Review Article

Maximilian Coblenz*, Oliver Grothe, Bolin Liu and David Weniger

Copulas and deep learning: a review

https://doi.org/10.1515/demo-2025-0017
Received July 14, 2025; accepted January 14, 2026; published online July 13, 2026

Abstract: In the last two decades, there has been a surge in the research on neural networks, and particularly on deep learning. At the same time, copulas as a statistical modeling tool for multivariate distributions became more and more popular. We survey how copulas are used in neural networks and deep learning and vice versa how neural networks and deep learning are used in the copula domain. For example, we highlight that copulas can be constructed from generative deep learning models and that neural networks can help in goodness-of-fit assessment of copulas. Moreover, we discuss how copulas amplify specific deep learning models and how copulas are harnessed to combine neural network outputs.

Keywords: copula; deep learning; neural network; vine copula; machine learning

MSC 2020: 62H10; 68T07

1 Introduction

Since the seminal article of Elidan (2013) on copulas in machine learning, more than a decade passed. During this time, application areas of deep learning – i.e., neural networks with many hidden layers – such as text, image, and video processing as well as data generation surged. Taking a statistical perspective, the aim of this survey article is to review developments in neural networks and deep learning in the light of copulas. We believe that these two research areas are largely unaware of each other, particularly as far as the statistics community is concerned – with some notable exceptions that are presented here. Our aim is to discuss how copulas profit from deep learning and vice versa. We hope to spark interest in researching the intersection of copulas and deep learning particularly stemming from the statistics community. On the one hand, we think that this is a fruitful avenue and on the other hand the field still seems to be in its infancy. We want to point out that much of the research presented relies on building and tuning specific neural networks. It is not our goal to discuss these in detail but rather to give the reader a rough understanding of how neural network concepts are applied to copulas and vice versa. Additionally, we do not aim to review deep learning models but rather their interplay with copulas. Thus, not all the latest deep learning architectures might be represented by the review and the detail in which we discuss a specific deep learning architecture is determined by the literature that already exists in conjunction with copulas.

Copulas are multivariate distribution functions with uniform margins on $[0, 1]$ that allow to model marginal distributions and dependence structure of multivariate random variables separately. Since arbitrary marginal distributions can be combined with a copula to form a valid multivariate distribution function, copulas are a

***Corresponding author: Maximilian Coblenz**, Department of Services and Consulting, University of Business and Society Ludwigshafen, Ernst-Boehe-Str. 4, 67059 Ludwigshafen, Germany, E-mail: maximilian.coblenz@hwg-lu.de
Oliver Grothe and Bolin Liu, Institute of Operations Research, Analytics and Statistics, Karlsruhe Institute of Technology, Kaiserstr. 12, 76131 Karlsruhe, Germany
David Weniger, Department of Services and Consulting, University of Business and Society Ludwigshafen, Ernst-Boehe-Str. 4, 67059, Ludwigshafen, Germany

powerful tool from a practical perspective. As such, they are applied in a wide range of areas such as finance (Aas 2016; Chollete et al. 2009; Cousin and Di Bernardino 2013; Czado et al. 2022; Patton 2006; Puccetti et al. 2016), hydrology (Salvadori and De Michele 2004; Salvadori 2004; Salvadori et al. 2016; Coblenz et al. 2018a,b), and engineering (Coblenz et al. 2020; Jiang et al. 2015; Salvadori et al. 2014, 2015; Schepsmeier and Czado 2016).

The usefulness of copulas was also recognized in the wider field of machine learning. In the last two decades, many developments in copulas in machine learning were made. For example, copulas are used in various regression techniques (Chang and Joe 2019; De Backer et al. 2017; Dette et al. 2014; Kraus and Czado 2017; Noh et al. 2013; Noh et al. 2015; Su et al. 2019), classification (Han et al. 2013; Mesfioui et al. 2023; Ozdemir et al. 2018), graphical models (Bauer and Czado 2016; Bauer et al. 2012; Dobra and Lenkoski 2011; Eban et al. 2013; Elidan 2010a,b; Elidan 2012a,b; Huang and Frey 2011; Karra and Mili 2016; Liu et al. 2012; Müller and Czado 2018; Pircalabelu et al. 2017; Silva et al. 2011; Tenzer and Elidan 2013, 2014; Veeramachaneni et al. 2015), mixture models and clustering (Burda and Prokhorov 2014; Fujimaki et al. 2011; Kim et al. 2013; Kirshner 2007; Rajan and Bhattacharya 2016; Rey and Roth 2012; Roy and Parui 2014; Sahin and Czado 2022; Silva and Gramacy 2009), variants of independent component analysis (Chen et al. 2015; Kirshner and Póczos 2008; Ma and Sun 2007), variational inference (Chi et al. 2022; Han et al. 2016; Hirt et al. 2019; Smith and Loaiza-Maya 2023; Tran et al. 2015), and other topics such as reinforcement learning (Kim 2025b,d; Wang et al. 2021b), transfer learning (Lopez-Paz et al. 2012; Salinas et al. 2020), and outlier detection (Li et al. 2020) to name a few. However, in this article we focus on neural networks and deep learning and do not treat machine learning as a whole, because this would simply exceed the scope of a review.

In order to find references primarily from the neural network community we were not aware of, we employed a loose search strategy. Particularly, we did a keyword search for “copula” and “dependence” in the titles and abstracts of major machine learning conferences and, additionally, in purely neural network related conferences. Moreover, we did this for major machine learning journals. Where possible, we considered proceedings and issues with a scope from 2000 until summer 2025. A full list of journals and conferences considered can be found in Appendix A. Finally, we did a Google Scholar search for “copula and deep learning” and went through the first 20 hit pages. We hasten to say that we make no claim to completeness but rather present the work we are aware of and that passed our check for soundness (also, see the conclusion) while keeping a clear scope to the article.

The article is structured as follows: In Section 2 we give concise theoretical introductions to both copulas and deep learning. Section 3 reviews how neural networks and deep learning are used in the domain of copulas, e.g., to construct new copulas. Complementing the latter, in Section 4 we discuss the use of copulas in neural network and deep learning models, e.g., to enhance generating capabilities of autoencoders. Throughout, some prominent deep learning architectures are briefly addressed and we provide references for in-depth treatments of these. Finally, we provide concluding remarks and discuss future research directions in Section 5. Some technical details can be found in Appendix B. A table grouping references by some overarching topics we thought might be interesting for some readers can be found in Appendix C.

2 Theoretical background

This section provides a brief theoretical introduction to both copulas and deep learning. First, we introduce the copula concept and touch on multivariate copula constructions and copula estimation. Second, we treat some basics of deep learning. Specific deep learning architectures are briefly introduced as needed in the subsequent sections. In this section, the two topics copulas and deep learning are treated separately and for the familiar reader the respective subsections are safe to skip.

2.1 Copulas

Throughout let $\mathbf{X} = \left(X_1, X_2, \ldots, X_d\right)$ be a d -dimensional random variable with distribution function $F_{\mathbf{X}}$ and marginal distribution functions F_i, $i = 1, \ldots, d$. According to Sklar's Theorem (Sklar 1959), $F_{\mathbf{X}}$ can be decomposed in the following way

$$F_{\mathbf{X}} = C\big(F_1(x_1), F_2(x_2), \ldots, F_d(x_d)\big), \tag{1}$$

where C is a copula. For continuous $\mathbf{X}$, the copula is unique. For discrete or mixed $\mathbf{X}$, the decomposition is unique on the image of $F_1(X_1), F_2(X_2), \ldots, F_d(X_d)$. Moreover, the second part of Sklar's Theorem (Sklar 1959) states that any Copula C and any $F_1(x_1), \ldots, F_d(x_d)$ may be combined as in Equation (1) to yield a valid joint distribution function of $\mathbf{X}$.

Equation (1) shows that the complete dependence structure of $\mathbf{X}$ is captured by the copula C. Furthermore, every copula is a multivariate distribution with uniform marginal distributions and vice versa. We want to point out that this is of utmost importance. As will be seen particularly in Section 3, some approaches in the deep learning literature see copulas as some kind of derivation trick that can be applied whenever it suits the method. Yet, having uniform marginal distributions is an essential property of copulas and not ensuring them delivers something else. This is especially relevant for vine copulas (see below and Appendix B) where non-copula like behavior on one level is propagated to all subsequent levels.

Copulas offer a general way to model, to construct, and to estimate a multivariate distribution since dealing with marginal distributions and dealing with the overall dependence structure is decoupled. Therefore, many specific copulas and functional copula families were developed and in the following we present some of the most well-known. For an in-depth theoretical introduction of copulas, the interested reader is referred to the excellent books by Nelsen (2006), Joe (2015), and Durante and Sempi (2015). From an applied perspective, Genest and Favre (2007) is a brilliant starting point.

To ease notation, let $u_i = F_i(x_i)$, $i = 1, \ldots, d$, for the remainder of the paper. The concept of stochastic independence is represented by the independence copula Π, also known as the product copula, where the copula function is defined as the product of its arguments:

$$\Pi(u_1, \ldots, u_d) = u_1 \cdots u_d. \tag{2}$$

This formula is intuitive to understand since the joint distribution function of independent random variables is the product of the marginal distributions.

From Equation (1), we can *extract* copulas of continuous random vectors by substituting the marginal quantile functions F_i^{-1}, $i = 1, \ldots, d$, into $F_{\mathbf{X}}$, i.e.,

$$C(u_1, \ldots, u_d) = F_{\mathbf{X}}\big(F_1^{-1}(u_1), \ldots, F_d^{-1}(u_d)\big). \tag{3}$$

This allows, e.g., for the derivation of the Gaussian copula and t-copula as the dependence structure of the multivariate normal distribution and the multivariate Student t-distribution, respectively. Denoting Φ_d as the d-dimensional standard normal cumulative distribution function (CDF) and Φ^{-1} as the inverse of the univariate standard normal CDF, along with $\mathbf{\Sigma}$ as the $d \times d$ dispersion matrix, the Gaussian copula is expressed as (Joe 2015)

$$C(u_1, \ldots, u_d) = \Phi_d\Big(\Phi^{-1}(u_1), \ldots, \Phi^{-1}(u_d); \mathbf{\Sigma}\Big). \tag{4}$$

Similarly, let $t_{d,\nu}$ be the d-dimensional Student-t CDF and t_ν^{-1} be the inverse of the univariate Student-t CDF, where ν represents the degrees of freedom parameter. The t-copula is defined by (Joe 2015)

$$C(u_1, \ldots, u_d) = t_{d,\nu}\big(t_\nu^{-1}(u_1), \ldots, t_\nu^{-1}(u_d); \mathbf{\Sigma}, \nu\big). \tag{5}$$

Besides the mentioned ones, there are many bivariate copula families available (for an overview, see, e.g., Nelsen (2006); Joe (2015). However, unlike the Gaussian, the t-, or Archimedean copulas (see Section 3.1) only few of them are extensible to dimensions greater than 2. Additionally, copula families that are available in more than two dimensions are usually inflexible in terms of the dependence structures they can model – often they have the property that all pairwise margins are of the same kind. Therefore, copula construction methods were developed, of which the vine copula will come up several times throughout the article. The interested reader can find more on vine copulas, e.g., in Aas (2016); Czado (2019); Czado and Nagler (2022), and Appendix B.

Lastly, we discuss the estimation of copulas. Copulas can be estimated in different fashions. Let $\mathbf{X}_1, \ldots, \mathbf{X}_n$ be an i.i.d. sample of $\mathbf{X}$ and let x_{ij} be the $i-$ th component of $\mathbf{X}_j$. The copula C of $\mathbf{X}$ may be estimated based on the so called pseudo-observations $\hat{\mathbf{U}}_j = (\hat{u}_{1j}, \ldots, \hat{u}_{dj})$, $j = 1, \ldots, n$. By specifying parametric distributions for each marginal variable, these can be obtained by

$$\hat{\mathbf{U}}_j = \left(\hat{F}_1(x_{1j}), \ldots, \hat{F}_d(x_{dj})\right), \tag{6}$$

where $\hat{F}_i(x)$ is an estimator of $F_i(x)$. By additionally estimating the copula parametrically, a fully parametric estimation procedure is obtained, also called inference functions for margins (Joe and Xu 1996).

Alternatively, the marginal distributions can be estimated non-parametrically by the empirical distribution function – or by ranks – as

$$\hat{\mathbf{U}}_j = \frac{1}{n+1}\mathbf{R}_j, \tag{7}$$

where $\mathbf{R}_j$ is the vector of componentwise ranks of $\mathbf{X}_j$ in $\mathbf{X}_1, \ldots, \mathbf{X}_n$ and, thus, contains numbers in $[1, n]$. Using a parametric copula, this yields a semi-parametric procedure that is usually employed in practice (Genest and Favre 2007). A fully non-parametric procedure can be implemented using, e.g., the so-called empirical copula, which is the empirical distribution of the pseudo-observations (Genest and Favre 2007)

$$\hat{C}(\mathbf{u}) = \frac{1}{n}\sum_{j=1}^{n} \mathbf{1}_{\{\hat{\mathbf{U}}_j \leq \mathbf{u}\}}. \tag{8}$$

The next section gives a short introduction to deep learning.

