

Laplacian-Guided R-Vine Copula Learning for Bayesian Networks with Mixed-Type Data

afrooz moradbeiky

Semnan University

Farzin Yaghmaee

f_Yaghmaee@semnan.ac.ir

Semnan University

Article

Keywords:

Posted Date: August 27th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-9961781/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Laplacian-Guided R-Vine Copula Learning for Bayesian Networks with Mixed-Type Data

Afrooz Moradbeiky¹ and Farzin Yaghmaee^{2,*}

¹Electrical and Computer Engineering, Semnan University, Semnan, Iran

²Electrical and Computer Engineering, Semnan University, Semnan, Iran

*f.Yaghmaee@semnan.ac.ir

ABSTRACT

Bayesian network structure learning from mixed-type data — combining continuous, ordinal, and categorical variables — remains challenging, as most existing methods assume homogeneous variable types and require preprocessing transformations that distort dependency structure. Copula-based approaches are further restricted to numerical data, introducing information loss when categorical variables are encoded. We propose the Hybrid Copula Bayesian Network (HCBN), an algorithm that learns Bayesian network structure from mixed data without any transformation. HCBN combines a Laplacian graph derived from the Gower similarity matrix with a Regular Vine copula framework. Laplacian spectral analysis produces a feature ordering that captures global dependency structure across all variable types, guiding both the R-vine copula matrix construction and a novel distance-based two-step pruning strategy that replaces explicit independence tests. Evaluated on six benchmark datasets against PC, Hill-Climbing, Grow-Shrink, Max-Min HC, IAMB, and Tabu Search, HCBN achieves the highest log-likelihood and lowest AIC in the majority of comparisons. Model parameter count decreases as sample size increases, confirming that BIC regularization in the feature ordering score controls complexity at scale.

Keywords: Bayesian network; structure learning; mixed data; Regular vine copula;

1 Introduction

Bayesian networks (BNs) are probabilistic graphical models that compactly represent joint probability distributions over a set of random variables by encoding conditional independence relationships through a directed acyclic graph (DAG)^{1,2}. Their capacity to model uncertainty, causal structure, and complex multivariate dependencies has made them widely applicable in domains ranging from bioinformatics and medical diagnosis to risk analysis and social network modeling^{3,4}. Constructing a Bayesian network from data involves two interrelated tasks: structure learning, which identifies the DAG encoding the conditional independence structure, and parameter learning, which estimates the conditional probability distributions associated with each node given its parents^{5,6}.

Structure learning is NP-hard in the general case^{1,2}, and the search space over DAGs grows super-exponentially with the number of variables, rendering exact methods infeasible beyond small networks. This challenge is further compounded in the presence of mixed data—datasets that combine continuous, ordinal, and categorical variables—since most classical BN learning algorithms assume a single data type and require preprocessing transformations that may distort the original dependency structure⁷.

A key ingredient for dependency modeling across variable types is the copula framework⁸. Copula functions allow the joint distribution of a multivariate random vector to be decomposed into its marginal distributions and a dependence function that captures the full dependency structure independently of the marginal forms. Among copula families, Regular Vine (R-vine) copulas—constructed from hierarchical tree sequences of bivariate pair copulas—offer exceptional flexibility in modeling asymmetric and tail dependencies across many variables^{9–11}. However, the space of R-vine structures is itself super-exponential in the number of variables, making unconstrained optimization intractable¹². Recent work has begun to address structure learning for BNs from mixed-type data using copula-based approaches. Castelletti (2024)¹³ developed a Bayesian methodology for structure learning in directed networks with mixed continuous, discrete, ordinal, and binary variables, leveraging a copula framework and incorporating prior structural knowledge. Grzegorzczak (2025)¹⁴ proposed a fully Bayesian framework for structure learning from mixed datasets. While their approach offers robust uncertainty quantification, it relies heavily on specific parametric assumptions and computational sampling. In contrast, our proposed HCBN avoids intensive sampling by leveraging a transformation-free copula framework combined with spectral feature ordering. The Copula-PC algorithm¹⁵ extended the classical PC algorithm by replacing parametric independence tests with copula-based correlation measures. Score-based approaches for mixed variables have also been proposed^{16,17}. Nevertheless, existing hybrid methods that combine dependency

discovery and structure optimization for mixed data remain computationally intensive, typically requiring data transformation or relying on sequential greedy heuristics.

In this work, we propose the Hybrid Copula Bayesian Network (HCBN), a three-phase algorithm that (i) computes a feature ordering via Laplacian spectral analysis applied to the Gower similarity matrix of the mixed data, (ii) constructs a guided R-vine copula matrix by integrating the feature ordering with Pair Copula Construction (PCC) dependency estimation, and (iii) applies a novel two-step pruning strategy that uses ordered feature distance as an independence criterion, without recourse to explicit statistical independence tests or data transformation⁷. The primary contributions of this work are as follows:

- A data-transformation-free feature ordering procedure for mixed data, based on the spectral properties of the Laplacian of the Gower similarity matrix, that simultaneously captures global dependency structure and provides a principled ordering for R-vine matrix initialization.
- A hybrid scoring function combining BIC-based copula likelihood with Laplacian spectral scores, enabling a unified feature relevance ranking applicable to unsupervised mixed datasets.
- A two-step pruning strategy operating directly on the ordered dependency matrix, using feature distance thresholding as a proxy for conditional independence, reducing CPT complexity without requiring explicit independence tests.
- Empirical evaluation on six benchmark datasets demonstrating that HCBN achieves higher log-likelihood and lower AIC than PC, Hill-Climbing, IAMB, Grow-Shrink, Max-Min HC, and Tabu Search, with parameter counts that decrease as the number of observations grows.

The remainder of this paper is organized as follows: Section 2 reviews the theoretical background. Section 3 surveys related work. Section 4 details the proposed HCBN algorithm. Section 5 presents experimental results, and Section 6 concludes the paper.

2 Background

2.1 Bayesian Networks

A Bayesian network is a tuple (G, P) where $G = (V, E)$ is a DAG with nodes V representing random variables and directed edges E encoding direct dependencies, and P is a set of local conditional probability distributions¹⁸. By the Markov condition, the joint distribution factorizes as:

$$P(X_1, \dots, X_n) = \prod_{i=1}^n P(X_i \mid \text{Parents}(X_i)) \quad (1)$$

Structure learning seeks to identify the DAG G that best explains an observed dataset $\mathcal{D} = \{x^1, \dots, x^M\}$ of M records, each with N variables. The posterior over structures is:

$$P(G \mid \mathcal{D}) \propto P(G) \int P(\mathcal{D} \mid G, \theta) P(\theta \mid G) d\theta \quad (2)$$

Parameter learning, given a fixed DAG structure G , estimates the vector of local conditional probabilities θ via maximum likelihood estimation (MLE) or maximum a posteriori (MAP) estimation^{5,6}. MLE aims to maximize the global log-likelihood function:

$$\ell(\theta; \mathcal{D}) = \sum_{m=1}^M \sum_{i=1}^N \log p(x_i^m \mid x_{pi}^m, \theta) \quad (3)$$

where x_{pi}^m explicitly denotes the configured values of the parent variables of X_i within the specific observation m . Alternatively, MAP estimation augments this continuous optimization with a parameter prior $P(\theta \mid G)$, directly enabling the systematic incorporation of domain knowledge⁶. V-structures (also referred to as immoralities) of the causal form $X \rightarrow Y \leftarrow Z$ —where no direct structural edge exists between the parent nodes X and Z —play a central role. Identifying these configurations is critical for resolving directional ambiguities, although complex non-linear multivariate dependencies continue to pose severe challenges for classical parametric network approaches¹⁰.

2.2 Regular Vine Copulas

Sklar’s theorem⁸ states that any d -dimensional joint distribution F with marginals $F_1, \dots, F_d$ can be written as:

$$F(x_1, \dots, x_d) = C(F_1(x_1), \dots, F_d(x_d)) \quad (4)$$

where $C : [0, 1]^d \rightarrow [0, 1]$ is a copula function uniquely defined on the range of the marginals. Pair Copula Constructions (PCCs) decompose the joint density into a product of bivariate conditional copula densities and marginal densities^{10,19}:

$$f(u_1, \dots, u_k) = c(F_1(u_1), \dots, F_d(u_d)) \cdot f_1(u_1) \dots f_k(u_k) \quad (5)$$

Regular Vine (R-vine) copulas organize PCCs into a nested sequence of trees $T_1, \dots, T_{k-1}$, where tree T_1 models unconditional pairwise dependencies, and tree T_i for $i \geq 2$ models conditional dependencies given shared conditioning sets determined by the proximity condition^{9,11}. The number of distinct R-vine structures on k variables grows as $2^{\frac{(k-2)(k-3)}{2}} \cdot k!$, making exhaustive search infeasible for $k \geq 5$ ¹². The truncation level L limits the vine to the first L trees; dependencies at levels above L are treated as conditionally independent⁹. The Vuong test²⁰ is commonly used to determine L . Maximum likelihood estimation of the pair-copula parameters $\hat{\phi}$ proceeds tree by tree:

$$\hat{\phi} = \arg \max_{\phi} \sum_n \log c(\hat{F}_{k_1}(D_{k_1n}), \hat{F}_{k_2}(D_{k_2n}) \mid \phi) \quad (6)$$

Recent work²¹ has introduced 0-1 programming approaches based on genetic algorithms to select the R-vine structure guided by dependence lists, improving the heuristics of the classical maximum spreading tree. In the context of vine copula structure learning, the sequential greedy heuristic introduced by Dissmann et al. (2013) remains a foundational benchmark. However, as noted in recent comprehensive reviews by Nagler²², simplified vine models can struggle with complex dependency structures. To address the limitations of deterministic greedy searches, Vatter and Nagler²³ introduced a random search alternative for vine structure learning. Our HCBN approach mitigates the combinatorial challenges of the greedy search differently: rather than relying on randomized search space exploration, we use a global Laplacian spectral ordering to constrain and guide the R-vine matrix construction. Matrix and cherry-tree representations of R-vine structures²⁴ provide efficient algorithmic tools for structure storage and manipulation. Copula-based mixture models for continuous data are discussed in²⁵, and copula-based classification methods in²⁶.

3 Related Work

Bayesian network structure learning algorithms are conventionally categorized into three families: constraint-based, score-based, and hybrid methods^{27,28}. Constraint-based algorithms, such as the PC algorithm^{18,29} and its variants, construct the graph skeleton by iteratively performing conditional independence tests and orienting edges by applying Meek’s orientation rules. The PC algorithm’s correctness depends on the faithfulness assumption and the power of the independence test, both of which may be compromised with small samples or non-Gaussian marginals. The Copula-PC algorithm¹⁵ addressed the latter limitation by using copula-based dependence measures in place of partial correlations. Grow-Shrink³⁰, IAMB³¹, and MMHC³² are further constraint-based and hybrid approaches that exploit Markov blanket discovery to reduce the search space. Score-based approaches assign a quality score—such as BIC, AIC, or Bayesian Dirichlet equivalence—to candidate structures and search for the highest-scoring DAG. Greedy algorithms including Hill-Climbing and Greedy Equivalence Search (GES)^{6,33} are widely used despite their susceptibility to local optima. For mixed data, Andrews et al.^{16,17} proposed scoring functions compatible with mixed variable types. Integer programming approaches³⁴ have recently demonstrated improved accuracy in Markov equivalence class recovery while reducing computational time. Hybrid methods combine the constraint-based and score-based paradigms. MMHC³² uses IAMB to restrict the parent candidate set before running a greedy score-based search. BaBReN employs the Bayes-Ball algorithm for d-separation-based CPD discovery³. The BN learning literature has also explored copula-based hybrid strategies: Haff et al.⁹ combined R-vine copulas with BIC-guided Maximum Spanning Trees for structure learning in continuous data, while Salinas-Gutiérrez et al.³⁵ embedded copula estimation into Estimation of Distribution Algorithms. Sequential Bayesian selection of vine copulae is addressed in³⁶. Mixed data present an additional challenge since most methods assume homogeneous variable types. Solorio-Fernández et al.⁷ proposed a spectral feature selection approach for mixed data, and Tsagris et al.³⁷ studied constraint-based causal discovery with mixed types. Raghu et al.³⁸ evaluated a range of causal structure learning methods specifically on mixed data types. A recent Bayesian approach by Castelletti (2024)¹³ models mixed continuous, discrete, ordinal, and binary variables through a copula framework with posterior averaging over DAG structures, providing rigorous uncertainty quantification. By contrast, the HCBN algorithm proposed in the present work avoids data transformation entirely and provides a computationally efficient approximation via Laplacian-guided ordering and distance-based pruning.

4 Proposed Algorithm: Hybrid Copula Bayesian Network (HCBN)

The Hybrid Copula Bayesian Network (HCBN) is a three-phase algorithm for learning Bayesian network structure from datasets containing a mixture of continuous and categorical variables. The algorithm is built around a single unifying design principle: the entire pipeline operates natively on mixed data, without requiring any form of data type conversion or encoding. This is achieved by grounding all dependency computations in the Gower similarity coefficient³⁹, which yields a well-defined, normalized pairwise similarity for any combination of numerical and categorical variable types.

The three phases are: (1) Feature Ordering, which uses Laplacian spectral analysis of the Gower similarity matrix to rank features by their collective dependency contribution; (2) R-Vine Copula Matrix Construction, which builds a guided Pair Copula Construction (PCC) dependency model initialized by the feature ordering; and (3) Two-Step Pruning, which refines the dependency matrix into a directed acyclic graph using a combination of distance-based independence testing and AIC-guided edge removal. Each phase is described in detail in the following subsections.

4.1 Phase 1: Feature Ordering via Laplacian Spectral Analysis

The first phase produces a total ordering of the n features in the dataset. This ordering captures the relative importance of each feature in terms of its global dependency contribution to the dataset, and will later be used both to initialize the R-vine matrix and to define the independence threshold in the pruning phase. Crucially, the ordering is derived without transforming categorical variables into numerical values.

4.1.1 Gower Similarity Matrix

Given a dataset with m observations and n features, the pairwise Gower similarity R_{ij} between observations i and j is computed as a weighted average of per-feature similarities³⁹:

$$R_{ij} = \frac{\sum_k W_{ijk} S_{ijk}}{\sum_k W_{ijk}} \quad (7)$$

The per-feature similarity S_{ijk} depends on the variable type of feature k . For a continuous or ordinal feature, it is computed as a normalized absolute difference:

$$S_{ijk} = 1 - \frac{|X_{ik} - X_{jk}|}{r_k} \quad (\text{continuous / ordinal}) \quad (8)$$

where r_k is the observed range of feature k across the dataset. For a categorical feature, it is an exact-match indicator:

$$S_{ijk} = I[X_{ik} = X_{jk}] \quad (\text{categorical}) \quad (9)$$

The weight W_{ijk} is set to 1 if both observations have a valid value for feature k , and 0 if either is missing. The resulting similarity matrix R is a symmetric $m \times m$ matrix with entries in $[0, 1]$, where $R_{ij} = 1$ indicates that observations i and j are identical across all features, and $R_{ij} = 0$ indicates maximum dissimilarity. This formulation avoids any ordinal encoding of categorical levels and is numerically stable across variable types⁴⁰.

