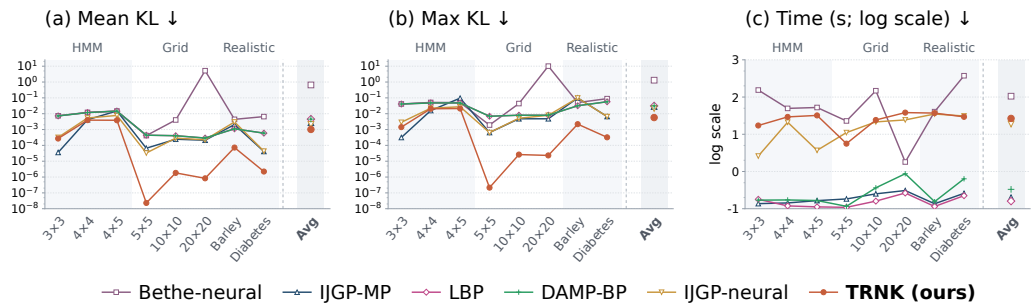


Figure 2: Singleton inference results comparisons on 8 BNs: (a) mean KL; (b) maximum KL; (c) inference time. "Avg": arithmetic mean across BNs.

results than factor graph-based methods. Overall, TRNK achieved the lowest mean KL on 7 out of 8 BNs and the lowest maximum KL on 6 out of 8, indicating that its region graph construction is more effective than IJGP and other methods. Ablation tests comparing TRI to TRC, found in Appendix E.2, demonstrate a KL improvement from 4.04×10^{-3} to 1.02×10^{-3} (a 75% reduction).

While the runtime (see Fig. 2 (c)) generally favors traditional message-passing algorithms, both LBP and DAMP-BP run on factor graphs, which have lower space complexity compared to region graphs. TRNK is more efficient than Bethe-neural, showing that its neural parameterization using the shared coefficient method is both more accurate and more efficient than the previous tuning method.

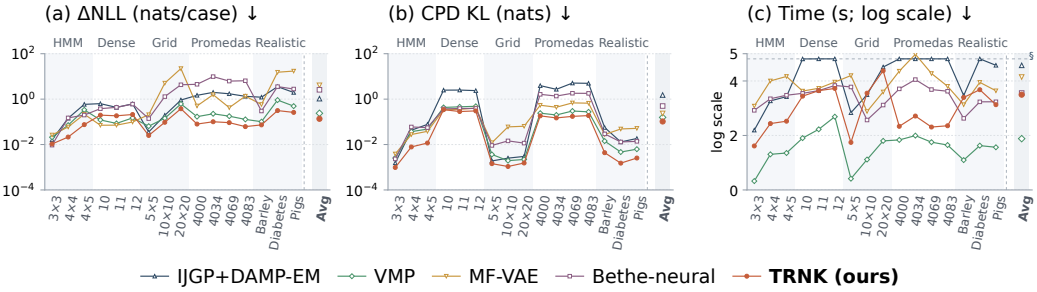


Figure 3: CPD learning on 16 BNs: (a) validation ΔNLL; (b) CPD KL; (c) learning time; lower is better. "Avg": arithmetic mean across BNs. §: includes unconverged IJGP runs.

Parameter Learning from Incomplete Data. We next evaluate TRNK learning with ELBO-based methods VMP (Winn et al., 2005; Luttinen, 2016) and categorical MF-VAE (Mnih & Gregor, 2014), and non-ELBO-based methods IJGP+DAMP-EM (Dempster et al., 1977) and Bethe-neural (Wiseman & Kim, 2019), for learning tabular CPDs on 16 BNs. All methods utilize 5000 partially observed training instances, with observation masks missingness probability of 0.3, alongside 500 complete validation instances per BN. To evaluate the quality of learned CPDs, we use two metrics. The difference in NLL (ΔNLL) is calculated as the validation NLL of the learned BN minus the NLL of the generating BN, expressed in nats per instance. The "CPD KL" is the averaged KL calculated across all CPDs in the learned BN. All amortized neural methods, including MF-VAE, Bethe-neural, and TRNK, utilize the same GPU processing capabilities.

Figure 3 (a) shows that TRNK achieved the lowest ΔNLL on 12 out of 16 BNs. In (b), it demonstrates that TRNK reached the lowest CPD KL on 15 out of 16 BNs. TRNK had the lowest average ΔNLL and CPD KL across all BNs. Compared to the ELBO-based methods, VMP and MF-VAE, TRNK reduces CPD KL by 34% relative to VMP and by 56% relative to MF-VAE, averaged across all BNs. TRNK performed slightly worse on ΔNLL in the Dense BN tests but outperformed them on CPD KL. This indicates that while the ΔNLL metric emphasizes the overall fit of the data, the CPD KL focuses on the quality of learning local dependencies. In comparison with other non-ELBO-based methods, Bethe-neural and IJGP+DAMP-EM, TRNK outperformed both in accuracy and efficiency

Table 1: Comparing VMP and TRNK on parameter learning using large datasets. Bolded items indicate better performance. The final row shows seconds per full CPD-update cycle.

Metric	Method	**HMM** 3×3			**Grid** 5×5			**Promedas_4000**		
		200k	500k	1m	200k	500k	1m	200k	500k	1m
ΔNLL	TRNK	**2.14E-4**	**-3.90E-4**	**-4.68E-4**	**1.04E-4**	**-5.43E-5**	**-9.93E-5**	**1.43E-2**	**4.57E-3**	**4.98E-3**
	VMP	9.01E-3	8.78E-3	9.18E-3	4.78E-2	4.72E-2	4.73E-2	2.65E-2	3.36E-2	2.50E+0
CPD KL	TRNK	**1.11E-4**	**5.69E-5**	**4.77E-5**	**1.70E-4**	**1.60E-4**	**1.88E-4**	**8.61E-2**	7.96E-2	**6.58E-2**
	VMP	1.36E-3	1.30E-3	1.35E-3	2.08E-3	2.02E-3	2.04E-3	1.13E-1	**7.15E-2**	7.20E-1
Total (s)	TRNK	**218.6**	**521.8**	**861.1**	196.4	**253.3**	**372.0**	**874.8**	**2262.6**	**3296.1**
	VMP	465.4	1535.3	2093.8	**92.0**	273.2	522.6	7200.2	7200.4	7200.0
s/full CPD cycle	TRNK	**0.38**	**0.67**	**1.18**	**0.36**	**0.53**	**0.83**	**4.32**	**7.28**	**8.96**
	VMP	28.11	95.19	137.62	5.93	17.73	34.02	252.46	750.67	1084.63

on most BNs. Fig. 3 (c) shows that TRNK is the second best in efficiency but slower than VMP. This is because the dataset is relatively small, and we run a series of large dataset tests shown in Table 1.

We compare TRNK and VMP across training dataset sizes ranging from 200k to 1 million instances using HMM, Grid, and Promedas BNs. Across all 9 tests, TRNK achieves a lower ΔNLL and secures lower CPD KL in 8 out of 9 cases. On the 5×5 Grid, increasing training size fivefold raises TRNK's time from 196.4 to 372.0 s ($1.89\times$), versus 92.0 to 522.6 s ($5.68\times$) for VMP. TRNK becomes faster at 500k and 1m while maintaining lower errors. On the 3×3 HMM, TRNK is faster at every size. These results confirm that TRNK amortized learning significantly improves both efficiency and accuracy over the VMP in large datasets. Appendix E.1 provides details of experimental protocols, including baseline adaptations, region budgets, metric definitions, Per-model settings, and timing boundaries.

5 Related work

Region-graph construction. Related region graph construction methods include IJGP (Mateescu et al., 2010) and LSRG (Gelfand & Welling, 2012). IJGP operates on a join graph but can be converted to an equivalent region graph. The i-bound parameter of IJGP aligns well with TRNK, as both offer bounded region control. In contrast, LSRG is not defined by a bounded region size but is defined by a tree-robustness cycle basis, making it not directly comparable to TRNK. However, Lin et al. (2020) discusses truncated LSRG and finds it inferior to TRC on BN models.

Amortized learning. Bethe and related free-energy objectives (Wiseman & Kim, 2019; Zhang et al., 2020) are parameterized with neural networks to enable neural inference, as opposed to traditional GBP message passing (Yedidia et al., 2005). Applying these objectives to BNs could serve as an alternative to classical variational inference (Winn et al., 2005; Blei et al., 2017). The amortized version (Wiseman & Kim, 2019) also performs favorably compared to amortized VI approaches such as MF-VAE (Kingma & Welling, 2013; Mnih & Gregor, 2014) and MF-IWAE (Burda et al., 2015). However, these non-ELBO methods currently lack stable neural parameterizations for region belief consistency regularization and are not scalable for obtaining region graph-based higher-order approximations. Previous amortized Bethe methods use a consistency penalty, which has two major drawbacks: finite penalty weights violate constraints (making lower objective values unreliable), and weights require per-model tuning. Instead, TRNK resolves this by eliminating disagreement directly within the mapping from proposals to beliefs.

6 Conclusion

We present the TRNK algorithm, which utilizes region graph-constrained neural inference to provide higher-order approximations with controlled space complexity for BNs. TRNK generalizes both the Bethe and Kikuchi free energy formulations, enabling non-ELBO-based amortized learning. Its associated TRI algorithm ensures sensible constructions of region graphs, and the contrast basis function mapping guarantees region consistency. The TRNK architecture is naturally compatible with GPU processing, facilitating amortized learning for massive data. Experimental results demonstrate that TRNK is an effective alternative to VI methods, achieving more accurate and efficient performance. Future work will explore using TRNK for hybrid BN inference and learning.

AI USE STATEMENT

In this work, we have not used generative AI tools for generating synthetic data sets, formulating mathematical claims, or assisting in the writing of proofs, and the rest of the required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools to edit the research paper to improve readability and summarize existing literature. We have reviewed all AI-assisted work. Specifically, all AI-edited text and summarized literature were manually verified and cross-referenced with the original sources by the authors to prevent misinterpretation or plagiarism. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Steen Andreassen, Roman Hovorka, Jonathan Benn, Kristian G Olesen, and Ewart R Carson. A model-based approach to insulin adjustment. In *AIME 91: Proceedings of the Third Conference on Artificial Intelligence in Medicine, Maastricht, June 24–27, 1991*, pp. 239–248. Springer, 1991.

David Barber. *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society: series B (methodological)*, 39(1): 1–22, 1977.

Gal Elidan, Amir Globerson, and Uri Heinemann. PASCAL 2011 probabilistic inference challenge. `http://www.cs.huji.ac.il/project/PASCAL/`, 2011.

Andrew E Gelfand and Max Welling. Generalized belief propagation on tree robust structured region graphs. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, pp. 296–305, 2012.

Alexander T Ihler, John W Fisher III, Alan S Willsky, and David Maxwell Chickering. Loopy belief propagation: convergence and effects of message errors. *Journal of Machine Learning Research*, 6 (5), 2005.

C. S. Jensen. Pedigree of breeding pigs, 1998. URL `https://www.cse.huji.ac.il/~galel/Repository/Datasets/pigs/pigs.htm`.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

K. Kristensen. A preliminary model for barley, 1998. URL `https://www.cse.huji.ac.il/~galel/Repository/Datasets/barley/barley.htm`.

HO Lancaster. The helmert matrices. *The American Mathematical Monthly*, 72(1):4–12, 1965.