2.2 Deep learning

Deep learning is a term to qualify artificial neural networks with many hidden layers. In essence, an artificial neural network is a composite function, which can be depicted as a network of connected nodes. The simplest form of a neural network is the so-called feedforward network, which consists of several layers of nodes and the inputs of nodes in one layer are the outputs of the nodes in the previous layer, cf. Figure 1. Let $\mathbf{h}_0 := \mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_d \end{pmatrix} \in \mathbb{R}^d$ be the input layer. Then, the output $\mathbf{h}_\ell \in \mathbb{R}^m$ of layer ℓ is computed as

$$\mathbf{h}_\ell = \sigma\left(\mathbf{W}_\ell \mathbf{h}_{\ell-1} + \mathbf{b}_\ell\right), \tag{9}$$

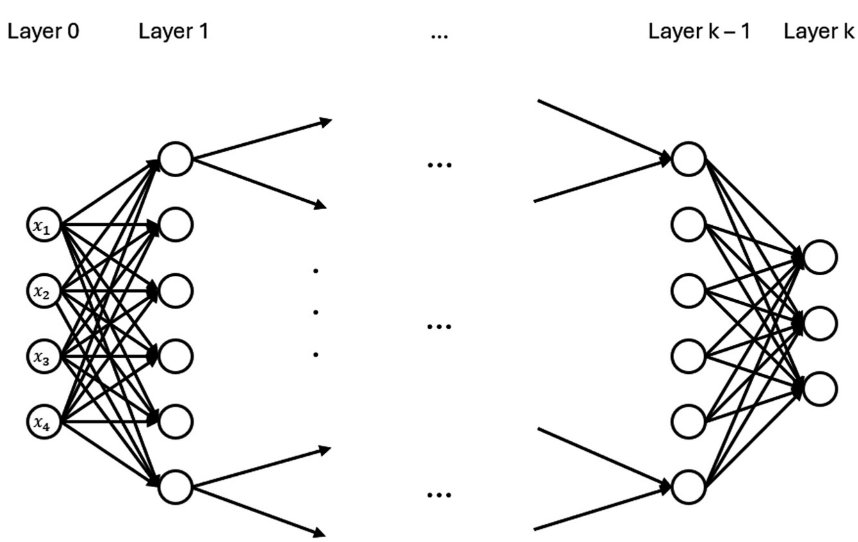

Figure 1: An exemplary feedforward neural network with four inputs x_1, x_2, x_3, x_4, $k+1$ layers, and three outputs in layer k.

where $\mathbf{h}_{\ell-1} \in \mathbb{R}^n$ are the outputs of the previous layer, $\mathbf{W}_\ell$ is an $m \times n$ weight matrix, $\mathbf{b}_\ell$ is an $m \times 1$ vector of bias terms, and $\sigma(\cdot)$ is the so-called activation function. There are many possible choices for $\sigma(\cdot)$ with ReLU $\sigma(x) = \max(0, x)$, sigmoid $\sigma(x) = \frac{1}{1+e^{-x}}$, tanh $\sigma(x) = \tanh(x)$, and softmax $\sigma_i(\mathbf{x}) = \frac{e^{x_i}}{\sum_{i=1}^d e^{x_i}}$, which is computed componentwise for vector $\mathbf{x}$, among the most popular ones. An overview of activation functions can be found, e.g. in Goodfellow et al. (2016).

In order to determine the weights and bias terms, neural networks are usually trained by backpropagation, which is a form of gradient descent minimizing a loss function. The loss function depends on the specific task. The interested reader is referred to Goodfellow et al. (2016) for a more detailed theoretical introduction and to Chollet (2021) for an applied perspective. Additionally, Schmidhuber (2015) and Nalisnick et al. (2023) provide excellent reviews of deep learning.

One of the strengths of neural networks is that relatively simple computing blocks can be combined into complex models and architectures that are capable of various tasks such as image, text, or audio processing. Recently, neural networks also became famous publicly as generative models, e.g., ChatGPT. The following sections shed light on how neural networks and deep learning are used in copulas and vice versa. Where needed, we briefly introduce the deep learning architectures that are dealt with.

3 Deep learning in copulas

This section gives an overview of how neural networks and deep learning are used to create and estimate copulas. We discuss, e.g., how Archimedean copulas are estimated with deep learning in Section 3.1 and how normalizing flows can estimate copulas from data in Section 3.2. Also, other deep learning architectures such as Generative Moment Matching networks can estimate copulas, see Section 3.3. How to alter the loss function of a neural network in order to obtain a valid copula is reviewed in Section 3.4. Finally, we address some further topics such as obtaining the vine copula structure from neural networks in Section 3.5. An overview of the cited works with some additional metadata can be found in Table 1.

3.1 Estimating Archimedean copulas with deep learning

A prominent family of copulas comprise the so-called Archimedean copulas. They are theoretically appealing and easy to handle, which makes them popular. Archimedean copulas follow the law (Joe 2015)

$$C(u_1, \dots, u_d) = \psi\left(\psi^{-1}(u_1) + \cdots + \psi^{-1}(u_d)\right), \tag{10}$$

where ψ is the so-called generator, or generator function, and ψ^{-1} is its (pseudo)-inverse. A generator ψ is a continuous, non-increasing function that maps $[0, \infty]$ to $[0, 1]$. It satisfies $\psi(0) = 1$ and $\psi(\infty) = 0$. Furthermore, it is strictly decreasing on $[0, \inf\{t\colon \psi(t) = 0\}]$. It is important to note that if and only if the generator ψ is completely monotone, i.e., $(-1)^k \frac{d^k}{du^k}\psi(u) \geq 0, k \in \mathbb{N}$, then it is the Laplace-Stieltjes transform (LST) of a positive random variable (Hofert 2008; Joe 2015). Parametric families of valid generator functions lead to subfamilies of Archimedean Copulas, e.g., the Clayton, Gumbel and Frank copulas. For an in-depth discussion of Archimedean copulas see, e.g., Hofert (2008); Joe (2015). Note that Archimedean copulas can be nested by imposing further restrictions on the generator functions involved. This leads to hierarchical Archimedean copulas, see, e.g., Hofert (2011).

Example 1. The Clayton copula (Hofert 2008; Joe 2015) has generator $\psi(u) = (1 + u)^{-\frac{1}{\theta}}$, where $\theta \in (0, \infty)$ is its parameter. The copula itself is given by $C(u_1, \dots, u_d) = \left(u_1^{-\theta} + \cdots + u_d^{-\theta} - (d-1)\right)^{-\frac{1}{\theta}}$. Furthermore, $\psi(0) = 1$, $\psi(\infty) = 0$, and $\frac{d}{du}\psi(u) = -\frac{1}{\theta}(1 + u)^{-\frac{1}{\theta}-1} < 0$ for $\theta > 0$ and $u \in [0, 1]$. One can show that $\psi(u)$ is also an LST.

One strand of literature in the deep learning community deals with estimating Archimedean copulas with neural networks. In Ling et al. (2020), the connection of the generator ψ of an Archimedean copula to LSTs is used.

Table 1: Overview of references for deep leaning in copulas sorted by publication year.

Reference	Deep learning architecture	Copula family	Application	Related references/ applied in
Wiese et al. (2019)	Normalizing flow	Any	Copula estimation	Mitskopoulos et al. (2022) McDonald et al. (2022) Liu et al. (2025a)
Sun et al. (2019)	LSTM	Vine	Vine structure estimation	
Ling et al. (2020)	Feedforward	Archimedean	Copula estimation	Ng et al. (2021) Zhang et al. (2024b) Tian et al. (2024)
Zhou et al. (2020)	Autoencoder	Vine	Vine copula regression Chemical engineering	Cui (2021)
Ng et al. (2021)	Feedforward, GAN	Archimedean Hierarchical Archimedean	Copula estimation	
Kamthe et al. (2021)	Normalizing flow	Any	Copula estimation	McDonald et al. (2022) Liu et al. (2025a)
Hofert et al. (2021)	GMM	Any	Quasi-random sampling	Hofert et al. (2023a) Hofert et al. (2023b)
Janke et al. (2021)	GMM	Any	Copula estimation	Huk et al. (2025)
Hasan et al. (2022)	Feedforward	Extreme-value	Copula estimation	
McDonald et al. (2022)	Normalizing flow	Any	Copula estimation	Liu et al. (2025a)
Mitskopoulos et al. (2022)	Normalizing flow	Vine	Copula estimation	Liu et al. (2025a)
Zeng and Wang (2022)	Feedforward	Any	Copula estimation	Li et al. (2025)
Chen et al. (2022)	GAN	Vine	Vine copula regression Chemical engineering	
Hofert et al. (2023a)	GMM	Any	Goodness-of-fit assessment	
Hofert et al. (2023b)	GMM	Any	Regression	
Zhang et al. (2024b)	Feedforward	Archimedean	Survival analysis	
Tian et al. (2024)	Feedforward	Archimedean	Kernel methods	
Jutras-Dubé et al. (2024)	GAN, autoencoder	Any	Synthetic population generation	
Liu et al. (2025a)	Normalizing flow	Any	Copula estimation	
Letizia et al. (2025)	Feedforward	Any	Copula estimation	

A completely monotone generator can be expressed as the expected value $\psi(u) = \mathbb{E}_M\left(\exp(-uM)\right)$, where M is a random variable with $\mathbb{P}(M > 0) = 1$. They construct a neural network to express the generator as a mixture of negative exponential functions and, thus, obtain a trainable generator function. Zhang et al. (2024b) apply this method to survival analysis, where they assume censoring and survival times to be dependent. The dependence structure is modeled with an Archimedean copula that in turn is estimated with the approach above. Another application of Ling et al. (2020) is Tian et al. (2024), where a spectral kernel network is estimated, which offers a way to improve scalability of kernel methods.

Complementing the method above, Ng et al. (2021) train a neural network to learn the distribution of the latent variable M of the LST. By this, some numerical issues in higher dimensions that came up in Ling et al. (2020) are circumvented. Additionally, Ng et al. (2021) extend their approach to hierarchical Archimedean copulas.

Extreme-value copulas can be used to model the dependence structure of extreme events. A copula C_{ext} is called extreme-value copula if there is a copula C such that (Gudendorf and Segers 2010)

$$C\left(u_1^{\frac{1}{n}}, \dots, u_d^{\frac{1}{n}}\right)^n \xrightarrow{n\to\infty} C_{ext}(u_1, \dots, u_d). \tag{11}$$

An example of an extreme-value copula is the Gumbel copula, which is also an Archimedean copula.

Extreme-value copulas can be characterized by the so-called Pickands dependence function, which is a d-max decreasing function (Gudendorf and Segers 2010). In a similar fashion as the generator functions of

Archimedean copulas above, Hasan et al. (2022) model extreme-value copulas by expressing the Pickands dependence function with a neural network that they ensure to be d-max decreasing by imposing restrictions on the architecture and on the activation functions of the neural network. Alternatively, for every Pickands dependence function there is a formulation via a spectral measure (Gudendorf and Segers 2010). Also, Hasan et al. (2022) utilize this with a generative neural network that learns the spectral measure of the Pickands dependence function as an alternative to the approach above.

3.2 Estimating copulas with normalizing flows

The core idea of a normalizing flow is to map a base density $p_{\mathbf{z}}$ to the data density – also called target density – $p_{\mathbf{x}}$ we want to model. To this end, a normalizing flow aims at representing a data point $\mathbf{x}$ as a transformation of a data point $\mathbf{z}$ coming from density $p_{\mathbf{z}}$, i.e., $\mathbf{x} = T(\mathbf{z})$. The transformation function T has to be a diffeomorphism, i.e., it is invertible and both T and its inverse T^{-1} are differentiable. With this, the change-of-variables formula for probability densities yields the density of $\mathbf{x}$ as

$$p_{\mathbf{x}}(\mathbf{x}) = p_{\mathbf{z}}(\mathbf{z})\left|\det J_T(\mathbf{z})\right|^{-1}, \tag{12}$$

where $\mathbf{z} = T^{-1}(\mathbf{x})$ and J_T is the Jacobian of T.

Example 2. Let the base density $p_{\mathbf{z}}$ be standard normal and let T be an invertible matrix $\mathbf{A}$ such that $\mathbf{x} = \mathbf{A}\mathbf{z}$. Then, we obtain a data density $p_{\mathbf{x}}$ that is normal again with covariance matrix $\mathbf{A}^t\mathbf{A}$.

In normalizing flows, the transformation function T is modeled by a composition of neural networks and the loss being minimized is the Kullback-Leibler divergence between the data distribution $p_{\mathbf{x}}$ and the distribution $p^*_{\mathbf{x}}$ imposed by the normalizing flow model. In order to make this feasible for training, the neural network is structured in a way such that T^{-1} as well as the determinant of the Jacobian of T are easy to compute, e.g., by T having an upper triangular Jacobian. Good starting points for normalizing flows and their numerous variants are the survey articles Kobyzev et al. (2021); Papamakarios et al. (2021).

Normalizing flows offer two ways to include copulas. One is to use a copula as the base distribution. This is pursued in a line of work discussed in Section 4. The other one is to estimate a copula as the target distribution of a normalizing flow, which is done in Wiese et al. (2019); Kamthe et al. (2021); McDonald et al. (2022). In order to obtain the full data distribution $F_{\mathbf{X}}$ usually two normalizing flows are chained together. First, each marginal distribution is modeled with a normalizing flow and, second, the copula is modeled with another normalizing flow. However, so far this line of research overlooked that copulas must have uniform marginal distributions, which is not enforced by the normalizing flow models in the cited works.

Copulas from normalizing flows are also used in Mitskopoulos et al. (2022) as building blocks in vine copulas. Here, each bivariate (conditional) copula is modeled via a normalizing flow, which together form a vine copula, see Appendix B for a brief introduction to vine copulas. However, the same shortcoming as above applies: Uniform marginal distributions for the copulas are not enforced by the normalizing flow models used.

A different route to copulas from normalizing flows take Liu et al. (2025a). Let $T : \mathbb{R}^d \to \mathbb{R}^d$, d ≥ 2, be an invertible measurable function such that the density transformation formula (12) holds. If the distribution function of the random vector $T(\mathbf{X})$ is a copula, i.e. $T(\mathbf{X})$ has standard uniform univariate margins, then the CDF of $T(\mathbf{X})$ is a copula via transformation. Liu et al. (2025a) construct a model for the copula via transformation by using a normalizing flow that estimates T. During the training of the normalizing flow, uniform marginal distributions are enforced by estimating the marginal distributions non-parametrically from synthetic data. Note that addressing this specific property of a copula explicitly requires an additional training step. As will become apparent from the following sections, it is an overarching theme that ensuring uniform marginal distributions requires additional effort when constructing copulas from neural networks.

3.3 Estimating copulas with generative moment matching networks

A generative moment matching (GMM) network is a neural network that maps a sample $\mathbf{z}$ from an input distribution to a sample $\mathbf{x}$ from a target distribution. In contrast to a normalizing flow, the loss optimized by the GMM network is different and also the architecture of normalizing flows and GMM networks differ. In a GMM network, the target distribution is the distribution of the data we want to obtain and the input distribution can be chosen arbitrarily, however, it is usually one that can be easily sampled from. The GMM network f creates an output sample $\mathbf{y} = f(\mathbf{z})$ and it is now of interest to train f such that $\mathbf{x}$ and $\mathbf{y}$ come from the same distribution. This is done by using the maximum mean discrepancy as the loss function, which compares the moments of the input distribution to the empirical data distribution using the kernel trick (Gretton et al. 2007). Thus, a GMM network obtains the parameters of f by minimizing the discrepancy between all empirical moments of $\mathbf{x}$ and $\mathbf{y}$, i.e.,

$$\frac{1}{n^2}\sum_{i}^{n}\sum_{i'}^{n} K\left(x_i, x_{i'}\right) - \frac{2}{nm}\sum_{i}^{n}\sum_{j}^{m} K(x_i, y_j) + \frac{1}{m^2}\sum_{j}^{m}\sum_{j'}^{m} K\left(y_j, y_{j'}\right), \tag{13}$$

where n and m are the sample sizes of the data and of the input distribution, respectively, and K is a kernel function. We refer the reader to Li et al. (2015); Dziugaite et al. (2015) for an in-depth treatment of GMM networks.

Example 3. Let $\phi(x) = x$ be the feature map that is represented by kernel K, i.e., we have a linear kernel (we refer the interested reader to Schölkopf and Smola (2002) for a more in-depth treatment of feature maps and kernels). Then, the maximum mean discrepancy of two distributions p_X and p_Y is given by

$$MMD(p_X, p_Y) = \|\mathbb{E}_{p_X}[\phi(X)] - \mathbb{E}_{p_Y}[\phi(Y)]\|_2 = \|\mathbb{E}_{p_X}[X] - \mathbb{E}_{p_Y}[Y]\|_2,$$

i.e., the empirical loss as defined above would measure the distance of the means of the two distributions. By choosing a more general kernel K that ensures that all moments are compared (not only the first one), the maximum mean discrepancy can measure whether all moments of two distributions are the same, i.e., whether the two distributions are the same.

Hofert et al. (2023b) estimate a copula with a GMM network and apply this in a regression context. This is done by using a multivariate standard normal distribution as the input distribution, which is mapped to the copula data by the GMM network with the maximum mean discrepancy loss above. Notable here is that the authors explicitly pay attention to map into a sample with uniform marginal distributions. GMM networks are further employed for quasi-random sampling of copulas in Hofert et al. (2021). Here, the network is trained as above, however, it is used to turn a randomized quasi-Monte Carlo point set into a quasi-random sample of the approximated distribution.

In Hofert et al. (2023a) a GMM network is used to obtain a lower dimensional representation of a high-dimensional copula in order to assess the goodness-of-fit. The idea followed is akin to the Rosenblatt transform (Rosenblatt 1952). The general scheme is, first, to estimate a candidate copula parametrically on a d-dimensional sample. Second, a GMM network is trained on the empirical copula of the data and the d'-dimensional uniform distribution as its target, where $d' < d$ (usually, $d' = 2$ for visual inspection). Now, the GMM network is utilized to evaluate the goodness-of-fit of the parametrically fitted copula. Therefore, samples are generated from the fitted copula and input to the GMM network. If the fit is good, the resulting output should be uniformly distributed in $[0, 1]^{d'}$.

An approach along the lines of GMM networks is followed by Janke et al. (2021), where a framework to fit a copula with an implicit generative neural network is developed. Examples in this category are Generative Adversarial Nets (see Section 4) and GMM networks (Mohamed and Lakshminarayanan 2017). First, multivariate standard normal noise is transformed via the neural network to a latent distribution. Second, by employing a softrank layer, and thus, ensuring uniform marginals, the latent distribution is mapped into the copula space. Third, the mapped latent distribution is tuned towards the data distribution in the copula space. In their experiments, Janke et al. (2021) employ a feedforward network using the energy distance as a loss, which is a special case of maximum mean discrepancy. Hence, the approach is similar to using a GMM network. An application of the previous approach can be found in Huk et al. (2025), where copulas are interpreted as ratio classifiers.

3.4 Estimating copulas via the neural network loss function

A different approach to copula modeling with neural networks take Zeng and Wang (2022), where the loss function that is minimized by the neural network is built as a linear combination of several losses of which each reflects a property of a copula. This is in contrast to the previous approaches, where the copula is modeled either by estimating the generator of an Archimedean copula or by minimizing some distance to the data distribution. Here, the copula is obtained by a maximum likelihood loss that is penalized for deviation from the copula properties. For example, one part of the loss function is set up such that the marginal CDFs are ensured to be uniform. Let $\tilde{C}$ be the neural network that estimates the copula and consider dimension $d = 2$ for notational convenience. Then, uniform marginal distributions can be incorporated into the loss function by considering the loss

$$\sum_{i=1}^{n_{uni}} \left|\tilde{C}(\tilde{u}_i, 0)\right| + \sum_{i=1}^{n_{uni}} \left|\tilde{C}(0, \tilde{v}_i)\right| + \sum_{i=1}^{n_{uni}} \left|\tilde{C}(\tilde{u}_i, 1) - \tilde{u}_i\right| + \sum_{i=1}^{n_{uni}} \left|\tilde{C}(1, \tilde{v}_i) - \tilde{v}_i\right|, \tag{14}$$

where $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{v}}$ are evenly spaced vectors in [0, 1] with n_{uni} elements. Here, the first two terms ensure that the neural network evaluates to 0 if one of the inputs is 0 and the last two terms ensure that the neural network evaluates to either $\tilde{\mathbf{u}}$ if $\tilde{\mathbf{v}} = 1$ or vice versa to $\tilde{\mathbf{v}}$ if $\tilde{\mathbf{u}} = 1$. Other losses reflect the fact, that the density of the estimated copula must not be negative and that it integrates to one. Finally, the fitness to the data is ensured by a maximum likelihood loss. An application of the approach above to remote sensing detection is given in Li et al. (2025).

3.5 Further topics on deep learning in copulas

This section briefly presents various works to complete the picture of deep learning and neural networks in copulas. Each is treated rather shortly and references are given as pointers for more detailed expositions.

In Letizia et al. (2025), a variational formulation of f-divergence via Fenchel conjugates (Rényi 1961) is used to estimate a copula density with a neural network. It is argued that the resulting estimate integrates to one and, therefore, is a normalized density. By operating in the copula domain and not in the original data space, uniformly distributed marginal distributions are ensured.

Recurrent neural networks (RNN) were developed for processing sequential data such as text, speech, and time series. Their defining feature is that outputs of a layer $\mathbf{h}_\ell$ are used as additional inputs to this same layer. RNNs come in many fashions and can make up building blocks of higher level architectures as discussed in the next subsections. A particular type of RNN, which became a standard in the literature, is the long short-term memory network (LSTM). An LSTM is comprised of so-called cells that store values. The cell itself has an input gate, forget gate and output gate that control which inputs to store, which information to discard, and which outputs to propagate, respectively. More on RNNs, LSTMs, and other variants can be found, e.g., in Hochreiter and Schmidhuber (1997); Goodfellow et al. (2016).

Vine copulas are treated in Sun et al. (2019) who want to estimate the vine structure. Estimation of the vine structure is a challenging task, also cf. Appendix B. Sun et al. (2019) tackle this by transforming the vine structure into a vector representation and by reformulating the problem into a reinforcement learning task, which is solved by an LSTM. A difference to the works presented so far is that the neural network does not estimate the copula but rather a part of the copula estimation process is enhanced by a neural network.

Staying with vine copulas, in chemical engineering and process control, vine copula regression plays a role. In order to reduce dimensions and, thus, to simplify the vine structure, features from the latent space of an autoencoder (cf. Section 4.3) can be used as the regression inputs. This idea is followed in Zhou et al. (2020) and Cui (2021). Moreover, additional samples can be generated from a generative deep learning model to enhance the predictive power of the vine regression. This is done in Chen et al. (2022) who use a generative adversarial network to generate samples (see Sections 4.2).

In Jutras-Dubé et al. (2024) the problem of synthetic population generation is addressed. First, the data is rank-transformed and the resulting distribution is estimated with a generative adversarial network and a variational autoencoder (see Sections 4.2 and 4.3). Even though, uniform marginals are not enforced by the method, the authors believe that the resulting distribution is a copula. Reflecting on the observations so far, this belief seems dubious. Ensuring the resulting estimate is a copula at least requires the setup of a loss function that penalizes deviation from the copula properties. In the light of the previous sections, merely fitting a network to the data, e.g., by Kullback-Leibler divergence, seems insufficient.

The next section surveys how copulas are used for neural networks and deep learning.

4 Copulas in deep learning

This section gives an overview of how copulas are utilized for neural networks and deep learning. First, we discuss how copulas are used to amplify several deep learning architectures such as normalizing flows, generative adversarial networks, (variational) autoencoders, and transformers in Sections 4.1–4.4. Second, we turn towards overarching topics such as how copulas are used at the input level of deep learning models in Section 4.5 and how to combine neural network outputs with copulas in Section 4.6. Lastly, we touch on further topics, where copulas help in deep learning such as graph neural networks and survival analysis in Section 4.7. An overview of the cited works with some additional metadata can be found in Table 2.

Table 2: Overview of references for copulas in deep leaning sorted by publication year.

Reference	Copula family	Deep learning architecture	Application
Yang et al. (2011)	Various	Feedforward	Output combination, climate science
Chatzis and Demiris (2012)	*t*, Archimedean	RNN	Output combination
Suh and Choi (2016)	Gaussian	Autoencoder	Data generation, mixed data
De Oliveira et al. (2016)	Gumbel, Gaussian	Feedforward	Error combination, time series
Walecki et al. (2017)	Frank	CNN	Output combination, facial expression estimation
Xu et al. (2017)	Various	CNN	Output combination, wind farm power output
Wieczorek et al. (2018)	Gaussian	Autoencoder	Information bottleneck
Sun et al. (2018)	Vine	Autoencoder	Input data generation, smart grids
Wang and Wang (2019)	Gaussian	Autoencoder	Data generation
Tagasovska et al. (2019)	Vine	Autoencoder	Data generation
Zhang et al. (2019)	Vine	Feedforward	Classification, sensor networks
Huang and Zhang (2019)	Unknown	Feedforward	Feature selection, drought prediction
Ouyang et al. (2019)	Gumbel	Deep belief network	Input data transformation, power systems
Salinas et al. (2019)	Gaussian	LSTM	Output distribution, time series
Toubeau et al. (2019)	Nonparametric	LSTM	Data generation, time series
Wen and Torkkola (2019)	Gaussian	RNN	Probabilistic forecasts, time series
Luo et al. (2020)	Vine	Deep belief network	Input data transformation, turbine failure
Chilinski and Silva (2020)	Gaussian	Feedforward	Output combination
Xu et al. (2020)	Vine	Feedforward	Output combination, predictive maintenance
Kuiry et al. (2020)	Various	Various	Ensemble building, image segmentation
Mukherjee et al. (2021)	Gaussian	RNN, LSTM	Input data generation, power systems
Silva et al. (2021)	Gaussian	CNN	Input data generation, medicine
Wang et al. (2021a)	Gaussian	LSTM, autoencoder	Finance
Song et al. (2021)	Gaussian	Transformers	Multi-head attention
Zhong et al. (2021)	Gaussian	Autoencoder	Collaborative filtering
Dimitriev and Zhou (2021a)	Gaussian, Dirichlet	Any	Neural network training
Dimitriev and Zhou (2021b)	Gaussian, Dirichlet	Any	Neural network training
Kom Samo (2021)	Maximum entropy	GAN	Mode collapse

Table 2: (continued)

Reference	Copula family	Deep learning architecture	Application
Ma et al. (2021)	Gaussian	Graph	Node modeling
Messoudi et al. (2021)	Various	Feedforward	Conformal prediction
Zhao et al. (2021)	Various	CNN	Output combination, wind speed prediction
Zhang et al. (2021)	Clayton	LSTM	Output combination, wind speed preditcion
Laszkiewicz et al. (2021)	Independence	Normalizing flows	Base distribution
Laszkiewicz et al. (2022)	Independence	Normalizing flows	Base distribution
Drouin et al. (2022)	Any	Transformer	Time series
Letizia and Tonello (2022)	Any	GAN	Data generation
Boulaguiem et al. (2022)	Extreme value	GAN	Climate science
Liu et al. (2022)	Empirical	Transformer	Feature selection, power systems
Li et al. (2022)	Empirical	RNN, LSTM	Feature selection, runoff prediction
Zhang et al. (2022)	Vine	Feedforward	Output combination, hydrology
Huang et al. (2022)	Clayton	LSTM	Output combination, wind speed prediction
Zhou et al. (2022)	Gaussian	Feedforward	Output combination, power prediction
Abbaszadeh et al. (2022)	Unknown	Various	Bayesian model averaging
Tagasovska et al. (2023)	Vine	Various	Output uncertainty
Cui et al. (2023)	Various	LSTM	Output uncertainty
Peng et al. (2023)	Gaussian	Feedforward	Feature selection
Gharibi et al. (2023)	Any	GAN	Charging electric vehicles
Kirchler et al. (2023)	Gaussian	Normalizing flows	Non-i.i.d. data
Xu and Cao (2023)	Vine	LSTM, autoencoder	Finance
Nafii et al. (2023)	Gaussian	Feedforward	Input data generation, water management
Dey et al. (2023)	Gaussian	CNN	Output combination, medicine
Yang et al. (2023)	Gaussian	LSTM	Output combination, time series
Foomani et al. (2023)	Clayton	Feedforward	Survival analysis
Zhang et al. (2024b)	Archimedean	Feedforward	Survival analysis
Sun and Yu (2024)	Empirical	RNN	Conformal prediction
Nabizadeh Chianeh et al. (2024)	Vine	Feedforward	Input data generation, chemistry
Ashok et al. (2024)	Any	Transformer	time series
Balgi et al. (2024)	Gaussian	Normalizing flows	Base distribution
Iqbal and Siddiqi (2024)	Various	RNN, LSTM	Feature selection, streamflow prediction
Xiao et al. (2024)	Vine	CNN	Output combination, reliability assessment
Kim et al. (2024)	Vine	LSTM	Output combination, control charts
Qiu et al. (2024)	Gaussian	Graph	Node modeling
Li et al. (2024b)	Gaussian	CNN	Loss function, medicine
Li et al. (2024a)	Gaussian	Transformer	Loss function, medicine
Sattari et al. (2024)	Various	Various	Bayesian model averaging
Kächele et al. (2025)	Various	Autoencoder	Data generation
Kim (2025e)	Gaussian	LSTM	Input data generation, survival analysis
Meng et al. (2025)	Empirical	Transformer	Feature selection, predictive maintenance
Akkepalli and K (2025)	Empirical	Autoencoder	Feature selection, network intrusion detection
Kim (2025a)	Empirical	CNN, LSTM	Input data transformation, survival analysis
Zhong et al. (2025)	Gaussian	CNN	Loss function, medicine
Panda and Garain (2025)	Gaussian	CNN	Multi-label classification
Liu et al. (2025b)	Hierarchical Archimedean	Feedforward	Survival analysis
Kwon et al. (2025)	Clayton	Feedforward	Survival analysis
Simchoni and Rosset (2025)	Gaussian	Feedforward	Mixed model estimation

Before starting, we want to point out that copulas have been used for comparison purposes and as competitor models in the neural network and deep learning community. For example, Kulkarni et al. (2018) generate mobility trajectories with generative neural networks and compare these to trajectories generated by a copula. Other examples, where copulas are used as competitor models include, e.g., Alokley et al. (2024); Hernández et al. (2025); Ling et al. (2020); Janke et al. (2021); Zhang et al. (2024a); Kim (2025c).

4.1 Copulas in normalizing flows

As already stated in Section 3.2, normalizing flows offer several ways to include copulas. One is to use a copula as the base distribution, which is pursued in Laszkiewicz et al. (2021, 2022). This is done, because in practice it was observed that normalizing flows face difficulties in modeling heavy-tailed distributions. However, copulas decouple marginal distributions from the dependence structure and, hence, open up the possibility to model each marginal distribution separately. This is useful since for heavy-tailed marginal distributions a tailored normalizing flow can be used. Laszkiewicz et al. (2021, 2022) use the independence copula as the base distribution and obtain good results when dealing with heavy-tailed marginal distributions. A different application of this principle is given in Balgi et al. (2024), where a Gaussian copula is used as the base distribution in a cause-effect modeling task.

In a further application of copulas to normalizing flows, Kirchler et al. (2023) consider a non-i.i.d. setting where the data are identically distributed but dependent. In this case, the objective function of the normalizing flow does not factor across observations. In order to solve this, Kirchler et al. (2023) derive the objective function for dependent data and use a Gaussian copula approach to model the dependence.

4.2 Copulas in generative adversarial networks

A generative adversarial network (GAN) comprises a so-called generator and discriminator network. These play the adversaries in a zero-sum game. The generator receives random noise and generates a fake example of the real data from it. The discriminator receives real data as well as the generator's fake data and classifies the input as either fake or real. By optimizing a minmax loss function, the generator learns to generate realistic data. In particular, GANs excel in image and video generation. More on GANs can be found in Goodfellow et al. (2014, 2020).

Letizia and Tonello (2022) use a copula approach in GANs. They do this in a two-stage scheme. First, the generator of a GAN fits the data in the copula domain. Second, a GAN learns the transformation from the unit space back into the original data space. They argue that using this decomposition enables a more robust generation process. A similar approach is taken by Boulaguiem et al. (2022) and Gharibi et al. (2023), where a GAN is trained in the copula domain. However, they do not use a GAN to learn the transformation back to the original data space but rather use empirical CDFs.

4.3 Copulas in (variational) autoencoders

Let $m < n$, $f : \mathbb{R}^n \to \mathbb{R}^m$ be an encoder function and $g : \mathbb{R}^m \to \mathbb{R}^n$ be a decoder function. An autoencoder tries to reconstruct its inputs $\mathbf{X}$ as close as possible, i.e., $\mathbf{X} \approx g(f(\mathbf{X}))$. On the one hand, this is done by choosing an appropriate neural network architecture that represents the encoder and decoder functions. Thus, the neural network, first, compresses the inputs into a lower dimensional space – the so-called latent space – and, second, decompresses the lower dimensional latent space into the input space. On the other hand, an appropriate loss function measures the discrepancy between inputs and outputs such that backpropagation can be used to train the autoencoder (Kramer 1991, 1992).

A variational autoencoder is a type of autoencoder where the latent space is represented by a probability distribution. The encoder maps the inputs to sample points $\mathbf{z}$ of the distribution in the latent space and the decoder maps the sampled latent space points to samples $\mathbf{x}$ of the distribution in the input space. One can show that this can be formulated via variational Bayesian methods such that the posterior latent space distribution $p_\theta(\mathbf{z}|\mathbf{x})$ is approximated by a distribution $q_\phi(\mathbf{z}|\mathbf{x})$. Due to the use of probability distributions, a specific loss function, called the evidence lower bound (ELBO), is used that maximizes the log-likelihood $p_\theta(\mathbf{x})$ of the observed data and minimizes the divergence of $p_\theta(\mathbf{z}|\mathbf{x})$ and $q_\phi(\mathbf{z}|\mathbf{x})$ at the same time. In order to achieve this, it is additionally necessary to reparametrize $\mathbf{z}$, e.g., as a Gaussian distribution, to be amenable for optimization, which is often referred to as the reparametrization trick. For detailed expositions of variational autoencoders, the interested reader is referred to Rezende et al. (2014); Kingma and Welling (2014, 2019).

Autoencoders offer a quite broad application area of copulas in deep learning. Recall that an autoencoder encodes input data to a lower-dimensional latent space. The distribution in this latent space can be modeled by a copula in order to draw samples from it and decode these via the decoder to obtain newly generated data. This approach is taken by Tagasovska et al. (2019) who use non-parametric vine copulas in the latent space. An application of this in an LSTM autoencoder architecture is given in Xu and Cao (2023) for financial cross-market dependence modeling. Another route to model the latent space with copulas is taken by Kächele et al. (2025) who use an empirical Beta copula. This paper also compares various latent space modeling techniques and it is argued that copula techniques offer several advantages compared to other methods. Finally, we want to point out that the generative methods described in the previous section, such as the implicit generative copula by Janke et al. (2021) or the approach by Liu et al. (2025a), can be used to model an autoencoder's latent space.

Suh and Choi (2016) employ a Gaussian copula in the latent space of a variational autoencoder, where they make a parametric assumption for the marginal distributions. This enables them to incorporate variables of mixed data type. A similar approach is taken in Wang and Wang (2019) and Zhong et al. (2021), where the posterior distribution is modeled with a Gaussian copula and the reparametrization trick is extended for it.

Wieczorek et al. (2018) try to solve the information bottleneck problem, which takes random vectors $\mathbf{x}$ and $\mathbf{y}$ and searches a third random vector $\mathbf{t}$ while compressing $\mathbf{x}$ and preserving information in $\mathbf{y}$. This can be framed as a variational autoencoder problem, where $\mathbf{t}$ plays the role of the latent space representation. Wieczorek et al. (2018) use a Gaussian copula to model the latent space distribution.

Wang et al. (2021a) use contrastive predictive coding (van den Oord et al. 2018) to predict stock prices. Contrastive predictive coding is similar to an autoencoder approach, however, applied to time series. An encoder projects the input sequence to a latent space, where a sequence model such as an LSTM models the time series of the codes. In Wang et al. (2021a) a Gaussian copula is used to model the distribution of macro-level variables such as interest rates, which are then transformed and input to the latent space as additional information.

4.4 Copulas in transformers

Transformers are neural networks that were originally developed for natural language processing tasks such as translation. Transformers rely on an encoder-decoder architecture (similarly to autoencoders) and use the so-called attention mechanism, which helps focusing on the relevant parts of the input (Vaswani et al. 2017). In a database analogue, attention is based on key, query, and value. For that, keys, values and queries are generated from the encoder and decoder by linear mappings of internal states. The queries and keys are then combined by dot products and softmax and the result is used to form a linear combination with the values, i.e.,

$$\mathbf{a} = \operatorname{softmax}\big((\mathbf{h}\mathbf{W}^q)(\mathbf{H}\mathbf{W}^k)'\big)(\mathbf{H}\mathbf{W}^v), \tag{15}$$

where $\mathbf{h}$ stems from the decoder and $\mathbf{H}$ from the encoder and the matrices $\mathbf{W}^q$, $\mathbf{W}^k$, and $\mathbf{W}^v$ contain trainable weights for queries, keys, and values respectively. This procedure can be done in parallel for different weight matrices, which is called multi-head attention. More on transformers and the attention mechanism can be read in, e.g., Vaswani et al. (2017).

Copulas can be applied in transformers. Song et al. (2021) use a Gaussian copula to model the dependence between heads in multi-head attention. In Drouin et al. (2022) and a subsequent development thereof (Ashok et al. 2024), a transformer is used to learn the conditional distribution of missing values – by noting on an abstract level that values to be forecast are missing, the model can be used for forecasting. This is done by employing a particular decoder structure in the transformer that, first, allows to train marginal distributions with normalizing flows (compare, e.g., Laszkiewicz et al. (2021, 2022) and, second, uses a factorization to learn a copula density $c(\mathbf{u})$ by

$$c(\mathbf{u}) = c_{\pi_1}(u_{\pi_1})c_{\pi_2}(u_{\pi_2}|u_{\pi_1})\cdots c_{\pi_d}(u_{\pi_d}|u_{\pi_1},\ldots,u_{\pi_{d-1}}), \tag{16}$$

where $(\pi_1,\ldots,\pi_d)$ is a permutation of $(1,\ldots,d)$. The univariate distributions c_{π_i} are estimated non-parametrically by a histogram approach, i.e., the space is binned and each bin is governed by a piecewise constant density (Ashok et al. 2024; Drouin et al. 2022).

4.5 Copulas to refine deep learning inputs and training

We now turn towards how copulas are utilized in neural networks and deep learning on a more conceptual level. For example, copulas can help in estimating gradients for discrete variables. During the training of a neural network it might be necessary to optimize an objective of the form $\mathbb{E}_{z\sim q_\phi(z)}[f(z)]$ and the gradient with respect to the distribution parameter(s) ϕ is needed. This comes up, e.g., when optimizing ELBO in variational autoencoders (Dimitriev and Zhou 2021a; Kingma and Welling 2014). However, when the variable is discrete this is difficult since the usual reparametrization does not work. Therefore, other estimators for the gradient have to be used which rely on sampling; for an overview see, e.g. Dimitriev and Zhou (2021a,b). In order to reduce the variance of these estimators, it is necessary to obtain antithetic variables. In Dimitriev and Zhou (2021a) binary antithetic variables are obtained by using copulas. This is further extended to categorical variables in Dimitriev and Zhou (2021b).

Also related to the training of a neural network, Kom Samo (2021) develops an optimization program to determine the maximum entropy copula from data, which is important when optimizing mutual information. This can be applied to prevent mode collapse during the training of GANs, which happens when the generator only samples near one mode of the data. The training objective of the generator can be regularized to mitigate this, where the maximum entropy copula can be used.

Additionally, copulas play a role at the input level of deep learning models. Since deep learning models require a large amount of training data to work well, a quite popular application area of copulas is to generate additional training data for these models. The general scheme is to model the data with a copula (mostly Gaussian or vine copulas) and sample points from it that are fed into the network during its training phase. Examples of this in various application areas such as water management, survival analysis, and power systems are Kim (2025e); Nabizadeh Chianeh et al. (2024); Nafii et al. (2023); Mukherjee et al. (2021); Silva et al. (2021); Sun et al. (2018).

Peng et al. (2023) deal with instance-wise feature selection in neural networks. For that, a copula is used to incorporate dependence between the features. The overall model is trained end-to-end such that for each data point a specific subset of the features is retained for the prediction task. Feature selection can also be done via copula entropy, which measures mutual information (Ma and Sun 2011). Mutual information in turn can be used to identify redundant features. This is applied, e.g., in Liu et al. (2022) to predict power grid frequency stability with a transformer model, in Huang and Zhang (2019) for drought prediction, in Meng et al. (2025) for remaining life prediction of rolling bearings, in Li et al. (2022) for runoff prediction, and in Akkepalli and K (2025) for network intrusion detection. A slightly different route to feature selection take Iqbal and Siddiqi (2024), where a copula Markov-Chain-Monte-Carlo model selects features, which are input to an LSTM for streamflow prediction.

Furthermore, copulas can be used for feature engineering. In Ouyang et al. (2019) a Gumbel copula is used to aggregate input variables which are fed into a neural network to forecast short-term power load. In a similar fashion, Luo et al. (2020) use a vine copula to aggregate input variables to a state parameter, which is fed into a neural network that predicts wind turbine failure. In Kim (2025a), the input data is rank-transformed as it is done when using the empirical copula (see Equation (8)) before it is fed into a Convolutional-LSTM neural network for heterogeneous treatment effect estimation.

4.6 Copulas to process deep learning output

At the output level of deep learning models, a popular application of copulas is to use them for combining neural network outputs. Yang et al. (2011) use extreme learning machines, which are specific types of single layer feedforward neural networks, to predict wind and ice loads. These predictions are then combined with a copula in order to predict failure probabilities for transmission lines and transmission towers.

In Chilinski and Silva (2020) a univariate CDF is created with a feedforward neural network employing non-negative weights and special activation functions so that the CDF properties are preserved. In the multivariate case, this network structure per variable is then combined by a Gaussian copula to create a multivariate CDF.

A similar approach is followed in Chatzis and Demiris (2012), where a so-called echo state network – an RNN, where all weights are initialized randomly and only the weights of the output layer are updated during training – estimates marginal distributions of the variables and a copula is used to combine these into a multivariate distribution. They use the t-copula and Archimedean copulas for this.

In De Oliveira et al. (2016) a target time series is considered that is forecast by multiple forecasters, in this case neural networks. From the forecasts the errors to the true value are obtained as well as the respective error distributions. These are then combined with a copula to obtain a joint model for the forecast errors.

In Zhang et al. (2019) a sensor network is used to classify human activity. Outputs for each sensor are input into a deep neural network in order to extract features. After estimating the marginal distributions of the neural network outputs via kernel density estimation, the joint distribution is created with a vine copula to solve the classification task.

Convolutional neural networks (CNN) were specifically developed for processing image(-related) data. They possess an architecture that enables them to process images efficiently. In particular, CNNs are composed of so-called convolutional layers that slide learnable kernels over the inputs. By that a small section of the input matrix is convolved part by part. This not only massively reduces the amount of weights that have to be optimized but also extracts important features efficiently. The CNN is made of several convolutional layers and further layers in between such as pooling layers and normalization layers. Over the years, many variants of CNNs were developed and they are a standard component in many architectures. For example, they can be used in autoencoders for image data (see Section 4.3). A more thorough treatment of CNNs can be found, e.g., in Goodfellow et al. (2016).

Outputs of CNNs are combined with copulas in Walecki et al. (2017) for facial action unit intensity estimation, in Xu et al. (2017) for wind farm power output prediction, in Zhao et al. (2021) for wind speed prediction, and in Dey et al. (2023) for tumor classification. Vine copulas are used in Zhang et al. (2022) to combine outputs of neural networks in a hydrology application and in Xiao et al. (2024) to model outputs of a hierarchical neural network for probabilistic reliability assessment of bridges. An ensemble approach is taken in Kuiry et al. (2020), where a copula builds an ensemble of neural networks for image segmentation.

A line of research deals with probabilistic forecasting of multivariate time series. Salinas et al. (2019) use LSTMs to model each univariate time series. The LSTM outputs are then used to estimate their CDFs non-parametrically. Finally, a Gaussian copula combines the CDF estimates in order to obtain the full multivariate distribution. Moreover, they show that the Gaussian copula parameters are trainable in an end-to-end model. Closely related, Toubeau et al. (2019) estimate the copula non-parametrically from the original data and generate samples from it. The samples are then used to turn probabilistic forecasts of the per variable LSTMs into samples from the original data space. Further references combining LSTM outputs using various copulas such as Gaussian copulas, Clayton copulas, and vine copulas are Huang et al. (2022); Kim et al. (2024); Zhang et al. (2021); Yang et al. (2023).

A similar approach to above is followed in Wen and Torkkola (2019), where an RNN is used that outputs quantile estimates, see, e.g., Wen et al. (2017), which are combined with a Gaussian copula to produce probabilistic multivariate time series forecasts. Along these lines, Zhou et al. (2022) also use a Gaussian copula to combine outputs of a quantile regression neural network for spatio-temporal forecasting of photovoltaic power. Furthermore, a vine copula to combine outputs of a quantile regression neural network is used in Xu et al. (2020).

Also, neural networks are combined by Copula Bayesian Model Averaging. Here several models are combined as a weighted sum, where the weight prior is modeled with a copula (Madadgar and Moradkhani 2014). Examples of this combining various neural network architectures are Sattari et al. (2024); Abbaszadeh et al. (2022).

In Tagasovska et al. (2023) output uncertainty estimation for neural networks is treated by using vine copulas. The method is independent of the neural network model and can be attached to any trained neural network. Output uncertainty is modeled by two components: epistemic uncertainty, i.e., uncertainty stemming from the model, and aleatoric uncertainty, i.e., uncertainty stemming from noise in the data. Epistemic uncertainty is modeled by fitting a vine copula to the activations of the last hidden layer of a neural network and simulating activations from the vine copula in order to obtain confidence intervals for the output. Aleatoric uncertainty is modeled by constructing prediction intervals with the help of vine copulas. Combining the two uncertainty sources yields an uncertainty estimate for the neural network over the data space. Uncertainty quantification of an LSTM architecture for flood prediction is treated in Cui et al. (2023) who also use a copula approach for this. In contrast to the latter approach, uncertainty quantification here is based on hydrological principles and Bayesian statistics.

4.7 Further topics on copulas in deep learning

This section briefly presents various topics to complete the picture of copulas in deep learning and neural networks. Each is treated rather shortly and references are given for more detailed expositions.

So-called graph neural networks, which model graph data, see, e.g., Ying et al. (2018); Kipf and Well (2017); Hamilton et al. (2017), offer a further application area for copulas. A graph is a set of nodes and edges that connect nodes. Graph neural networks are limited, when correlation between nodes has to be incorporated. For that, Ma et al. (2021) utilize a Gaussian copula to link the marginal distributions of the nodes that are modeled with a graph neural network. If the nodes are not connected, the correlation parameter in the Gaussian copula is set to 0. All other parameters are learned during the training phase end-to-end. An application of this to temporally changing graphs can be found in Qiu et al. (2024).

Tweaking loss functions to learn a valid copula is described in Section 3.4. Complementing the latter, copulas can be used to adjust the loss function of neural networks. This plays a role in Li et al. (2024b) who want to screen myopia from images of the patient's two eyes. For that they use the Gaussian copula in a bi-channel CNN setup in order to define a loss function for the alignment of the two CNN channels. This is further extended for categorical data in Zhong et al. (2025) and in Li et al. (2024a), where the authors use transformers instead of a bi-channel CNN. Related to amending loss functions with copulas is Panda and Garain (2025) who deal with multi-label classification by considering label dependence with copulas. They derive a loss function that can be used as a regularizer in neural network training.

Foomani et al. (2023) deal with survival analysis and model the hazard function of the censoring and event time with a neural network. Since they drop the assumption of conditional independence of censoring and event time, they derive a likelihood function with a copula to model the dependence of the two times. This is akin to the procedure in Zhang et al. (2024b), however, Foomani et al. (2023) use parametric copulas. Related to Zhang et al. (2024b) is Liu et al. (2025b), where competing risks are modeled with hierarchical Archimedean copulas and marginal distributions are estimated with neural networks as in Chilinski and Silva (2020). Similar to the approaches above, Kwon et al. (2025) deal with clustered survival data and model the within cluster event times with a Clayton copula. The hazard function is estimated with a neural network.

Simchoni and Rosset (2025) treat mixed model estimation with neural networks. They extend the approach by incorporating a Gaussian copula to model the dependence of random effects in the model. By doing so, they relax the assumption that random effects follow a normal distribution, where each marginal is normally distributed.

On a final note, we want to point out that copulas can be used in conformal prediction (Shafer and Vovk 2008), when more than one variable is to be predicted. However, this applies not only to neural networks but also to any other prediction method. Thus, the following references are only provided for a complete picture. Sun and Yu (2024) use this approach in conformal prediction for multi-step time series forecasting, where RNNs predict the time series. This is akin to Messoudi et al. (2021), where a copula is used for conformal prediction in a multivariate multiple regression framework that is carried out by neural networks.

The next section provides some concluding remarks.

5 Looking backward and forward

Sections 3 and 4 show that there already is a notable amount of research in copulas and deep learning. Not only is deep learning used to estimate copulas, see, e.g., Ling et al. (2020); Ng et al. (2021), but also copulas are used in neural networks and deep learning, e.g., to serve as a base distribution in normalizing flows (Laszkiewicz et al. 2021, 2022). Moreover, it is exciting to see that copulas are used in a wider machine learning perspective; we refer to the examples in the introduction.

When looking at the cited literature, it is striking that the Gaussian copula is used very often. It even has a special name in the machine learning community and is known as the *non-paranormal* (Liu et al. 2009). We suspect that its wide usage is due to two reasons: First, we suppose that many researchers are familiar with the multivariate normal distribution and, thus, are drawn towards the Gaussian copula. Second, the Gaussian copula has a closed-form density (including standard normal quantile functions), which makes it appealing to use in an end-to-end deep learning model that has to have trainable parameters. It would be interesting to see how other copulas perform in these applications. For example, the t-copula has a closed-form density and additionally allows for tail dependence, which the Gaussian copula does not. Additionally, some Archimedean copulas have a closed-form expression for the density, see, e.g., Nelsen (2006); Joe (2015).

Another point we want to mention is that at times the literature in the deep learning community lacks clearness of exposition. This might be due to the publication in conferences, where space is limited. Hence, we urge researchers to be more precise and clear so that ideally the method can be implemented by reading the paper alone. In addition to that, often a copula is used as some kind of "derivation trick", where someone can just decouple marginal distributions and dependence structure without paying attention to copula properties such as uniform marginal distributions on [0, 1]. This is most prominent in the literature on copulas from normalizing flows. Merely fitting a network to the data, e.g., by Kullback-Leibler divergence, is insufficient and at least requires the setup of a loss function that penalizes deviation from the copula properties. Yet, there are some stellar examples, where this is explicitly taken into account and it can be seen from these examples that ensuring uniform marginal distributions is not obvious or easy to achieve but requires adapting the method. Examples of the latter can be found, e.g., in Janke et al. (2021); Zeng and Wang (2022); Hofert et al. (2023a); Liu et al. (2025a).

On a last note, there are papers related to neural networks and deep learning applications that claim to use copulas for feature selection. On closer inspection, however, it turns out that these papers use Kendall's τ. Yet, instead of using the usual sample formula to estimate Kendall's τ, see, e.g. Joe (2015), the authors proceed to fit a copula to the data and use its parameter estimate to compute Kendall's τ. Note that many copulas have formulas that connect their parameter(s) with Kendall's τ (again, see Joe (2015)). However, from a statistical perspective, it is clear that this procedure is biased from copula function misspecification and obtains additional variance from the copula parameter estimation. Therefore, we refrained from citing these works here.

Looking forward, we see a lot of potential and little explored areas in deep learning and copulas. For example, it would be interesting to see, whether other deep learning models such as energy-based models (Song and Kingma 2021), score-based generative models (Song and Ermon 2019, 2020) relying on score matching (Hyvärinen 2005), or denoising diffusion models (Ho et al. 2020; Sohl-Dickstein et al. 2015) can be used to express new copulas. In score-based generative models, the so-called score function, i.e., the gradient of the logarithmized density, instead of the density is estimated. The advantage of this is that one does not need to estimate the normalizing constant during training. It would be interesting to see whether score-based models can be applied to copula estimation. Denoising diffusion models on the other hand add noise to data and learn the reverse process of generating data from noise. Even though the above models are linked theoretically (Song et al. 2020), denoising diffusion models work by adding Gaussian noise to data, which is difficult to adapt in the bounded copula domain, also see works on kernel density estimation for copulas (Coblenz et al. 2022; Geenens et al. 2017; Omelka et al. 2009). Therefore, it might be challenging to combine copulas with denoising diffusion models, even though noise based on a Gaussian copula might translate directly into the diffusion model framework.

From an application perspective, it is striking that copulas and deep learning are applied in several works on survival analysis, see Table 3 in Appendix C. Additionally, there are many applications in engineering, power systems, climate science, and hydrology that rely on copulas and deep learning, cf. Table 2. In contrast to that, traditional areas of application for copulas such as finance and insurance are still underrepresented at the interface of copulas and deep learning. Hence, these areas of application might profit from introducing deep learning methods into the copula domain. In addition to that, for deep learning driven areas of application such as bioinformatics and computer vision it can be fruitful to incorporate copulas as a useful tool, see e.g., Walecki et al. (2017); Li et al. (2024a,b). Therefore, we encourage researchers to leave their tooling comfort zone and explore what else there is in the copula or deep learning world.

With reference to the last point, we think that the "copula trick", i.e., decoupling marginal distributions and dependence structure, is not applied enough or not well known enough in the deep learning and wider machine learning community. In principle, a copula can be utilized whenever a multivariate distribution comes up. However, attention should be paid to uniform marginal distributions of the copula and also the usefulness of the approach should always be questioned sharply, since copula modeling is not an end in itself. Complementing the latter, we see a lack of theoretical treatment on what the copulas from deep learning models are capable of. For example, a theoretical analysis of whether tail dependence (Joe 2015) can be modeled would be interesting. An example where this is possible, is the copula estimation approach with normalizing flows by Liu et al. (2025a). We are excited to see what copulas can do in neural networks and deep learning and vice versa in the future.

Acknowledgements: We thank the reviewers and the Associate Editor for valuable comments that improved the manuscript.
Research ethics: Not applicable.
Informed consent: Not applicable.
Author contributions: All authors have accepted responsibility for the entire content of this manuscript and consented to its submission to the journal, reviewed all the results, and approved the final version of the manuscript.
Use of Large Language Models, AI and Machine Learning Tools: None declared.
Conflict of interest: The authors state no conflict of interest.
Research funding: MC, BL, and DW gratefully acknowledge support from Forschungsstart project P2022-13-009 of the Carl-Zeiss-Stiftung.
Data availability: Not applicable.

A List of journals and conferences

Here, we list the journals and conferences we considered alphabetically. Proceedings and issues roughly from 2000 until summer 2025 were included.

Journals

- Journal of Artificial Intelligence Research
- Journal of Machine Learning Research
- Machine Learning
- Neural Computation
- Neurocomputing
- Pattern Recognition
- Transactions on Machine Learning Research

Conferences

- All Proceedings in https://proceedings.mlr.press/
- AAAI Conference on Artificial Intelligence (AAAI)
- Conference on Learning Theory (COLT)
- Conference on Uncertainty in Artificial Intelligence (UAI)
- European Conference on Computer Vision (ECCV)
- IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)
- IEEE/CVF International Conference on Computer Vision (ICCV)
- IEEE International Conference on Data Mining (ICDMW)
- International Conference on Artificial Intelligence and Statistics (AISTATS)
- International Conference on Artificial Neural Networks (ICANN)
- International Conference on Learning Representations (ICLR)
- International Conference on Machine Learning (ICML)
- International Conference on Machine Learning and Applications (ICMLA)

- International Conference on Pattern Recognition (ICPR)
- International Joint Conference on Artificial Intelligence (IJCAI)
- International Joint Conference on Neural Networks (IJCNN)
- Knowledge Discovery and Data Mining (KDD)
- Neural Information Processing Systems (NIPS)
- SIAM International Conference on Data Mining (SDM)

B Vine copulas

Vine copulas are a copula construction method that allows to combine bivariate copulas to obtain a multivariate copula. They are most easily represented in terms of densities and in the following we assume that these all exist. Let $f_{\mathbf{X}}(x_1, \ldots, x_d)$ be the joint density of a random variable $\mathbf{X} = (X_1, \ldots, X_d)$ and $f_1(x_1), \ldots, f_d(x_d)$ its univariate marginal densities. Moreover, let $c(u_1, \ldots, u_d)$ denote the copula density. Subsequently, for given indices i, j, $i_1, \ldots, i_k$, we write for short $f_i := f_i(x_i)$, $c_{ij} := c_{ij}(u_i, u_j)$, $u_{i|i_1,\ldots,i_k} := F_{i|i_1,\ldots,i_k}(x_i | x_{i_1}, \ldots, x_{i_k})$, and $c_{ij|i_1,\ldots,i_k} := c_{ij|i_1,\ldots,i_k}(u_{i|i_1,\ldots,i_k}, u_{j|i_1,\ldots,i_k}; u_{i_1}, \ldots, u_{i_k})$.

Vine copulas have a graphical representation as a multi-layered tree structure, which is called a vine and is thus the namesake of vine copulas. It was developed in Cooke (1997) and Bedford and Cooke (2001, 2002) and it follows specific construction rules to obtain a valid copula density, see, e.g., Czado (2019). Figure 2 shows the graphical representation of an exemplary 5-dimensional vine copula with the following density decomposition

$$f_{\mathbf{X}}(x_1, \ldots, x_5) = c_{12} \cdot c_{23} \cdot c_{34} \cdot c_{35} \cdot c_{13|2} \cdot c_{24|3} \cdot c_{25|3} \cdot c_{14|23} \cdot c_{45|23} \cdot c_{15|234} \cdot \prod_{k=1}^{5} f_k. \tag{17}$$

However, note that this decomposition is not unique, e.g., exchanging variables 3 and 5 leads to a different vine copula. In general, in d dimensions there are $2^{\frac{1}{2}(d-2)(d-3)} \cdot \frac{d!}{2}$ possible vines (Morales-Nápoles 2011), which is a prohibitive large search space. How to determine a good vine structure is an open problem and, thus, heuristics are used for this (Dißmann et al. 2013; Kurowicka 2011).

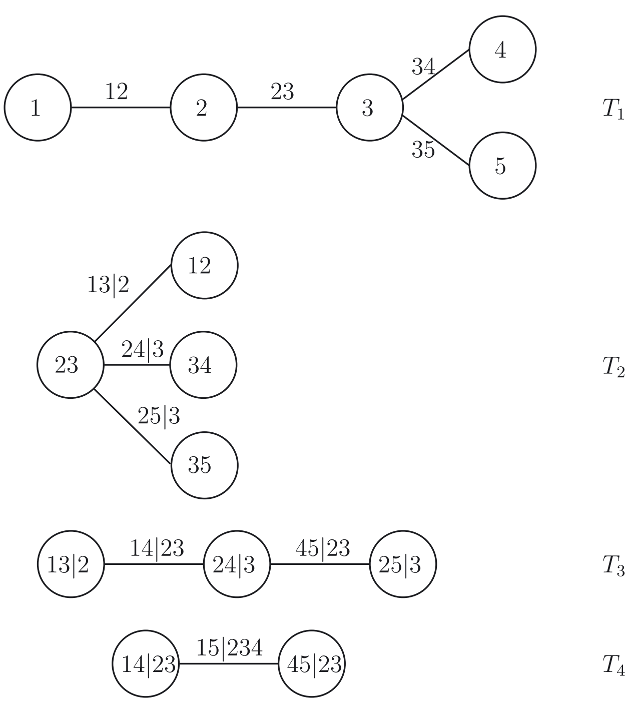

Figure 2: Example of a 5-dimensional vine copula.

C References by topic

Table 3: Overview of references by topic.

Topic	References
Copula estimation	Wiese et al. (2019); Ling et al. (2020); Ng et al. (2021) Kamthe et al. (2021); Janke et al. (2021); Zeng and Wang (2022) Hasan et al. (2022); Mitskopoulos et al. (2022) McDonald et al. (2022); Zhang et al. (2024b); Tian et al. (2024) Liu et al. (2025a); Letizia et al. (2025)
Vine copula estimation	Sun et al. (2019); Zhou et al. (2020); Chen et al. (2022)
Miscellaneous deep learning in copulas	Hofert et al. (2021, 2023b,a); Jutras-Dubé et al. (2024)
Copulas in normalizing flows	Laszkiewicz et al. (2021, 2022); Kirchler et al. (2023) Balgi et al. (2024)
Copulas in GANs	Letizia and Tonello (2022); Boulaguiem et al. (2022) Gharibi et al. (2023)
Copulas in (variational) autoencoders	Suh and Choi (2016); Wieczorek et al. (2018) Wang and Wang (2019); Tagasovska et al. (2019) Wang et al. (2021a); Zhong et al. (2021) Xu and Cao (2023); Kächele et al. (2025)
Copulas in transformers	Song et al. (2021); Drouin et al. (2022); Liu et al. (2022) Ashok et al. (2024); Li et al. (2024a)
Deep learning input	Sun et al. (2018); Huang and Zhang (2019) Ouyang et al. (2019); Luo et al. (2020); Mukherjee et al. (2021) Silva et al. (2021); Liu et al. (2022); Li et al. (2022) Peng et al. (2023); Nafii et al. (2023) Nabizadeh Chianeh et al. (2024); Iqbal and Siddiqi (2024) Kim (2025e); Meng et al. (2025); Akkepalli and K (2025) Kim (2025a)
Processing deep learning output	Yang et al. (2011); Chatzis and Demiris (2012) De Oliveira et al. (2016); Walecki et al. (2017); Xu et al. (2017) Zhang et al. (2019); Salinas et al. (2019); Toubeau et al. (2019) Wen and Torkkola (2019); Chilinski and Silva (2020) Xu et al. (2020); Kuiry et al. (2020); Zhao et al. (2021) Zhang et al. (2021, 2022); Huang et al. (2022) Zhou et al. (2022); Tagasovska et al. (2023); Cui et al. (2023) Dey et al. (2023); Yang et al. (2023); Xiao et al. (2024) Kim et al. (2024)
Survival analysis	Foomani et al. (2023); Zhang et al. (2024b); Kim (2025e,a) Liu et al. (2025b); Kwon et al. (2025)
Miscellaneous copula in deep learning	Dimitriev and Zhou (2021a,b); Kom Samo (2021) Ma et al. (2021); Messoudi et al. (2021); Abbaszadeh et al. (2022) Sun and Yu (2024); Qiu et al. (2024); Li et al. (2024b,a) Sattari et al. (2024); Zhong et al. (2025); Panda and Garain (2025) Simchoni and Rosset (2025)

References

Aas, K. (2016). Pair-copula constructions for financial applications: a review. *Econometrics* 4: 43.

Abbaszadeh, P., Gavahi, K., Alipour, A., Deb, P., and Moradkhani, H. (2022). Bayesian multi-modeling of deep neural nets for probabilistic crop yield prediction. *Agric. For. Meteorol.* 314: 108773.

Akkepalli, S. and K, S. (2025). Copula entropy regularization transformer with C2 variational autoencoder and fine-tuned hybrid DL model for network intrusion detection. *Telemat. Inform. Rep.* 17: 100182.

Alokley, S. A., Araichi, S., and Alomair, G. (2024). Exploring the relationship and predictive accuracy for the Tadawul all share index, oil prices, and bitcoin using copulas and machine learning. *Energies* 17: 3241.
Ashok, A., Marcotte, É., Zantedeschi, V., Chapados, N., and Drouin, A. (2024). TACTiS-2: better, faster, simpler attentional copulas for multivariate time series. In: *International conference on learning representations*, 2024. pp. 22897—22924.
Balgi, S., Pe, J. M., and Daoud, A. (2024). ρ-GNF: a copula-based sensitivity analysis to unobserved confounding using graphical normalizing flows. *Proc. Mach. Learn. Res.* 246: 20—37.
Bauer, A. and Czado, C. (2016). Pair-copula Bayesian networks. *J. Comput. Graph Stat.* 25: 1248—1271.
Bauer, A., Czado, C., and Klein, T. (2012). Pair-copula constructions for non-Gaussian DAG models. *Can. J. Stat.* 40: 86—109.
Bedford, T. and Cooke, R. M. (2001). Probability density decomposition for conditionally dependent random variables modeled by vines. *Ann. Math. Artif. Intell.* 32: 245—268, https://doi.org/10.1023/a:1016725902970.
Bedford, T. and Cooke, R. M. (2002). Vines: a new graphical model for dependent random variables. *Ann. Stat.* 30: 1031—1068.
Boulaguiem, Y., Zscheischler, J., Vignotto, E., van der Wiel, K., and Engelke, S. (2022). Modeling and simulating spatial extremes by combining extreme value theory with generative adversarial networks. *Environmental Data Sci.* 1: 1—18.
Burda, M. and Prokhorov, A. (2014). Copula based factorization in Bayesian multivariate infinite mixture models. *J. Multivariate Anal.* 127: 200—213.
Chang, B. and Joe, H. (2019). Prediction based on conditional distributions of vine copulas. *Comput. Stat. Data Anal.* 139: 45—63.
Chatzis, S. P. and Demiris, Y. (2012). The copula echo state network. *Pattern Recogn.* 45: 570—577.
Chen, R. B., Guo, M., Härdle, W. K., and Huang, S. F. (2015). COPICA — independent component analysis via copula techniques. *Stat. Comput.* 25: 273—288.
Chen, H., Jiao, L., and Li, S. (2022). A soft sensor regression model for complex chemical process based on generative adversarial nets and vine copula. *J. Taiwan Inst. Chem. Eng.* 138: 104483.
Chi, J., Ouyang, J., Zhang, A., Wang, X., and Li, X. (2022). Fast copula variational inference. *J. Exp. Theor. Artif. Intell.* 34: 295—310.
Chilinski, P. and Silva, R. (2020). Neural likelihoods via cumulative distribution functions. In: *Conference on uncertainty in artificial intelligence*, pp. 420—429.
Chollet, F. (2021). *Deep learning with python*. Manning, Shelter Island, NY.
Chollete, L., Heinen, A., and Valdesogo, A. (2009). Modeling international financial returns with a multivariate regime-switching copula. *J. Financ. Econom.* 7: 437—480.
Coblenz, M., Dyckerhoff, R., and Grothe, O. (2018a). Confidence regions for multivariate quantiles. *Water* 10: 996.
Coblenz, M., Dyckerhoff, R., and Grothe, O. (2018b). Nonparametric estimation of multivariate quantiles. *Environmetrics* 29: e2488.
Coblenz, M., Grothe, O., Herrmann, K., and Hofert, M. (2022). Smooth bootstrapping of copula functionals. *Elec. J. Stat.* 16: 2550—2606.
Coblenz, M., Holz, S., Bauer, H.-J., Grothe, O., and Koch, R. (2020). Modelling fuel injector spray characteristics in jet engines by using vine copulas. *J. R. Stat. Soc. Series C: Appl. Stat.* 69: 863—886.
Cooke, R. M. (1997). Markov and entropy properties of tree and Vine dependent variables. In: *Proceedings of the ASA section of Bayesian statistical science*.
Cousin, A. and Di Bernardino, E. (2013). On multivariate extensions of value-at-risk. *J. Multivariate Anal.* 119: 32—46.
Cui, C. (2021). Nonlinear non-Gaussian and multimode probabilistic weighted copula regression model based deep neural network. *Can. J. Chem. Eng.* 99: S432—S442.
Cui, Z., Guo, S., Zhou, Y., and Wang, J. (2023). Exploration of dual-attention mechanism-based deep learning for multi-step-ahead flood probabilistic forecasting. *J. Hydrol.* 622: 129688.
Czado, C. (2019). *Analyzing dependent data with vine copulas: a practical guide with R*. Springer, Cham.
Czado, C., Bax, K., Sahin, O., Nagler, T., Min, A., and Paterlini, S. (2022). Vine copula based dependence modeling in sustainable finance. *J. Financ. Data Sci.* 8: 309—330.
Czado, C. and Nagler, T. (2022). Vine copula based modeling. *Ann. Rev. Stat. Appl.* 9: 453—477.
De Backer, M., El Ghouch, A., and Van Keilegom, I. (2017). Semiparametric copula quantile regression for complete or censored data. *Elec. J. Stat.* 11: 1660—1698.
De Oliveira, R. T., De Assis, T. F. O., Firmino, P. R. A., Ferreira, T. A., and Oliveira, A. L. (2016). Copulas-based ensemble of artificial neural networks for forecasting real world time series. In: *Proceedings of the international joint conference on neural networks*, pp. 4089—4096.
Dette, H., Van Hecke, R., and Volgushev, S. (2014). Some comments on copula-based regression. *J. Am. Stat. Assoc.* 109: 1319—1324.
Dey, S., Mitra, S., Chakraborty, S., Mondal, D., Nasipuri, M., and Das, N. (2023). GC-EnC: a copula based ensemble of CNNs for malignancy identification in breast histopathology and cytology images. *Comput. Biol. Med.* 152: 106329.
Dimitriev, A. and Zhou, M. (2021a). ARMS: antithetic-reinforce-multi-sample gradient for binary variables. In: *International conference on machine learning*, pp. 2717—2727.
Dimitriev, A. and Zhou, M. (2021b). CARMS: categorical-antithetic-reinforce multi-sample gradient estimator. In: *Advances in Neural Information Processing Systems*, pp. 13217—13229.
Dißmann, J., Brechmann, E., Czado, C., and Kurowicka, D. (2013). Selecting and estimating regular Vine copulae and application to financial returns. *Comput. Stat. Data Anal.* 59: 52—69.

Dobra, A. and Lenkoski, A. (2011). Copula Gaussian graphical models and their application to modeling functional disability data. *Ann. Appl. Stat.* 5: 969–993.

Drouin, A., Marcotte, É., and Chapados, N. (2022). TACTiS : Transformer-attentional copulas for time series. In: *Proceedings of the 39th International Conference on Machine Learning*, pp. 5447–5493.

Durante, F. and Sempi, C. (2015). *Principles of Copula Theory*. CRC/Chapman & Hall, Boca Raton, FL.

Dziugaite, G. K., Roy, D. M., and Ghahramani, Z. (2015). Training generative neural networks via maximum mean discrepancy optimization. *Conf. Uncertain. Artif. Intell.*: 258–267.

Eban, E., Rothschild, G., Mizrahi, A., Nelken, I., and Elidan, G. (2013). Dynamic copula networks for modeling real-valued time series. *J. Mach. Learn. Res.* 31: 247–255.

Elidan, G. (2010a). Copula bayesian networks. In: *Advances in neural information processing*, pp. 1–9.

Elidan, G. (2010b). Inference-less density estimation using Copula Bayesian Networks. In: *Conference on uncertainty in artificial intelligence*, pp. 151–159.

Elidan, G. (2012a). Copula network classifiers (CNCs). In: *International Conference on Artificial Intelligence and Statistics*, pp. 346–354.

Elidan, G. (2012b). Lightning-speed structure learning of nonlinear continuous networks. *J. Mach. Learn. Res.* 22: 355–363.

Elidan, G. (2013). Copulas in machine learning. In: *Copulae in mathematical and quantitative finance*, pp. 39–60.

Foomani, A. H. G., Cooper, M., Greiner, R., and Krishnan, R. G. (2023). Copula-Based deep survival models for dependent censoring. In: *Proceedings of the 39th conference on uncertainty in artificial intelligence*, pp. 669–680.

Fujimaki, R., Sogawa, Y., and Morinaga, S. (2011). Online heterogeneous mixture modeling with marginal and copula selection. In: *Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 645–653.

Geenens, G., Charpentier, A., and Paindaveine, D. (2017). Probit transformation for nonparametric kernel estimation of the copula density. *Bernoulli* 23: 1848–1873.

Genest, C. and Favre, A.-C. (2007). Everything you always wanted to know about Copula modeling but were afraid to ask. *J. Hydrol. Eng.* 12: 347–368, https://doi.org/10.1061/(asce)1084-0699(2007)12:4(347).

Gharibi, M. A., Nafisi, H., Askarian-abyaneh, H., and Hajizadeh, A. (2023). Deep learning framework for day-ahead optimal charging scheduling of electric vehicles in parking lot. *Appl. Energy* 349: 121614.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT Press, Cambridge, MA.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In: Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger. (Eds.). *Advances in neural information processing systems*, Vol. 27. Curran Associates, Inc., New York. https://proceedings.neurips.cc/paper_files/paper/2014/file/f033ed80deb0234979a61f95710dbe25-Paper.pdf.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2020). Generative adversarial networks. *Commun. ACM* 63: 139–144.

Gretton, A., Borgwardt, K. M., Rasch, M., Schölkopf, B., and Smola, A. J. (2007). A kernel method for the two-sample-problem. In: *Advances in neural information processing*, pp. 513–520.

Gudendorf, G. and Segers, J. (2010). Extreme-value copulas. In: Jaworski, P., Durante, F., Härdle, W. K., and Rychlik, T. (Eds.). *Copula theory and its applications*. Springer, Berlin, Heidelberg, pp. 127–145.

Hamilton, W. L., Ying, R., and Leskovec, J. (2017). Inductive representation learning on large graphs. In: *Conference on neural information processing systems*, pp. 1–11.

Han, S., Liao, X., Dunson, D. B., and Carin, L. (2016). Variational Gaussian copula inference. In: *International conference on artificial intelligence and statistics*, pp. 829–838.

Han, F., Zhao, T., and Liu, H. (2013). CODA: high dimensional copula discriminant analysis. *J. Mach. Learn. Res.* 14: 629–671.

Hasan, A., Elkhalil, K., Ng, Y., Pereira, J. M., Farsiu, S., Blanchet, J., and Tarokh, V. (2022). Modeling extremes with d-max-decreasing neural networks. In: *Conference on uncertainty in artificial intelligence*, pp. 759–768.

Hernández, H., Díaz-Viera, M. A., Alberdi, E., and Goti, A. (2025). Comparison of trivariate copula-based conditional quantile regression versus machine learning methods for estimating copper recovery. *Mathematics* 13: 1–22.

Hirt, M., Dellaportas, P., and Durmus, A. (2019). Copula-like variational inference. In: *Conference on neural information processing systems*, pp. 1–13.

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. In: *Advances in neural information processing systems*, pp. 1–12.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.* 9: 1735–1780.

Hofert, M. (2008). Sampling Archimedean copulas. *Comput. Stat. Data Anal.* 52: 5163–5174.

Hofert, M. (2011). Efficiently sampling nested Archimedean copulas. *Comput. Stat. Data Anal.* 55: 57–70.

Hofert, M., Prasad, A., and Zhu, M. (2021). Quasi-random sampling for multivariate distributions via generative neural networks. *J. Comput. Graph Stat.* 30: 647–670.

Hofert, M., Prasad, A., and Zhu, M. (2023a). Dependence model assessment and selection with DecoupleNets. *J. Comput. Graph Stat.* 32: 1272–1286.

Hofert, M., Prasad, A., and Zhu, M. (2023b). RafterNet: probabilistic predictions in multi-response regression. *Am. Stat.* 77: 406–416.

Huang, J. C. and Frey, B. J. (2011). Cumulative distribution networks and the derivative-sum-product algorithm: models and inference for cumulative distribution functions on graphs. *J. Mach. Learn. Res.* 12: 301—348.
Huang, C. Y. and Zhang, Y. P. (2019). Prediction based on copula entropy and general regression neural network. *Appl. Ecol. Environ. Res.* 17: 14415—14424.
Huang, Y., Zhang, B., Pang, H., Wang, B., Lee, K. Y., Xie, J., and Jin, Y. (2022). Spatio-temporal wind speed prediction based on Clayton Copula function with deep learning fusion. *Renew. Energy* 192: 526—536.
Huk, D., Steel, M., and Dutta, R. (2025). Your copula is a classifier in disguise: classification-based copula density estimation. In: *The 28th International conference on artificial intelligence and statistics*, pp. 1—12.
Hyvärinen, A. (2005). Estimation of non-normalized statistical models by score matching. *J. Mach. Learn. Res.* 6: 695—709.
Iqbal, A. and Siddiqi, T. A. (2024). Multi-phase hybrid bidirectional deep learning model integrated with Markov chain Monte Carlo bivariate copulas function for streamflow prediction. *Stoch. Environ. Res. Risk Assess.* 38: 1351—1382.
Janke, T., Ghanmi, M., and Steinke, F. (2021). Implicit generative copulas. In: *Advances in neural information processing systems*, pp. 26028—26039.
Jiang, C., Zhang, W., Han, X., Ni, B. Y., and Song, L. J. (2015). A vine-copula-based reliability analysis method for structures with multidimensional correlation. *J. Mech. Des.* 137: 061405.
Joe, H. (2015). *Dependence Modeling with Copulas*. Chapman & Hall, Boca Raton, FL.
Joe, H. and Xu, J. J. (1996). *The estimation method of inference functions for margins for multivariate models*, Technical report no. 166. Department of Statistics, University of British Columbia.
Jutras-Dubé, P., Al-Khasawneh, M. B., Yang, Z., Bas, J., Bastin, F., and Cirillo, C. (2024). Copula-based transferable models for synthetic population generation. *Transport. Res. Part C: Emerg. Technol.* 169: 104830.
Kächele, F., Coblenz, M., and Grothe, O. (2025). A comparison of latent space modeling techniques in a plain-vanilla autoencoder setting. *Mach. Learn.* 114: 1—35.
Kamthe, S., Assefa, S., and Deisenroth, M. (2021). Copula flows for synthetic data generation, Working paper.
Karra, K. and Mili, L. (2016). Hybrid copula Bayesian networks. *J. Mach. Learn. Res.* 52: 240—251.
Kim, J. M. (2025a). A copula-driven CNN-LSTM framework for estimating heterogeneous treatment effects in multivariate outcomes. *Mathematics* 13: 1—20.
Kim, J. M. (2025b). Gaussian process with Vine Copula-based context modeling for contextual multi-armed bandits. *Mathematics* 13: 1—18.
Kim, J. M. (2025c). Integrating copula-based random forest and deep learning approaches for analyzing heterogeneous treatment effects in survival analysis. *Mathematics* 13: 1—29.
Kim, J. M. (2025d). LLM-guided ensemble learning for contextual bandits with copula and Gaussian process models. *Mathematics* 13: 1—18.
Kim, J.-M. (2025e). Treatment effect estimation in survival analysis using copula-based deep learning models for causal inference. *Axioms* 14: 458.
Kim, J. M., Ha, I. D., and Kim, S. (2024). Copula deep learning control chart for multivariate zero inflated count response variables. Statistics 58: 749—769.
Kim, J. M., Kim, D., Liao, S. M., and Jung, Y. S. (2013). Mixture of D-vine copulas for modeling dependence. *Comput. Stat. Data Anal.* 64: 1—19.
Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In: *International conference on learning representations*, pp. 1—14.
Kingma, D. P. and Welling, M. (2019). An introduction to variational autoencoders. *Found. Trends Mach. Learn.* 12: 307—392.
Kipf, T. N. and Well (2017). Semi-supervised classification with graph convolutional networks. In: *International conference on learning representations*, pp. 1—11.
Kirchler, M., Lippert, C., and Kloft, M. (2023). Training normalizing flows from dependent data. In: *Proceedings of the 40th international conference on machine learning*, pp. 17105—17121.
Kirshner, S. (2007). Learning with tree-averaged densities and distributions. In: *Advances in neural information processing systems*, pp. 761—768.
Kirshner, S. and Póczos, B. (2008). ICA and ISA using Schweizer-Wolff measure of dependence. In: *International conference on machine learning*, pp. 464—471.
Kobyzev, I., Prince, S. J., and Brubaker, M. A. (2021). Normalizing flows: an introduction and review of current methods. *IEEE Trans. Pattern Anal. Mach. Intell.* 43: 3964—3979.
Kom Samo, Y.-L. (2021). Inductive mutual information estimation: a convex maximum-entropy copula approach. In: *Proceedings of the 24th international conference on artificial intelligence and statistics*, pp. 2242—2250.
Kramer, M. A. (1991). Nonlinear principal component analysis using autoassociative neural networks. *AIChE J.* 37: 233—243.
Kramer, M. (1992). Autoassociative neural networks. *Comput. Chem. Eng.* 16: 313—328.
Kraus, D. and Czado, C. (2017). D-vine copula based quantile regression. *Comput. Stat. Data Anal.* 110: 1—18.
Kuiry, S., Das, N., Das, A., and Nasipuri, M. (2020). EDC3: ensemble of deep-classifiers using class-specific copula functions to improve semantic image segmentation. Working paper.
Kulkarni, V., Tagasovska, N., Vatter, T., and Garbinato, B. (2018). Generative models for simulating mobility trajectories. In: *Workshop on modeling and decision-making in the Spatiotemporal Domain, 32nd conference on neural information processing systems*.

Kurowicka, D. (2011). Optimal truncation of vines. In: Kurowicka, D., and Joe, H. (Eds.). *Dependence modeling: vine copula handbook*. World Scientific, Singapore, pp. 233—247.

Kwon, J., Emura, T., and Ha, I. D. (2025). Copula-based deep neural networks for clustered survival data. *Commun. Stat. Appl. Methods* 32: 389—400.

Laszkiewicz, M., Lederer, J., and Fischer, A. (2021). Copula-based normalizing flows. In: *Third Workshop on invertible neural networks, normalizing flows, and explicit likelihood models*.

Laszkiewicz, M., Lederer, J., and Fischer, A. (2022). Marginal tail-adaptive normalizing flows. In: *Proceedings of the 39th international conference on machine learning*, pp. 12020—12048.

Letizia, N. A., Novello, N., and Tonello, A. M. (2025). Copula density neural estimation. IEEE Transact. Neural Networks Learn. Syst 36: 19452—19459.

Letizia, N. A. and Tonello, A. M. (2022). Segmented generative networks: data generation in the uniform probability space. *IEEE Transact. Neural Networks Learn. Syst.* 33: 1338—1347.

Li, Y., Deng, J., Zhong, C., Yang, D., Li, M., Welsh, A. H., Liu, A., Zhou, X., Liu, C. C., and Fu, B. (2024a). OU-CoViT: Copula-Enhanced Bi-Channel Multi-Task Vision Transformers with Dual Adaptation for OU-UWF Images. Working paper.

Li, Y., Huang, Q., Zhong, C., Yang, D., Li, M., Welsh, A. H., Fu, B., Liu, C. C., and Zhou, X. (2024b). OUCopula: bi-channel multi-label copula-enhanced adapter-based CNN for myopia screening based on OU-UWF images. In: *International joint conference on artificial intelligence*, pp. 5927—5935.

Li, Y., Swersky, K., and Zemel, R. (2015). Generative moment matching networks. In: *International conference on machine learning*, pp. 1718—1727.

Li, W., Wang, X., Li, G., Geng, B, and Varshney, PK (2025). NN-Copula-CD: a copula-guided interpretable neural network for change detection in heterogeneous remote sensing images. *IEEE Trans. Geosci. Rem. Sens.* 63: 1—17.

Li, X., Zhang, L., Zeng, S., Tang, Z., Liu, L., Zhang, Q., Tang, Z., and Hua, X. (2022). Predicting monthly runoff of the upper Yangtze River based on multiple machine learning models. *Sustainability* 14: 11149.

Li, Z., Zhao, Y., Botta, N., Ionescu, C., and Hu, X. (2020). COPOD: Copula-based outlier detection. In: *IEEE International conference on data mining*, pp. 1118—1123.

Ling, C. K., Fang, F., and Kolter, J. Z. (2020). Deep archimedean copulas. In: *Advances in neural information processing systems*, pp. 1535—1545.

Liu, B., Coblenz, M., and Grothe, O. (2025a). The copula via transformation representation and its estimation with normalizing flows. Working paper, submitted for publication.

Liu, P., Han, S., Rong, N., and Fan, J. (2022). Frequency stability prediction of power systems using vision transformer and copula entropy. *Entropy* 24: 1165.

Liu, H., Han, F., Yuan, M., Lafferty, J., and Wasserman, L. (2012). High-dimensional semiparametric Gaussian copula graphical models. *Ann. Stat.* 40: 2293—2326.

Liu, H., Lafferty, J., and Wasserman, L. (2009). The nonparanormal: semiparametric estimation of high dimensional undirected graphs. *J. Mach. Learn. Res.* 10: 2295—2328.

Liu, X., Zhang, W., and Zhang, M.-L. (2025b). HACSurv: a hierarchical copula-based approach for survival analysis with dependent competing risks. In: *28th International conference on artificial intelligence and statistics*, pp. 1—9.

Lopez-Paz, D., Hernández-Lobato, J. M., and Schölkopf, B. (2012). Semi-supervised domain adaptation with non-parametric copulas. In: *Advances in neural information processing systems*, pp. 665—673.

Luo, Z., Liu, C., and Liu, S. (2020). A novel fault prediction method of Wind Turbine Gearbox based on pair-copula construction and BP neural network. *IEEE Access* 8: 91924—91939.

Ma, J., Chang, B., Zhang, X., and Mei, Q. (2021). Copulagnn: towards integrating representational and correlational roles of graphs in graph neural networks. In: *International conference on learning representations*.

Ma, J. and Sun, Z. (2007). Copula component analysis. In: *Independent component analysis and signal separation*, pp. 73—80.

Ma, J. and Sun, Z. (2011). Mutual information is copula entropy. *Tsinghua Sci. Technol.* 16: 51—54.

Madadgar, S. and Moradkhani, H. (2014). Improved Bayesian multimodeling: integration of copulas and Bayesian model averaging. *Water Resour. Res.* 50: 9586—9603.

McDonald, A., Tan, P. N., and Luo, L. (2022). COMET flows: towards generative modeling of multivariate extremes and tail dependence. In: *International joint conference on artificial intelligence*, pp. 3328—3334.

Meng, Z., Ma, S., Cao, W., Li, J., Cao, L., Fan, F., and Wang, X. (2025). A remaining useful life prediction method of rolling bearings by RSA-BAFT combined with Copula Entropy feature selection. *Expert Syst. Appl.* 275: 127100.

Mesfioui, M., Bouezmarni, T., and Belalia, M. (2023). Copula-based link functions in binary regression models. *Stat. Pap.* 64: 557—585.

Messoudi, S., Destercke, S., and Rousseau, S. (2021). Copula-based conformal prediction for multi-target regression. *Pattern Recogn.* 120: 108101.

Mitskopoulos, L., Amvrosiadis, T., and Onken, A. (2022). Mixed Vine copula flows for flexible modeling of neural dependencies. *Front. Neurosci.* 16: 910122.

Mohamed, S. and Lakshminarayanan, B. (2017). Learning in implicit generative models. Working paper.

Morales-Nápoles, O. (2011). Counting vines. In: Kurowicka, D., and Joe, H. (Eds.). *Dependence modeling: vine copula handbook*. World Scientific, Singapore, pp. 189—218.

Mukherjee, D., Chakraborty, S., Ghosh, S., and Mishra, R. K. (2021). Application of deep learning for power system state forecasting. *Int. Trans. Electr. Energy Syst.* 31: 1—26.

Müller, D. and Czado, C. (2018). Representing sparse gaussian DAGs as sparse R-Vines allowing for non-Gaussian dependence. *J. Comput. Graph Stat.* 27: 334—344.

Nabizadeh Chianeh, F., Valikhan Anaraki, M., Mahmoudian, F., and Farzin, S. (2024). A new methodology for the prediction of optimal conditions for dyes' electrochemical removal; Application of copula function, machine learning, deep learning, and multi-objective optimization. *Process Saf. Environ. Prot.* 182: 298—313.

Nafii, A., Lamane, H., Taleb, A., and El Bilali, A. (2023). An approach based on multivariate distribution and Gaussian copulas to predict groundwater quality using DNN models in a data scarce environment. *MethodsX* 10: 102034.

Nalisnick, E., Smyth, P., and Tran, D. (2023). A brief tour of deep learning from a statistical perspective. *Ann. Rev. Stat. Appl.* 10: 219—246.

Nelsen, R. B. (2006). *An introduction to copulas*. Springer Series in Statistics. Springer New York, New York, NY.

Ng, Y., Hasan, A., Elkhalil, K., and Tarokh, V. (2021). Generative archimedean copulas. In: *Conference on uncertainty in artificial intelligence*, pp. 643—653.

Noh, H., Ghouch, A. E., and Bouezmarni, T. (2013). Copula-Based regression estimation and inference. *J. Am. Stat. Assoc.* 108: 676—688.

Noh, H., Ghouch, A. E., and Van Keilegom, I. (2015). Semiparametric conditional quantile estimation through copula-based multivariate models. *J. Bus. Econ. Stat.* 33: 167—178.

Omelka, M., Gijbels, I., and Veraverbeke, N. (2009). Improved kernel estimation of copulas: weak convergence and goodness-of-fit testing. *Ann. Stat.* 37: 3023—3058.

Ouyang, T., He, Y., Li, H., Sun, Z., and Baek, S. (2019). Modeling and forecasting short-term power load with copula model and deep belief network. *IEEE Trans. Emerg. Top. Comput. Intell.* 3: 127—136.

Ozdemir, O., Allen, T. G., Choi, S., Wimalajeewa, T., and Varshney, P. K. (2018). Copula based classifier fusion under statistical dependence. *IEEE Trans. Pattern Anal. Mach. Intell.* 40: 2740—2748.

Panda, A. and Garain, U. (2025). Copula based trainable calibration error estimator of multi-label classification with label interdependencies. In: *The 28th international conference on artificial intelligence and statistics*, pp. 1—9.

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. (2021). Normalizing flows for probabilistic modeling and inference. *J. Mach. Learn. Res.* 22: 1—64.

Patton, A. J. (2006). Modelling asymmetric exchange rate dependence. *Int. Econ. Rev.* 47: 527—556.

Peng, H., Fang, G., and Li, P. (2023). Copula for instance-wise feature selection and ranking. In: *Proceedings of the 39th conference on uncertainty in artificial intelligence*, pp. 1651—1661.

Pircalabelu, E., Claeskens, G., and Gijbels, I. (2017). Copula directed acyclic graphs. *Stat. Comput.* 27: 55—78.

Puccetti, G., Rüschendorf, L., and Manko, D. (2016). VaR bounds for joint portfolios with dependence constraints. *Depend. Model.* 4: 368—381.

Qiu, X., Qian, J., Wang, H., Tan, X., and Jin, Y. (2024). An attentive Copula-based spatio-temporal graph model for multivariate time-series forecasting. *Appl. Soft Comput.* 154: 111324.

Rajan, V. and Bhattacharya, S. (2016). Dependency clustering of mixed data with Gaussian mixture copulas. In: *International joint conference on artificial intelligence*, pp. 1967—1973.

Rényi, A. (1961). On measures of entropy and information. In: *Proceedings of the fourth Berkeley symposium on mathematics, statistics, and probability*, pp. 547—561.

Rey, M. and Roth, V. (2012). Copula mixture model for dependency-seeking clustering. In: *International conference on machine learning*, pp. 927—934.

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In: *International conference on machine learning*, pp. 3057—3070.

Rosenblatt, M. (1952). Remarks on a multivariate transformation. *Ann. Math. Stat.* 23: 470—472.

Roy, A. and Parui, S. K. (2014). Pair-copula based mixture models and their application in clustering. *Pattern Recogn.* 47: 1689—1697.

Sahin, Ö. and Czado, C. (2022). Vine copula mixture models and clustering for non-Gaussian data. *Econom. Stat.* 22: 136—158.

Salinas, D., