

Article

Bayesian Estimation of R-Vine Copula with Gaussian-Mixture GARCH Margins: An MCMC and Machine Learning Comparison

Rewat Khanthaporn ¹ and Nuttanan Wichitaksorn ^{2,*}

¹ Assurance Audit and Risk Group, Health New Zealand, Auckland 1023, New Zealand; ray.khanthaporn@tewhatuora.govt.nz

² Department of Mathematical Sciences, Auckland University of Technology, Auckland 1010, New Zealand

* Correspondence: nuttanan.wichitaksorn@aut.ac.nz

Abstract

This study proposes Bayesian estimation of multivariate regular vine (R-vine) copula models with generalized autoregressive conditional heteroskedasticity (GARCH) margins modeled by Gaussian-mixture distributions. The Bayesian estimation approach includes Markov chain Monte Carlo and variational Bayes with data augmentation. Although R-vines typically involve computationally intensive procedures limiting their practical use, we address this challenge through parallel computing techniques. To demonstrate our approach, we employ thirteen bivariate copula families within an R-vine pair-copula construction, applied to a large number of marginal distributions. The margins are modeled as exponential-type GARCH processes with intertemporal capital asset pricing specifications, using a mixture of Gaussian and generalized Pareto distributions. Results from an empirical study involving 100 financial returns confirm the effectiveness of our approach.

Keywords: regular vine copulas; variational bayes with data augmentation; exponential-type generalised autoregressive conditional heteroskedasticity model; intertemporal capital asset pricing model; mixture distribution; Markov chain Monte Carlo

MSC: 62E; 62F; 62H; 62J



Academic Editor: Quanxin Zhu

Received: 30 October 2025

Revised: 28 November 2025

Accepted: 1 December 2025

Published: 4 December 2025

Citation: Khanthaporn, R.; Wichitaksorn, N. Bayesian Estimation of R-Vine Copula with Gaussian-Mixture GARCH Margins: An MCMC and Machine Learning Comparison. *Mathematics* **2025**, *13*, 3886. <https://doi.org/10.3390/math13233886>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Over the past few decades, following the work of [1], who demonstrated that copula functions can capture non-linear comovement between variables, research on copula-based dependence structure models has grown rapidly. One of the key advantages of copulas, as multivariate distribution tools, is their ability to link different marginal distributions. As a result, copulas have been widely applied across various fields, including hydrology [2], energy [3], agricultural and forest meteorology [4], oceanology [5], computer science [6], insurance [7], and more widely in economics and finance [8,9].

A substantial body of literature demonstrates that copula models, particularly due to their ability to capture tail asymmetry, are highly effective for modeling dependence in quantitative risk management. Tail modeling plays a crucial role in risk assessment, where common measures include Value-at-Risk (VaR) and Conditional Value-at-Risk (CVaR), also known as expected shortfall. For formal definitions of risk measures, see [10]. Among multivariate dependence models, standard copulas are widely used. These typically fall into two main families: elliptical and Archimedean copulas (see [11–14] for details). However, standard multivariate copulas come with certain limitations—most notably,

they rely on a single parameter to govern all pairwise tail dependencies. This becomes problematic in high-dimensional settings, where such a simplification can be unrealistic. This limitation was identified by [15], who introduced a more flexible class of models known as regular vine (R-vine) copulas. Two notable subclasses within the vine copula framework are the canonical vine (C-vine) and the drawable vine (D-vine) copulas [16]. Further discussion on vine copulas can be found in Section 2, and additional details are provided in Chapter 3 of [17].

It consists of two copula families: one is elliptical copulas—see Wichitaksorn et al. [11], Liang and Rosen [13] for recent studies—and the other is Archimedean copulas; see Michaelides et al. [14] and Górecki et al. [12] for recent studies.

In the existing literature, research on vine copulas—particularly R-vines—remains relatively limited. This is likely due to their structural complexity and the intensive computational effort required. Specifically, R-vines allow for a significantly larger number of possible tree structures compared to C-vines and D-vines, especially as the dimensionality increases. Despite these challenges, R-vines offer considerable flexibility as dependence models. Their wide range of possible structures makes them highly adaptable, and importantly, they overcome the parameter limitation of standard multivariate copulas. This added flexibility is particularly advantageous for analyzing tail dependence in high-dimensional financial data.

To our knowledge, there are few studies in the financial literature on R-vines with high dimensions. For instance, ref. [18] studied up to 398 dimensions on S&P500 constituents and ref. [19] studied 2131 dimensions, while other R-vine studies have examined the lower dimensions (less than 25). See, among others, the studies of R-vines in lower dimensions in [20–25]. For studies on copula models, see, among others, [26–29]. Hence, the study of R-vines in high dimensions is still scarce and needs more investigation. This observation is the starting point of our study. The model most closely related to ours is that of [30], which studies R-vines in application to 16-dimensional financial indices using the quasi-maximum likelihood estimation (QMLE) method.

Given standardized residuals, vine copula models can be applied across five different classes: mixed R-vines, mixed C-vines, mixed D-vines, all- t R-vines, and the standard multivariate Gaussian copula. This study utilizes seven bivariate copula families. Based on the test proposed by [31] and likelihood comparisons, previous research has identified mixed R-vines as the best-performing models—an approach we adopt in this study.

Our first contribution is to extend this analysis to 100 dimensions of financial data using constituents of the S&P500, which, to our knowledge, has not been explored in this context. The second contribution is the expansion of the copula family set from seven to thirteen bivariate copulas, including two-parameter BB families and additional Archimedean copulas. This broader selection allows for a more accurate capture of tail asymmetries observed in financial data. Further details are provided in Section 4. Our third contribution lies in the modeling of the marginal distributions. We employ the intertemporal capital asset pricing model (ICAPM), originally proposed by [32], with an asymmetric GARCH model featuring a mixture innovation. This approach enhances the filtration of volatility clustering and yields deeper insights into risk premia across individual financial returns. Unlike [30], who used an ARMA-GARCH model with a single Student- t innovation, we incorporate a mixture of one Gaussian and two heavy-tailed distributions, offering a more flexible representation of return dynamics.

Another key contribution of this study lies in the estimation methodology. In the existing literature, quasi-maximum likelihood estimation (QMLE) is widely used for copula models; see, for example, [33–35]. However, QMLE has notable limitations—it is highly sensitive to initial parameter values and often suffers from convergence issues [36]. To

address these shortcomings, we employ more efficient estimation techniques, specifically Bayesian Markov chain Monte Carlo (MCMC) and variational Bayes (considered as machine learning) methods, to estimate our proposed R-vine dependence models. These methods are also compatible with the two main subclasses of vine copulas: C-vines and D-vines. While Bayesian MCMC and machine learning approaches are computationally intensive, we mitigate the associated costs by incorporating parallel computing. Additionally, we use QMLE estimates as initial values to improve convergence speed in the Bayesian MCMC and machine learning algorithms.

Our study explores these advanced methods in two distinct ways. First, we apply the random walk Metropolis–Hastings (rw-MH) algorithm for Bayesian MCMC estimation. Second, we implement the variational Bayes method—both with and without data augmentation (VBDA)—as proposed by [37], to estimate the multivariate dependence structure. Furthermore, we propose an alternative, more efficient variational approximation to reduce computational time, the details of which are presented in the simulation study in the Supplementary Material. For related surveys on MLE, Bayesian, and machine learning methods, see [38–41], among others.

The remainder of the paper is structured as follows. Section 2 introduces the graph-theoretical framework for high-dimensional regular vine (R-vine) copulas, including the two special subclasses—C-vines and D-vines—as well as the marginal models. Section 3 outlines the Bayesian inference and machine learning algorithms used to estimate general R-vine dependence models. Section 4 presents empirical results based on 100-dimensional financial data. Section 5 offers concluding remarks. Details on the simulation study and data generation algorithms are provided in the Supplementary Material.

2. Graph Theory

2.1. Regular Vines in n -by- n Matrix Representation

This section provides the theoretical background on regular vine (R-vine) copulas, along with two important subclasses: canonical vine (C-vine) and drawable vine (D-vine) copulas, all represented in an n -dimensional graphical model framework. In graph theory, an R-vine is represented as a sequence of tree structures composed of nodes N and edges E . An R-vine (\mathcal{V}) in n -dimensions is defined as a nested set of $n - 1$ trees. The first tree ($T = 1$) consists of n nodes and $n - 1$ edges, each representing a pair of variables. Each edge in $T = 1$ becomes a node in the second tree ($T = 2$), which then has $n - 1$ nodes and $n - 2$ edges. The construction follows the proximity condition, meaning that connected nodes in $T = 2$ must share a common node from $T = 1$.

This nesting continues: in the third tree ($T = 3$), there are $n - 2$ nodes and $n - 3$ edges, and again, any two connected nodes must share a common node from $T = 2$. This process continues until the final tree (T_{n-1}), which has two nodes and a single edge. These two nodes must share all remaining nodes from the previous tree. For further definitions and examples—including complete unions, conditioning sets, and conditional sets of an edge—refer to [30]. Additional theoretical details and properties of vine copulas can be found in [17,42–44].

A convenient approach for developing an algorithm in a vine graphical model is to store the trees in an n -by- n matrix. The authors of [30] proposed using a lower triangular matrix \mathcal{M} to represent vines as a tree structure in n dimensions; see Figure 1.

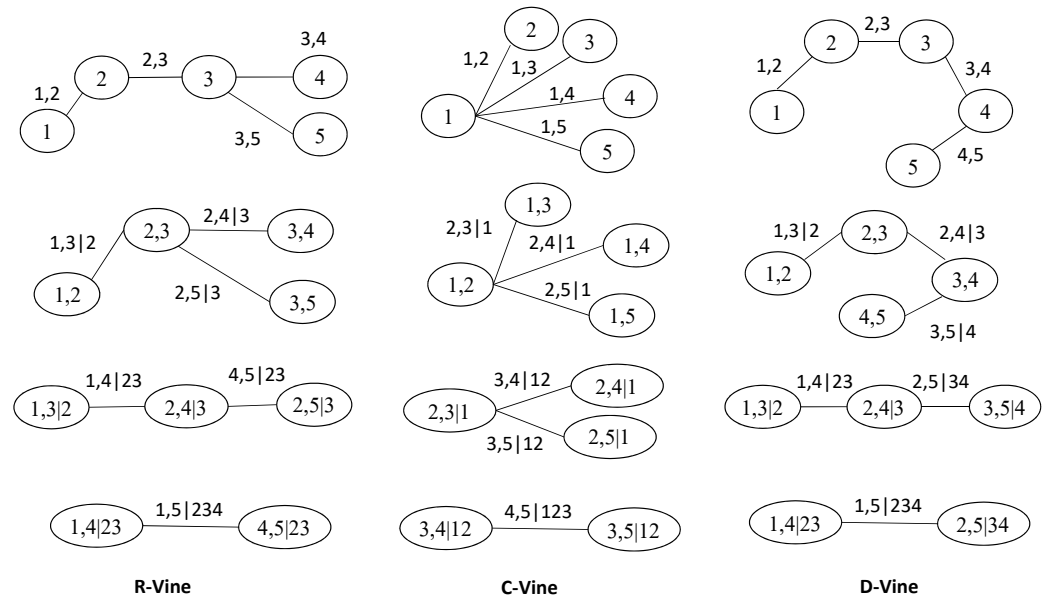


Figure 1. An illustrative example of vine copula graphical model in five dimensions.

Definition 1 (Matrix Constraint Set). Let $\mathcal{M} = (m_{i,k})$ be a lower triangular n -by- n matrix where $i, k = 1, \dots, n$. The i^{th} constraint set for \mathcal{M} is

$$C\mathcal{M}(i) = \{(\{m_{k,k}, m_{i,k}\}, \mathcal{D}) \mid i = k + 1, \dots, n, \mathcal{D} = \{m_{i+1,k}, \dots, m_{n,k}\}\}$$

for $i = n - 1, \dots, 1$. Otherwise, $\mathcal{D} = \emptyset$ (if $i = n$). Hence, \mathcal{D} is the conditioning set and $\{m_{k,k}, m_{i,k}\}$ is the conditioned set.

In other words, every pair variable within the constraint set is built up from a diagonal element $m_{k,k}$ plus another element in the same column $m_{i,k}$ conditioning on set of $\mathcal{D} = \{m_{i+1,k}, \dots, m_{n,k}\}$. Otherwise, $\mathcal{D} = \emptyset$ when i equal to n . To illustrate Definition 1, the 5-by-5 matrix \mathcal{M} of Figure 1 can be constructed as

$$\mathcal{M}^R = \begin{pmatrix} 1 & & & & \\ 5 & 5 & & & \\ 4 & 4 & 2 & & \\ 3 & 2 & 4 & 3 & \\ 2 & 3 & 3 & 4 & 4 \end{pmatrix}, \quad \mathcal{M}^C = \begin{pmatrix} 5 & & & & \\ 4 & 4 & & & \\ 3 & 3 & 3 & & \\ 2 & 2 & 2 & 2 & \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}, \quad \mathcal{M}^D = \begin{pmatrix} 5 & & & & \\ 1 & 4 & & & \\ 2 & 1 & 3 & & \\ 3 & 2 & 1 & 2 & \\ 4 & 3 & 2 & 1 & 1 \end{pmatrix}$$

Once a 5-by-5 matrix is constructed, it becomes straightforward to generalize the construction to an \mathcal{M} matrix for any dimension n . For different constructions of upper triangular matrices representing vines in graphical theory, see [17] (Chapter 3).

2.2. Multivariate Regular Vines Density

This section introduces the matrix \mathcal{M} within a numerical framework for computing multivariate regular vine copula densities, including both multivariate canonical vine (C-vine) and drawable vine (D-vine) structures in n -dimensions. In the previous section, we generated an n -by- n matrix \mathcal{M} satisfying a proximity condition. Building on this, the n -dimensional multivariate vine copula density is now formulated using two additional n -by- n matrices: the bivariate copula type matrix $\mathcal{T} = (t_{i,k})$ and its associated parameter matrix $\mathcal{P} = (p_{i,k})$ where $i = n, \dots, 2$ and $k = n - 1, \dots, 1$.

We begin by expressing the vine copula density as a product of pairwise copula functions. According to Sklar’s theorem, a copula C is a multivariate distribution function

that links a joint distribution to its marginals $F_1(x_1), \dots, F_n(x_n)$ [1]. Thus, the multivariate copula distribution can be written as:

$$F(x_1, \dots, x_n) = C\{F_1(x_1), \dots, F_n(x_n)\}$$

for n -dimensional data. After that, vine copula density in n -dimensions can be expressed in graph theory. In product indexing, the C-vine density is

$$\prod_{j=1}^{n-1} \prod_{i=1}^{n-j} c_{j,j+i|1,\dots,j-1} \left\{ F(x_j|x_1, \dots, x_{j-1}), F(x_{j+i}|x_1, \dots, x_{j-1}) \right\}. \tag{1}$$

The D-vine density is

$$\prod_{j=1}^{n-1} \prod_{i=1}^{n-j} c_{i,i+j|i+1,\dots,i+j-1} \left\{ F(x_i|x_{i+1}, \dots, x_{i+j-1}), F(x_{i+j}|x_{i+1}, \dots, x_{i+j-1}) \right\}. \tag{2}$$

The R-vine density is

$$\prod_{i=1}^{n-1} \prod_{e \in E_i} c_{C_{e,a}, C_{e,b}|D_e} (F_{C_{e,a}|D_e}(X_{C_{e,a}}|\mathbf{X}_{D_e}), F_{C_{e,b}|D_e}(X_{C_{e,b}}|\mathbf{X}_{D_e})). \tag{3}$$

In a D-vine, each node in a given tree T_i can be connected to only one edge. In contrast, within a C-vine, a specific node in tree T_i must have $n - j$ edges. The D-vine structure allows for more independence assumptions compared to the C-vine. However, the C-vine can be advantageous when there is prior knowledge of a particular variable that strongly influences the dataset. The R-vine represents the most general form, offering significantly greater flexibility in modeling complex dependency structures.

Figure 1 provides an example of a five-dimensional regular vine copula density. The complete R-vine copula density depicted in this figure can be expressed as follows:

$$f(x_1, x_2, x_3, x_4, x_5) = c_{1,2} \cdot c_{2,3} \cdot c_{3,4} \cdot c_{3,5} \text{ (unconditional pairs)} \\ \times c_{1,3|2} \cdot c_{2,4|3} \cdot c_{2,5|3} \cdot c_{1,4|23} \cdot c_{4,5|23} \cdot c_{1,5|234} \text{ (conditional pairs)}$$

where $c_{u,v|k}$ stands for $c_{u,v|k}(F_{u|k}(x_u|x_k), F_{v|k}(x_v|x_k))$. All definitions and further details can be found, among others, in [15] for C-vines and D-vines, and in [30] for R-vines. Next, we present R-vine density using matrix representation \mathcal{M}^* , \mathcal{T}^* , and \mathcal{P}^* in Figure 2, which can be used for its special cases.

$$\mathcal{M}^* = \begin{pmatrix} m_{1,1} & & & & \\ m_{2,1} & m_{2,2} & & & \\ \vdots & \vdots & \ddots & & \\ m_{n,1} & m_{n,2} & \dots & & m_{n,n} \end{pmatrix},$$

$$\mathcal{T}^* = \begin{pmatrix} t_{2,1} & & & & \\ t_{3,1} & t_{3,2} & & & \\ \vdots & \vdots & \ddots & & \\ t_{n,1} & t_{n,2} & \dots & t_{n,n-1} & \end{pmatrix}, \mathcal{P}^* = \begin{pmatrix} p_{2,1} & & & & \\ p_{3,1} & p_{3,2} & & & \\ \vdots & \vdots & \ddots & & \\ p_{n,1} & p_{n,2} & \dots & p_{n,n-1} & \end{pmatrix}$$

Figure 2. Vine copula specification in n -by- n matrix representation including \mathcal{T}^* , \mathcal{P}^* and \mathcal{M}^* .

In Figure 2, the red label corresponds to the last tree, $n - 1$. The conditioned set is $\{m_{1,1}, m_{2,1}\} \in \mathcal{M}^*$, and the conditioning set \mathcal{D} includes all elements $m_{i,1}$, $i = 3, \dots, n$. The associated bivariate copula type and parameter are $t_{2,1}$ and $p_{2,1}$, respectively.

The blue label represents the first tree, T_1 . Here, the conditioned set is $\{m_{2,2}, m_{n,2}\} \in \mathcal{M}^*$, and the conditioning set \mathcal{D} is \emptyset . The corresponding bivariate copula type and parameter are $t_{n,2}$ and $p_{n,2}$, respectively.

From a graph-theoretical perspective, the regular vine copula density can therefore be described as:

$$f_{1,\dots,n} = \prod_{k=1}^n f_k \prod_{k=n-1}^1 \prod_{i=n}^{k+1} c_{m_{k,k}, m_{i,k} | m^*} (F_{m_{k,k} | m^*}, F_{m_{i,k} | m^*}), \tag{4}$$

where m^* is $m_{i+1,k}, \dots, m_{n,k} \in \mathcal{M}^*$ with bivariate copula types \mathcal{T}^* , bivariate copula parameters \mathcal{P}^* , and the corresponding densities in Equations (1)–(3). For the R-vine density algorithm, see Algorithm 2.1 of [30].

2.3. Marginal Model and Distribution

This section presents our proposed marginal model, which combines elements of the ICAPM and an asymmetric GARCH model with a mixture distribution. This hybrid approach is designed to capture key stylized facts of financial returns, such as volatility clustering and time-varying heteroskedasticity. The model offers several advantages, including the incorporation of risk premium information and the ability to account for asymmetric shocks and fat-tailed behavior in financial data.

Our marginal model is similar to that of [45], who employed Gaussian innovations; however, we adopt a more flexible mixture innovation to better reflect the complexity of financial return distributions. Furthermore, we extend this framework to the multivariate setting in order to analyze dependence structures using regular vine copulas.

The ICAPM we use in this study follows [32], which is a linear-in-variance function of financial returns and is given by

$$y_t = \lambda_1 + \lambda_2 h_t + \varepsilon_t, \tag{5}$$

where y_t is the financial returns at time $t = \{1, \dots, T\}$, λ_1 is a risk-free rate, λ_2 is a risk premium, h_t is a conditional volatility, $\varepsilon_t = \sqrt{h_t} z_t$ is an innovation with zero mean and the conditional variance (or volatility) h_t , and z_t is the independent and identically distributed (iid) random variable with zero mean and unit variance from a mixture distribution. To deal with the heteroskedastic returns, we use an exponential-type GARCH model or EGARCH(p,q,r) given by

$$\log h_t = \omega + \sum_{i=1}^p \gamma_i z_{t-i} + \sum_{j=1}^q \delta_j |z_{t-j}| + \sum_{k=1}^r \nu_k \log h_{t-k}. \tag{6}$$

where ω is a constant, γ_i measures the direct innovation effects, δ_j is the asymmetry parameter for the leverage effects, and ν_k is the volatility persistence parameter. For details on the EGARCH model, see [46], as well as the surveys on asymmetric GARCH models, including EGARCH, in [47,48]. Regarding the distribution of innovations in the marginal model, we employ a mixture distribution originally proposed by [49] for use with the ARMA-GARCH model and standard copulas. In contrast, our study applies this mixture distribution within an ICAPM-EGARCH framework combined with regular vine copulas.

The mixture distribution consists of three components: one Gaussian distribution and two generalized Pareto distributions (GPDs) derived from extreme value theory (EVT). The

GPD components are specifically designed to capture asymmetric fat tails—one of the most significant features observed in financial return data [50].

The mixture distribution has a flexible property. It can be represented by either a finite set of distribution functions (cdf), $P_k(x_t)$, or density functions (pdf), $p_k(x_t)$, $k = 1, \dots, K$. The mixture distribution and its density are a weighted sum such that $F(x_t) = \sum_{k=1}^K w_k P_k(x_t)$ and $f(x_t) = \sum_{k=1}^K w_k p_k(x_t)$, respectively. $w > 0$ is a weight and $\sum w = 1$. If the mixing weight w is pre-specified, it can also represent the quantile level of the mixture distribution. Given the three components, the mixture pdf is given by

$$p(x_t) = \begin{cases} \frac{1}{\beta^L} \left(1 + \zeta^L \frac{(x_t - \mu^L)}{\beta^L} \right)^{-\frac{1}{\zeta^L} - 1} & , x_t \leq \Phi^{-1}(\alpha^L) \\ \phi(x_t) & , \Phi^{-1}(\alpha^L) < x_t < \Phi^{-1}(\alpha^R) \\ \frac{1}{\beta^R} \left(1 + \zeta^R \frac{(x_t - \mu^R)}{\beta^R} \right)^{-\frac{1}{\zeta^R} - 1} & , x_t \geq \Phi^{-1}(\alpha^R), \end{cases} \quad (7)$$

where α^L and α^R are a pre-specified quantile, $\mu = (-\infty, \infty)$ is a location parameter, $\beta = (0, \infty)$ is a scale parameter, $\zeta = (-\infty, \infty)$ is a shape parameter, and L and R denote the left and right tails, respectively. The support of $x_t \sim GPD(\mu, \beta, \zeta)$ is $x_t \geq 0$ when $\zeta = [0, \infty)$ and $x_t = [\mu, \mu - \beta/\zeta]$ when $\zeta = (-\infty, 0)$, and $\zeta = [0, 1/2]$ represents a fat tail and exists at least up to the second moment. For more details on the moment property, see [51,52].

The GPDs, in a sense of generalization, consist of an ordinary Pareto distribution, an exponential distribution, and a short-tailed, Pareto type II distribution. Further details of the GPD used in the peak-over-threshold method are found in ([10] Chapter 5). Note that $\phi(\cdot)$ and $\Phi^{-1}(\cdot)$ are the standard Gaussian pdf and its inverse cdf, respectively.

3. Bayesian Inference and Machine Learning in the Graphical Model

This section outlines the parameter estimation techniques used in our study, which include both Bayesian and machine learning approaches. For the Bayesian framework here, we refer to the Markov chain Monte Carlo (MCMC), while the variational Bayes is considered a machine learning method. Though both methods are computationally demanding and mathematically intricate, they are well-suited for cases where the posterior distribution of the marginal models is intractable—as is the case in our analysis.

Specifically, we apply the random walk Metropolis–Hastings (rw-MH) MCMC algorithm proposed by [53], along with the variational Bayes method developed by [37], both with and without data augmentation. These methods are used to estimate model parameters and to analyze the R-vine copula dependence structure.

To support and enhance the efficiency of these estimation procedures, we also incorporate plug-in parallel computing and the quasi-maximum likelihood estimation (QMLE) method. These additions help reduce computational time while preserving estimation accuracy, thereby facilitating the integration of Bayesian inference and machine learning techniques in a computationally feasible manner.

3.1. Full Likelihood Function

It is important to first discuss a likelihood function since this is a connection between Bayesian MCMC and machine learning methods. Given the observed n -dimensional data, the full log-likelihood function of our model is

$$\begin{aligned}
 \Lambda(\theta|y_{k,i,t}) &= \sum_{k=1}^n \log(f_k) \sum_{k=n-1}^1 \sum_{i=n}^{k+1} \log(c_{m_{k,k},m_{i,k}|m^*}(F_{m_{k,k}|m^*}, F_{m_{i,k}|m^*})) \\
 &= \sum_{k=1}^n \sum_{t=1}^T \log[q_k^L p_L^{GPD}(x_{k,t}) + (1 - q_k^L - q_k^R) p^G(x_{k,t}) + q_k^R p_R^{GPD}(x_{k,t})] \quad (8) \\
 &\quad \sum_{k=n-1}^1 \sum_{i=n}^{k+1} \sum_{t=1}^T \log(c_{t_{m_{k,k},m_{i,k}|m^*}}(F_{t_{m_{k,k}|m^*}}, F_{t_{m_{i,k}|m^*}))),
 \end{aligned}$$

where

$$\begin{aligned}
 p_L^{GPD}(x_{k,t}) &= \frac{1}{\sqrt{h_t^k(1 - \xi^{L,k})(1 - 2\xi^{L,k})^{1/2}}} \left(1 + \frac{\xi^{L,k}(y_{k,t} - \lambda_1^k - \lambda_2^k h_t^k)}{\sqrt{h_t^k(1 - \xi^{L,k})(1 - 2\xi^{L,k})^{1/2}}} \right)^{-\frac{1}{\xi^{L,k}} - 1}, \\
 p^G(x_{k,t}) &= \frac{1}{(2\pi h_t^k)^{1/2}} \exp\left[-\frac{(y_{k,t} - \lambda_1^k - \lambda_2^k h_t^k)^2}{2h_t^k} \right], \\
 p_R^{GPD}(x_{k,t}) &= \frac{1}{\sqrt{h_t^k(1 - \xi^{R,k})(1 - 2\xi^{R,k})^{1/2}}} \left(1 + \frac{\xi^{R,k}(y_{k,t} - \lambda_1^k - \lambda_2^k h_t^k)}{\sqrt{h_t^k(1 - \xi^{R,k})(1 - 2\xi^{R,k})^{1/2}}} \right)^{-\frac{1}{\xi^{R,k}} - 1},
 \end{aligned}$$

which is compatible with Equation (4). c_t is a unique bivariate copula pdf. F_t is a mixture cdf which is defined in Section 2.3. x_t is a standardized residual given by Equation (5). The parameter θ in parameter space Θ is $\theta = \{\lambda_1^k, \lambda_2^k, \xi^{L,k}, \xi^{R,k}, \omega^k, \gamma^k, \delta^k, \nu^k, \mathcal{P}^*\}$. In n dimensions, there are at least $\frac{n(n+15)}{2}$ parameters that need to be estimated. The number of parameters grows proportionally to the number of dimensions. From our R-vine copula-based models, there are at least 606 parameters in 30 dimensions and 6094 parameters in 100 dimensions, respectively. Notably, the trade-off between flexibility and parsimony needs to be taken into account. This leads to a research area of regularization or shrinkage method where the most common shrinkage methods are, for instance, lasso and ridge—see [54–56], among others. However, we leave this for further research.

3.2. Bayesian MCMC Estimation

Due to the intractable nature of the likelihood, a Bayesian approach is a good candidate for the estimation. Hence, this section discusses the rw-MH algorithm, enhanced with the QMLE and parallel computing to improve computational efficiency. Given the large number of parameters in vine copula models, we adopt the Inference for Margins (IFM) approach [57], which enables the parallelization of marginal model estimation and significantly accelerates the overall computation. Regarding the choice of prior distributions, we use an informative prior, especially the uniform distribution, in the estimation with the hyperparameter values following previous studies.

In summary, Algorithm 1 outlines the procedure for obtaining the parameter estimates $\hat{\mathcal{P}}^*$ for the R-vine copula model and $\hat{\theta}$ for the marginal models. The process begins by constructing the maximum matrix \mathbb{M} from \mathcal{M}^* , where each element $m_{i,k}$ in \mathbb{M} represents the maximum value among all entries in the k -th column from row i to row n .

The algorithm proceeds in several parallelized steps. The first loop performs data standardization in parallel. In the second loop, the algorithm computes the independent pairwise copulas in the first tree, T_1 , of the R-vine model—also in parallel. The third loop handles the dependent pairs in the subsequent trees T_i , for $i = 2, \dots, n - 1$.

Algorithm 1 Parallel algorithm of regular vine copulas model using random walk chain Metropolis–Hastings sampling MCMC and maximum likelihood estimation.

Require: y_t

- 1: Calculate $\mathbb{M}_{i,k} = \mathbf{m}_{i,k} | i = n, k = 1, \dots, n$. $\mathbf{m}_{n,k} = \max\{m_{i,k}, \dots, m_{n,k}\}$ for all $k = n, \dots, 1$. m is spanning tree maximisation.
- 2: **parfor** $i_g \leftarrow 1, n$ **do** \triangleright estimate margin parameters from Equations (5)–(7), do parallel
- 3: Perform $\hat{\theta}_{i_g}^{QMLE, Margins} \leftarrow \arg \max_{\theta \in \Theta} \log(f_{i_g}(\theta | y_{i_g}))$.
- 4: Perform $\hat{\theta}_{i_g}^{rmMH, Margins} | (\hat{\theta}_{i_g}^{QMLE, Margins}, y_{i_g})$ via call Algorithm 2
- 5: Calculate z_{t,i_g} via standardized $y_{t,i_g} | \hat{\theta}_{i_g}^{rmMH, Margins}$.
- 6: Calculate $u_{i_g} = F_{i_g}(z_{t,i_g}) \sim Unif(0, 1)$.
- 7: **end parfor**
- 8: Allocate $V^{direct} = (v_{i,k}^{direct})$ where $i = n$ and $k = 1, \dots, n$.
- 9: Allocate $(v_{n,1}^{direct}, \dots, v_{n,n}^{direct}) = (u_1, \dots, u_n)$.
- 10: **parfor** $k \leftarrow 1, n - 1$ **do** \triangleright first tree, do parallel
- 11: Perform $\hat{\mathbf{p}}_{n,k}^{QMLE}, BIC^{QMLE} \leftarrow \arg \max_{\theta \in \Theta} \log(c(z_{n,k}^{(2)}, z_{n,k}^{(1)}; t_{n,k}))$ using Equation (8) for copula candidate.
- 12: Find $\hat{\mathbf{p}}_{n,k}^{QMLE}, t_{n,k}$ by $\min\{BIC^{QMLE}\}$
- 13: Set $z_{n,k}^{(1)} = v_{n,k}^{direct}$ and $z_{n,k}^{(2)} = v_{i,(n-\mathbf{m}_{n,k}+1)}^{direct}$.
- 14: Perform $\hat{\mathbf{p}}_{n,k}^{rmMH} | (z_{n,k}^{(1)}, z_{n,k}^{(2)}, \hat{\mathbf{p}}_{n,k}^{QMLE})$ using Equation (8) via call Algorithm 2
- 15: Calculate $v_{n-1,k}^{direct} = h(z_{n,k}^{(1)}, z_{n,k}^{(2)} | t_{n,k}, \hat{\mathbf{p}}_{n,k}^{rmMH})$
- 16: Calculate $v_{n-1,k}^{indirect} = h(z_{n,k}^{(2)}, z_{n,k}^{(1)} | t_{n,k}, \hat{\mathbf{p}}_{n,k}^{rmMH})$
- 17: **end parfor**
- 18: **for** $k \leftarrow n - 2, 1$ **do** \triangleright column iteration, iteration for tree $n - 1$ to tree 2
- 19: **for** $i \leftarrow n - 1, k + 1$ **do** \triangleright row iteration
- 20: Calculate $\mathbb{M}_{i,k} = \mathbf{m}_{i,k}$. $\mathbf{m}_{i,k} = \max\{m_{i,k}, \dots, m_{n,k}\}$.
- 21: Set $z_{i,k}^{(1)} = v_{i,k}^{direct}$.
- 22: **if** $\mathbf{m}_{i,k} = m_{i,k}$ **then**
- 23: $z_{i,k}^{(2)} = v_{i,(n-\mathbf{m}_{i,k}+1)}^{direct}$
- 24: **else**
- 25: $z_{i,k}^{(2)} = v_{i,(n-\mathbf{m}_{i,k}+1)}^{indirect}$
- 26: **end if**
- 27: Perform $\hat{\mathbf{p}}_{i,k}^{QMLE}, BIC^{QMLE} \leftarrow \arg \max_{\theta \in \Theta} \log(c(z_{i,k}^{(2)}, z_{i,k}^{(1)}; t_{i,k}))$ using Equation (8) for copula candidate.
- 28: Find $\hat{\mathbf{p}}_{i,k}^{QMLE}, t_{i,k}$ by $\min\{BIC^{QMLE}\}$
- 29: Perform $\hat{\mathbf{p}}_{i,k}^{rmMH} | (z_{i,k}^{(2)}, z_{i,k}^{(1)}, t_{i,k}, \hat{\mathbf{p}}_{i,k}^{QMLE})$ using Equation (8) via call Algorithm 2
- 30: Calculate $v_{i-1,k}^{direct} = h(z_{i,k}^{(1)}, z_{i,k}^{(2)} | t_{i,k}, \hat{\mathbf{p}}_{i,k}^{rmMH})$
- 31: Calculate $v_{i-1,k}^{indirect} = h(z_{i,k}^{(2)}, z_{i,k}^{(1)} | t_{i,k}, \hat{\mathbf{p}}_{i,k}^{rmMH})$
- 32: **end for**
- 33: **end for**
- 34: **return** $\hat{\mathcal{P}}^*, \hat{\mathcal{T}}^*, \hat{\mathcal{M}}^*, \hat{\theta}$

To ensure better convergence, the algorithm initially employs the QMLE to provide starting values for the rw-MH sampler, described in Algorithm 2, which functions as a subroutine within Algorithm 1.

The function $c(u, v; t, \mathbf{p})$ denotes a specific bivariate copula density, parameterized by copula type t and parameter \mathbf{p} . The terms $v_{i,k}^{direct}$ and $v_{i,k}^{indirect}$ refer to recursive conditional cumulative distribution functions (cdfs), often denoted by $h(u, v | t, \mathbf{p})$, which are fundamental components in the construction of conditional copulas in vine structures.

$$\begin{aligned}
 v_{i,k}^{direct} &= C(v_{m_{k,k}} | u_{m_{i+1,k}}, u_{m_{i+2,k}}, \dots, u_{m_{n,k}}), \\
 v_{i,k}^{indirect} &= C(v_{m_{i+1,k}} | u_{m_{k,k}}, u_{m_{i+2,k}}, \dots, u_{m_{n,k}}),
 \end{aligned}
 \tag{9}$$

for all $n \geq i > 1$. Denote that $h(u, v | t, \mathbf{p}) = C_{2|1}(v | u; \theta, \delta)$. $C(u, v; \theta, \delta) = G(x, y; \theta, \delta)$ where $G(x, y; \theta, \delta)$ is a bivariate survival function and x, y are monotonically decreasing transforms. Therefore, the conditional *cdf* is $C_{2|1}(v | u; \theta, \delta) = \frac{\partial G}{\partial x} \cdot \frac{\partial x}{\partial u}$. See [30] for further discussion.

Algorithm 2 presents the rw-MH sampling method used as part of the MCMC procedure in Algorithm 1. In general, the performance of the Metropolis–Hastings algorithms is sensitive to the choice of initial values (θ, Ω) where Ω is a covariance matrix. The first stage of the rw-MH algorithm involves a burn-in period, during which early samples of θ are discarded. A longer burn-in period increases computational time, and its length largely depends on how close the initial values are to the target distribution.

To address this, our approach initializes the algorithm with parameter estimates obtained via the QMLE. This helps achieve a reasonable acceptance rate more quickly—approximately 50% in our implementation.

The proposal step involves an increment random variable z , which follows a Gaussian distribution, $z \sim N(0, c\Omega)$, where c is a constant scaling factor. This constant is tuned to ensure an appropriate acceptance rate for the sampler. For further details on the MH algorithm, see [53] (Exercise 17.5 and Exercise 11.18).

Algorithm 2 Random walk Metropolis–Hastings algorithm

Require: $\Lambda(\theta | y), \hat{\theta}^{QMLE}, \hat{\Sigma}^{QMLE}, c, nburn, nrep$.

- 1: Calculate $\Sigma^* = c \times \hat{\Sigma}^{QMLE}$.
- 2: Set $\theta^1 = \hat{\theta}^{QMLE}$.
- 3:
- 4: **for** $i \leftarrow 2, nburn + nrep$ **do**
- 5: Draw a candidate $\theta^* = \theta^{i-1} + z$
- 6: Draw an acceptance probability

$$\alpha(\theta^{i-1}, \theta^*) = \min \left[\frac{\Lambda(\theta^* | y)}{\Lambda(\theta^{i-1} | y)}, 1 \right]$$

- 7: Set $\theta^i = \theta^*$ with probability $\alpha(\theta^{i-1}, \theta^*)$. Otherwise, set $\theta^i = \theta^{i-1}$.
 - 8: **end for**
 - 9:
 - 10: Calculate posterior means $\hat{\theta}^{rwMH}$ from the draws θ and whatever posterior inference of interest are needed.
 - 11: **return** $\hat{\theta}^{rwMH}$
-

3.3. Variational Bayes Estimation

This section introduces an alternative algorithm based on the variational Bayes (VB) method, which integrates Bayesian inference with machine learning to estimate n -dimensional general regular vine copula models. Specifically, we apply the variational Bayes with (and without) data augmentation (VBDA) approach originally developed for a drawable vine structure by [37,58].

In this study, we extend the VBDA method to accommodate the more general class of regular vines by incorporating an additional stopping criterion. This enhancement is designed to reduce the number of iterative learning steps, thereby improving computational efficiency.

We explore three versions of the variational Bayes estimation method: one without data augmentation (VBDA0), and two with data augmentation (VBDA1 and VBDA2). All three methods rely on latent variables during estimation. To address convergence

challenges and reduce computational time, we also incorporate plug-in QMLE estimates and leverage parallel computing techniques.

The augmented posterior density is

$$p(\theta, \mathbf{u}|y) = \frac{f(y, \mathbf{u}, \theta)}{f(y)} = (c(\mathbf{u}|\theta)p(\theta) \prod_{t=1}^T \mathcal{I}(a_t \leq u_t < b_t)) / f(y).$$

where $p(\theta)$ is the prior density. $f(y)$ is the marginal likelihood function. $c(\mathbf{u}|\theta)$ is the copula density. \mathcal{I} is the indicator function. VB estimation approximates $p(\theta, \mathbf{u}|y)$ by a tractable density $q_\lambda(\theta, \mathbf{u})$ in a so-called variational approximation. This estimation corresponds to minimizing the Kullback–Leibler (KL) divergence

$$KL(q_\lambda(\theta, \mathbf{u})||p(\theta, \mathbf{u}|y)) = \int \log\left(\frac{q_\lambda(\theta, \mathbf{u})}{p(\theta, \mathbf{u}|y)}\right) q_\lambda(\theta, \mathbf{u}) d\theta du$$

or maximizing the lower bound of the logarithm of the marginal likelihood (or the evidence lower bound (ELBO)), given by

$$\mathcal{L}(\lambda) = \int \log\left(\frac{p(\theta)f(y, \mathbf{u}|\theta)}{q_\lambda(\theta, \mathbf{u})}\right) q_\lambda(\theta, \mathbf{u}) d\theta du,$$

where θ and \mathbf{u} are independent. For further details and examples of VB methods, see, among others, [59,60]. Also, see [61,62], among others, for reviews of VB inference in machine learning.

The density approximation of q_λ can be written as $q_\lambda(\theta, \mathbf{u}) = q_{\lambda^a}(\theta)q_{\lambda^b}(\mathbf{u})$. The specification of q_{λ^a} and q_{λ^b} is crucial. q_{λ^a} has parameter λ^a while q_{λ^b} has parameter λ^b with the support on $[a_1, b_1) \times \dots \times [a_T, b_T)$. This study selects the choice of \mathbf{u} approximation as follows:

$$\text{VBDA0} : q_{\lambda^b}(u) = 0,$$

$$\text{VBDA1} : q_{\lambda^b}(u) = \prod_{t=1}^T \frac{1}{b_t - a_t} \mathcal{I}(a_t \leq u_t < b_t),$$

$$\text{VBDA2} : q_{\lambda^b}(u) = \prod_{t=1}^T \frac{\phi(x_t; \mu_t, \Omega_t)}{(b_t - a_t)\phi(x_t)}.$$

VBDA2 nests VBDA1, which covers a wide range of data dependencies with an effective mean field approximation. VBDA1 is expected to be more accurate for data with low dependence. $\lambda^b = \{\mu_1, \dots, \mu_T, \log \omega_1, \dots, \log \omega_T\}$, $\Omega_t = \text{diag}(\omega_1, \dots, \omega_T)$, and ϕ is the standard normal density.

To approximate θ , we adopt a commonly used variational distribution q_{λ^a} , specifically the multivariate normal (Gaussian) distribution with mean M and covariance matrix Σ , denoted as $N(M, \Sigma)$. In the context of R-vine models, a large number of parameter estimates are required. The use of a tractable multivariate normal distribution makes it relatively straightforward to compute the gradient of $\log q_{\lambda^a}(\theta)$.

Moreover, the structure of Σ contributes to improving the accuracy of the gradient estimates. In particular, when Σ is expressed using its factorized form $\Sigma = BB' + D^2$, the gradient of this factorization has been derived and shown to be computationally efficient [63].

Algorithm 3 shows the VBDA algorithm with stochastic gradient ascent optimization. Note that, in Algorithm 1, Algorithm 2 is replaced by Algorithm 3 in order to perform VBDA method in the R-vine model. This approach is tractable posterior distribution.

Algorithm 3 Variational Bayes data augmentation with control variates and ADADELTA learning rate

Require: $\Lambda(\theta|y), DA, \lambda^{(0)}, \mu^{QMLE}, S, nVB, c, t_w, P, \rho, \epsilon.$

- 1: Set $k = 1.$
- 2: Initialize $\lambda | \mu^{QMLE}.$
- 3: Generate $(\theta_s^{(k)}, \mathbf{u}_s^{(k)}) \sim q_{\lambda^k}(\theta, u)$ for $s = 1, \dots, S$ where
- 4: $q_{\lambda^b}(\mathbf{u})$ could be *i. DA0 ii. DA1* or *iii. DA2* and $q_{\lambda^a}(\theta) \sim N(M_\theta, \Sigma_\theta).$
- 5: Calculate $\zeta_i^{(k)} = (\zeta_1^{(k)}, \dots, \zeta_m^{(k)})$ with

$$\zeta_i^{(k)} = \frac{\text{cov}\left([\log h(\theta, \mathbf{u}) - \log q_\lambda(\theta, \mathbf{u})] \times \nabla_{\lambda_i} \log q_\lambda(\theta, \mathbf{u}), \nabla_{\lambda_i} \log q_\lambda(\theta, \mathbf{u})\right)}{\text{var}(\nabla_{\lambda_i} \log q_\lambda(\theta, \mathbf{u}))}$$

- 6: Set $k = k + 1.$
- 7: **while** *stopping rule* is not satisfied **do**
- 8: Generate $(\theta_s^{(k)}, \mathbf{u}_s^{(k)}) \sim q_{\lambda^k}(\theta, u)$ for $s = 1, \dots, S.$
- 9: Calculate $\nabla_\lambda \left(\widehat{\mathcal{L}}(\lambda^{(k)})\right) = \left(g_{\lambda_1}^{(k)}, \dots, g_{\lambda_m}^{(k)}\right)'$

$$g_{\lambda_i}^{(k)} = \frac{1}{S} \sum_{s=1}^S \left(\log h(\theta_s^{(k)}, \mathbf{u}_s^{(k)}) - \log q_\lambda(\theta_s^{(k)}, \mathbf{u}_s^{(k)}) - \zeta_i^{(k-1)} \right) \times \nabla_{\lambda_i} \log q_\lambda(\theta_s^{(k)}, \mathbf{u}_s^{(k)})$$

- 10: Calculate $\zeta_i^{(k)} = (\zeta_1^{(k)}, \dots, \zeta_m^{(k)}).$
 - 11: Perform $\Delta\lambda^{(k)} \leftarrow$ the ADADELTA method.
 - 12: Set $\lambda^{(k+1)} = \lambda^{(k)} + \Delta\lambda^{(k)}$
 - 13: Calculate $\widehat{LB}(\lambda^{(k+1)}) = \frac{1}{S} \sum_{s=1}^S \left(\log h(\theta_s^{(k+1)}, \mathbf{u}_s^{(k+1)}) - \log q_\lambda(\theta_s^{(k+1)}, \mathbf{u}_s^{(k+1)}) \right)$
 - 14: **if** $k > c$ **then**
 - 15: Calculate $\overline{LB}_{k-t_w} = 1/t_w \sum_{i=1}^{t_w} \widehat{LB}(\lambda^{(k-i+1)})$
 - 16: **if** $\widehat{LB}(\lambda^{(k-t_w)}) > \max(\overline{LB})$ **then**
 - 17: *patience* = 0.
 - 18: Collect $\lambda^{(*)} \leftarrow \lambda^{(k+1)}.$
 - 19: **else**
 - 20: *patience* = *patience* + 1.
 - 21: **end if**
 - 22: **if** *patience* > P **then**
 - 23: *stop.*
 - 24: **end if**
 - 25: **end if**
 - 26: Set $k = k + 1.$
 - 27: **end while**
 - 28: **return** $\lambda^{(*)}$
-

A brief overview of the procedure is as follows:

Three VB estimators for regular vine dependence models are introduced in Line 4, where the specification of $q_{(u)}$ can be one of DA0, DA1, or DA2. To initiate the algorithm, an initial value for the variational parameter, $\lambda^{(0)}$, is required, typically initialized using QMLE estimates μ^{QMLE} .

In Line 12, the lower bound (LB) is updated iteratively as follows: $\lambda^{(k+1)} = \lambda^{(k)} + \rho^{(k)} \nabla_\lambda \widehat{\mathcal{L}}(\lambda^{(k)})$. Here, $\rho^{(k)}$ is the learning rate, and $\nabla_\lambda \widehat{\mathcal{L}}(\lambda^{(k)})$ is an unbiased estimate of the gradient of the lower bound \mathcal{L} , computed using the ADADELTA optimization method (Line 11). For further details on adaptive learning rate techniques like ADADELTA, see [64].

The gradient of the lower bound, $\nabla_\lambda \mathcal{L}(\lambda)$, is expressed as the expectation:

$$\nabla_{\lambda} \mathcal{L}(\lambda) = E_q(\nabla_{\lambda} \log q_{\lambda}(\theta, \mathbf{u}) \{ \log h(\theta, \mathbf{u}) - \log q_{\lambda}(\theta, \mathbf{u}) \})$$

where $p(\theta, \mathbf{u}|y) \propto f(y, \mathbf{u}|\theta)p(\theta) = h(\theta, \mathbf{u})$ represents the unnormalized posterior distribution.

To reduce the variance of this unbiased gradient estimator, a control variate vector ζ is employed. Consequently, the unbiased estimator of the gradient is given by $\nabla_{\lambda} \widehat{\mathcal{L}}(\lambda^{(k)}) = (g_{\lambda_1}, \dots, g_{\lambda_m})'$ where $g_{\lambda} = \frac{1}{S} \sum_{s=1}^S (\log h(\theta_s, \mathbf{u}_s) - \log q_{\lambda}(\theta_s, \mathbf{u}_s)) \nabla_{\lambda} \log q_{\lambda}(\theta_s, \mathbf{u}_s)$ where $g_{\lambda} = \frac{1}{S} \sum_{s=1}^S (\log h(\theta_s, \mathbf{u}_s) - \log q_{\lambda}(\theta_s, \mathbf{u}_s)) \nabla_{\lambda} \log q_{\lambda}(\theta_s, \mathbf{u}_s)$ and m is the number of elements in λ , and SS is the number of Monte Carlo samples.

For a more detailed discussion of the VBDA algorithm, refer to [30] (Section 3).

The stopping rule for the algorithm terminates the iterative procedure if

- (i.) There is a fixed number of stochastic gradient ascent steps, nVB , or
- (ii.) The moving average \widehat{LB} is not improved, $\overline{LB}_{k-t_w} = 1/t_w \sum_{i=1}^{t_w} \widehat{LB}(\lambda^{(k-i+1)})$, which is referred to as the *patience* parameter and $patience = patience + 1$ in Line 20 and the whole procedure stops when $patience > \text{maximum patience } P$ in Line 22–23, and
- (iii.) All of the above conditions will be checked when the k^{th} iteration is greater than a constant c .

Finally, $\lambda^{(*)}$ is collected when \overline{LB}_{k-t_w} is greater than $\max(\overline{LB})$ [60]. Note that the simulation study in Bayesian inference and machine learning including the rw-MH algorithm and the VBDA algorithm can be found in the Supplementary Material. The results show that, in general, the VB methods perform better than the others.

4. Empirical Study

4.1. Data

This study analyzes financial market time series data with up to 100 dimensions. For the training set, we selected the daily adjusted closing prices of the top 100 Nasdaq-listed stocks by market capitalization, covering the period from 2 January 2009 to 20 September 2021, where we intentionally avoided the impact of the 2008 global financial crisis. Only stocks listed before 2009 were included, resulting in a total of 3201 daily observations. The test set spans from 21 September 2021 to 21 December 2021, comprising 65 daily observations. All data were obtained from Yahoo Finance using MATLAB 2022a’s data feed function. The experiments were implemented in MATLAB and conducted on a laptop equipped with an 11th Gen Intel(R) Core(TM) i7-1185G7 processor (3.00 GHz) and 32 GB of RAM.

After completing data cleaning, we obtained 100 stock indices for use in our empirical study. A full list of these indices is provided in Appendix C. To evaluate the performance of the proposed models described in Section 2, we conducted two empirical experiments using two datasets: one consisting of all 100 time series and another with only the last 30 stock indices. The goal of these experiments was to assess the scalability and effectiveness of the proposed model as the dimensionality increases.

The descriptive statistics of the dataset are summarized as follows. The top five business sectors represented in our analysis are technology (22 firms), health care (18 firms), finance (11 firms), consumer services (11 firms), and capital goods (8 firms). The total market capitalization of the selected stocks is USD 30,814 billion, accounting for approximately 46% of the total market capitalization on the Nasdaq exchange.

All stocks exhibit a positive mean return, with return values ranging from -39.02% to 52.29% . The standard deviation ranges from 1.06% to 3.67% , while skewness values range between -1.04 and 1.56 . Kurtosis spans from 5.95 to 47.26 . These statistics indicate that the return distributions are asymmetric and exhibit heavy tails.

Additionally, the Jarque–Bera test statistics for all returns strongly reject the null hypothesis of normality at the 1% significance level, confirming that the return distributions deviate significantly from normality.

4.2. Regular Vine Distribution Selection

Multivariate copula models can be fitted to n -dimensional time series data using a two-stage sequential approach [30]. In the first stage, a univariate marginal model is selected and fitted separately to each time series to obtain standardized residuals, which are then transformed into marginally uniform variables. In this study, we apply our proposed marginal model, detailed in Section 2.3, and compare its performance to a widely used benchmark: the ARMA(1,1)-GARCH(1,1) model with Student- t innovations.

In the second stage, the dependence structure among the transformed residuals is modeled using a multivariate copula. It is well established in the literature that different pairs of financial variables exhibit varying degrees of asymmetry and tail dependence. Traditional multivariate copulas are generally unable to capture such heterogeneous dependence structures. Vine copulas, however, are specifically designed to address this limitation.

While D-vine and C-vine copulas are effective in modeling complex dependencies, they are constrained by their predefined path structures. To overcome this limitation, we investigate the use of regular vine (R-vine) copulas—a more flexible and general class that encompasses both D-vines and C-vines as special cases. R-vines offer significantly greater modeling capacity due to their extensive variety of possible tree structures and pair-copula combinations.

In fact, for high-dimensional datasets, the number of possible R-vine structures grows rapidly, with the total number given by $\frac{n!}{2} 2^{\binom{n-2}{2}}$ [65], underscoring their flexibility and complexity compared to standard copula models and their vine subclasses.

The following tasks are the sequential steps of fitting an R-vine copula with marginal specification:

- (a) Standardize the residuals of the returns using the univariate marginal model.
- (b) Select the R-vine path structure of the tree, $\mathcal{M} = (N, E)$, given by maximized empirical Kendall's τ via
 - b.1 Formulation of empirical Kendall's τ of all possible pairs variable, $\hat{\tau}_{i,k}$, $1 \leq i < k \leq n$;
 - b.2 Performing the spanning tree maximization (the sum of absolute empirical Kendall's τ)

$$\max_{e=\{i,k\} \text{ in spanning tree}} \sum |\hat{\tau}_{i,k}|;$$
 - b.3 For each edge, selection of bivariate copula family $t_{i,k}$ and its parameter(s) $p_{i,k}$ from all possible bivariate copula candidates via Bayesian information criterion (BIC) value.
- (c) Iterate Step (b) from the first tree T_1 (unconditional tree, $\hat{\tau}_{i,k}$) to the last tree T_{n-1} (conditional tree, $\hat{\tau}_{i,k|D}$ where D is the proximity condition).

The methodology described above yields a complete specification of a regular vine copula, including $\mathcal{M}, \mathcal{T}, \mathcal{P}$. The QMLE is employed in Step b.3 to estimate all relevant parameters. This full specification is then used as the foundation for our Bayesian inference and machine learning procedures.

It is important to note that, for the first tree in the vine structure, parallel computation is feasible due to the independence of the path structure in R-vines. The method proceeds as follows:

- Step 1 (Tree 1): A parallel computation technique is applied to estimate $\hat{\tau}_{1,k}$ for $k = 1, \dots, n - 1$.
- Step 2: A FOR loop is used to compute $\hat{\tau}_{i,k}$ where $i < k, i = n - 1, \dots, 2, k = 1, \dots, n - 2$.

For more details on the maximum spanning tree algorithm used in this context, see [30].

Our study employs a range of bivariate copula functions to capture the diverse (a)symmetrical dependence structures present in financial time series data. While the core families considered include the Gaussian, Student- t , Gumbel, and Frank copulas—consistent with the approach of [30]—we extend the analysis by incorporating a total of 13 copula families. The complete list of candidate bivariate copula families used in this study is provided below:

- Gaussian/Normal copula presents symmetric and no tail dependence.
- Student- t copula presents symmetric and upper and lower tail dependence.
- Frank copula presents symmetric and tail independence.
- Clayton copula presents asymmetric and lower tail dependence.
- B5/Joe, Gumbel, and Galambos copulas present asymmetric and different upper and lower tail dependence.
- BB1, BB3, BB4, BB5, BB7 and BB10 copulas present asymmetric and different upper and lower tail dependence.

Details of the bivariate copula distributions and their key dependence structure properties are summarized in Appendix A. These include the cumulative distribution functions, lower and upper tail dependence orders, and rank-based dependence measures. For additional discussion on copula properties, see, among others, [17] (Chapter 4) and [66] (Appendix).

We also conducted Ljung–Box tests on the standardized residuals from all proposed models. The results confirm the independence of the residuals. Full details of these tests are provided in Appendix B.

4.3. Empirical Results

In this section, we present the empirical results obtained from our proposed Bayesian models and estimation methods for R-vine copulas. A comprehensive set of 13 bivariate copula families is employed within the regular vine structure to capture complex dependence patterns.

For the univariate margins, we apply the mixture distribution defined in Equation (7) to the ICAPM-EGARCH models, which are designed to capture key features of financial time series data such as volatility clustering and heavy tails.

As a benchmark, we use the traditional QMLE method and compare its performance with our proposed approaches, including the rw-MH algorithm and the VBDA methods. These methods are implemented in the R-vine copula framework for dimensions of up to 100. In total, the empirical study involves the estimation of up to 4950 bivariate copula functions and 6138 parameters.

The experimental results, summarized in Tables 1 and 2, indicate that the VBDA1 method delivers the best performance in terms of marginal likelihood on the training set. On the test set, the variational Bayes approaches—particularly VBDA1—demonstrate substantial improvements in forecasting accuracy as the dimensionality increases, making them strong candidates for high-dimensional modeling. Specifically, as both the number of dimensions and the forecast horizon grow, VBDA1 provides more accurate variability estimates. In contrast, the QMLE method performs relatively poor in higher dimensions and longer forecast horizons. However, in the case of 30 dimensions with a three-month forecast window, QMLE outperforms the other methods.

Table 1. Performance comparison of 30 dimensions.

Model	Method	Training Data					Test Data				
		MarLik	AIC	BIC	Loglik	Last nVB Step	Time	1 Month		3 Months	
								MAD	RMSE	MAD	RMSE
MD1	QMLE	n.a.	302,512	302,955	−150,347	n.a.	23	1.22	1.63	1.43	1.94
MD1	rw-MH	1,381,159	300,006	302,968	−149,107	n.a.	1484	1.22	1.63	1.42	2.03
MD1	VBDA0	1,395,092	320,038	323,005	−159,122	944	3586	1.50	2.57	1.54	2.26
MD1	VBDA1	1,402,133	328,476	331,339	−163,360	489	1465	1.45	2.31	1.48	2.21
MD1	VBDA2	1,395,652	318,248	321,177	−158,234	728	10,530	1.35	2.01	1.54	2.19
MD2	QMLE	n.a.	298,389	298,677	−148,359	n.a.	28	1.22	1.63	1.41	2.04

Notes: MD1 = R-vine-ICAPM-EGARCH-Mixture model. MD2 = R-vine-ARMA-GARCH-Student-*t* model. The acceptance rate of the rw-MH method is 49%. Time is computational time in minutes. n.a. = not available. MarLik = marginal log-likelihood. MAD = mean absolute deviation. RMSE = root mean square error. Bold font indicates better results.

Table 2. Performance comparison of 100 dimensions.

Model	Method	Training Data					Test Data				
		MarLik	AIC	BIC	Loglik	Last nVB Step	Time	1 Month		3 Months	
								MAD	RMSE	MAD	RMSE
MD1	QMLE	n.a.	969,425	1,015,504	−477,779	n.a.	1.18	1.59	1.35	1.85	
MD1	rw-MH	4,372,594	906,846	948,133	−446,490	n.a.	1.17	1.58	1.34	1.85	
MD1	VBDA0	4,483,822	915,862	956,847	−451,037	1116	1.19	1.60	1.37	1.90	
MD1	VBDA1	4,485,991	963,895	1,005,219	−475,009	567	1.21	1.63	1.34	1.86	
MD1	VBDA2	4,482,233	917,936	959,083	−452,053	1086	1.19	1.61	1.37	1.89	
MD2	QMLE	n.a.	964,942	1,010,414	−475,638	n.a.	1.20	1.61	1.34	1.88	

Notes: MD1 = R-vine-ICAPM-EGARCH-Mixture model. MD2 = R-vine-ARMA-GARCH-Student-*t* model. The acceptance rate of the rw-MH method is 48%. Time is computational time in minutes. n.a. = not available. MarLik = marginal log-likelihood. MAD = mean absolute deviation. RMSE = root mean square error. Bold font indicates better results.

For 100-dimensional data, VBDA1 achieves the best forecasting performance based on the mean absolute deviation (MAD). Meanwhile, both the rw-MH and QMLE methods yield the best performance in terms of root mean square error (RMSE) for the three-month test set, while rw-MH shows the strongest performance in one-month forecasts at the 100-dimensional level.

Figure 3 illustrates a representative example of the empirical results for a normalized R-vine copula contour plot and rank-based dependence measures. These results are based on the univariate ICAPM-EGARCH-mixture marginal model using the VBDA2 estimation method in a 30-dimensional setting.

Due to space, we present a normalized 10-dimensional subset of the R-vine contour plot for selected stock indices. The stock names appear along the diagonal of the matrix. The lower triangular part of the matrix displays the R-vine copula contours, where each row corresponds to the *i*th tree (*i* = 1, ..., 9) derived from the full 30-dimensional dataset. The upper triangular matrix shows the empirical Kendall’s $\hat{\tau}^k$ and Spearman’s $\hat{\rho}_s^k$, where the *k*th column (*k* = 10, ..., 2) corresponds to tree \hat{T}_i for *i* = 1, ..., 9).

Similarly, Figure 4 provides an empirical example of the normalized 10-dimensional R-vine contour plots and rank-based dependence measures from the 100-dimensional analysis, using the VBDA1 method. Complete empirical results—including Kendall’s $\hat{\tau}$, Spearman’s $\hat{\rho}_s$, and the estimated vine matrices $\hat{\mathcal{M}}$ and $\hat{\mathcal{T}}$ for both figures—are available in the Supplementary Materials.

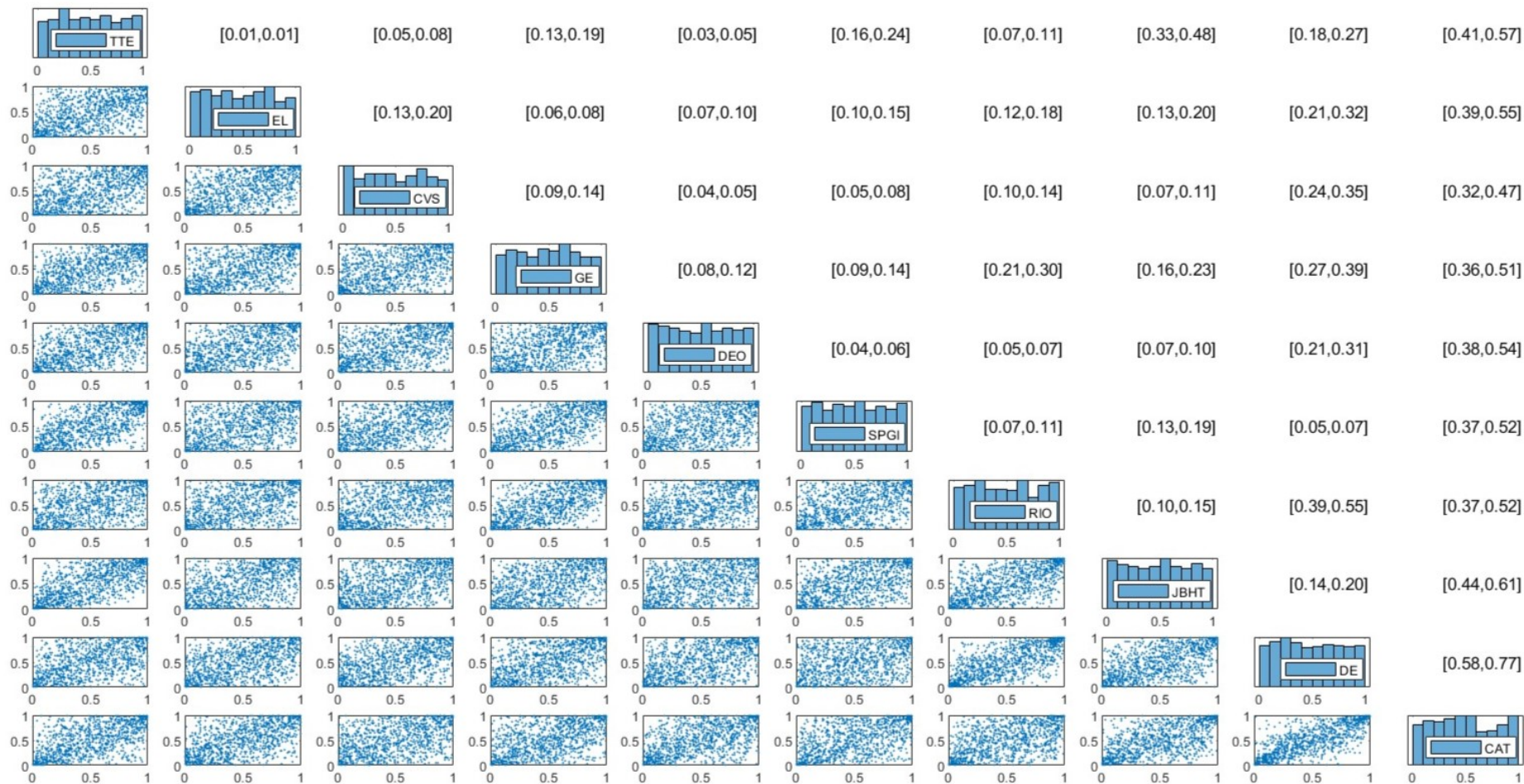


Figure 3. Estimation results of normalized mixed R-vine copula contour plot, Kendall's $\hat{\tau}$, and Spearman's $\hat{\rho}_s$ for 10 stock indices in the 30-dimensional cases using R-vine-ICAPM-EGARCH-Mixture model and VBDA2 method. Notes: Due to space, only 10 representative stocks are shown. Contour plots are shown in the lower triangular matrix while [Kendall's $\hat{\tau}$, Spearman's $\hat{\rho}_s$] are shown in the upper triangular matrix.



Figure 4. Estimation results of normalized mixed R-vine copula contour plot, Kendall’s $\hat{\tau}$, and Spearman’s $\hat{\rho}_s$ for 10 stock indices in the 100-dimensional cases using R-vine–ICAPM–EGARCH–Mixture model and VBDA1 method. Notes: Due to space, only 10 representative stocks are shown. Contour plots are shown in the lower triangular matrix while [Kendall’s $\hat{\tau}$, Spearman’s $\hat{\rho}_s$] are shown in the upper triangular matrix.

Table 1 summarizes the model performance for the 30-dimensional case. Based on the training dataset, the proposed VBDA1–R-vine–ICAPM–EGARCH–mixture model delivers the best performance in terms of marginal likelihood. The next best performers are the VBDA2 and VBDA0 methods, respectively. Among all estimation methods, VBDA2 is the most computationally intensive, which explains its longer runtime for R-vine copula estimation. VBDA1, while still relatively complex, requires less computing time than VBDA2 and converges more efficiently than simpler methods such as VBDA0 due to its favorable convergence properties.

An illustration of the convergence behavior of the evidence lower bound (ELBO) for all VBDA methods is shown in Figure 5, using the stock index pair (DE, CAT) as an example. Additionally, Figure 6 presents the first tree structure from the spanning tree maximization algorithm for the VBDA2–R-vine–ICAPM–EGARCH–mixture model. In contrast, when evaluating by Akaike information criterion (AIC), Bayesian information criterion (BIC), and log-likelihood values, the QMLE–R-vine–ARMA–GARCH–Student-*t* model achieves the best performance. The second-best performance under these criteria is the rw-MH-based R-vine–ICAPM–EGARCH–mixture model, which achieves a favorable acceptance rate in the MH algorithm.

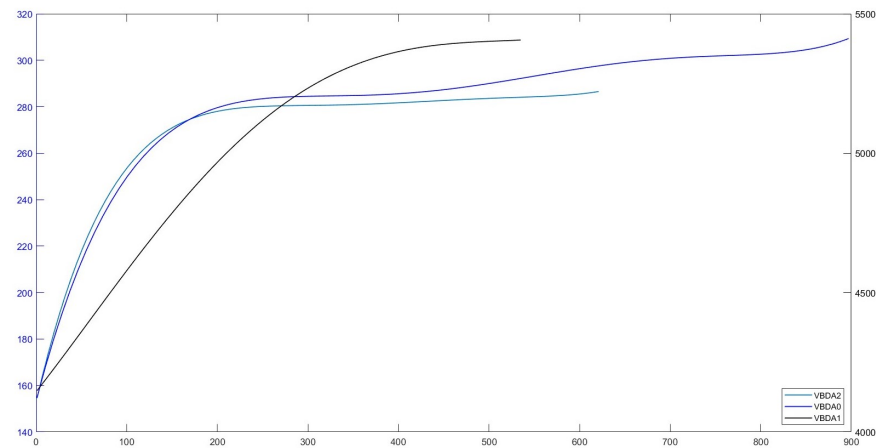


Figure 5. An example of ELBO convergence across VBDA methods for a pair of stocks (DE, CAT), using a Student-*t* copula in the 30-dimensional case. Note: The ELBO values of VBDA0 are plotted against the right y-axis while those of VBDA1 and VBDA2 are plotted against the left y-axis.

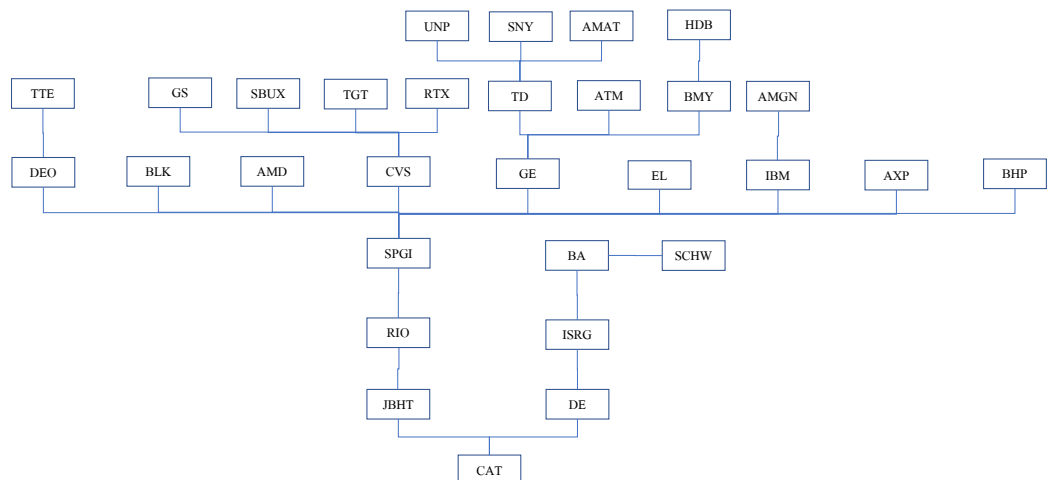


Figure 6. An example of Tree 1 in the 30-dimensional case of an R-vine copula using VBDA2 within the R-vine–ICAPM–EGARCH–Mixture model.

For the test dataset, forecasting performance was evaluated over one-month and three-month horizons using the MAD and RMSE metrics. In terms of forecast errors, the QMLE and rw-MH methods show comparable predictive performance and emerge as the best-performing models overall. Interestingly, within the variational Bayes framework, VBDA1 demonstrates improved performance—especially as the forecast horizon increases—outperforming the other VBDA variants based on MAD values. This trend is further supported by the experimental results.

Table 2 summarizes the performance of the proposed models in the 100-dimensional setting, using both training and test datasets. Overall, as the number of dimensions increases to 100, the performance of variational Bayes methods improves noticeably.

Among all approaches, the Bayesian inference and machine learning methods—particularly the Variational Bayes with Data Augmentation Type 1 (VBDA1)—outperform the traditional QMLE method, especially in terms of the mean absolute deviation (MAD) for the three-month forecast horizon. The VBDA1–R-vine–ICAPM–EGARCH–mixture model achieves the lowest three-month MAD, accurate to three decimal places, indicating superior forecasting accuracy.

These findings suggest that variational Bayes—specifically, the VBDA1 algorithm—performs effectively in high-dimensional settings. Both the training and test results confirm that the VBDA1–R-vine–ICAPM–EGARCH–mixture model surpasses all competing models in terms of marginal likelihood and three-month MAD forecast error.

Based on the model comparisons, the rw-MH algorithm combined with the R-vine–ICAPM–EGARCH–Mixture model emerges as the most outstanding, particularly due to its reasonable acceptance rate. Variational Bayes inference also performs exceptionally well, with the VBDA0–R-vine–ICAPM–EGARCH–Mixture model ranking as the second most preferable according to the AIC and BIC values. While the traditional QMLE method shows a decline in performance on the training data, it still remains a viable alternative for the test data.

Tables 3 and 4 summarize the copula families and their parameter counts across models and methods for 30- and 100-dimensional settings, as selected by Algorithm 1. In total, there are 435 pair-copula functions in 30 dimensions and 4950 in 100 dimensions. In the 30-dimensional case, the best-performing method—VBDA1—estimates 638 parameters, which is the second-lowest among all methods. Conversely, in the 100-dimensional case, VBDA1 estimates 6138 parameters—the highest. This indicates that better model performance does not necessarily correspond to a smaller number of parameters. Among the selected bivariate copula families, the elliptical copulas—Student- t and Gaussian—fit the empirical data best, followed by Archimedean families. The Student- t , Gaussian, and Frank copulas together account for at least 68% of all copulas used, implying that approximately 32% of the data exhibit asymmetric or tail dependence characteristics. In 30 dimensions, the most frequently selected families are Student- t and Frank, while in 100 dimensions, the elliptical copulas dominate. Notably, the BB6 copula was not selected in the 30-dimensional case, and the BB10 copula was not used in the 100-dimensional case.

In this study, we apply the ICAPM–EGARCH–mixture with R-vine copula, which we believe is a useful model. However, this might be subject to the model misspecification, which can be a limitation of this study. The computing performance is another possible limitation, which can be improved, but we leave for future research. A practical implementation from the practitioner’s viewpoint poses an interesting challenge to be validated in the future.

Table 3. Summary of copula families and their parameter numbers across models and methods for 30 dimensions.

Model	R-Vine–ICAPM–EGARCH–Mixture					Competing Model *
Estimation Method	VBDA0	VBDA1	VBDA2	rw-MH	QMLE	QMLE
Total Parameter Estimates **	657	638	650	656	682	606
Copula Family	Number of Parameters					
Gaussian	53	73	54	37	39	44
Student- <i>t</i>	156	119	151	156	143	118
Frank	111	111	114	124	121	178
Clayton	19	23	24	29	31	24
Gumbel	17	12	12	14	16	6
B5/Joe	8	4	8	3	7	6
Galambos	5	9	8	7	3	6
BB1	6	10	9	4	7	3
BB3	1	1	3	1	1	0
BB4	1	5	1	1	0	0
BB6	0	0	0	0	0	0
BB7	12	15	14	12	12	1
BB10	46	53	37	47	55	49

Notes: * The competing model is the R-vine–ARMA–GARCH–Student-*t* model. ** Including the copula and marginal distribution parameters.

Table 4. Summary of copula families and their parameter numbers across models and methods for 100 dimensions.

Model	R-Vine–ICAPM–EGARCH–Mixture					Competing Model *
Estimation Method	VBDA0	VBDA1	VBDA2	rw-MH	QMLE	QMLE
Total Parameter Estimates **	6094	6138	6115	6133	6133	6133
Total Copula Family	4950					
Gaussian	1537	1659	1535	1527	1527	1527
Student- <i>t</i>	1226	1526	1215	1207	1207	1207
Frank	617	777	625	633	633	633
Clayton	528	254	526	526	526	526
Gumbel	440	228	446	452	452	452
B5/Joe	196	70	193	193	193	193
Galambos	178	185	174	173	173	173
BB1	141	157	142	141	141	141
BB3	51	40	49	43	43	43
BB4	24	47	29	34	34	34
BB6	6	3	10	11	11	11
BB7	6	4	6	10	10	10
BB10	0	0	0	0	0	0

Notes: * The competing model is R-vine–ARMA–GARCH–Student-*t* model. ** Including the copula and marginal distribution parameters.

5. Concluding Remarks

This study introduces the variational Bayes methods—with and without data augmentation types 1 and 2—for Bayesian inference and machine learning, alongside the rw-MH Bayesian MCMC approach and the competing QMLE method, applied to regular vine copula models in graph theory for *n*-dimensional data. Standard multivariate copulas often face limitations due to having a single control parameter, a problem that worsens as dimensionality increases. Although C-vine and D-vine copulas offer more flexibility by allowing different bivariate copula choices and tree constructions, they impose restrictions on the tree path structure. To overcome this, we employ the more general R-vine copula, which eliminates the tree path structure constraint and provides richer insights into the relationships within financial data. This study considers 13 candidate bivariate copula families for the R-vine model, though the algorithm easily accommodates additional families. Furthermore, our proposed model is fully compatible with the special subclasses of C-vine and D-vine copulas and can be applied to *n*-dimensional data analysis.

Furthermore, this study is the first to introduce a mixture distribution into the R-vine model, enhancing its flexibility. The mixture distribution combines a Gaussian distribution with two generalized Pareto distributions within a univariate marginal intertemporal capital asset pricing model and an asymmetric GARCH framework, capturing additional risk premium stylized facts of specific financial securities. Additionally, the study presents a parallel R-vine with margins algorithm for Bayesian inference and machine learning, a bivariate copula selection algorithm within the R-vine framework, and an R-vine with margins data simulation algorithm. Importantly, our proposed algorithm can compute the joint distribution for arbitrary R-vine copulas.

In conclusion, our study produces the R-vine tree path dependence structure, pair-copula types, and parameter estimates in n -dimensions. The univariate marginal model captures individual stock risk premiums and risk-free rates. Our proposed model, particularly the VBDA1–R-vine–ICAPM–EGARCH–mixture model, outperforms competing approaches. Moreover, we demonstrate that the traditional QMLE estimation method significantly deteriorates as dimensionality increases.

For future research, there are numerous avenues to explore depending on specific interests. For instance, the model could be applied to higher-frequency data or extended with alternative copula families such as factor copulas or rotated copulas. The copula selection criteria could also be modified—using Spearman’s ρ_s or other relevant measures tailored to particular financial contexts instead of Kendall’s τ . Additionally, the framework can be adapted to other domains, such as high-dimensional portfolio optimization and higher-dimensional datasets. From a computational perspective, further development could involve implementing the algorithm on high-performance computing platforms to reduce computational time in Bayesian inference and machine learning.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/math13233886/s1>.

Author Contributions: Conceptualization, R.K. and N.W.; methodology, R.K.; software, R.K.; validation, R.K. and N.W.; formal analysis, R.K.; data curation, R.K.; writing—original draft preparation, R.K.; writing—review and editing, N.W.; visualization, R.K.; supervision, N.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Acknowledgments: We thank the reviewers for their useful comments and feedback that help improve the manuscript.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Some Parametric Copula Families and Their Properties

This appendix presents the dependence structure of all copula families used in this study including one- and two-parameter copulas. The elliptical copulas are the Gaussian (normal) copula and Student- t copula. The bivariate copula distributions of Gaussian and Student- t , respectively, are

$$C_{\Sigma}^{Ga}(u) = \Phi_{\Sigma}(\Phi^{-1}(u_1), \Phi^{-1}(u_2))$$

$$C_{v,\Sigma}^t(u) = t_{v,\Sigma}\left(t_v^{-1}(u_1), t_v^{-1}(u_2)\right)$$

where Φ^{-1} is an inverse cumulative Gaussian distribution function. Σ is a 2×2 correlation matrix; $u \sim Unif(0, 1)$ is a continuous standard uniform random variable; t^{-1} is an inverse

cumulative Student-*t* distribution function; $t_{v,\Sigma}$ is the cdf of bivariate $t_{v,\Sigma}$ distribution where $v > 2$ degree of freedom. The copula is denoted as $C(u, v) = \varphi(\varphi^{-1}(u) + \varphi^{-1}(v))$. It is worth mentioning that $C(u, v; \theta, \delta) = G(x, y; \theta, \delta)$ where $G(x, y; \theta, \delta)$ is a bivariate survival function and x, y are monotonically decreasing transforms. Therefore, the conditional cdf and copula pdf are:

$$C_{2|1}(v|u; \theta, \delta) = \frac{\partial G}{\partial x} \cdot \frac{\partial x}{\partial u}$$

$$c(u, v; \theta, \delta) = \frac{\partial^2 G}{\partial x \partial y} \cdot \frac{\partial x}{\partial u} \cdot \frac{\partial y}{\partial v}$$

The bivariate cdf copula families in this study including one and two parameters are presented in Table A1 while their tail dependence measures are shown in Table A2.

Table A1. Some parametric copula distributions.

Copula	Bivariate Function $C(u, v)$	Parameter θ, δ
One-parameter		
Gumbel	$\exp[-((-\log(u))^\theta + (-\log(v))^\theta)^{1/\theta}]$	$\theta \geq 1$
Clayton	$(u^{-\theta} + v^{-\theta} - 1)^{-1/\theta}$	$\theta \geq -1, \theta \neq 0$
Frank	$-\frac{1}{\theta} \ln\left(1 + \frac{(e^{-\theta u} - 1)(e^{-\theta v} - 1)}{e^{-\theta} - 1}\right)$	$-\infty < \theta < \infty, \theta \neq 0$
Joe/B5	$1 - [(1-u)^\theta + (1-v)^\theta - (1-u)^\theta(1-v)^\theta]^{1/\theta}$	$\theta \geq 1$
Galambos	$uv \exp\{[-(-\log u)^{-\theta} + (-\log v)^{-\theta}]^{-1/\theta}\}$	$0 \leq \theta < \infty$
Two-parameters		
BB1	$(1 + [(u^{-\theta} - 1)^\delta + (v^{-\theta} - 1)^\delta]^{1/\delta})^{-1/\theta}$	$\theta > 0, \delta \geq 1$
BB3	$\exp\{-[\delta^{-1} \log(e^{\delta \bar{u}^\theta} + e^{\delta \bar{v}^\theta} - 1)]^{1/\theta}\}, \bar{u} = -\log(u), \bar{v} = -\log(v)$	$\theta \geq 1, \delta > 0$
BB4	$(u^{-\theta} + v^{-\theta} - 1 - [(u^{-\theta} - 1)^{-\delta} + (v^{-\theta} - 1)^{-\delta}]^{-1/\delta})^{-1/\theta}$	$\theta \geq 0, \delta > 0$
BB5	$\exp\{-[x^\theta + y^\theta - (x^{-\theta\delta} + y^{-\theta\delta})^{-1/\delta}]^{1/\theta}\}, x = -\log(u), y = -\log(v)$	$\theta \geq 1, \delta > 0$
BB7	$1 - (1 - [(1 - \bar{u}^\theta)^{-\delta} + (1 - \bar{v}^\theta)^{-\delta} - 1]^{-1/\delta})^{1/\theta}, \bar{u} = 1 - u, \bar{v} = 1 - v$	$\theta \geq 1, \delta > 0$
BB10	$uv[1 - \delta(1 - u^\theta)(1 - v^\theta)]^{-1/\theta}$	$\theta > 0, 0 \leq \delta \leq 1$

Table A2. Tail dependence measures of some parametric copulas.

Copula	K_L or λ_L	K_U or λ_U
One-parameter		
Gaussian/Normal	$K_L = 2/(1 + \rho)$	$K_U = K_L$
Gumbel (extreme value)	$K_L = 2^{1/\theta}$	$K_U = 2 - 2^{1/\theta}$
Clayton	$K_L = 2^{-1/\theta}$	$K_U = 2$
Frank	$K_L = 2$	$K_U = 2$
Joe/B5	$K_L = 2$	$\lambda_U = 2 - 2^{1/\theta}$
Galambos (extreme value)	$K_L = 2 - 2^{-1/\theta}$	$\lambda_U = 2^{-1/\theta}$
Two-parameters		
Student-t	$\lambda_L = \lambda_U$	$\lambda_U = 2T_{v+1}\left(-\sqrt{\frac{(v+1)(1-\rho)}{1+\rho}}\right)$
BB1	$\lambda_L = 2^{-1/\theta\delta}$	$\lambda_U = 2 - 2^{1/\delta}$
BB3	$\lambda_L = 1$	$\lambda_U = 2 - 2^{1/\theta}$
BB4	$\lambda_L = (2 - 2^{-1/\delta})^{-1/\theta}$	$\lambda_U = 2^{-1/\delta}$
BB5 (extreme value)	$K_L = 2 - \lambda_U$	$\lambda_U = 2 - (2 - 2^{-1/\delta})^{1/\theta}$
BB7	$\lambda_L = 2^{-1/\delta}$	$\lambda = 2 - 2^{1/\theta}$
BB10	$K_L = 2$	$K_U = 2$

A rank-based measurement of the dependence measure of the copula families can be computed either from a closed-form solution (if available) or numerical approximation via

a Monte Carlo simulation through two-variable integrals as in Equation (A1) for Kendall’s τ and Equation (A2) for Spearman ρ_s , respectively.

$$\hat{\tau}(C) = 1 - 4 \int_0^1 \int_0^1 C_{1|2}(u|v)C_{2|1}(v|u)dudv \tag{A1}$$

$$\hat{\rho}_s(C) = 3 - 12 \int_0^1 \int_0^1 uC_{2|1}(v|u)dudv \tag{A2}$$

Note that a closed-form Kendall’s $\hat{\tau}$ and Spearman $\hat{\rho}_s$ of Gaussian copula, t copula, Gumbel copula, Frank copula, and Clayton copula are used. Otherwise, we use Monte Carlo approximation.

Appendix B. Independence Test Results of the Standardized Residuals

This appendix discusses the Ljung–Box independence testing of the marginal model residuals of this study including ICAPM-EGARCH with mixture innovation and the competing ARMA-GARCH with Student- t innovation with the estimation method of rw-MH, variational Bayes and QMLE. For the 100 stocks, standardized residuals of all the proposed models indicated independence where $\alpha = 0.01$ meant failure to reject the null hypothesis that the residuals are independently distributed, except for the stock abbreviations ‘V’, ‘MA’, ‘TXN’, ‘QCOM’, ‘SCHW’ and ‘DEO’ in ICAPM-EGARCH(1,1)-Mixture-rw-MH model. For the ICAPM-EGARCH(1,1)-Mixture-VBDA0 model, the stock abbreviations are ‘V’, ‘TXN’, ‘QCOM’ and ‘DEO’. For the ICAPM-EGARCH(1,1)-Mixture-VBDA1 model, the stock abbreviations are ‘V’, ‘MA’, ‘TXN’, ‘QCOM’ and ‘DEO’. For ICAPM-EGARCH(1,1)-Mixture-VBDA2 model, the stock abbreviations are ‘V’, ‘TXN’, ‘QCOM’ and ‘DEO’. For ICAPM-EGARCH(1,1)-Mixture-QMLE model, the stock abbreviations are ‘V’, ‘MA’, ‘TXN’, ‘QCOM’, ‘SCHW’ and ‘DEO’. For ARMA(1,1)-GARCH(1,1)-t-QMLE model, the stock abbreviations are ‘QCOM’ and ‘HON’.

Appendix C. List of 100 Nasdaq Stocks Used in the Study (The Last 30 Stocks Used in the 30-Dimensional Analysis)

No	Symbol	Name	Sector
1	AAPL	Apple Inc. Common Stock	Technology
2	MSFT	Microsoft Corporation Common Stock	Technology
3	GOOG	Alphabet Inc. Class C Capital Stock	Technology
4	GOOGL	Alphabet Inc. Class A Common Stock	Technology
5	AMZN	Amazon.com Inc. Common Stock	Consumer Services
6	MCHP	Microchip Technology Incorporated Common Stock	Technology
7	TSM	Taiwan Semiconductor Manufacturing Company Ltd.	Technology
8	NVDA	NVIDIA Corporation Common Stock	Technology
9	V	Visa Inc.	Miscellaneous
10	JPM	JP Morgan Chase & Co. Common Stock	Finance
11	IDXX	IDEXX Laboratories Inc. Common Stock	Health Care
12	JNJ	Johnson & Johnson Common Stock	Health Care
13	WMT	Walmart Inc. Common Stock	Consumer Services
14	UNH	United Health Group Incorporated Common Stock (DE)	Health Care
15	ADI	Analog Devices Inc. Common Stock	Technology
16	HD	Home Depot Inc. (The) Common Stock	Consumer Services
17	ASML	ASML Holding N.V. New York Registry Shares	Technology
18	PG	Procter & Gamble Company (The) Common Stock	Consumer Non-Durables

19	BAC	Bank of America Corporation Common Stock	Finance
20	MA	Mastercard Incorporated Common Stock	Miscellaneous
21	KLAC	KLA Corporation Common Stock	Capital Goods
22	DIS	Walt Disney Company (The) Common Stock	Consumer Services
23	ADBE	Adobe Inc. Common Stock	Technology
24	CMCSA	Comcast Corporation Class A Common Stock	Consumer Services
25	NFLX	Netflix Inc. Common Stock	Consumer Services
26	CRM	Salesforce.com Inc Common Stock	Technology
27	PFE	Pfizer Inc. Common Stock	Health Care
28	TM	Toyota Motor Corporation Common Stock	Capital Goods
29	NKE	Nike Inc. Common Stock	Consumer Non-Durables
30	CSCO	Cisco Systems Inc. Common Stock (DE)	Technology
31	ORCL	Oracle Corporation Common Stock	Technology
32	KO	Coca-Cola Company (The) Common Stock	Consumer Non-Durables
33	TMO	Thermo Fisher Scientific Inc Common Stock	Capital Goods
34	DHR	Danaher Corporation Common Stock	Health Care
35	NVO	Novo Nordisk A/S Common Stock	Health Care
36	XOM	Exxon Mobil Corporation Common Stock	Energy
37	VZ	Verizon Communications Inc. Common Stock	Public Utilities
38	LLY	Eli Lilly and Company Common Stock	Health Care
39	ABT	Abbott Laboratories Common Stock	Health Care
40	INTC	Intel Corporation Common Stock	Technology
41	NTES	NetEase Inc. American Depositary Shares	Miscellaneous
42	PEP	PepsiCo Inc. Common Stock	Consumer Non-Durables
43	ACN	Accenture plc Class A Ordinary Shares (Ireland)	Technology
44	COST	Costco Wholesale Corporation Common Stock	Consumer Services
45	T	AT&T Inc.	Consumer Services
46	WFC	Wells Fargo & Company Common Stock	Finance
47	NVS	Novartis AG Common Stock	Health Care
48	CVX	Chevron Corporation Common Stock	Energy
49	MRK	Merck & Company Inc. Common Stock (new)	Health Care
50	AZN	AstraZeneca PLC American Depositary Shares	Health Care
51	MS	Morgan Stanley Common Stock	Finance
52	MCD	McDonald's Corporation Common Stock	Consumer Services
53	TXN	Texas Instruments Incorporated Common Stock	Technology
54	MDT	Medtronic plc. Ordinary Shares	Health Care
55	UPS	United Parcel Service Inc. Common Stock	Transportation
56	SAP	SAP SE ADS	Technology
57	NEE	NextEra Energy Inc. Common Stock	Public Utilities
58	PM	Philip Morris International Inc Common Stock	Consumer Non-Durables
59	TMUS	T-Mobile US Inc. Common Stock	Public Utilities
60	LIN	Linde plc Ordinary Share	Basic Industries
61	INTU	Intuit Inc. Common Stock	Technology
62	QCOM	QUALCOMM Incorporated Common Stock	Technology
63	HON	Honeywell International Inc. Common Stock	
64	LOW	Lowe's Companies Inc. Common Stock	Consumer Services
65	UL	Unilever PLC Common Stock	Consumer Non-Durables
66	RY	Royal Bank Of Canada Common Stock	

67	ILMN	Illumina Inc. Common Stock	Health Care
68	BHP	BHP Group Limited American Depositary Shares (Each representing two Ordinary Shares)	Basic Industries
69	C	Citigroup Inc. Common Stock	Finance
70	SONY	Sony Group Corporation American Depositary Shares	Consumer Non-Durables
71	HDB	HDFC Bank Limited Common Stock	Finance
72	BMJ	Bristol-Myers Squibb Company Common Stock	Health Care
73	AMT	American Tower Corporation (REIT) Common Stock	Finance
74	SBUX	Starbucks Corporation Common Stock	
75	BLK	BlackRock Inc. Common Stock	Finance
76	SCHW	Charles Schwab Corporation (The) Common Stock	Finance
77	UNP	Union Pacific Corporation Common Stock	Transportation
78	BBL	BHP Group PlcSponsored ADR	Energy
79	AXP	American Express Company Common Stock	Finance
80	GS	Goldman Sachs Group Inc. (The) Common Stock	Finance
81	RTX	Raytheon Technologies Corporation Common Stock	Capital Goods
82	AMD	Advanced Micro Devices Inc. Common Stock	Technology
83	BA	Boeing Company (The) Common Stock	Capital Goods
84	AMAT	Applied Materials Inc. Common Stock	Capital Goods
85	AMGN	Amgen Inc. Common Stock	Health Care
86	ISRG	Intuitive Surgical Inc. Common Stock	Health Care
87	IBM	International Business Machines Corporation Common Stock	Technology
88	SNY	Sanofi ADS	Health Care
89	TGT	Target Corporation Common Stock	Consumer Services
90	TD	Toronto Dominion Bank (The) Common Stock	
91	TTE	TotalEnergies SE	Energy
92	EL	Estee Lauder Companies Inc. (The) Common Stock	Consumer Non-Durables
93	CVS	CVS Health Corporation Common Stock	Health Care
94	GE	General Electric Company Common Stock	Consumer Durables
95	DEO	Diageo plc Common Stock	Consumer Non-Durables
96	SPGI	S&P Global Inc. Common Stock	Technology
97	RIO	Rio Tinto Plc Common Stock	Basic Industries
98	JBHT	J.B. Hunt Transport Services Inc. Common Stock	Transportation
99	DE	Deere & Company Common Stock	Capital Goods
100	CAT	Caterpillar Inc. Common Stock	Capital Goods

References

1. Sklar, M. Fonctions de repartition an dimensions et leurs marges. *Publ. L'Institut Stat. L'Universit'E Paris* **1959**, *8*, 229–231.
2. Achite, M.; Bazrafshan, O.; Pakdaman, Z.; Simsek, O.; Caloiero, T. Multivariate uncertainty analysis of severity-duration-magnitude frequency curves using Khoudraji copula and bootstrap method. *Hydrol. Sci. J.* **2025**. [[CrossRef](#)]
3. Deng, T.; Zeng, Z.; Xu, J. A copula-based statistical method of estimating production profile from temperature measurement for perforated horizontal wells. *Pet. Sci. Technol.* **2024**, *42*, 4589–4609. [[CrossRef](#)]
4. Zhou, C.; van Nooijen, R.; Kolechkina, A.; Gargouri, E.; Slama, F.; van de Giesen, N. The uncertainty associated with the use of copulas in multivariate analysis. *Hydrol. Sci. J.* **2023**, *68*, 2169–2188. [[CrossRef](#)]
5. Aghatise, O.; Khan, F.; Ahmed, S. Reliability assessment of marine structures considering multidimensional dependency of the variables. *Ocean. Eng.* **2021**, *230*, 109021. [[CrossRef](#)]
6. Chen, P.; Zhang, J. Copula-based multivariate control charts for monitoring multiple dependent Weibull processes. *Commun. Stat.-Simul. Comput.* **2025**, *0*, 1–22. [[CrossRef](#)]

7. Farkas, S.; Lopez, O. Semiparametric copula models applied to the decomposition of claim amounts. *Scand. Actuar. J.* **2024**, *2024*, 1065–1092. [[CrossRef](#)]
8. Wang, Q.; Wu, X. A Nonparametric Bayesian Estimator of Copula Density with Applications to Financial Market. *J. Bus. Econ. Stat.* **2025**, *43*, 1021–1033. [[CrossRef](#)]
9. Kim, J.M.; Ha, I.D.; Kim, S. Copula deep learning control chart for multivariate zero inflated count response variables. *Statistics* **2024**, *58*, 749–769. [[CrossRef](#)]
10. McNeil, A.J.; Frey, R.; Embrechts, P. *Quantitative Risk Management: Concepts, Techniques and Tools*; Princeton Series in Finance; Princeton University Press: Princeton, NJ, USA, 2015.
11. Wichitaksorn, N.; Gerlach, R.; Choy, S.T.B. Efficient MCMC estimation of some elliptical copula regression models through scale mixtures of normals. *Appl. Stoch. Model. Bus. Ind.* **2019**, *35*, 808–822. [[CrossRef](#)]
12. Górecki, J.; Hofert, M.; Okhrin, O. Outer power transformations of hierarchical Archimedean copulas: Construction, sampling and estimation. *Comput. Stat. Data Anal.* **2021**, *155*, 107109. [[CrossRef](#)]
13. Liang, P.; Rosen, O. Flexible Bayesian estimation of elliptical copulas. *J. Comput. Graph. Stat.* **2025**, *0*, 1–18. [[CrossRef](#)]
14. Michaelides, M.; Cossette, H.; Pigeon, M. Simulations of Bivariate Archimedean Copulas from Their Nonparametric Generators for Loss Reserving under Flexible Censoring. *N. Am. Actuar. J.* **2025**, *0*, 1–28. [[CrossRef](#)]
15. Aas, K.; Czado, C.; Frigessi, A.; Bakken, H. Pair-copula constructions of multiples dependence. *Insur. Math. Econ.* **2009**, *44*, 182–198. [[CrossRef](#)]
16. Kurowicka, D.; Cooke, R.M. Distribution-free continuous Bayesian belief nets. In *Modern Statistical and Mathematical Methods in Reliability*; World Scientific: Singapore, 2005; Volume 10, pp. 309–322.
17. Joe, H. *Dependence Modelling with Copulas*; Monographs on Statistics and Applied Probability: 134; CRC Press: Boca Raton, FL, USA, 2015.
18. Müller, D.; Czado, C. Representing sparse Gaussian DAGs as sparse R-vines Allowing for Non-Gaussian Dependence. *J. Comput. Graph. Stat.* **2018**, *27*, 334–344. [[CrossRef](#)]
19. Müller, D.; Czado, C. Dependence modelling in ultra high dimensions with vine copulas and the Graphical Lasso. *Comput. Stat. Data Anal.* **2019**, *137*, 211–232. [[CrossRef](#)]
20. Brechmann, E.C.; Czado, C.; Aas, K. Truncated regular vines in high dimensions with application to financial data. *Can. J. Stat.* **2012**, *40*, 68–85. [[CrossRef](#)]
21. Maya, R.A.L.; Gomez-Gonzalez, J.E.; Velandia, L.F.M. Latin American exchange rate dependencies: A regular vine copula approach. *Contemp. Econ. Policy* **2015**, *33*, 535–549. [[CrossRef](#)]
22. Li, H.; Liu, Z.; Wang, S. Vines climbing higher: Risk management for commodity futures markets using a regular vine copula approach. *Int. J. Financ. Econ.* **2022**, *27*, 2438–2457. [[CrossRef](#)]
23. Xiao, Q.; Yan, M.; Zhang, D. Commodity market financialization, herding and signals: An asymmetric GARCH R-vine copula approach. *Int. Rev. Financ. Anal.* **2023**, *89*, 102743. [[CrossRef](#)]
24. Čeryová, B.; Árendáš, P. Vine copula approach to the intra-sectoral dependence analysis in the technology industry. *Financ. Res. Lett.* **2024**, *60*, 104889. [[CrossRef](#)]
25. Hamza, T.; Ben Haj Hamida, H.; Mili, M.; Sami, M. High inflation during Russia–Ukraine war and financial market interaction: Evidence from C-Vine Copula and SETAR models. *Res. Int. Bus. Financ.* **2024**, *70*, 102384. [[CrossRef](#)]
26. Czado, C. Pair-Copula Constructions of Multivariate Copulas. In *Copula Theory and Its Applications*; Jaworski, P., Durante, F., Härdle, W.K., Rychlik, T., Eds.; Springer: Berlin/Heidelberg, Germany, 2010; pp. 93–109.
27. Joe, H.; Li, H.; Nikoloulopoulos, A.K. Tail dependence functions and vine copulas. *J. Multivar. Anal.* **2010**, *101*, 252–270. [[CrossRef](#)]
28. Patton, A.J. A review of copula models for economic time series. *J. Multivar. Anal.* **2012**, *110*, 4–18. [[CrossRef](#)]
29. Fan, Y.; Patton, A.J. Copulas in econometrics. *Annu. Rev. Econ.* **2014**, *6*, 179–200. [[CrossRef](#)]
30. Dißmann, J.; Brechmann, E.C.; Czado, C.; Kurowicka, D. Selecting and estimating regular vine copulae and application to financial returns. *Comput. Stat. Data Anal.* **2013**, *59*, 52–69. [[CrossRef](#)]
31. Vuong, Q.H. Likelihood ratio tests for model selection and non-nested Hypotheses. *Econometrica* **1989**, *57*, 307–333. [[CrossRef](#)]
32. Merton, R.C. An intertemporal capital asset pricing model. *Econometrica* **1973**, *41*, 867–887. [[CrossRef](#)]
33. Sun, F.; Fu, F.; Liao, H.; Xu, D. Analysis of multivariate dependent accelerated degradation data using a random-effect general Wiener process and D-vine Copula. *Reliab. Eng. Syst. Saf.* **2020**, *204*, 107168. [[CrossRef](#)]
34. Nagler, T.; Bumann, C.; Czado, C. Model selection in sparse high-dimensional vine copula models with an application to portfolio risk. *J. Multivar. Anal.* **2019**, *172*, 180–192. [[CrossRef](#)]
35. Chang, B.; Joe, H. Prediction based on conditional distributions of vine copulas. *Comput. Stat. Data Anal.* **2019**, *139*, 45–63. [[CrossRef](#)]
36. Mantel, N.; Myers, M. Problems of convergence of maximum likelihood iterative procedures in multiparameter situations. *J. Am. Stat. Assoc.* **1971**, *66*, 484–491. [[CrossRef](#)]
37. Loaiza-Maya, R.; Smith, M.S. Variational Bayes estimation of discrete-margined copula models with application to time series. *J. Comput. Graph. Stat.* **2019**, *28*, 523–539. [[CrossRef](#)]

38. Chang, C.L.; McAleer, M.; Wong, W.K. Big data, computational science, economics, finance, marketing, management, and psychology: Connections. *J. Risk Financ. Manag.* **2018**, *11*, 15. [CrossRef]
39. Oh, D.H.; Patton, A.J. High-dimensional copula-based distributions with mixed frequency data. *J. Econom.* **2016**, *193*, 349–366. [CrossRef]
40. Pareek, M.K.; Thakkar, P. Surveying stock market portfolio optimization techniques. In Proceedings of the 2015 5th Nirma University International Conference on Engineering (NUiCONE), Ahmedabad, India, 26–28 November 2015; pp. 1–5.
41. Kolm, P.N.; Tütüncü, R.; Fabozzi, F.J. 60 Years of portfolio optimization: Practical challenges and current trends. *Eur. J. Oper. Res.* **2014**, *234*, 356–371. [CrossRef]
42. Bedford, T.; Cooke, R.M. Probability density decomposition for conditionally dependent random variables modeled by vines. *Ann. Math. Artif. Intell.* **2001**, *32*, 245–268. [CrossRef]
43. Bedford, T.; Cooke, R.M. Vines: A new graphical model for dependent random variables. *Ann. Stat.* **2002**, *30*, 1031–1068. [CrossRef]
44. Kurowicka, D.; Joe, H. (Eds.) *Dependence Modeling: Vine Copula Handbook*; World Scientific: Singapore, 2010.
45. Hafner, C.M.; Kyriakopoulou, D. Exponential-Type GARCH models with linear-in-variance risk premium. *J. Bus. Econ. Stat.* **2021**, *39*, 589–603. [CrossRef]
46. Nelson, D.B. Conditional heteroskedasticity in asset returns: A new approach. *Econometrica* **1991**, *59*, 347–370. [CrossRef]
47. Hentschel, L. All in the family nesting symmetric and asymmetric GARCH models. *J. Financ. Econ.* **1995**, *39*, 71–104. [CrossRef]
48. Jose Rodriguez, M.; Ruiz, E. Revisiting several popular GARCH models with leverage effect: Differences and similarities. *J. Financ. Econom.* **2012**, *10*, 637–668. [CrossRef]
49. Sahamkhadam, M.; Stephan, A.; Östermark, R. Portfolio optimization based on GARCH-EVT-Copula forecasting models. *Int. J. Forecast.* **2018**, *34*, 497–506. [CrossRef]
50. Engle, R. Risk and volatility: Econometric models and financial practice. *Am. Econ. Rev.* **2004**, *94*, 405–420. [CrossRef]
51. Hosking, J.R.M.; Wallis, J.R. Parameter and Quantile Estimation for the generalized Pareto Distribution. *Technometrics* **1987**, *29*, 339–349. [CrossRef]
52. Singh, V.P.; Guo, H. Parameter estimation for 3-parameter generalized Pareto distribution by the principle of maximum entropy (POME). *Hydrol. Sci. J.* **1995**, *40*, 165–181. [CrossRef]
53. Koop, G.; Poirier, D.J.; Tobias, J.L. *Bayesian Econometric Methods*; Econometric Exercises; Cambridge University Press: Cambridge, UK, 2007.
54. Scherr, S.; Zhou, J. Automatically Identifying Relevant Variables for Linear Regression with the Lasso Method: A Methodological Primer for its Application with R and a Performance Contrast Simulation with Alternative Selection Strategies. *Commun. Methods Meas.* **2020**, *14*, 204–211. [CrossRef]
55. Nezamdoust, S.; Eskandari, F. Ridge Shrinkage Estimators in Finite Mixture of Generalized Estimating Equations. *J. Math. Model. Financ.* **2022**, *2*, 91–106.
56. Liang, C.; Xu, Y.; Chen, Z.; Li, X. Forecasting China’s stock market volatility with shrinkage method: Can Adaptive Lasso select stronger predictors from numerous predictors? *Int. J. Financ. Econ.* **2022**, *28*, 3689–3699. [CrossRef]
57. Al Janabi, M.A.M.; Arreola Hernandez, J.; Berger, T.; Nguyen, D.K. Multivariate dependence and portfolio optimization algorithms under illiquid market scenarios. *Eur. J. Oper. Res.* **2017**, *259*, 1121–1131. [CrossRef]
58. Yu, W.; Bondell, H.D. Variational Bayes for Fast and Accurate Empirical Likelihood Inference. *J. Am. Stat. Assoc.* **2024**, *119*, 1089–1101. [CrossRef]
59. Gunawan, D.; Tran, M.N.; Suzuki, K.; Dick, J.; Kohn, R. Computationally efficient Bayesian estimation of high-dimensional Archimedean copulas with discrete and mixed margins. *Stat. Comput.* **2019**, *29*, 933–946. [CrossRef]
60. Tran, M.N.; Nguyen, T.N.; Dao, V.H. A practical tutorial on variational Bayes. *arXiv* **2021**, arXiv:2103.01327. [CrossRef]
61. Blei, D.M.; Kucukelbir, A.; McAuliffe, J.D. Variational inference: A review for statisticians. *J. Am. Stat. Assoc.* **2017**, *112*, 859–877. [CrossRef]
62. Smith, M.S.; Loaiza-Maya, R.; Nott, D.J. High-dimensional copula variational approximation through transformation. *J. Comput. Graph. Stat.* **2020**, *29*, 729–743. [CrossRef]
63. Ong, V.M.H.; Nott, D.J.; Smith, M.S. Gaussian variational approximation with a factor covariance structure. *J. Comput. Graph. Stat.* **2018**, *27*, 465–478. [CrossRef]
64. Zeiler, M.D. ADADELTA: An Adaptive Learning Rate Method. *arXiv* **2012**, arXiv:1212.5701. [CrossRef]
65. Morales Napoles, O.; Cooke, R.M.; Kurowicka, D. About the Number of Vines and Regular Vines on n Nodes (Submitted for Publication). 2010. Available online: <https://resolver.tudelft.nl/uuid:912abf55-8112-48d2-9cca-323f7f6aacc7> (accessed on 17 November 2025).
66. Nikoloulopoulos, A.K.; Joe, H.; Li, H. Vine copulas with asymmetric tail dependence and applications to financial return data. *Comput. Stat. Data Anal.* **2012**, *56*, 3659–3673. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.