Steffen L Lauritzen and David J Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 50(2):157–194, 1988.

Peng Lin, Martin Neil, and Norman Fenton. Improved high dimensional discrete bayesian network inference using triplet region construction. *Journal of Artificial Intelligence Research*, 69:231–295, 2020.

Jaakko Luttinen. Bayespy: variational bayesian inference in python. *The Journal of Machine Learning Research*, 17(1):1419–1424, 2016.

Radu Marinescu. Merlin: An extensible C++ library for probabilistic inference in graphical models. https://github.com/radum2275/merlin, 2019.

Robert Mateescu, Kalev Kask, Vibhav Gogate, and Rina Dechter. Join-graph propagation algorithms. *Journal of Artificial Intelligence Research*, 37:279–328, 2010.

Thomas P. Minka. Expectation propagation for approximate bayesian inference. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, UAI'01, pp. 362–369. Morgan Kaufmann Publishers Inc., 2001. ISBN 1558608001.

Andriy Mnih and Karol Gregor. Neural variational inference and learning in belief networks. In *International Conference on Machine Learning*, pp. 1791–1799. PMLR, 2014.

Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

Radford M Neal and Geoffrey E Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pp. 355–368. Springer, 1998.

Martin Neil, Xiaoli Chen, and Norman Fenton. Optimizing the calculation of conditional probability tables in hybrid bayesian networks using binary factorization. *IEEE Transactions on Knowledge and Data Engineering*, 24(7):1306–1312, 2011.

Marco Pretti. A message-passing algorithm with damping. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(11):P11008–P11008, 2005.

Guy Van den Broeck, Karthika Mohan, Arthur Choi, Adnan Darwiche, and Judea Pearl. Efficient algorithms for bayesian network parameter learning from incomplete data. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, UAI'15, pp. 161–170, 2015.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

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

Max Welling. On the choice of regions for generalized belief propagation. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pp. 585–592, 2004.

John Winn, Christopher M Bishop, and Tommi Jaakkola. Variational message passing. *Journal of Machine Learning Research*, 6(4):661, 2005.

Sam Wiseman and Yoon Kim. Amortized bethe free energy minimization for learning mrfs. *Advances in Neural Information Processing Systems*, 32, 2019.

Jonathan S Yedidia, William T Freeman, and Yair Weiss. Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Transactions on information theory*, 51(7): 2282–2312, 2005.

Alan L Yuille. Cccp algorithms to minimize the bethe and kikuchi free energies: Convergent alternatives to belief propagation. *Neural computation*, 14(7):1691–1722, 2002.

Zhen Zhang, Fan Wu, and Wee Sun Lee. Factor graph neural networks. *Advances in Neural Information Processing Systems*, 33:8577–8587, 2020.

Appendix guide. Appendix A contains the two-level region graph and the pseudocode of the algorithms. Appendix B provides the details of the TRC and demonstrates Propositions 1 and 2, as well as Equation (4). Appendix C includes updates to the reference distribution, learning derivations, and implementation details. Appendix D illustrates the TRC and TRI constructions using a complete DAG. Lastly, Appendix E outlines the experimental protocols and presents the results for each model.

A ALGORITHMS

This section presents the CVM and the two-level region graph, followed by pseudocode for constructing TRI regions and TRNK inference and learning.

A.1 CVM AND TWO-LEVEL REGION GRAPH

Given the TRI outer regions $\mathcal{R}_0$, we use the CVM to generate inner regions by iteratively intersecting the outer regions. Let the initial inner-region set be $\mathcal{I}^{(0)} = \mathcal{R}_0$. At each iteration $t \geq 1$, define

$$\mathcal{I}^{(t)} = \left\{ r_i \cap r_j \;\middle|\; r_i, r_j \in \mathcal{I}^{(t-1)},\ r_i \cap r_j \neq \emptyset,\ r_i \cap r_j \notin \bigcup_{s<t} \mathcal{I}^{(s)} \right\}. \tag{14}$$

The process terminates when $\mathcal{I}^{(t)} = \emptyset$. All inner regions $\mathcal{I} = \bigcup_{t \geq 1} \mathcal{I}^{(t)}$ are sorted in descending order of cardinality. Counting numbers follow the recursion of Section 2.1; regions with $c_r = 0$ are discarded.

The resulting CVM region graph $\mathcal{G}_1$ can contain multiple inner-region levels. As the outer-region size increases, these levels create additional containment paths and cycles among parent-child edges. This multi-level graph can lead to numerical instability and increase the cost of neural belief marginalization. We therefore construct a two-level representation $\mathcal{G}_2$ that removes these redundant paths while preserving the region counting numbers.

Definition 4. ***Two-level region graph.*** *Given a CVM region graph $\mathcal{G}_1$ with outer regions $\mathcal{R}_0$ and inner regions $\mathcal{I}$ with $c_r \neq 0$, the two-level region graph $\mathcal{G}_2$ is constructed as follows:*

- ***(i)*** *Level 1 of $\mathcal{G}_2$ consists of $\mathcal{R}_0$ and Level 2 consists of $\mathcal{I}$. Each region retains its counting number from $\mathcal{G}_1$.*
- ***(ii)*** *For each $r_c \in \mathcal{I}$ and $r_p \in \mathcal{R}_0$, add the directed edge $r_p \to r_c$ if $r_c \subsetneq r_p$.*
- ***(iii)*** *For each $r_c \in \mathcal{I}$ and each $r_q \in \mathcal{I}$ with $r_c \subsetneq r_q$, keep the edge to r_c from the first Level 1 region (in a fixed order) that contains r_q, and remove the edges to r_c from the other Level 1 regions that contain r_q.*
- ***(iv)*** *If r_c is left with one parent, restore the edge from one more Level 1 region that contains r_c.*

$\mathcal{G}_2$ keeps the regions and counting numbers of $\mathcal{G}_1$ rather than recomputing them from its edges, so both define the same F_{KFE}; the edges of $\mathcal{G}_2$ carry the running-intersection constraints used in Section 3. Figure 4 illustrates the CVM and the resulting two-level representation.

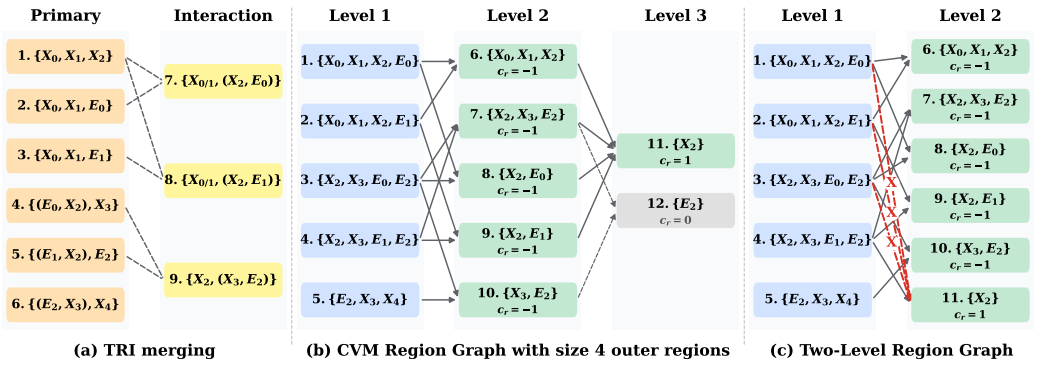


Figure 4: Region graphs for the BF model of Figure 1 (b). (a) TRI merging; (b) CVM region graph $\mathcal{G}_1$; (c) two-level representation $\mathcal{G}_2$, with removed edges dashed in red.

A.2 TRI Region Construction

Algorithm 1 builds the TRI outer regions of Section 2.2 and the region graph used by TRNK.

Algorithm 1 TRI region construction

Require: BN G
Ensure: Outer regions $\mathcal{R}_0$ with assigned CPDs, inner regions $\mathcal{I}$, counting numbers $\{c_r\}$, and $\mathcal{G}_2$
Binary factorize G into G' as specified in Appendix B.1 (2)
Initialize the TRC outer regions with the primary triplet regions $\mathcal{P}_v$ of G'
for each moral edge $e = (X_t, E_k) \in \mathcal{E}_m$ **do**
    Form the candidate interaction triplets $\{X_s, X_t, E_k\}$, $X_s \in \partial(X_t, E_k)$, and retain those whose three pairs are edges of $M[G']$
    For symmetric interaction triplets, retain the one with the smaller ΔH^j (15)
    If a candidate is retained, add it to the TRC outer regions as $\mathcal{U}_e$
end for
Form $\mathcal{X}(\mathcal{U}_e) \cup \mathcal{X}(\mathcal{P}_v)$ for every admissible triplet pair (Def. 2), retaining each distinct scope once
Let $\mathcal{R}_0$ consist of these quadruplets and the TRC outer regions not contained in a quadruplet
Assign each CPD to exactly one containing outer region
Obtain $\mathcal{I}$ and $\{c_r\}$ with the CVM, and construct $\mathcal{G}_2$ (14), Def. 4

A.3 TRNK Inference and Parameter Learning

Algorithm 2 runs on the region graph of Algorithm 1. When CPDs are fixed, it performs inference (Section 3.1); otherwise it learns the CPDs from incomplete data (Section 3.2).

Algorithm 2 TRNK inference and parameter learning

Require: Regions, $\{c_r\}$, and $\mathcal{G}_2$ from Algorithm 1; fixed CPDs θ, or N training and validation instances
Ensure: Approximate marginals $\{b_i\}$, or learned CPDs θ
Form the coefficient classes along the edges of $\mathcal{G}_2$
if the CPDs θ are fixed **then** *// inference*
    Share single-variable coefficients across all regions; set $\boldsymbol{\omega}$ at random, $z = 0$, π uniform
    repeat
        Compute the proposals $\widetilde{b}_r$ (5), (6)
        Compute the class values a_κ with uniform γ and construct the tables $\hat{b}_r$ (8), (9)
        Select the largest $\rho \in [0, 1]$ with $b_r \geq \epsilon/|\Omega_r|$ for $b = \pi + \rho(\hat{b} - \pi)$ (10)
        Update $(\boldsymbol{\omega}, z)$ using the KFE with a potential floor, treating ρ as a constant App. C.1
        Set π to a convex combination of b and the uniform distribution
    until the beliefs stabilize
    return b_i from any region containing X_i
else *// parameter learning*
    Initialize $(\boldsymbol{\omega}, \gamma)$ and the CPDs of G uniformly; fix the CPDs of the intermediate variables
    For each instance s, compute $\Omega^{(s)}$ and the initial reference App. C.2.2
    repeat
        Compute the proposals $\widetilde{b}_r^{(s)}$ and the beliefs $b^{(s)}$ for all instances (8), (9), (18)
        Take one gradient step on $(\boldsymbol{\omega}, \gamma)$ with θ fixed (11)
        Recompute $b^{(s)}$ and obtain the expected counts $N_v(y, k)$ (12)
        Update the CPDs of G (13)
        Update the reference of each instance
    until the validation NLL no longer improves
    return the CPDs θ with the lowest validation NLL
end if

B Theoretical Analysis and Proofs

B.1 Review of TRC outer regions

This subsection reviews the TRC (Lin et al., 2020) algorithm, which is utilized in Section 2.2.

CPD table size bound for the parent grouping. For the parent grouping used here, BF factorize a group of a child node's parents using intermediate variables whose states represent the group's joint states. As a result, the CPD of the node in graph G' will have the same number of entries as in graph G. The deterministic CPD of an intermediate variable whose state space has D joint states contains D^2 entries: its two parent state spaces have sizes D_1 and D_2 with $D_1 D_2 = D$.

For a node with k binary parents, whose CPD has 2^{k+1} entries, BF splits the parents into two groups: one group containing $\lceil k/2 \rceil$ variables and the other containing $\lfloor k/2 \rfloor$ variables. Each intermediate variable encodes a subset from one of these groups, therefore, its CPD has at most $2^{2\lceil k/2 \rceil} \leq 2^{k+1}$ entries. Figure 5 illustrates the relationships for nodes with three to six parents. For nodes with an odd number of parents, as shown in (a) and (c), the maximum CPD table size of an intermediate variable matches the CPD of the node itself. In contrast, for nodes with an even number of parents, as depicted in (b) and (d), the maximum CPD table size of an intermediate variable is half the size of the node's CPD. In our experiments, all variables are binary, and in every BN model we analyzed, the maximum CPD table size of G' is equal to that of G. This is the parent grouping used by Algorithm 1 and in the experiments.

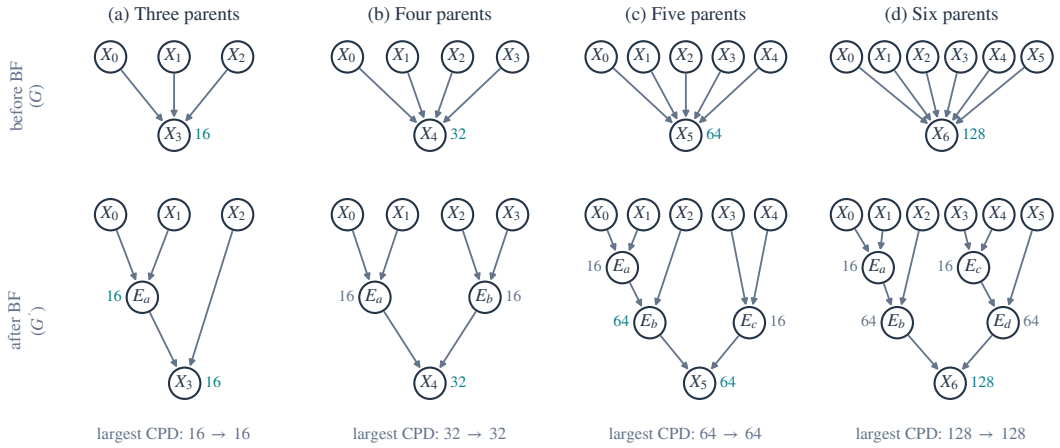


Figure 5: The maximum CPD table size before and after BF for a node with three to six parents. Top: the family of the node in G. Bottom: its BF model G'; each intermediate variable encodes the joint states of its two parents.

Complete-DAG construction for general structural analysis. For the general structural analysis, we can complete the BF model G' in a fixed topological order by adding arcs whose extra parent variables are ignored by the CPDs, and apply the BF construction for a complete DAG of Lin et al. (2020). Treating every node of G', including its intermediate variables, as an input node, this construction introduces only auxiliary variables. Marginalizing them recovers the joint distribution of G', and marginalizing the intermediate variables of G' then recovers the joint distribution of G on its original variables. We use this complete-DAG construction to index the regions in the proof of Proposition 1 in Appendix B.2. Algorithm 1 constructs its regions from the BF model G' obtained from the supplied BN with the parent grouping specified above. Note that this construction will not introduce more variables other than the input nodes in the TRC outer regions, as the auxiliary variables are only used as replicas of the input nodes.

Edge-relative sufficient interaction triplets selection. For a moral edge $e = (X_t, E_k)$, the CMB $\partial(X_t, E_k)$ is the set of neighbors of X_t or E_k in $M[G']$. Each $X_s \in \partial(X_t, E_k)$ provides a candidate $\{X_s, X_t, E_k\}$ with the uniform potential $\mu \equiv 1$. A candidate region equal to the scope of a primary

triplet is excluded. A candidate is considered a complete interaction triplet if each of its three node pairs is connected by an edge in $M[G']$. Candidates that do not meet this criterion are discarded. The candidate pool is edge-relative sufficient because the third variable of a complete interaction triplet is adjacent to both X_t and E_k, placing it in $\partial(X_t, E_k)$ and ensuring that no complete region is overlooked. The TRI method merges the selected triplets without altering this pool, thus maintaining alignment with the edge-relative sufficiency property.

Symmetric interaction triplets. For a node pair $\{X_t, E_k\}$ connected by a moral edge, if both X_t and E_k have two common parents, they will yield complete region candidates. These candidates are referred to as symmetric interaction triplets. In the candidate region involving a third variable X_{s_j}, where $j \in \{1, 2\}$ and $X_{s_j} \notin \{X_t, E_k\}$, the TRC computes

$$\Delta H^j = H(X_t \mid X_{s_j}) + H(E_k \mid X_{s_j}) \tag{15}$$

from the local pairwise marginals of the primary triplet regions. TRC determines the candidate with the smaller ΔH^j without performing inference. Each moral edge is thus associated with one interaction triplet $\mathcal{U}_e$.

B.2 Proof of Proposition 1: Orthogonality of region interactions

We describe the orthogonality of region interactions from Section 2.2 in the following way. The newly induced region interactions consist of inner regions that contain a pair of nodes connected by a moral edge and do not originate from primary triplet regions. In other words, such an inner region is not the scope of a primary triplet region, and at most one primary triplet region contains it. Each of these regions is assigned a counting number of -1, so that each of them is counted only once. With this formulation, the newly induced interaction regions are counted only once in the region graph.

For the complete-DAG construction defined by Eq. (16), with distinct intermediate variables for different CPDs, the following proof establishes the stated orthogonality property. We analyze the dense case of a complete DAG G and its BF model G', as illustrated in Figure 1; Algorithm 1 uses the BF model constructed with the parent grouping specified above.

Notation. In Figure 1 (b), the intermediate variables E_0, E_1, E_2 are numbered in the order of their creation. However, this numbering does not indicate which CPD each intermediate variable is associated with, nor does it specify the parents that each variable represents. To clarify this for the proof, we assign each intermediate variable two indices: one that corresponds to the original variable whose CPD it factorizes and another that indicates the parent variable it adds.

Let the original variables be $X_0, \ldots, X_{n-1}$ arranged in topological order. For $j \geq 2$, this construction replaces the parents $X_0, \ldots, X_{j-1}$ of X_j with intermediate variables that successively add one parent. Each intermediate variable encodes one additional parent than the previous one. Specifically, $E_{j,\ell}$ has as its parents $E_{j,\ell-1}$ and X_ℓ, which means its states represent the joint states of $(X_0, \ldots, X_\ell)$ (Eq. (2)), we can express

$$E_{j,0} = X_0, \qquad \mathrm{pa}(E_{j,\ell}) = \{E_{j,\ell-1}, X_\ell\}, \quad 1 \leq \ell \leq j-1. \tag{16}$$

The construction ends with X_j, specifically $E_{j,j-1} = X_j$, which has parents $E_{j,j-2}$ and X_{j-1}. The intermediate variables are $E_{j,1}, \ldots, E_{j,j-2}$. For instance, in Figure 1 (b), where $n = 5$, E_0 is $E_{3,1}$, E_1 is $E_{4,1}$, E_2 is $E_{4,2}$, and $E_{3,2}$ and $E_{4,3}$ denote X_3 and X_4.

We can express regions using the index defined in (16). The primary triplet region of $E_{j,\ell}$ is defined as $\mathcal{P}_{j,\ell} = \{E_{j,\ell-1}, X_\ell, E_{j,\ell}\}$. For $\ell = 1$, the two parents X_0 and X_1 are adjacent. However, for $\ell \geq 2$, the two parents X_ℓ and $E_{j,\ell-1}$ are not adjacent, which results in the addition of a moral edge:

$$e_{j,\ell} = (X_\ell, E_{j,\ell-1}), \quad 3 \leq j \leq n-1, \quad 2 \leq \ell \leq j-1,$$

where $X_t = X_\ell$, $E_k = E_{j,\ell-1}$, and the common child is $E_{j,\ell}$. For example, in Figure 1 (b), the moral edges are represented as $(X_2, E_0) = e_{3,2}$, $(X_2, E_1) = e_{4,2}$, and $(X_3, E_2) = e_{4,3}$. We denote the interaction triplet selected by TRC for $e_{j,\ell}$ as $\mathcal{U}_{j,\ell}$. With this notation, under this construction, we have:

(a) For TRC: each node pair connected by a moral edge e forms an inner region that has exactly two outer region parents, with the counting number $c_e = -1$.
(b) For TRI (Proposition 1): the newly induced region interactions involve pairwise $e_{j,\ell}$ and the triplets $\mathcal{U}_{j,\ell}$ with $\ell \geq 3$. Each of them is an inner region with counting number -1, and each $\mathcal{U}_{j,\ell}$ has exactly two outer-region parents.

Statement (b) describes how TRI merges regions. For each moral edge $e_{j,\ell}$, TRI combines the selected interaction triplet $\mathcal{U}_{j,\ell}$ with the primary triplet regions of the two nodes, X_ℓ and $E_{j,\ell-1}$, associated with the edge. However, it does not merge with the primary triplet region $\mathcal{P}_{j,\ell}$ of their common child $E_{j,\ell}$. As a result, the node pair $e_{j,\ell}$ appears as the intersection of the merged quadruplets and the outer regions containing $\mathcal{P}_{j,\ell}$.

For $\ell \geq 3$, the interaction triplet $\mathcal{U}_{j,\ell}$ manifests as the intersection of its two merged quadruplets. In the case of $\ell = 2$, the two merges yield the same quadruplet, which means $\mathcal{U}_{j,2}$ does not become an inner region.

Interaction triplets selected by TRC. First, we identify the interaction triplet $\mathcal{U}_{j,\ell}$ that the TRC selects for the moral edge $e_{j,\ell} = (X_\ell, E_{j,\ell-1})$. According to Equation (16), the intermediate variable $E_{j,\ell-1}$ is adjacent in $M[G']$ to exactly four nodes: its parents $E_{j,\ell-2}$ and $X_{\ell-1}$, its child $E_{j,\ell}$, and X_ℓ, the other parent of this child. A candidate triplet $\{X_s, X_\ell, E_{j,\ell-1}\}$ is considered complete only if X_s is one of these adjacent nodes and is also connected to X_ℓ.

The node $X_{\ell-1}$ is a parent of X_ℓ, which ensures it is always adjacent to X_ℓ. The node $E_{j,\ell}$ is a child of X_ℓ; however, the candidate $\{E_{j,\ell}, X_\ell, E_{j,\ell-1}\}$ corresponds to the primary triplet region $\mathcal{P}_{j,\ell}$ and is therefore excluded from consideration.

The node $E_{j,\ell-2}$ is adjacent to X_ℓ only when $\ell = 2$. In this specific case, it is the original variable $E_{j,0} = X_0$, which serves as a parent of X_2. Thus, for $\ell \geq 3$, the only complete candidate is $\mathcal{U}_{j,\ell} = \{X_{\ell-1}, X_\ell, E_{j,\ell-1}\}$.

In the case of $\ell = 2$, both variables X_0 and X_1 are common parents of X_2 and $E_{j,1}$. This results in two symmetric interaction triplets: $\{X_0, X_2, E_{j,1}\}$ and $\{X_1, X_2, E_{j,1}\}$. The TRC selects one of these according to Equation (15). In both scenarios, we denote the third variable in $\mathcal{U}_{j,\ell}$ as X_s, with $X_s = X_{\ell-1}$ for $\ell \geq 3$ and X_s being one of $\{X_0, X_1\}$ for $\ell = 2$.

Proof of (a). In the TRC region graph, the outer regions are the primary triplet regions and the interaction triplets. The intermediate variable $E_{j,\ell-1}$ belongs to exactly three of them: its own primary triplet region $\mathcal{P}_{j,\ell-1} = \{E_{j,\ell-2}, X_{\ell-1}, E_{j,\ell-1}\}$, the primary triplet region $\mathcal{P}_{j,\ell}$ of its child, and the interaction triplet $\mathcal{U}_{j,\ell}$. Since the moral edge $e_{j,\ell}$ contains $E_{j,\ell-1}$, every outer region that contains $e_{j,\ell}$ is one of these three regions. Their pairwise intersections are

$$\mathcal{P}_{j,\ell-1} \cap \mathcal{P}_{j,\ell} = \{E_{j,\ell-1}\}, \qquad \mathcal{P}_{j,\ell-1} \cap \mathcal{U}_{j,\ell} = \{X_s, E_{j,\ell-1}\}, \qquad \mathcal{P}_{j,\ell} \cap \mathcal{U}_{j,\ell} = e_{j,\ell}.$$

Hence $e_{j,\ell}$ is contained in exactly two outer regions, $\mathcal{P}_{j,\ell}$ and $\mathcal{U}_{j,\ell}$, whose intersection is $e_{j,\ell}$ itself, and no other inner region contains $e_{j,\ell}$. By the recursion for counting numbers in Section 2.1, $c_{e_{j,\ell}} = 1 - 2 = -1$. This also explains why TRC keeps only one of the two symmetric interaction triplets for $\ell = 2$: if both were kept, $e_{j,2}$ would be contained in three outer regions and would receive $c = 1 - 3 = -2$.

Outer regions of TRI. We next outline the outer regions produced by TRI. For the moral edge $e_{j,\ell}$, the node X_ℓ, which corresponds to X_t in Definition 2, has the primary triplet region $\mathcal{P}_{\ell,\ell-1} = \{E_{\ell,\ell-2}, X_{\ell-1}, X_\ell\}$. Meanwhile, the node $E_{j,\ell-1}$, which corresponds to E_k, has the primary triplet region $\mathcal{P}_{j,\ell-1} = \{E_{j,\ell-2}, X_{\ell-1}, E_{j,\ell-1}\}$. Each of these regions shares two variables with $\mathcal{U}_{j,\ell}$, resulting in both forming admissible triplet pairs with $\mathcal{U}_{j,\ell}$ (as defined in Definition 2).

The primary triplet region $\mathcal{P}_{j,\ell}$ of the common child also shares two variables with $\mathcal{U}_{j,\ell}$, specifically the moral edge $e_{j,\ell}$. However, this region is not the primary triplet region for X_t or E_k; thus, according to Definition 2, this pair is excluded. Merging it would not provide any additional pairwise information about $e_{j,\ell}$. The two admissible pairs yield the quadruplets.

$$\mathcal{Q}^1_{j,\ell} = \mathcal{U}_{j,\ell} \cup \mathcal{P}_{\ell,\ell-1} = \{X_{\ell-1}, X_\ell, E_{j,\ell-1}, E_{\ell,\ell-2}\},$$
$$\mathcal{Q}^2_{j,\ell} = \mathcal{U}_{j,\ell} \cup \mathcal{P}_{j,\ell-1} = \{X_{\ell-1}, X_\ell, E_{j,\ell-1}, E_{j,\ell-2}\},$$

where $\ell \geq 3$.

They are distinct because $j > \ell$: $E_{\ell,\ell-2}$ is an intermediate variable of the CPD of X_ℓ, whereas $E_{j,\ell-2}$ is an intermediate variable of the CPD of X_j, so they are different variables even though both encode $(X_0, \ldots, X_{\ell-2})$. When considering the case where $(j, \ell) = (4, 3)$, we have the quadruplets $\mathcal{Q}_1$ and $\mathcal{Q}_2$ as discussed in Section 2.2. For $\ell = 2$, the two primary triplet regions, $\mathcal{P}_{2,1} = \{X_0, X_1, X_2\}$ and $\mathcal{P}_{j,1} = \{X_0, X_1, E_{j,1}\}$, both include X_0 and X_1. Consequently, both unions result in the same quadruplet, $\mathcal{Q}_{j,2} = \{X_0, X_1, X_2, E_{j,1}\}$, regardless of which symmetric interaction triplet TRC is retained.

Every primary triplet region and every interaction triplet is included within one of these quadruplets, with the exception of the primary triplet region $\mathcal{P}_{n-1,n-2}$ concerning the last variable X_{n-1}, which does not serve as a node for any moral edge. The outer regions of the TRI consist of the quadruplets $\mathcal{Q}^1_{j,\ell}$ and $\mathcal{Q}^2_{j,\ell}$ for $\ell \geq 3$, the quadruplet $\mathcal{Q}_{j,2}$, and the region $\mathcal{P}_{n-1,n-2}$.

For instance, in Figure 4, regions 1 and 2 correspond to $\mathcal{Q}_{3,2}$ and $\mathcal{Q}_{4,2}$, while regions 3 and 4 represent $\mathcal{Q}^1_{4,3}$ and $\mathcal{Q}^2_{4,3}$, respectively. Region 5 is designated as $\mathcal{P}_{4,3}$.

A rule for counting numbers. The proof of (b) uses the following consequence of the recursion $c_r = 1 - \sum_{r' \supsetneq r} c_{r'}$ in Section 2.1. Suppose that a variable set S is contained in exactly $m \geq 2$ outer regions and that any two of these outer regions intersect exactly in S. Then no inner region lies strictly between these outer regions and S, so $c_S = 1 - m$. Moreover, these m outer regions and S together contribute $m + (1 - m) = 1$ to the sum over the ancestors of any subset of S. If $m = 1$, S is not an inner region, and the single outer region also contributes 1.

Proof of (b). Fix a moral edge $e = e_{j,\ell}$. Every inner region is an intersection of outer regions, and an inner region that contains e can only be an intersection of outer regions that contain e. We therefore first list the outer regions that contain e. They fall into two groups:

- Group A consists of the quadruplets merged from $\mathcal{U}_{j,\ell}$, namely $\mathcal{Q}^1_{j,\ell}$ and $\mathcal{Q}^2_{j,\ell}$ for $\ell \geq 3$, and $\mathcal{Q}_{j,2}$ for $\ell = 2$.
- Group B consists of the outer regions that contain the primary triplet region $\mathcal{P}_{j,\ell}$ of the common child. If $\ell < j - 1$, the child $E_{j,\ell}$ is an intermediate variable, and group B is the single quadruplet $\mathcal{Q}^2_{j,\ell+1}$. If $\ell = j - 1 < n - 2$, the child is X_j, and group B consists of the quadruplets $\mathcal{Q}^1_{h,j}$ with $h > j$, which pairwise intersect in $\mathcal{P}_{j,j-1}$. If $\ell = j - 1 = n - 2$, group B is the outer region $\mathcal{P}_{n-1,n-2}$ itself.

These are all outer regions that contain e, because every other outer region lacks $E_{j,\ell-1}$ or X_ℓ. A region of group A and a region of group B intersect exactly in e.

We now apply the counting rule to each group. In Group A, where $\ell \geq 3$, we have the intersection $\mathcal{Q}^1_{j,\ell} \cap \mathcal{Q}^2_{j,\ell} = \{X_{\ell-1}, X_\ell, E_{j,\ell-1}\} = \mathcal{U}_{j,\ell}$. Since no region in Group B contains $X_{\ell-1}$, it follows that $\mathcal{U}_{j,\ell}$ is contained solely in these two outer regions. As a result, $\mathcal{U}_{j,\ell}$ is classified as an inner region with exactly two outer-region parents, and we find that $c_{\mathcal{U}_{j,\ell}} = 1 - 2 = -1$. Thus, Group A contributes $2 - 1 = 1$.

For Group A when $\ell = 2$, $\mathcal{U}_{j,2}$ is contained only within $\mathcal{Q}_{j,2}$, which means it is not considered an inner region, so Group A contributes 1 in this case.

In Group B, applying the counting rule to $S = \mathcal{P}_{j,\ell}$ shows that Group B also contributes 1. When Group B has $m \geq 2$ regions, $\mathcal{P}_{j,\ell}$ is classified as an inner region with $c = 1 - m$. Therefore, the ancestors of e contribute a total of $1 + 1 = 2$, and consequently, $c_e = 1 - 2 = -1$.

Finally, the inner regions that contain e are the intersections of regions from groups A and B. Specifically, these are the regions $\mathcal{U}_{j,\ell}$ for $\ell \geq 3$, the region $\mathcal{P}_{j,\ell}$ when group B has two or more regions, and the region e itself.

Since the region $\mathcal{P}_{j,\ell}$ is a primary triplet scope and does not represent a newly induced region interaction, the newly induced region interactions consist solely of the node pairs $e_{j,\ell}$ and the interaction triplets $\mathcal{U}_{j,\ell}$ for $\ell \geq 3$, each assigned a counting number of -1.

B.3 Proof of Eq. (4): KL improvement

This subsection proves Eq. (4), which compares the local approximations of TRC and TRI around the moral edge (X_3, E_2) of Figure 1. We use the regions of Eq. (3): the interaction triplet $\mathcal{U} = \{X_2, X_3, E_2\}$, the primary triplet region $\mathcal{P}_1 = \{X_2, X_3, E_0\}$ of X_3, and the primary triplet region $\mathcal{P}_2 = \{X_2, E_1, E_2\}$ of E_2. TRI merges them into $\mathcal{Q}_1 = \mathcal{U} \cup \mathcal{P}_1$ and $\mathcal{Q}_2 = \mathcal{U} \cup \mathcal{P}_2$. Let p be the true distribution on the five variables $V = \{X_2, X_3, E_0, E_1, E_2\}$, and let p_r denote its marginal on a region r.

In the CVM region graph of the TRC regions $\{\mathcal{P}_1, \mathcal{U}, \mathcal{P}_2\}$, the outer regions have $c_r = 1$. The inner regions $\{X_2, X_3\}$ and $\{X_2, E_2\}$ have $c_r = -1$, while the region $\{X_2\}$ has $c_r = 0$.

In the CVM region graph of the TRI regions $\{\mathcal{Q}_1, \mathcal{Q}_2\}$, the outer regions also have $c_r = 1$, and the region $\mathcal{U}$ has $c_r = -1$.

The corresponding local approximations of p are given by the products $\prod_r p_r^{c_r}$.

$$q_3 = \frac{p_{\mathcal{P}_1}\, p_{\mathcal{U}}\, p_{\mathcal{P}_2}}{p_{\{X_2,X_3\}}\, p_{\{X_2,E_2\}}}, \qquad q_4 = \frac{p_{\mathcal{Q}_1}\, p_{\mathcal{Q}_2}}{p_{\mathcal{U}}}.$$

The regions in each region graph can be arranged as a junction tree: $\mathcal{P}_1$–$\mathcal{U}$–$\mathcal{P}_2$ with separators $\{X_2, X_3\}$ and $\{X_2, E_2\}$, and $\mathcal{Q}_1$–$\mathcal{Q}_2$ with separator $\mathcal{U}$. The first arrangement is valid because $\mathcal{P}_1 \cap \mathcal{P}_2 = \{X_2\}$ is contained in $\mathcal{U}$. Hence q_3 and q_4 are normalized distributions, and their marginals on the regions equal p_r. Since each p_r is a marginal of p, taking the expectation of $\log(p/q)$ under p gives, for both $q = q_3$ and $q = q_4$,

$$D_{\mathrm{KL}}(p\|q) = \sum_r c_r H(p_r) - H(p),$$

which is the error of the entropy approximation in Eq. (1) when the region beliefs equal the true marginals. Subtracting the two errors and grouping the terms yields

$$\begin{aligned} D_{\mathrm{KL}}(p\|q_3) - D_{\mathrm{KL}}(p\|q_4) = &\big[H(p_{\mathcal{P}_1}) + H(p_{\mathcal{U}}) - H(p_{\{X_2,X_3\}}) - H(p_{\mathcal{Q}_1})\big] \\ &+ \big[H(p_{\mathcal{P}_2}) + H(p_{\mathcal{U}}) - H(p_{\{X_2,E_2\}}) - H(p_{\mathcal{Q}_2})\big]. \end{aligned}$$

Since $\mathcal{P}_1 \cap \mathcal{U} = \{X_2, X_3\}$ and $\mathcal{P}_1 \cup \mathcal{U} = \mathcal{Q}_1$, the first bracket is the conditional mutual information $I_p(E_0; E_2 \mid X_2, X_3)$. Likewise, the second bracket is $I_p(E_1; X_3 \mid X_2, E_2)$. Both terms are nonnegative, which proves Eq. (4).

The proof uses only the fact that the five variables are distinct. In the notation of Appendix B.2, every moral edge $e_{j,\ell}$ with $\ell \geq 3$ has the same configuration, with $(X_2, X_3, E_2, E_0, E_1)$ replaced by $(X_{\ell-1}, X_\ell, E_{j,\ell-1}, E_{\ell,\ell-2}, E_{j,\ell-2})$. Therefore, Eq. (4) holds for every such moral edge.

B.4 Proof of Proposition 2: Consistent region beliefs

This subsection proves Proposition 2 of Section 3.1. We first give the Helmert contrasts and the marginalization identity that the proof uses.

Helmert contrasts. Number the states of X_i as $1, \ldots, d_i$. The scaled Helmert contrasts (Lancaster, 1965) are

$$\chi_{i,\ell}(x) = \sqrt{\tfrac{d_i}{\ell(\ell+1)}} \times \begin{cases} 1, & x \leq \ell, \\ -\ell, & x = \ell + 1, \\ 0, & x > \ell + 1, \end{cases} \qquad \ell = 1, \ldots, d_i - 1.$$

Each contrast has mean zero, and the contrasts are orthonormal under the uniform distribution on the states of X_i. The uniform distribution on Ω_r is the product of these distributions, so the products $\Phi_{S,\boldsymbol{\ell}_S}$ and the constant function form an orthonormal basis of the functions on Ω_r. Expanding a distribution q in this basis gives Eq. (7), in which the coefficient of $\Phi_{S,\boldsymbol{\ell}_S}$ is $\mathbb{E}_q[\Phi_{S,\boldsymbol{\ell}_S}]$.

Marginalization. Let u be a region with $\mathcal{X}(u) \subseteq \mathcal{X}(r)$. Summing $\Phi_{S,\boldsymbol{\ell}_S}$ over a variable in S gives zero, because the contrast of that variable has mean zero. Summing over a variable outside S

multiplies it by the number of states of that variable. Marginalizing the table in Eq. (9) onto $\mathcal{X}(u)$ therefore removes every term with $S \not\subseteq \mathcal{X}(u)$ and keeps the others with the same class values:

$$\hat{b}_r^{\downarrow u}(\mathbf{x}_u) = \frac{1}{|\Omega_u|}\Big[1 + \sum_{\varnothing \neq S \subseteq \mathcal{X}(u)} \sum_{\boldsymbol{\ell}_S} a_{\kappa(r,S,\boldsymbol{\ell}_S)}\, \Phi_{S,\boldsymbol{\ell}_S}(\mathbf{x}_S)\Big].$$

On an RG edge $r \to u$, the coefficients of every $S \subseteq \mathcal{X}(u)$ are in the same class for r and u, so $\kappa(r, S, \boldsymbol{\ell}_S) = \kappa(u, S, \boldsymbol{\ell}_S)$ and $\hat{b}_r^{\downarrow u} = \hat{b}_u$.

Proof. Summing $\hat{b}_r$ over all of Ω_r removes every term except the constant, so $\sum_{\mathbf{x}_r} \hat{b}_r(\mathbf{x}_r) = 1$, and $\hat{b}$ satisfies $\hat{b}_r^{\downarrow u} = \hat{b}_u$ on every edge by the identity above. The reference π satisfies the same linear equations by assumption, so $b = \pi + \rho(\hat{b} - \pi)$ satisfies them for every scalar ρ shared by all regions. For positivity, assume $\pi_r > \epsilon/|\Omega_r|$ for all regions and configurations; the reference update in Appendix C.1 guarantees this. An entry with $\hat{b}_r(\mathbf{x}_r) \geq \pi_r(\mathbf{x}_r)$ satisfies $b_r(\mathbf{x}_r) \geq \pi_r(\mathbf{x}_r) > \epsilon/|\Omega_r|$ for every $\rho \in [0, 1]$. An entry with $\hat{b}_r(\mathbf{x}_r) < \pi_r(\mathbf{x}_r)$ decreases linearly in ρ and stays at least $\epsilon/|\Omega_r|$ exactly when $\rho \leq (\pi_r(\mathbf{x}_r) - \epsilon/|\Omega_r|)/(\pi_r(\mathbf{x}_r) - \hat{b}_r(\mathbf{x}_r))$. The largest $\rho \in [0, 1]$ with $b_r \geq \epsilon/|\Omega_r|$ everywhere is therefore the minimum of 1 and these bounds.

C Technical Details

C.1 Consistent Region Beliefs for Inference

This subsection gives the update of the reference and the implementation details of Section 3.1 for inference. The proof of Proposition 2 is given in Appendix B.4.

Reference update. After each parameter update, the reference is set to $\pi_r = (1 - m)\, b_r + m/|\Omega_r|$, a convex combination of the current beliefs and the uniform distribution with weight m. Both are normalized and consistent, so π is as well, and $\pi_r \geq ((1 - m)\epsilon + m)/|\Omega_r| > \epsilon/|\Omega_r|$. The uniform distribution also satisfies this bound, since $\epsilon < 1$. Hence every reference meets the conditions of Proposition 2, and the proposition holds at every step. The weight starts at $m = 10^{-3}$; every 20 steps it is multiplied by $\exp\big(2(0.99 - \rho)\big)$ and clipped to $[10^{-3}, 0.2]$, so it grows only while $\rho < 0.99$.

Single-variable marginals. Proposition 2 covers the edges of $\mathcal{G}_2$. For inference, the coefficients of each variable X_i are shared by all regions that contain X_i (Algorithm 2, line 3), so by the marginalization identity of Appendix B.4 all tables $\hat{b}_r$ with $X_i \in \mathcal{X}(r)$ have the same marginal on X_i. The uniform distribution has this property, and it is preserved by Eq. (10) and by the reference update. Every region containing X_i therefore gives the same marginal b_i, which we report.

Implementation. The weights γ are uniform, and the parameters are optimized separately for each BN from a random initialization, with the offsets z_κ initialized to zero. The proposal bias in Eq. (6) uses $w = 2.4$ and $\log \max(\varphi_r, 10^{-9})$, and the KFE energy is $E_r = -\log \max(\varphi_r, 10^{-4})$. This floor makes the energy finite at zero potentials but also changes the objective for positive potentials below 10^{-4}. We use $\epsilon = 10^{-7}$. The scale ρ is treated as a constant during back-propagation, so the loss adds a penalty on the entries of $\hat{b}$ below $\epsilon/|\Omega_r|$,

$$\mu \sum_r \Big[\sum_{\mathbf{x}_r} \big(\epsilon - |\Omega_r|\, \hat{b}_r(\mathbf{x}_r)\big)_+^2 + \big(\epsilon - |\Omega_r| \min_{\mathbf{x}_r} \hat{b}_r(\mathbf{x}_r)\big)_+^2\Big], \qquad \mu = 10,$$

which is zero when $\rho = 1$ is admissible. The loss also adds $0.55\, \frac{1}{|\mathcal{R}_0|} \sum_{r \in \mathcal{R}_0} D_{\mathrm{KL}}(b_r \| \bar{b}_r)$, where $\bar{b}_r$ is the belief of the previous step and receives no gradient; this term is zero when the beliefs stop changing.

C.2 Parameter Learning Details

The learning network, its hyperparameters, and the stopping rule are given in Appendix E.1.3.

C.2.1 LIKELIHOOD APPROXIMATION

Each CPD is a normalized table, so $\sum_{\mathbf{x}} \prod_v p_\theta(x_v \mid \mathbf{x}_{\mathrm{pa}(X_v)}) = 1$ and no partition-function term is needed. Missing values are assumed ignorable, as for the value-independent masks in our experiments. For an instance with observed values $\mathbf{x}_O$, missing variables H, and any distribution q on $\mathbf{x}_H$,

$$\sum_{\mathbf{x}_H} q(\mathbf{x}_H) \log \frac{q(\mathbf{x}_H)}{p_\theta(\mathbf{x}_O, \mathbf{x}_H)} = D_{\mathrm{KL}}(q \| p_\theta(\mathbf{X}_H \mid \mathbf{x}_O)) - \log p_\theta(\mathbf{x}_O) \tag{17}$$

(Dempster et al., 1977; Neal & Hinton, 1998). Replacing the entropy of q by the CVM entropy $\sum_r c_r H(b_r)$ of consistent region beliefs gives the evidence-clamped KFE of Eq. (11). Two approximations remain: the CVM entropy, and the restriction to beliefs produced by the network. Consistent region beliefs on a loopy region graph need not be the marginals of any q, so the evidence-clamped KFE is neither an upper bound on the negative log-likelihood nor the negative of an ELBO (Wainwright & Jordan, 2008). Unlike the MRF objective of Wiseman & Kim (2019), no free-energy term or maximization step is needed.

C.2.2 EVIDENCE-CONDITIONED BELIEFS

For instance s, the beliefs of Section 3.2 are

$$b^{(s)} = \pi + \rho_s \, P\big(\hat{b}^{(s)} - \pi\big), \tag{18}$$

where the reference π, the projection P, and the scale ρ_s are specified below.

Shared coefficients under evidence. The expansion base for instance s is the product of point masses on $\mathcal{X}(r) \cap O_s$ and uniform distributions on the remaining variables, so the contrasts of the remaining variables are orthonormal under it and the marginalization argument of Appendix B.4 applies unchanged. Coefficients whose S contains an observed variable are dropped. The class weights are $\gamma_{\kappa,r'} = \exp(6 \tanh g_{r'}) / \sum_{r''} \exp(6 \tanh g_{r''})$, with one parameter $g_{r'}$ per outer-region occurrence, initialized to zero and shared across instances. Single-variable coefficients are shared only along edges, and there is no offset z.

Support and reference. $\Omega_r^{(s)}$ contains the configurations of $\mathcal{X}(r)$ that agree with the observed values and with the deterministic CPD of every intermediate variable, pruned by arc consistency over the region graph. The initial reference is the set of region marginals of the joint distribution that fixes the observed variables, is uniform on the missing original variables, and sets each intermediate variable to its deterministic value. After each epoch the reference π is updated to $0.9\, b^{(s)}$ plus 0.1 times its initial value. Both terms satisfy the constraints and are positive on $\Omega^{(s)}$, so the reference does as well.

Projection. Stack all region tables of an instance into one vector. The matrix A has one row per region, which sums its entries, and one row per edge $r \to u$ and configuration $\mathbf{x}_u$, which takes the difference between the marginal of r and the entry of u. A difference δ with $A\delta = 0$ has zero sum in every region and satisfies $\delta_r^{\downarrow u} = \delta_u$ on every edge, so adding it preserves normalization and consistency. Let W be the diagonal matrix with entries π on $\Omega^{(s)}$ and zero elsewhere. For a vector v restricted to $\Omega^{(s)}$,

$$Pv = v - WA^\top (AWA^\top)^+ Av,$$

so that $APv = 0$ and Pv vanishes outside $\Omega^{(s)}$. The linear system is solved by Jacobi-preconditioned conjugate gradients with relative tolerance 10^{-10}, and the backward pass applies $P^\top$ with the same solver. The constraint residual is checked after every E-step. With $\delta^{(s)} = P(\hat{b}^{(s)} - \pi)$ and $M_s = \max_{r,\, \mathbf{x}_r \in \Omega_r^{(s)}} [-\delta_r^{(s)}(\mathbf{x}_r)/\pi_r(\mathbf{x}_r)]_+$, the scale is $\rho_s = (1-\epsilon)/\max\{1, M_s\}$ with $\epsilon = 10^{-7}$. It is not detached. When G' has no intermediate variables, P is the identity.

Properties. Since π satisfies the constraints and the correction lies in the null space of A, the beliefs are normalized and consistent on every edge. Both terms vanish outside $\Omega^{(s)}$. On $\Omega_r^{(s)}$, entries with a nonnegative correction satisfy $b_r^{(s)} \geq \pi_r$, and the others satisfy $b_r^{(s)} \geq \pi_r(1 - \rho_s M_s) \geq \epsilon\, \pi_r$. These statements hold up to the linear-solver tolerance.

C.2.3 CPD UPDATE

Each CPD of G is assigned to one outer region, outer regions have $c_r = 1$, and inner regions carry no factors, so $\sum_r c_r \log \varphi_r = \sum_v \log p_\theta(x_v \mid \mathbf{x}_{\mathrm{pa}(X_v)})$. Taking expectations under consistent beliefs gives the energy term of Eq. (11), with b_{P_v} read from an outer region that contains the family of X_v in G'. When the parents of X_v are encoded by intermediate variables, their fixed deterministic CPDs map this family marginal to b_{P_v} on the original family.

With the beliefs fixed, the entropy does not depend on θ, and the minimizer over a CPD row with $\sum_k N_v(y,k) > 0$ is $N_v(y,k)/\sum_j N_v(y,j)$. Eq. (13) smooths this target with β and damps the step with α; entries are then floored at 10^{-9} and each row is renormalized. The beliefs are not exact posteriors and change between epochs, so the alternation is not exact EM and does not guarantee a monotone increase of the likelihood.

C.2.4 INITIALIZATION AND EVALUATION

The CPDs of the original variables start uniform and the network starts from a random initialization; neither uses the generating CPDs. The proposal bias $w \log \varphi_r$ of Eq. (6) is not used, since the CPDs change during training. Only the CPDs of the original variables are learned, the CPDs of the intermediate variables stay fixed, and all metrics are computed on the original variables. For complete held-out instances, the validation negative log-likelihood is

$$\mathrm{NLL}_{\mathrm{val}}(\theta) = -\frac{1}{N_{\mathrm{val}}} \sum_{s=1}^{N_{\mathrm{val}}} \sum_v \log p_\theta\big(x_v^{(s)} \mid \mathbf{x}_{\mathrm{pa}(X_v)}^{(s)}\big), \tag{19}$$

which is exact and requires no inference; checkpoints are selected by it. CPD KL averages $D_{\mathrm{KL}}(p_\star(\cdot \mid y) \| p_\theta(\cdot \mid y))$ over the parent configurations y of each node and then over nodes. The generating CPDs $p_\star$ are used only to simulate data and to compute this diagnostic.

C.3 IMPLEMENTATION DETAILS

Inference network. Each variable of G' has a learnable embedding of dimension 16. The encoder has $L = 2$ blocks, each with single-head scaled dot-product attention and a SwiGLU feed-forward layer of hidden width 32, both preceded by RMSNorm. The shared head f of Eq. (6) is a two-layer SwiGLU MLP with RMSNorm applied to the concatenated representations of the variables of an outer region. There is no head for inner regions.

Inference optimization. We use AdamW with learning rate 5×10^{-3} and weight decay 0.01 (none on z), multiplied by 0.8 every 750 updates. Gradients are clipped at norm 5; once ten gradient norms are recorded, the threshold becomes three times their recent median, and at least 1. Optimization stops when the mean over outer regions of the mean squared change of their beliefs stays below 10^{-10} for 60 consecutive checks, with a cap of 10^6 updates. The seed is 10.

Parameter learning. The learning network, schedule, and stopping rule are given in Appendix E.1.3.

Compute. Region construction (Algorithm 1) runs on CPU and is excluded from the reported times. The neural solver is implemented in PyTorch; each run uses one NVIDIA A800 GPU unless stated otherwise.

D EXAMPLE

D.1 TRI CONSTRUCTION ON A SEVEN-VARIABLE COMPLETE DAG

Figure 6 illustrates the edge-wise TRC selection and the subsequent TRI merges for a seven-variable complete DAG. Repeated labels in panels (c)–(d) denote the same BF variables; the symbols $\mathcal{U}$, $\mathcal{Q}_i$, and $\mathcal{P}_{X_6}$ are scope annotations, not additional nodes. The detailed candidate and merging rules are given below so that the figure itself remains readable.

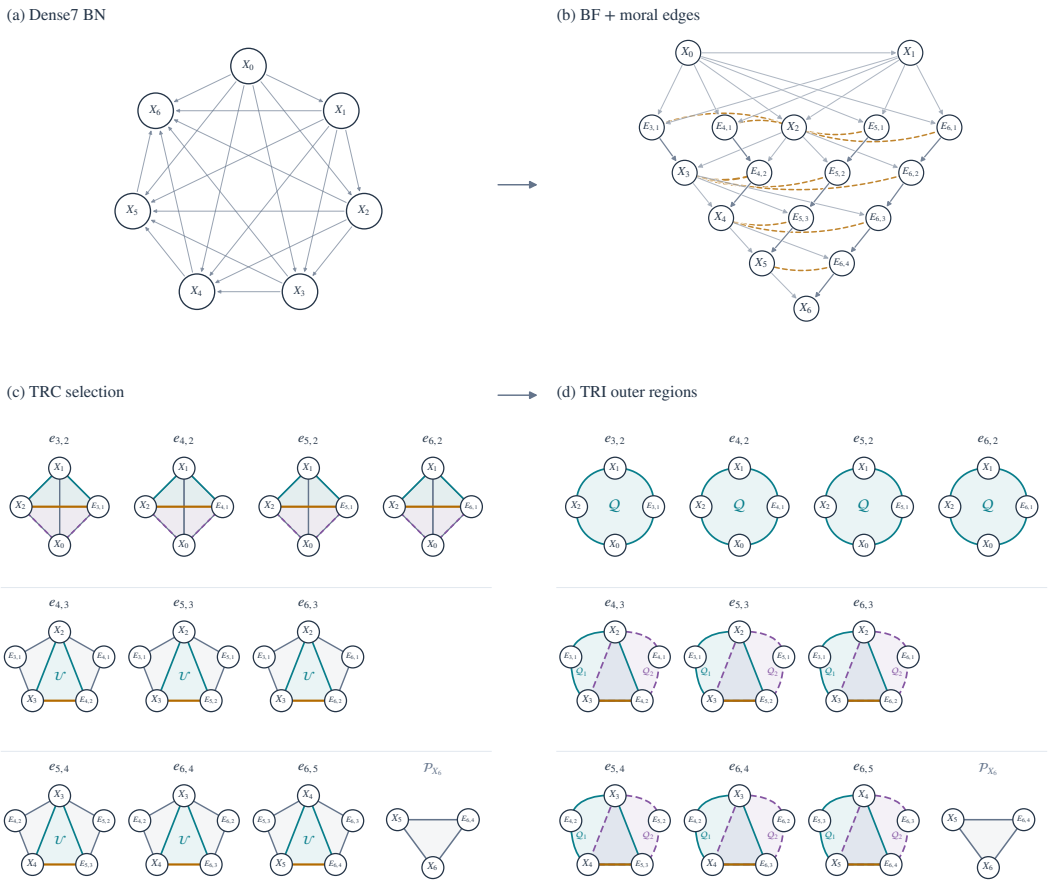


Figure 6: (a) A 7-dimensional complete DAG; (b) BF model with moral edges dashed; (c) TRC interaction triplets; (d) TRI quadruplets.

BF representation and moral edges. The original variables are $X_0, \ldots, X_6$. To illustrate the complete-DAG construction used above, for each $3 \le j \le 6$ let $E_{j,0} = X_0$, $E_{j,j-1} = X_j$, and $\mathrm{pa}(E_{j,\ell}) = \{E_{j,\ell-1}, X_\ell\}$ for $1 \le \ell \le j-1$. Thus the display contains 7 original and 10 intermediate variables, 31 directed arcs, and the 10 moral edges

$$e_{j,\ell} = \{X_\ell, E_{j,\ell-1}\}, \qquad 3 \le j \le 6, \quad 2 \le \ell \le j-1.$$

The amber dashed links in panel (b) are these moral edges; they are added to make the TRC candidates visible, not to replace any directed BF arc. The display illustrates this complete-DAG construction; Algorithm 1 uses the parent grouping specified above for the supplied BN.

TRC region selection. There are 15 primary triplets, one for each two-parent BF family. TRC selects one interaction triplet for each moral edge. At $\ell = 2$ the candidates have third variable X_0 or X_1 and are ranked by the conditional-entropy criterion ΔH^j in Eq. (15); panel (c) uses X_1 only to illustrate the construction without specifying CPDs. For $\ell \ge 3$, the unique candidate is $\mathcal{U}_{j,\ell} = \{X_{\ell-1}, X_\ell, E_{j,\ell-1}\}$. In panel (c), teal marks the candidate shown as selected and purple marks the other complete boundary candidate; the colors indicate the selection logic, not an additional region or a numerical score.

TRI merging. Let $\mathcal{P}_1$ and $\mathcal{P}_2$ be the primary triplets with child X_ℓ and $E_{j,\ell-1}$, respectively. TRI forms both endpoint unions $\mathcal{Q}_i = \mathcal{U} \cup \mathcal{P}_i$, $i = 1, 2$, excluding the common-child primary triplet. The 20 endpoint unions therefore contain four duplicate unions from the $\ell = 2$ cases, leaving 16 distinct size-four regions. The only retained primary triplet is $\mathcal{P}_{X_6} = \{E_{6,4}, X_5, X_6\}$; each model factor is assigned once. In panel (d), teal and purple outline the two endpoint scopes $\mathcal{Q}_1$ and $\mathcal{Q}_2$, while

the gray region marks the shared BF context. The choice between X_0 and X_1 at every $\ell = 2$ edge changes only the local TRC label: after the endpoint unions, all such choices yield the same final TRI scope set. The drawing shows these local outer scopes and is not a depiction of the complete CVM or its counting-number graph.

E EXPERIMENTS

E.1 EXPERIMENTAL PROTOCOL AND REPORTING

We give the method settings, evaluation metrics, and timing boundaries for fixed-parameter inference and CPD learning from incomplete data. Per-model results and the large-sample learning study follow these protocols.

E.1.1 INFERENCE COMPARISONS

Methods and shared settings. Our method (TRNK) utilizes TRI regions in conjunction with the neural solver discussed in Section 3. The IJGP-neural method provides IJGP regions (Mateescu et al., 2010) to the same solver and settings outlined in Appendix C.3. It runs on the join graph without requiring CVM conversion. TRI regions contain at most four variables. The Bethe-neural approach adapts the method described by Wiseman & Kim (2019) and includes its own neural network and a soft consistency penalty.

IJGP-MP applies join-graph propagation (Mateescu et al., 2010) using Merlin (Marinescu, 2019), while LBP employs loopy belief propagation on the factor graph. DAMP-BP is a variant of LBP that incorporates probability-space damping, defined as $m_{t+1} = 0.5\widehat{m}_{t+1} + 0.5m_t$, which is applied to normalized messages. All methods utilize the same inputs and reference marginals and are initialized with a seed of 10.

Region-capacity rule and bound conventions. IJGP-MP chooses its bound from the supplied model representation, independently of inference KL. For BN g, let F_g be the maximum original CPD table size and T_g the largest TRI region table, both measured in numbers of entries rather than numbers of variables:

$$F_g = \max_{f \in \mathcal{F}_g^{\mathrm{orig}}} \prod_{v \in \mathrm{scope}(f)} d_v^{\mathrm{orig}}, \qquad T_g = \max_{R \in \mathcal{R}_g^{\mathrm{TRI}}} \prod_{v \in R} d_v^{\mathrm{BF}}, \qquad \rho_g = \frac{T_g}{F_g}.$$

The capacity target is $F_g \rho_g = T_g$. For binary inputs, the IJGP i-bound, the largest number of variables in a cluster, is

$$b_g = \lceil \log_2(F_g \rho_g) \rceil = \lceil \log_2 T_g \rceil \,.$$

IJGP-neural uses b_g directly. For IJGP-MP and IJGP+DAMP-EM, Merlin's `--ibound` is set to $b_g - 1$: a merged clique may contain one more variable, while larger input factors are retained.

Table 2 lists the number n of original variables, an upper bound w on the treewidth of the moral graph, F_g, T_g, and the i-bound b_g. Dense$k$ has $k+1$ binary variables: $X_0, \ldots, X_{k-2}$ form a complete DAG, X_{k-1} has the same parents as X_{k-2}, and X_k has the parents X_{k-2} and X_{k-1}; for example, Dense10 has $n = 11$ and $F_g = 2^9 = 512$. The bound w is the induced width of the best of 100 randomized min-fill elimination orderings, computed with the generalized belief propagation code that also builds the IJGP region graphs; we report the lowest value over seeds 1, 2, 3, 10, and 42. For IJGP-MP and IJGP-neural, the largest table actually built is measured by a construction check with the same input, bound, and seed, excluded from inference time; it equals T_g for both methods on every BN. Equal largest tables do not imply equal region counts, total memory, or runtime. LBP and DAMP-BP have no region bound.

Reference marginals and evaluation. Inference KL is recomputed from the saved marginals of the supplied runs on original BN variables only. For a fixed set $\mathcal{V}_{\mathrm{eval}}$ of evaluation variables, define

$$\mathrm{KL}_{\mathrm{node}} = \frac{1}{|\mathcal{V}_{\mathrm{eval}}|} \sum_{v \in \mathcal{V}_{\mathrm{eval}}} D_{\mathrm{KL}}(p_{\mathrm{ref},v} \| b_v) \,.$$

Table 2: BN statistics. n: original variables; w: treewidth upper bound; F_g and T_g: largest original CPD and largest TRI region, in table entries; b_g: IJGP i-bound.

	HMM			Dense			Grid			Promedas				Realistic		
BN	3×3	4×4	4×5	10	11	12	5×5	10×10	20×20	4000	4034	4069	4083	Barley	Diabetes	Pigs
n	9	16	20	11	12	13	25	100	400	461	894	414	374	48	413	441
w	5	7	7	9	10	11	5	13	29	22	10	25	25	5	5	11
F_g	16	32	32	512	1 024	2 048	8	8	8	8	8	8	8	16	8	8
T_g	32	64	64	512	1 024	2 048	16	16	16	8	8	8	8	64	16	16
b_g	5	6	6	9	10	11	4	4	4	3	3	3	3	6	4	4

The maximum is taken over the same set. $\mathcal{V}_{\text{eval}}$ contains the original variables and excludes the intermediate variables introduced by BF. All methods use the same evaluation variables within each row. Predictions are floored at 10^{-9} and renormalized, and KL uses natural logarithms.

Figure and timing. Figure 2 shows the eight models in Table 3. Its "Avg" is the equal-BN arithmetic mean over these eight models. Each Bethe-neural, IJGP-neural, and TRNK run uses one NVIDIA A800 GPU, and the message-passing methods run on CPU. Neural times are the recorded execution times of setup, optimization, and readout, excluding region construction and KL evaluation; Bethe-neural time covers its optimization loop. Merlin times are subprocess wall times, excluding reference generation and clique-construction checks. IJGP-neural and TRNK stop after 60 consecutive checks with mean per-region belief MSE below 10^{-10}; Bethe-neural stops after 60 consecutive maximum node-marginal changes below 10^{-5}; Merlin uses message tolerance 10^{-10} and a cap of 10^6 sweeps. All runs stop by their convergence rule.

E.1.2 Per-model inference results

Table 3 gives the eight models shown in Figure 2. Each cell lists mean KL, maximum KL, and time in seconds, from top to bottom; lower is better. Bold marks the lowest unrounded mean KL, and shading identifies TRNK. The † and $*$ on BN names indicate IJGP-neural bounds of five and six; otherwise the bound is four. The last row is the mean over the eight BNs with equal weight, the "Avg" of Figure 2; its maximum KL is the mean of the per-BN maxima.

Table 3: Fixed-CPD inference on original BN variables.

	Neural		*Traditional*		*Region-based*	
BN	Bethe-neural	IJGP-MP	LBP	DAMP-BP	IJGP-neural	**TRNK (Ours)**
HMM						
$3\times3^{\dagger}$	7.39E-3 4.12E-2 154.2	**3.57E-5** 3.14E-4 0.137	7.25E-3 4.09E-2 0.179	7.25E-3 4.09E-2 0.170	3.20E-4 2.86E-3 2.6	2.70E-4 1.42E-3 17.2
$4\times4^{*}$	1.21E-2 5.15E-2 49.6	4.17E-3 1.58E-2 0.143	1.18E-2 4.79E-2 0.119	1.18E-2 4.79E-2 0.172	5.35E-3 2.22E-2 21.1	**3.93E-3** 2.05E-2 29.3
$4\times5^{*}$	1.48E-2 5.05E-2 52.4	1.49E-2 9.62E-2 0.165	1.45E-2 4.79E-2 0.112	1.45E-2 4.79E-2 0.166	7.77E-3 2.74E-2 3.7	**3.85E-3** 2.12E-2 32.1
Grid						
5×5	4.14E-4 1.92E-3 22.8	6.62E-5 6.46E-4 0.183	4.40E-4 6.90E-3 0.109	4.40E-4 6.90E-3 0.116	3.29E-5 6.43E-4 11.1	**2.32E-8** 2.15E-7 5.6
10×10	4.02E-3 4.37E-2 147.8	2.44E-4 4.78E-3 0.251	4.08E-4 8.02E-3 0.160	4.08E-4 8.02E-3 0.365	2.87E-4 5.17E-3 21.5	**1.83E-6** 2.63E-5 24.2
20×20	5.19E+0 1.01E+1 1.8	2.11E-4 4.78E-3 0.308	2.90E-4 8.02E-3 0.264	2.90E-4 8.02E-3 0.870	2.25E-4 7.90E-3 24.3	**8.31E-7** 2.32E-5 38.2
Realistic						
Barley*	4.41E-3 5.04E-2 40.5	2.05E-3 9.86E-2 0.136	1.11E-3 3.19E-2 0.115	1.11E-3 3.19E-2 0.155	3.01E-3 9.81E-2 35.0	**7.29E-5** 2.19E-3 37.1
Diabetes	6.55E-3 8.92E-2 373.8	4.12E-5 6.63E-3 0.259	5.91E-4 5.68E-2 0.224	5.91E-4 5.68E-2 0.636	4.31E-5 6.62E-3 31.0	**2.23E-6** 3.23E-4 29.4
Avg (8 BNs)	6.55E-1 1.30E+0 105.4	2.71E-3 2.85E-2 0.198	4.55E-3 3.10E-2 0.160	4.55E-3 3.10E-2 0.331	2.13E-3 2.14E-2 18.8	**1.02E-3** 5.71E-3 26.6

E.1.3 PARAMETER-LEARNING RESULTS ON ORIGINAL BNS

Data. The 16-BN comparison contains the eight BNs in Table 3, three additional Dense BNs, four Promedas models, and Pigs. Each BN has 5000 training instances and 500 complete validation instances, with missing probability 0.3 on the original variables. All methods use the same instances and masks; training and validation-mask seeds are 42 and 1234. All metrics score original BN variables.

Baselines. The settings are:

- *IJGP+DAMP-EM* applies IJGP (Mateescu et al., 2010), implemented in Merlin (Marinescu, 2019), to each incomplete instance after adding indicator factors for the observed values, and uses the normalized cluster belief covering each family as expected statistics for a row-normalized M-step. The E-step has a finite i-bound and message budget, so this is approximate EM. It uses uniform initial CPDs, 10 inner sweeps per instance, probability-space damping 0.5, model-specific i-bounds (Table 2), the first-pass elimination order of each instance, and seed 42; it stops after three consecutive maximum CPD changes below 10^{-5}, with a cap of 100000 outer iterations. Seven entries (Dense11–12, Diabetes, and the four Promedas models) use the last complete iteration before the 18-hour limit and are marked §.
- *VMP* uses the Dirichlet–Categorical nodes of BayesPy 0.6.6 with a 10^{-3} symmetric Dirichlet prior, at most 50 updates, and tolerance 10^{-5}, and reports posterior-mean CPDs.
- *MF-VAE* is the categorical mean-field VAE of Wiseman & Kim (2019): a shared recognition network, a node-factorized posterior, REINFORCE with a learned baseline for the recognition network, and the exact expected CPD gradient under that posterior. It uses batch size 5000, node and hidden dimensions 16 and 64, and learning rate 0.002.
- *Bethe-neural* is the BN adaptation of Wiseman & Kim (2019): a two-layer Transformer (embedding and hidden sizes 100, dropout 0.2) predicts factor and singleton beliefs under an L2 consistency penalty, with learning rate 0.001. Because the CPDs are normalized, the free partition term is analytic and the MRF maximization step is omitted. Its batch size is 5000, except 512 on the four Promedas models, Barley, Diabetes, Pigs, and Grid 10×10/20×20 (marked †); its penalty multiplier is 1.5, except 1.0 on Diabetes.

Both neural baselines keep the CPDs with the best validation score, with an epoch cap of 100000 and patience 100 (improvement thresholds 10^{-5} for MF-VAE and 10^{-8} for Bethe).

TRNK. TRNK uses a three-layer, width-64 graph encoder over the BF model, including its intermediate variables. Each layer averages messages from parents, children, and co-parents in a residual update; observed nodes retain their input representations. The inputs are node embeddings, masked observations, and current CPDs. A mixture of $K = 8$ factorized distributions supplies outer-region proposals, restricted to configurations compatible with the observations and deterministic BF CPDs and then renormalized. Mixture probabilities and contrast reconstruction use double precision. The reference update retains 0.9 of its previous value, and projection tolerance is 10^{-10}.

The original CPDs start uniform. We use Adam with initial learning rate 0.005 and Eq. (13) with $\beta = 0.01$ and $\alpha = 0.5$. The learning rate halves after two unsuccessful 20-check validation windows, with a 20-check cooldown and a floor of 1% of its initial value. Training stops after 60 checks without a validation-NLL improvement greater than 10^{-5}, with a cap of 2000 epochs; we report the best-validation checkpoint. Hyperparameters were selected on validation data, with no separate test set.

Metrics. Each cell reports complete-data validation ΔNLL (learned minus generating-BN NLL on the same instances, in nats per instance) and node-averaged CPD KL, computed from the same saved CPDs. ΔNLL can be negative on a finite validation sample when the learned BN assigns higher likelihood to those instances than the generating BN. In Table 4, each cell lists ΔNLL and CPD KL from top to bottom; bold marks the lowest unrounded value. The last row is the equal-BN mean over all 16 models, the "Avg" of Figure 3; bold marks the lowest mean.

Timing. Table 5 reports training times and selected/stopping epochs. TRNK time runs from the start of the epoch loop to termination and includes in-loop validation; it excludes setup, region construction, and final scoring. Each MF-VAE, Bethe-neural, and TRNK run uses one NVIDIA A800 GPU. For the seven IJGP runs reaching the 18-hour limit, Table 5 reports 64800 s. Selected and Stop are the validation-selected and final TRNK epochs. The last row is the mean over the 16 BNs; for IJGP+DAMP-EM it uses the displayed times.

Table 4: Original-BN parameter-learning errors on 16 BNs.

BN	IJGP+ DAMP-EM	VMP	MF-VAE	Bethe-neural	**TRNK** (Ours)
HMM					
3×3	1.62E-2	2.10E-2	2.64E-2	**9.41E-3**	1.06E-2
	1.56E-3	2.54E-3	3.85E-3	2.33E-3	**9.77E-4**
4×4	1.44E-1	7.16E-2	5.97E-2	1.49E-1	**2.14E-2**
	4.27E-2	3.70E-2	2.76E-2	5.95E-2	**7.84E-3**
4×5	5.88E-1	3.36E-1	2.16E-1	1.99E-1	**7.53E-2**
	7.72E-2	4.78E-2	3.82E-2	5.56E-2	**1.17E-2**
Dense					
10	6.33E-1	1.21E-1	**7.09E-2**	3.80E-1	1.97E-1
	2.49E+0	4.31E-1	**3.36E-1**	3.72E-1	3.44E-1
11	4.33E-1[§]	8.52E-2	**7.13E-2**	4.35E-1	1.84E-1
	2.51E+0	4.58E-1	3.55E-1	3.87E-1	**2.84E-1**
12	6.18E-1[§]	1.49E-1	**9.84E-2**	6.04E-1	2.10E-1
	2.37E+0	4.84E-1	4.04E-1	3.90E-1	**3.03E-1**
Grid					
5×5	3.56E-2	6.44E-2	2.06E-1	1.39E-1	**2.52E-2**
	1.90E-3	3.66E-3	1.21E-2	9.12E-3	**1.44E-3**
10×10	1.93E-1	1.44E-1	5.08E+0	1.29E+0[†]	**9.55E-2**
	2.52E-3	1.95E-3	6.00E-2	1.45E-2	**1.08E-3**
20×20	9.12E-1	6.14E-1	2.22E+1	4.29E+0[†]	**3.78E-1**
	3.00E-3	2.20E-3	6.39E-2	1.16E-2	**1.54E-3**
Promedas					
4000	1.47E+0[§]	1.69E-1	5.02E-1	4.50E+0[†]	**7.98E-2**
	3.94E+0	2.41E-1	5.39E-1	1.63E+0	**1.83E-1**
4034	1.98E+0[§]	2.26E-1	1.57E+0	9.82E+0[†]	**1.01E-1**
	2.71E+0	2.00E-1	4.40E-1	1.37E+0	**1.49E-1**
4069	1.72E+0[§]	1.77E-1	4.22E-1	6.22E+0[†]	**9.21E-2**
	5.09E+0	3.00E-1	6.83E-1	1.81E+0	**1.74E-1**
4083	1.33E+0[§]	1.28E-1	1.35E+0	6.45E+0[†]	**6.16E-2**
	4.97E+0	2.75E-1	6.62E-1	1.79E+0	**1.88E-1**
Realistic					
Barley	1.22E+0	1.02E-1	5.81E-1	3.00E-1[†]	**7.34E-2**
	5.63E-2	1.42E-2	2.76E-2	2.95E-2	**4.37E-3**
Diabetes	3.54E+0[§]	9.11E-1	1.53E+1	3.56E+0[†]	**3.13E-1**
	1.30E-2	4.68E-3	4.86E-2	1.30E-2	**1.52E-3**
Pigs	1.97E+0	4.92E-1	1.72E+1	2.79E+0[†]	**2.60E-1**
	1.85E-2	6.25E-3	5.16E-2	1.38E-2	**2.55E-3**
Avg (16 BNs)	1.05E+0	2.38E-1	4.06E+0	2.57E+0	**1.36E-1**
	1.52E+0	1.57E-1	2.35E-1	4.97E-1	**1.04E-1**

E.1.4 LARGE-SAMPLE CPD LEARNING

Table 1 uses HMM 3×3, Grid 5×5, and Promedas_4000, each with 200000, 500000, and 1000000 training instances and missing probability 0.3. TRNK and VMP use the same data and masks, and the same 500 complete validation instances are used across sizes within each BN; learning and mask seeds are 42. Scores evaluate original BN variables on the validation set.

TRNK. The runs use five A800 GPUs. Each epoch takes one neural step on at most 40000 instances and recomputes the beliefs of the full training set for the counts. The learning rate starts at 0.005; validation patience is 60 checks with minimum improvement 10^{-5}, and the checkpoint with the lowest validation NLL is reported.

VMP. VMP uses BayesPy 0.6.6 with a 10^{-3} Dirichlet prior, tolerance 10^{-5}, and posterior-mean CPDs; it runs on CPU. HMM and Promedas have a two-hour budget: HMM converges after 16, 16,

Table 5: Recorded parameter-learning time (s) and TRNK checkpoint epochs.

BN	IJGP+ DAMP-EM	VMP	MF-VAE	Bethe-neural	**Ours** loop	Selected	Stop
HMM							
3×3	157	2.12	1 200	835	41.2	277	380
4×4	1 860	20.3	10 000	2 230	273	724	784
4×5	2 660	22.7	14 700	2 990	334	600	660
Dense							
10	64 300	80.1	4 260	3 520	2 770	569	629
11	64 800[§]	168	5 380	4 420	4 410	719	779
12	64 800[§]	488	9 140	6 820	5 320	494	554
Grid							
5×5	683	2.59	15 800	6 030	55.6	597	660
10×10	3 120	13.1	759	369	3 570	1 199	1 259
20×20	32 400	63.3	3 960	1 310	23 700	1 780	1 840
Promedas							
4000	64 800[§]	68.5	22 900	5 130	216	68	160
4034	64 800[§]	98.8	95 300	11 200	516	107	200
4069	64 800[§]	56.7	18 800	4 840	204	66	160
4083	64 800[§]	44.5	6 360	4 190	227	70	190
Realistic							
Barley	2 920	12.5	1 350	424	2 490	1 224	1 283
Diabetes	64 800[§]	42.4	8 910	1 720	4 780	1 012	1 072
Pigs	37 600	36.8	4 360	1 720	1 350	877	940
Avg (16 BNs)	37 500	76.3	13 900	3 610	3 140	649	722

and 15 sweeps at the three sizes, and Promedas reaches the limit after 25, 7, and 4 complete sweeps, so its scores use partial snapshots. Grid has a 50-update limit and converges at sweep 15 at every size.

Timing. TRNK time includes setup, training, validation, and final scoring, and excludes region construction; VMP time is the recorded duration until the result is saved. The per-cycle time is one TRNK epoch (the maximum training-loop time across GPUs divided by the number of epochs) or one complete VMP sweep; each cycle updates every CPD once.

E.2 TRC vs. TRI Region Construction

Protocol. We compare TRC with TRI on the eight BNs of Table 3. The region-comparison runs use seed 10, state dimension 16, two attention layers, learning rate 5×10^{-3}, potential floor 10^{-4}, and stopping tolerance 10^{-10} with 60 consecutive convergence checks and a cap of 10^6 updates.

Table 6: TRC–TRI inference comparison on 8 BNs.

	Mean KL ↓		Max KL ↓		Time (s) ↓	
BN	TRC	TRI	TRC	TRI	TRC	TRI
HMM 3×3	7.06E-3	**2.70E-4**	3.97E-2	**1.42E-3**	23.2	**17.2**
HMM 4×4	1.17E-2	**3.93E-3**	4.69E-2	**2.05E-2**	**28.7**	29.3
HMM 4×5	1.35E-2	**3.85E-3**	4.66E-2	**2.12E-2**	**31.2**	32.1
Grid 5×5	**1.29E-8**	2.32E-8	**1.05E-7**	2.15E-7	**3.8**	5.6
Grid 10×10	**1.56E-6**	1.83E-6	**1.84E-5**	2.63E-5	**19.5**	24.2
Grid 20×20	**7.00E-7**	8.31E-7	**2.31E-5**	2.32E-5	**32.3**	38.2
Barley	7.30E-5	**7.29E-5**	2.49E-3	**2.19E-3**	45.9	**37.1**
Diabetes	4.42E-6	**2.23E-6**	8.31E-4	**3.23E-4**	58.2	**29.4**
All (8 BNs)	4.04E-3	**1.02E-3**	1.71E-2	**5.71E-3**	30.3	**26.6**

Results. Table 6 shows lower mean node KL for TRI on 5 of 8 BNs and lower maximum node KL on 5 of 8. Giving each BN equal weight, mean KL decreases from 4.04×10^{-3} to 1.02×10^{-3} (74.9%), and the average of per-BN maximum KL decreases from 1.71×10^{-2} to 5.71×10^{-3} (66.5%). The largest relative mean-KL gains occur on the HMM series. On the three Grid BNs, TRC has slightly lower KL, but both constructions are close to exact (mean KL at most 1.83×10^{-6}), so these differences are of no practical significance. The average recorded inference times are 30.3 s and 26.6 s for TRC and TRI, respectively.