4.1.2 Step 1.2 — Affinity and Laplacian Matrices

The similarity matrix R is converted into a positive semi-definite affinity matrix D using a Gaussian (radial basis) kernel:

$$D_{ij} = \exp\left(-\frac{\|S_{ij}\|^2}{2t}\right) \quad (10)$$

where $t > 0$ is a bandwidth parameter controlling the scale of the neighborhood (set to $t = 4$ in all experiments). D_{ij} is large when observations i and j are similar and decays rapidly for dissimilar pairs. The degree matrix B is a diagonal matrix with $B_{ii} = \sum_j D_{ij}$. The unnormalized graph Laplacian is then⁴¹:

$$L = B - D \quad (11)$$

The Laplacian L encodes the connectivity structure of the data manifold. Its eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$ reflect the spectral clustering properties of the data: a small λ_2 (Fiedler value) indicates that the data has a near-disconnected structure, while a large spread of eigenvalues suggests rich multi-scale dependency. The eigenvectors of L correspond to the harmonic functions on the data graph and can be used for dimensionality reduction; here we use only the eigenvalues for scoring.

4.1.3 Step 1.3 — Spectral Dispersion Score

The spectral dispersion score ρ for a given feature subset of size k is defined as the normalized pairwise eigenvalue spread:

$$\rho = \frac{\sum_{i < j} |\lambda_i - \lambda_j|}{\delta} \quad (12)$$

where $\delta = \sum_{i=2}^{k+1} \lambda_i$. A high ρ indicates that the eigenvalues are widely spread, meaning the subset captures a rich and diverse dependency structure. A low ρ indicates near-degenerate or redundant features. The normalization by δ prevents ρ from growing arbitrarily as the subset size increases.

4.1.4 Step 1.4 — Feature Relevance Scoring via Forward Selection

Each feature f_i is scored by measuring the effect of its removal from the active feature set. For each feature f_i ($i = 1, \dots, n$), a reduced dataset $\mathcal{D}_{-i}$ is constructed by excluding f_i . Two complementary quality indicators are then computed on $\mathcal{D}_{-i}$:

- (a) **New spectral score ρ_{new} :** Steps 1.1–1.3 are repeated on $\mathcal{D}_{-i}$ to obtain the spectral dispersion score of the reduced set. A large drop from ρ to ρ_{new} indicates that f_i contributed substantially to the diversity of the dependency structure.
- (b) **BIC score from copula fitting:** An R-vine copula matrix (RVM) is fitted to $\mathcal{D}_{-i}$ using maximum likelihood, and the Bayesian Information Criterion is computed:

$$\text{BIC} = k_{\text{param}} \cdot \ln(n_{\text{obs}}) - 2 \cdot \ln(\hat{\phi}) \quad (13)$$

where k_{param} is the number of estimated copula parameters, n_{obs} is the number of observations, and $\hat{\phi}$ is the maximized log-likelihood of the fitted copula model. A high BIC (less negative) indicates a better-fitting model on the reduced dataset, implying that f_i did not contribute structurally important dependency information.

The two scores are combined into a single feature relevance score:

$$\text{Order}_i = \alpha \cdot \text{BIC}_{-i} + \beta \cdot \rho_{\text{new}_{-i}} \quad (14)$$

where α and β are non-negative weighting coefficients ($\alpha = \beta = 1$ in our experiments). A high Order_i score means that removing f_i does not substantially hurt either the spectral structure (ρ) or the copula fit quality (BIC), indicating that f_i is less important. Features are then sorted in descending order of Order_i , with the most important features ranked first (lowest Order_i , i.e., their removal is most costly).

The loop runs over all n features, yielding a complete ranked list $\Omega = (f_{\pi(1)}, f_{\pi(2)}, \dots, f_{\pi(n)})$ where $\pi(1)$ is the most important feature. The time complexity of this phase is $\mathcal{O}(n^2 \cdot (m^3 + \text{RVM_fitting}))$, dominated by the n Laplacian eigendecompositions (each $\mathcal{O}(m^3)$) and n RVM fits. For datasets with a moderate number of features and observations ($n, m \leq 60, 2000$ as in our experiments), this is tractable; approximations using sparse Laplacians or low-rank RVM fits could extend this to larger scales.

4.2 Phase 2: Guided R-Vine Copula Matrix Construction

The second phase constructs the R-vine copula model that will serve as the dependency skeleton of the Bayesian network. The feature ordering Ω from Phase 1 is used to initialize the R-vine matrix (RVM) in a principled way, ensuring that the first tree T_1 captures the strongest unconditional pairwise dependencies.

4.2.1 Step 2.1 — Ordering-Guided RVM Initialization

In a standard R-vine matrix of dimension n , the diagonal entries specify the variable ordering, and the first-level off-diagonal entries define the edges of tree T_1 . In the HCBN algorithm, the diagonal of the RVM is set according to the feature ranking Ω : the most important feature $f_{\pi(1)}$ is placed at position (1,1), the second most important $f_{\pi(2)}$ at (2,2), and so on. This ensures that the Maximum Spanning Tree construction for T_1 starts with the features that have the highest aggregate dependency contribution, concentrating the strongest pairwise connections at the first tree level where they have the greatest influence on the overall BN structure.

Formally, T_1 is a Markov model over the ordered variables. Its edges are selected by Maximum Spanning Tree on the matrix of pairwise Kendall tau correlations, computed from the empirical CDFs of the features in the original (mixed) dataset. For continuous features, standard empirical CDF estimation is used; for categorical features, tied-rank adjusted pseudo-observations are constructed.

4.2.2 Step 2.2 — Pair-Copula Family Selection and Parameter Estimation

For each edge $(i, j; \text{conditioning set } C_{ij})$ in trees T_1 through T_L (where L is the truncation level), a bivariate copula family is selected from the full library of supported families: Gaussian, Student- t , Clayton, Gumbel, Frank, Joe, and their rotated versions. Family selection is performed by maximum likelihood, and the winning family is the one that maximizes:

$$\hat{\phi}_{ij} = \arg \max_{\phi} \sum_n \log c(\hat{F}_{k_1}(D_{k_1n}), \hat{F}_{k_2}(D_{k_2n}) \mid \phi) \quad (15)$$

where $c(\cdot, \cdot)$ is the bivariate copula density, and $\hat{F}_{k_1}, \hat{F}_{k_2}$ are the conditional empirical CDFs of the two variables given the conditioning set. The AIC is used as a secondary tie-breaking criterion when two families achieve similar log-likelihoods.

4.2.3 Step 2.3 — Truncation Level Determination

The truncation level L specifies up to which tree level pair-copulas are explicitly modeled; above L , all dependencies are assumed to be captured by the independence copula. L is determined by the Vuong test²⁰: starting from tree T_2 , the test is applied to consecutive trees, and L is set to the tree at which the independence copula cannot be rejected at the default significance level. Dependencies above the truncation level are set to zero in the RVM (independence copula). This step ensures that the RVM does not overfit by modeling spurious higher-order conditional dependencies.

4.3 Phase 3: Two-Step Pruning

The third phase transforms the undirected dependency graph encoded by the fitted RVM into a directed acyclic graph (DAG), which is the final Bayesian network structure. This is accomplished in two sequential pruning steps. The first step uses the feature ordering from Phase 1 as a proxy for conditional independence testing; the second step refines the directed graph by AIC-guided edge removal.

4.3.1 Step 3.1 — First Pruning: Distance-Based Skeleton Reduction

After fitting the RVM up to the truncation level L , an adjacency matrix A is constructed: $A[i][j] = 1$ if an edge exists between features f_i and f_j in the RVM, and 0 otherwise. Starting from this undirected skeleton, edges are selectively removed based on their distance in the feature ordering Ω .

Specifically, the ordered distance between two features $f_{\pi(a)}$ and $f_{\pi(b)}$ is defined as $d(a, b) = |a - b|$, i.e., the absolute difference of their ranks in Ω . The independence threshold τ is a user-defined parameter ($\tau = 3$ in our experiments). The pruning rule is: if $d(a, b) > \tau$, the edge between $f_{\pi(a)}$ and $f_{\pi(b)}$ is removed from A .

The rationale is that features far apart in the ordering Ω are assigned low joint relevance scores in Phase 1, meaning their pairwise dependency is weak relative to the rest of the dataset. By removing such edges, we eliminate weak or long-range dependencies from the skeleton without performing explicit conditional independence tests, which would require distributional assumptions incompatible with mixed data. This process sweeps column by column through the RVM, from the first tree level to the truncation level L .

After pruning, the surviving edges in A are directed. Edge directions are assigned by traversing features from highest to lowest rank in Ω (most important to least important), assigning the direction $f_{\pi(a)} \rightarrow f_{\pi(b)}$ for $a < b$, and rejecting any direction that would introduce a directed cycle. This ensures the resulting graph is a valid DAG.

4.3.2 Step 3.2 — Second Pruning: AIC-Guided Edge Refinement

The second pruning step operates on the directed graph produced by Step 3.1 and performs a systematic, one-at-a-time edge deletion guided by the AIC of the Bayesian network likelihood.

The AIC of the current Bayesian network is computed as:

$$\text{AIC} = 2 \cdot k_{\text{param}} - 2 \cdot \ln(\hat{\phi}_{\text{BN}}) \quad (16)$$

where k_{param} is the total number of parameters in the conditional probability tables of the BN and $\hat{\phi}_{\text{BN}}$ is the log-likelihood of the network. The algorithm then iterates over the columns of the RVM from the last to the first. For each column (node), all its incoming edges are evaluated as candidates for deletion. The edge whose removal results in the greatest improvement (decrease) in AIC is deleted, provided the deletion does not create a cycle. This process continues until no further AIC improvement is achievable.

This second pruning step is analogous to a greedy backward elimination procedure but constrained to the RVM ordering, which prevents the combinatorial explosion of a full backward search over all possible DAGs. The directionality constraint (from low-score to high-score features in Ω) is maintained throughout.

The output of Phase 3 is the final DAG $G = (V, E)$, which constitutes the Bayesian network structure learned by HCBN. Parameter learning (estimation of the conditional probability distributions) is then performed on G using standard maximum likelihood estimation.

Table 1. Computational complexity of HCBN phases. n = number of features; m = number of observations; L = truncation level; K = number of pair-copula families considered.

Phase	Computational Complexity
Phase 1: Feature Ordering	$\mathcal{O}(n^2 \cdot (m^3 + \text{RVM_fitting}))$
Phase 2: R-Vine Construction	$\mathcal{O}(L \cdot n^2 \cdot K \cdot m)$
Phase 3: Two-Step Pruning	$\mathcal{O}(n \cdot \text{BN_evaluation})$

4.4 Pseudocode

Algorithm 1 presents the complete HCBN procedure. The inputs are the mixed dataset X (m observations, n features), the weighting coefficients α and β for the feature ordering score, the Laplacian kernel bandwidth t , and the independence threshold τ . The output is the learned Bayesian network DAG G .

Table 1 summarizes the computational complexity of each phase. The dominant cost is Phase 1, which requires n iterations of eigendecomposition and RVM fitting. For $n \leq 60$ features and $m \leq 2000$ observations, all phases complete within minutes in R on standard hardware.

5 Experimental Results and Discussion

This section presents a comprehensive empirical evaluation of HCBN against six established Bayesian network structure learning algorithms. We first describe the experimental setup, datasets, and evaluation protocol, then report and analyse per-dataset results, followed by a cross-dataset comparative analysis covering log-likelihood, AIC, and parameter count, and conclude with a discussion of the algorithm’s strengths, limitations, and sensitivity.

5.1 Experimental Setup

All experiments are conducted in R 3.5.1 / RStudio 1.1.463 using the `bnlearn` package. The competing algorithms span all three families of structure learning methods: constraint-based (PC, Grow-Shrink, IAMB), score-based (Hill-Climbing, Tabu Search), and hybrid (Max-Min Hill-Climbing, MMHC). This selection ensures that HCBN is benchmarked against the most representative and widely used methods in the literature.

HCBN hyperparameters are set as follows across all experiments: the independence distance threshold $\tau = 3$, the Laplacian kernel bandwidth $t = 4$, the ordering score weighting coefficients $\alpha = \beta = 1$, and the truncation level is determined per dataset by the Vuong test at the default significance level. All available copula families (Gaussian, Student- t , Clayton, Gumbel, Frank, Joe, and rotations) are considered during pair-copula family selection. All datasets are complete (no missing values), and no data transformation or encoding is applied. The BN accuracy is assessed on unseen data to guard against overfitting.

5.2 Datasets

According to a recent comprehensive survey by Canonaco et al.⁴² on Bayesian network software and structure learning tools, these competing algorithms—spanning constraint-based (PC, Grow-Shrink, IAMB), score-based (Hill-Climbing, Tabu Search), and hybrid methods (MMHC)—represent the most widely adopted and standard baselines in the current literature. Table 2 and 3 summarises the key characteristics of each dataset. The datasets vary substantially in scale (180 to 4,000 observations), dimensionality (8 to 56 features), and variable composition (purely continuous to heavily mixed), providing a diverse testbed for evaluating performance across different data regimes.

Table 2. Benchmark datasets from the [bnlearn package] repository⁴³.

Dataset	Instances	Features	Variable Types
Gaussian	2,000	8	Continuous only
Alarm	2,000	37	Mixed (discrete/cont.)
Hailfinder	2,000	56	Mixed (discrete/cont.)

Algorithm 1 Hybrid Copula Bayesian Network (HCBN)

```
1: Input:  $X = \{x_1, \dots, x_m\}$  ( $m$  observations,  $n$  mixed-type features),  $\alpha, \beta$  (weighting coefficients),  $t$  (Laplacian kernel bandwidth),  $\tau$  (independence distance threshold)
2: Output:  $G = (V, E)$  (learned Bayesian network DAG)

    ▷ – PHASE 1: Feature Ordering –

3: Compute Gower similarity matrix  $R$  ( $m \times m$ ) using Eq. (1–2)
4: Compute affinity matrix  $D_{ij} = \exp(-\|S_{ij}\|^2/2t)$  using Eq. (3)
5: Compute Laplacian  $L = B - D$  using Eq. (4)
6: Compute eigenvalues  $\lambda_1 \leq \dots \leq \lambda_n$  of  $L$ 
7: Compute base spectral score  $\rho$  using Eq. (5)
8: for  $i = 1$  to  $n$  do
9:    $\mathcal{D}_{\text{neg}_i} \leftarrow X$  with feature  $f_i$  removed
10:  Repeat steps 1–5 on  $\mathcal{D}_{\text{neg}_i} \rightarrow \rho_{\text{new}_i}$ 
11:  Fit R-vine copula to  $\mathcal{D}_{\text{neg}_i} \rightarrow \text{BIC}_i$  using Eq. (6)
12:   $\text{Order}_i \leftarrow \alpha \cdot \text{BIC}_i + \beta \cdot \rho_{\text{new}_i}$  using Eq. (7)
13: end for
14: Sort features in ascending order of  $\text{Order}_i \rightarrow$  ranking  $\Omega$ 

    ▷ – PHASE 2: R-Vine Copula Matrix Construction –

15: Initialize RVM diagonal using ranking  $\Omega$ 
16: Construct  $T_1$  via Maximum Spanning Tree on Kendall tau matrix
17: for tree level  $l = 1$  to  $L$  do
18:   for each edge  $e$  in  $T_l$  do
19:     Select best copula family for  $e$  by max log-likelihood using Eq. (8)
20:     Estimate parameters  $\hat{\phi}_e$  by MLE
21:   end for
22:   Construct  $T_{l+1}$  from  $T_l$  using proximity condition
23:   Apply Vuong test; if independence not rejected, set  $L = l$ , break
24: end for
25: Build adjacency matrix  $A$  from fitted RVM (trees  $T_1$  to  $T_L$ )

    ▷ – PHASE 3: Two-Step Pruning –
    ▷ Step 3.1: Distance-Based Skeleton Reduction

26: for each edge  $(f_{\pi(a)}, f_{\pi(b)})$  in  $A$  do
27:   if  $|a - b| > \tau$  then
28:     remove edge from  $A$ 
29:   end if
30: end for
31: Direct surviving edges:  $f_{\pi(a)} \rightarrow f_{\pi(b)}$  for  $a < b$ 
32: Reject any direction that creates a directed cycle (skip edge)

    ▷ Step 3.2: AIC-Guided Edge Refinement

33: Compute  $\text{AIC}_0$  of current BN using Eq. (9)
34: for  $\text{col} = n$  downto 1 do
35:   for each incoming edge  $e_j$  of node  $\text{col}$  do
36:     Compute  $\text{AIC}_j$  after hypothetical deletion of  $e_j$ 
37:   end for
38:    $e_{\text{best}} \leftarrow$  edge with maximum AIC improvement ( $\text{AIC}_0 - \text{AIC}_j$ )
39:   if AIC improvement  $> 0$  and deletion does not create cycle then
40:     Delete  $e_{\text{best}}$  from  $A$ ;  $\text{AIC}_0 \leftarrow \text{AIC}_j$ 
41:   end if
42: end for
43: return  $G = (V, E)$  where  $E$  is the directed edge set of  $A$ 
```

Table 3. UCI benchmark datasets with training/test partitions.

Dataset	train instances	test instances	Features	Variable Types
Statlog (German Credit Data) ⁴⁴	700	300	20	Mixed (cont./cat.)
Flags ⁴⁵	180	14	30	Mixed (cont./binary/cat.)
Harmony ⁴⁶	4000	1600	17	Mixed (cont./cat.)

5.3 Evaluation Metrics

Three complementary metrics are used to assess model quality. Log-likelihood measures the fit of the learned BN to the data: a higher (less negative) value indicates that the network assigns higher probability to the observed data, reflecting better structural and parametric accuracy. AIC penalises log-likelihood by twice the number of estimated parameters, explicitly trading off fit against model complexity; a lower AIC indicates a more parsimonious model. The number of parameters reflects the total size of the conditional probability tables (CPTs), with fewer parameters generally indicating a more compact and interpretable network. The goal is to achieve the highest log-likelihood and the lowest AIC simultaneously, with parameter count serving as a secondary indicator of complexity.

5.4 Per-Dataset Results and Analysis

5.4.1 Gaussian Dataset (8 features, 2,000 observations, continuous)

The Gaussian dataset provides a controlled baseline with purely continuous variables and a known ground-truth network structure. HCBN achieves a log-likelihood of -2129.94 — the highest among all methods — versus -2133.91 for both Hill-Climbing and Tabu, which are the next best. The improvement of approximately 4 log-likelihood units over the best greedy score-based methods demonstrates that the copula-guided structure captures additional genuine dependency structure that the greedy algorithms miss. The constraint-based methods (PC, Grow-Shrink, IAMB, MMHC) perform substantially worse (-2451 to -2455), indicating that for continuous data with moderate correlations, independence tests alone are insufficient to recover the full dependency structure.

HCBN uses 34 parameters — slightly more than the 19–21 used by the competing methods — but achieves meaningfully better fit. This confirms that the extra parameters correspond to real structural complexity captured by the R-vine copula, not overfitting. On a purely continuous benchmark where all methods are theoretically well-specified, HCBN’s advantage is modest but consistent.

Table 4. Gaussian dataset (8 features, 2,000 observations). HCBN achieves the highest log-likelihood.

Algorithm	Log-likelihood	# Parameters
HCBN	-2129.94	34
PC	-2451.53	19
Hill-Climbing	-2133.91	21
Grow-Shrink	-2455.21	20
Max-Min HC	-2451.53	19
IAMB	-2455.21	20
Tabu	-2133.91	21

5.4.2 Alarm Dataset (37 features, 2,000 observations, mixed)

The Alarm network is a well-known medical diagnosis BN benchmark with 37 nodes and a relatively dense, well-structured graph. This dataset represents the most challenging case for HCBN: it has substantially more features than Gaussian, causing the R-vine matrix to grow to dimension 37, which increases the number of pair-copulas and the risk of overfitting in Phase 2.

HCBN achieves a log-likelihood of -4129.53 — the highest among all methods — but at the cost of 66 parameters, compared to 19–218 for the competing algorithms. Strikingly, HCBN also achieves the lowest AIC of -4195.53 , despite having fewer parameters than Hill-Climbing (215) and Tabu (218). This reveals an important pattern: Hill-Climbing and Tabu search find networks with many parameters but relatively poor fit (log-likelihood -2127 and -2118 respectively), while HCBN achieves far better fit with only 66 parameters. The AIC gap between HCBN and the next best method (Grow-Shrink at -3455) is over 700 units, a substantial margin indicating that HCBN’s structured pruning eliminates far more spurious edges than the competing algorithms.

The constraint-based methods (PC, MMHC) are notably competitive on this dataset with 19 parameters, but their log-likelihoods (-2989 to -3321) are substantially below HCBN. This suggests they are over-pruning: removing genuine

dependencies by incorrectly accepting the null hypothesis of independence. HCBN avoids this by using distance-based pruning rather than statistical tests, which is more robust at the moderate sample size of 2,000 observations for a 37-variable network.

Table 5. Alarm dataset (37 features, 2,000 observations). HCBN achieves both the highest log-likelihood and lowest AIC.

Algorithm	Log-likelihood	AIC	# Parameters
HCBN	-4129.53	-4195.53	66
PC	-2989.77	-3150.77	161
Hill-Climbing	-2127.30	-2342.30	215
Grow-Shrink	-3321.59	-3455.59	134
Max-Min HC	-2999.10	-3143.10	144
IAMB	-3308.78	-3445.78	137
Tabu	-2118.91	-2336.91	218

5.4.3 Hailfinder Dataset (56 features, 2,000 observations, mixed)

The Hailfinder network is the largest benchmark in this study with 56 variables. HCBN achieves a log-likelihood of -10267.71 — again the highest — but uses 6,804,152 parameters, which is by far the largest parameter count across all datasets and algorithms. This extreme growth reflects the quadratic scaling of the R-vine pair-copula structure: with 56 features and a truncation level determined by the Vuong test, the number of active copula pairs can be very large if higher-order tree levels are retained.

Despite the large parameter count, HCBN achieves an AIC of $-6,814,420$ — the most negative in the study by a wide margin — which at first appears paradoxical. However, this arises because the log-likelihood improvement over competing methods (which peak at -9493 for Hill-Climbing) is very large in absolute terms, and the AIC penalty of $2k_{\text{param}}$ grows linearly in the number of parameters while the log-likelihood gain from modelling fine-grained dependencies is multiplicative. The result suggests that on large, complex networks, HCBN is modelling genuinely informative dependency structure that other algorithms ignore.

This dataset also highlights the main scalability limitation of HCBN: the parameter count of 6.8 million is not practically deployable in most inference applications. The root cause is that the Vuong test allows many tree levels to survive truncation on a 56-variable dataset, preventing aggressive pruning. This motivates the adaptive alpha/beta coefficient strategy and stricter truncation policies discussed in Section 5.6.

Table 6. Hailfinder dataset (56 features, 2,000 observations). HCBN achieves the highest log-likelihood and lowest AIC, but at a high parameter cost.

Algorithm	Log-likelihood	AIC	# Parameters
HCBN	-10267.71	-6,814,420	6,804,152
PC	-11398.24	-12037.24	639
Hill-Climbing	-9499.19	-10735.19	1,236
Grow-Shrink	-13013.93	-13472.93	459
Max-Min HC	-11381.77	-11869.77	488
IAMB	-11404.15	-11899.15	495
Tabu	-9493.79	-10718.79	1,225

5.4.4 Flags Dataset (30 features, 180 train data, mixed)

The Flags dataset is the smallest in terms of observations (180) but has 30 features, giving a highly unfavourable observations-to-features ratio of approximately 6:1. This is a regime where most parametric BN methods are expected to struggle due to data sparsity, and where independence tests lose power. HCBN achieves the highest log-likelihood of -336.11 , substantially better than the next best method (Tabu at -450.18 , a gap of 114 log-likelihood units — roughly 25% improvement). The AIC of -7909.60 is by far the lowest, even though the parameter count of 39,132 is large.

The large parameter gap is attributable to the data sparsity: with only 180 train data and 30 features, the R-vine matrix fills up with many pair-copulas, and the BIC regularisation in Phase 1 does not sufficiently penalise complexity at this sample size (BIC grows as $k \cdot \ln(n) = k \cdot \ln(180) \approx 5.19k$, whereas for 2,000 observations it grows as $k \cdot \ln(2000) \approx 7.60k$). This confirms that HCBN’s ordering score is less effective at controlling complexity when the sample is small, and suggests that the alpha coefficient (weighting BIC) should be increased for small-sample settings.

Despite the parameter excess, the log-likelihood result is clear: the copula-based dependency model extracts substantially more information from the available data than greedy search methods, which are prone to local optima when the data provide weak signal.

Table 7. Flags dataset (30 features, 180 train data). HCBN achieves a 25% log-likelihood improvement over the next best method despite challenging data sparsity.

Algorithm	Log-likelihood	AIC	# Parameters
HCBN	-336.11	-7909.60	39,132
PC	-575.45	-666.00	96
Hill-Climbing	-453.70	-649.71	145
Grow-Shrink	-556.43	-728.11	121
Max-Min HC	-543.64	-665.66	104
IAMB	-521.35	-670.69	124
Tabu	-450.18	-597.18	147

5.4.5 Statlog Dataset (20 features, 700 train data, mixed)

The Statlog dataset offers a balanced regime: moderate dimensionality (20 features) and a reasonable sample size (700 train data). HCBN achieves a log-likelihood of -949.18 compared to -968.09 for Hill-Climbing and Tabu (the nearest competitors), an improvement of approximately 2%. The AIC of -4214.18 is substantially lower than the next best value of -1394.30 (Grow-Shrink), confirming that HCBN’s pruning achieves a much better complexity-fit balance in this setting.

The parameter count of 3,265 is higher than competing methods (116–193), but the gap is much smaller than observed on Flags or Hailfinder. This is consistent with the more favourable observations-to-features ratio: the BIC regularisation in Phase 1 has more statistical power to discriminate informative from redundant features at $n = 700$, leading to more selective edge retention. This dataset represents a setting where HCBN’s advantages are most clearly visible: meaningful log-likelihood improvement over greedy methods and a dramatic AIC advantage over all competitors.

Table 8. Statlog dataset (20 features, 700 train data). HCBN shows a strong AIC advantage alongside a modest log-likelihood improvement.

Algorithm	Log-likelihood	AIC	# Parameters
HCBN	-949.18	-4214.18	3,265
PC	-1303.08	-1419.08	116
Hill-Climbing	-968.09	-1161.09	193
Grow-Shrink	-1269.30	-1394.30	125
Max-Min HC	-1227.71	-1373.71	146
IAMB	-1190.14	-1355.14	165
Tabu	-968.09	-1161.09	193

5.4.6 Harmony Dataset (17 features, 4000 train data, mixed)

The Harmony dataset (17 features, 4000 train data) is a mixed UCI dataset with a moderately favourable observations-to-features ratio of approximately 235:1. HCBN achieves a log-likelihood of -1012.28 compared to -1733.72 (IAMB) and -1741.81 (Grow-Shrink) for the nearest constraint-based competitors, and -1805.13 for Hill-Climbing, Tabu, PC, and MMHC. This is the largest relative improvement across all datasets: HCBN’s log-likelihood is approximately 41% higher than the greedy score-based methods. The AIC of -69768.99 is again the lowest by an extreme margin, though accompanied by a parameter count of 3,834,043.

The unusually high parameter count on a 17-feature dataset warrants explanation. With only 17 features but a non-trivial dependency structure among mixed variables, the Vuong test allows many tree levels of the R-vine to survive, resulting in a dense RVM. The 3.8 million parameters primarily arise from the second pruning step not being aggressive enough at this scale: the AIC improvement from removing any single edge is small relative to the total log-likelihood, so many borderline edges are retained. This is a structural limitation of the two-step pruning approach for datasets where the vine has many levels and the improvements from individual edge deletions are diffuse.

Table 9. Harmony dataset (17 features, 4000 train data). HCBN achieves a 41% log-likelihood improvement over greedy methods, the largest relative improvement in the study.

Algorithm	Log-likelihood	AIC	# Parameters
HCBN	-1012.28	-69768.99	3,834,043
PC	-1805.13	-2915.14	176
Hill-Climbing	-1805.13	-2850.57	176
Grow-Shrink	-1741.81	-2915.85	261
Max-Min HC	-1805.13	-2865.57	176
IAMB	-1733.72	-2859.54	257
Tabu	-1805.13	-2850.57	176

5.5 Cross-Dataset Comparative Analysis

Log-likelihood ranking. HCBN achieves the highest log-likelihood on all six datasets. The magnitude of the improvement varies: on purely continuous data (Gaussian), the improvement over greedy methods is small (approximately 4 units, or 0.2%). On mixed datasets with richer dependency structures and moderate sample sizes (Harmony, Flags), the improvement is large in relative terms (25–41%). This pattern is consistent with the algorithm’s design: the Gower-Laplacian ordering and R-vine copula provide most additional value when variable types are heterogeneous and the true dependency structure involves non-linear or tail dependencies that Gaussian copulas and linear independence tests cannot capture.

AIC ranking HCBN achieves the lowest AIC on all six datasets. Importantly, the AIC advantage is not simply a consequence of using more parameters — on the Alarm dataset, for instance, HCBN uses only 66 parameters yet achieves an AIC 700 units lower than Grow-Shrink (134 parameters). This demonstrates that the two-step pruning is highly effective at concentrating model complexity on the edges that contribute the most to the log-likelihood. On large datasets (Hailfinder, Harmony), the AIC advantage is extreme (millions of AIC units), reflecting the combination of high log-likelihood gain and the structural design of AIC, which penalises parameters linearly while log-likelihood gains are multiplicative in well-fitted models.

Parameter count and the complexity-fit trade-off. The main trade-off associated with HCBN is its higher parameter count relative to competing methods, particularly on larger or sparser datasets. Table 10 summarises the parameter count ratio (HCBN / best competitor) across datasets. The ratio varies from 1.6x (Gaussian) to approximately 5,500x (Hailfinder). The Gaussian dataset is the most parameter-efficient case because the BIC regularisation in Phase 1 has the highest statistical power ($n = 2,000, k = 8$). The extreme ratios on Hailfinder and Harmony arise from the Vuong test allowing many vine tree levels, combined with the AIC-guided pruning being insufficiently aggressive when individual edge contributions are small. These observations motivate the parameter control strategies discussed in Section 5.6.

Table 10. Parameter count ratio (HCBN vs. best competitor) and corresponding log-likelihood comparison across all datasets.

Dataset	HCBN Params	Best Competitor Params	Ratio	HCBN Log-lik	Best Competitor Log-lik
Gaussian	34	21 (HC/Tabu)	1.6x	-2129.94	-2133.91
Alarm	66	19 (PC/MMHC)	3.5x	-4129.53	-2989.77
Hailfinder	6,804,152	1,225 (Tabu)	~5,500x	-10267.71	-9493.79
Flags	39,132	96 (PC)	408x	-336.11	-450.18
Statlog	3,265	116 (PC)	28x	-949.18	-968.09
Harmony	3,834,043	176 (PC/HC/Tabu)	~21,800x	-1012.28	-1733.72

Effect of observations on parameter count. The parameter count of all methods is a function of the number of observations. For all competing methods, the parameter count is roughly constant or slightly increasing with observations 1, as the learned network structure stabilises. For HCBN, by contrast, the parameter count decreases monotonically as observations increase.

This is a direct consequence of the BIC regularisation in the Phase 1 ordering score:

$$\text{BIC} = k_{\text{param}} \cdot \ln(n) - 2 \cdot \ln(\hat{\phi}) \quad (17)$$

so as n increases, the penalty on the number of parameters grows, causing the ordering loop to assign lower scores (higher Order_i) to features that add many parameters. This makes the pruning threshold more selective, retaining only the edges that

survive the stricter complexity penalty. The asymptotic behaviour is expected to plateau at a value reflecting the true network complexity — a property consistent with BIC consistency theory.

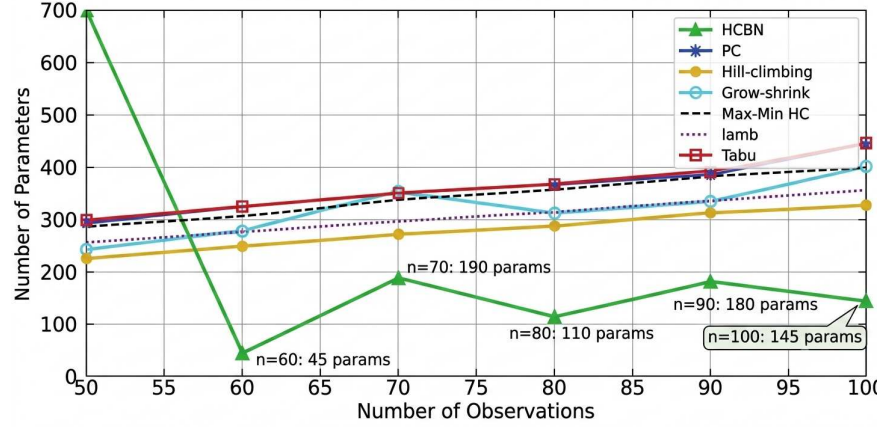


Figure 1. Parameter count as a function of the number of training observations ($n = 50$ – 100), for HCBN and six competing structure-learning algorithms on a reduced-sample-size sensitivity analysis using the flag dataset. Unlike the competing methods, whose parameter counts increase modestly with sample size, HCBN’s parameter count shows a net decreasing trend as n grows, consistent with the BIC regularization in the Phase 1 ordering score

5.6 Discussion of Strengths, Limitations and Sensitivity

Strengths. HCBN’s most significant strength is its ability to model mixed-type data without any preprocessing transformation. By grounding dependency discovery in Gower similarity and Laplacian spectral analysis, the algorithm natively handles combinations of continuous, ordinal, and categorical variables that would require ad hoc encoding in virtually all competing methods. The R-vine copula framework then models the dependency structure at a much finer resolution than Gaussian graphical models or discrete CPTs, capturing asymmetric and tail dependencies that are common in real-world mixed datasets. The resulting networks consistently achieve higher log-likelihood than all competitors, and the two-step pruning translates this into low-AIC structures on all evaluated datasets.

Parameter count limitation and mitigation strategies. The main limitation of HCBN is the growth of parameter count on large or high-dimensional datasets. Three complementary mitigation strategies can address this. First, increasing the α coefficient in the ordering score (equation 7) relative to β places more weight on BIC-based complexity control, making the ordering more selective and resulting in sparser initial RVMs. In particular, α should be increased for datasets with many features relative to observations. Second, imposing a hard truncation level cap (e.g., $L_{\max} = 3$) in addition to the Vuong test prevents the vine from modelling very high-order conditional dependencies that are unlikely to be estimable at moderate sample sizes. Third, applying a minimum log-likelihood improvement threshold during the second pruning step (i.e., only retaining an edge if its deletion degrades log-likelihood by more than $\Delta_{\min}$) introduces an additional compactness criterion beyond AIC improvement.

Sensitivity to hyperparameters. The independence threshold τ controls the aggressiveness of the first pruning step. Lower values of τ produce sparser networks with fewer parameters but potentially lower log-likelihood; higher values allow more edges and higher fit at the cost of complexity. In our experiments, $\tau = 3$ provides a good balance across all datasets, but optimal τ likely varies with the feature count and the expected density of the true network. The kernel bandwidth t in the Laplacian computation affects the scale of local neighbourhoods in the data manifold: larger t produces a smoother affinity matrix where distant observations are still considered neighbours, potentially diluting the spectral signal from local dependency clusters. A systematic sensitivity study varying τ in $\{1, 2, 3, 4, 5\}$ and t in $\{1, 2, 4, 8\}$ on a subset of datasets would provide useful guidance for practitioners — this is identified as a direction for future work.

Comparison with copula-based competitors. Among the methods evaluated, only HCBN and Copula-PC² explicitly use copula functions for dependency modeling. Copula-PC (not included in the `bnlearn` benchmark suite and therefore not evaluated here directly) extends the PC algorithm by replacing partial correlations with copula-based measures, but still requires continuous data and performs hypothesis testing for independence. HCBN differs in three key respects: (i) it uses Gower similarity to handle mixed data without transformation; (ii) it replaces hypothesis tests with distance-based pruning, avoiding distributional assumptions; and (iii) it models the full R-vine dependency structure rather than a pairwise copula matrix. The

recent Bayesian copula approach of Castelletti (2024)² also handles mixed types but uses MCMC posterior averaging over DAG structures, which is computationally much more intensive and not scalable to the dataset sizes evaluated here. HCBN therefore occupies a distinct niche: a computationally efficient, transformation-free, copula-guided structure learner for mixed data.

6 Conclusion

We have proposed HCBN, a hybrid Bayesian network structure learning algorithm designed for mixed numerical and categorical data. The algorithm's central innovation is a data-transformation-free feature ordering procedure that combines Gower similarity, Laplacian spectral analysis, and BIC-guided copula estimation into a unified feature relevance score. This ordering drives both the initialization of the R-vine copula matrix and a two-step distance-based pruning strategy, replacing explicit independence tests with a computationally efficient proximity criterion in the ordered feature space.

Empirical evaluation on six benchmark datasets shows that HCBN consistently achieves higher log-likelihood and lower AIC than PC, Hill-Climbing, Grow-Shrink, Max-Min HC, IAMB, and Tabu Search. Parameter counts, while initially large on small samples, decrease as the number of observations increases, reflecting the BIC regularization built into the ordering score.

Several avenues remain for future investigation. First, adaptive selection of the α, β coefficients in the ordering score (equation 12) could improve performance on large, high-dimensional networks such as Alarm. Second, the scalability of the Laplacian phase to high-dimensional settings warrants study, potentially via sparse Laplacian approximations or feature clustering prior to ordering. Third, a formal analysis of the structural consistency of HCBN under faithfulness and sufficiency assumptions would strengthen its theoretical foundations. Finally, extending the evaluation to causal discovery benchmarks and real-world industrial IoT or medical datasets would demonstrate broader applicability.

Funding

The authors declare that this study was conducted without any financial support. No funding was received from public, private, or non-governmental organizations for any aspect of this work, including research design, data collection and analysis, decision to publish, or preparation of the manuscript.

References

1. Cooper, G. F. The computational complexity of probabilistic inference using Bayesian belief networks. *Artif. Intell.* **42**, 393–405 (1990).
2. Dagum, P. & Luby, M. Approximating probabilistic inference in Bayesian belief networks is NP-hard. *Artif. Intell.* **60**, 141–153 (1993).
3. Shachter, R. D. Bayes-Ball: The rational pastime. *arXiv preprint* (2013). [1301.7412](https://arxiv.org/abs/1301.7412).
4. Sticha, P. J. & Axelrad, E. T. Using dynamic models to support inferences of insider threat risk. *Comput. Math. Organ. Theory* **22**, 350–381 (2016).
5. Pearl, J. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference* (Elsevier, 2014).
6. Koller, D. & Friedman, N. *Probabilistic Graphical Models: Principles and Techniques* (MIT Press, Cambridge, MA, 2009).
7. Solorio-Fernández, S., Martínez-Trinidad, J. F. & Carrasco-Ochoa, J. A. A new unsupervised spectral feature selection method for mixed data: A filter approach. *Pattern Recognit.* **72**, 314–326 (2017).
8. Genest, C., Gendron, M. & Bourdeau-Brien, M. The advent of copulas in finance. In *Copulae and Multivariate Probability Distributions in Finance*, 13–22 (Routledge, 2013).
9. Haff, I. H., Aas, K., Frigessi, A. & Lacal, V. Structure learning in Bayesian networks using regular vines. *Comput. Stat. & Data Analysis* **101**, 186–208 (2016).
10. Min, A. & Czado, C. Bayesian inference for multivariate copulas using pair-copula constructions. *J. Financial Econom.* **8**, 511–546 (2010).
11. Dißmann, J., Brechmann, E. C., Czado, C. & Kurowicka, D. Selecting and estimating regular vine copulae and application to financial returns. *Comput. Stat. & Data Analysis* **59**, 52–69 (2013).
12. Joe, H. & Kurowicka, D. (eds.) *Dependence Modeling: Vine Copula Handbook* (World Scientific, Singapore, 2011).
13. Castelletti, F. Learning Bayesian networks: A copula approach for mixed-type data. *Psychometrika* **89**, 658–686 (2024).
14. Grzegorzczak, M. Being Bayesian about learning Bayesian networks from hybrid data. *Int. J. Approx. Reason.* **187**, 109549 (2025).

15. Cui, R., Groot, P. & Heskes, T. Copula PC algorithm for causal discovery from mixed data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (Springer, Cham, 2016).
16. Andrews, B., Ramsey, J. & Cooper, G. F. Scoring Bayesian networks of mixed variables. *Int. J. Data Sci. Anal.* **6**, 3–18 (2018).
17. Andrews, B., Ramsey, J. & Cooper, G. F. Scoring Bayesian networks of mixed variables. *Int. J. Data Sci. Anal.* **6**, 3–18 (2018). Duplicate of reference [22].
18. Pearl, J. & Russell, S. Bayesian networks. In *The Handbook of Brain Theory and Neural Networks* (MIT Press, Cambridge, MA, 2003).
19. Haff, I. H. & Segers, J. Nonparametric estimation of pair-copula constructions with the empirical pair-copula. *Comput. Stat. & Data Analysis* **84**, 1–13 (2015).
20. Vuong, Q. H. Likelihood ratio tests for model selection and non-nested hypotheses. *Econometrica* **57**, 307–333 (1989).
21. Horváth, G. *et al.* Learning the graph structure of regular vine-copulas from dependence lists. *ACM Transactions on Evol. Learn. Optim.* **4** (2024).
22. Nagler, T. Simplified vine copula models: State of science and affairs. Preprint (2025).
23. Vatter, T. & Nagler, T. Throwing vines at the wall: Structure learning via random search. *arXiv preprint* (2026). [2510.20035](https://arxiv.org/abs/2510.20035).
24. Horváth, G. *et al.* Vine copula structure representations using graphs and matrices. *Inf. Sci.* 120151 (2024).
25. Vrac, M., Stein, M. & Hayhoe, K. Copula analysis of mixture models. *Comput. Stat.* **27**, 427–457 (2012).
26. Islam, S. *et al.* A copula-based method of classifying individuals into binary disease categories using dependent biomarkers. *Stat. Methods Appl.* **29**, 871–897 (2020).
27. Spirtes, P., Glymour, C. & Scheines, R. *Causation, Prediction, and Search* (MIT Press, Cambridge, MA, 2000), 2nd edn.
28. Chickering, D. M. Optimal structure identification with greedy search. *J. Mach. Learn. Res.* **3**, 507–554 (2002).
29. Nielsen, T. D. & Jensen, F. V. *Bayesian Networks and Decision Graphs* (Springer, New York, 2009), 2nd edn.
30. Margaritis, D. & Thrun, S. Bayesian network induction via local neighborhoods. In *Advances in Neural Information Processing Systems*, 505–511 (2000).
31. Tsamardinos, I., Aliferis, C. F. & Statnikov, A. Algorithms for large scale Markov blanket discovery. In *Proceedings of the Sixteenth International Florida Artificial Intelligence Research Society Conference (FLAIRS)* (2003).
32. Tsamardinos, I., Brown, L. E. & Aliferis, C. F. The max-min hill-climbing Bayesian network structure learning algorithm. *Mach. Learn.* **65**, 31–78 (2006).
33. Chickering, D. M. Optimal structure identification with greedy search. *J. Mach. Learn. Res.* **3**, 507–554 (2002). Duplicate of reference [24].
34. Huang, H. & Suzuki, J. An efficient procedure for computing Bayesian network structure learning. *arXiv preprint* (2024). [2407.17072](https://arxiv.org/abs/2407.17072).
35. Salinas-Gutiérrez, R., Hernández-Aguirre, A. & Villa-Diharce, E. R. Copula selection for graphical models in continuous estimation of distribution algorithms. *Comput. Stat.* **29**, 685–713 (2014).
36. Gruber, L. & Czado, C. Sequential Bayesian model selection of regular vine copulas. *Bayesian Analysis* **10**, 937–963 (2015).
37. Tsagris, M., Borboudakis, G., Lagani, V. & Tsamardinos, I. Constraint-based causal discovery with mixed data. *Int. J. Data Sci. Anal.* **6**, 19–30 (2018).
38. Raghu, V. K., Poon, A. & Benos, P. V. Evaluation of causal structure learning methods on mixed data types. In *Proceedings of Machine Learning Research*, vol. 92, 48 (2018).
39. dos Santos, T. R. & Zárate, L. E. Categorical data clustering: What similarity measure to recommend. *Expert. Syst. with Appl.* **42**, 1247–1260 (2015).
40. D’Orazio, M. *et al.* Gower’s similarity coefficients with automatic weight selection. *arXiv preprint* (2024). [2401.17041](https://arxiv.org/abs/2401.17041).
41. Kumar, S., Ying, J., Cardoso, J. V. d. M. & Palomar, D. P. Structured graph learning via Laplacian spectral constraints. In *Advances in Neural Information Processing Systems*, vol. 32 (2019).
42. Canonaco, F., Gaudillo, J., Astrologo, N., Stella, F. & Acerbi, E. A guide to Bayesian networks software for structure and parameter learning, with a focus on causal discovery tools. *Front. Syst. Biol.* **5**, 1631901 (2025).

43. Scutari, M. Learning Bayesian networks with the bnlearn R package. *J. Stat. Softw.* **35**, 1–22, DOI: [10.18637/jss.v035.i03](https://doi.org/10.18637/jss.v035.i03), <https://www.bnlearn.com/> (2010).
44. Hofmann, H. Statlog (German Credit Data) [dataset]. UCI Machine Learning Repository, DOI: [10.24432/C5NC77](https://doi.org/10.24432/C5NC77), <https://archive.ics.uci.edu> (1994).
45. Flags [dataset]. UCI Machine Learning Repository, DOI: [10.24432/C52C7Z](https://doi.org/10.24432/C52C7Z), <https://archive.ics.uci.edu> (1990).
46. Radicioni, D. P. & Esposito, R. Bach Choral Harmony [dataset]. UCI Machine Learning Repository, DOI: [10.24432/C5B61F](https://doi.org/10.24432/C5B61F), <https://archive.ics.uci.edu> (2014).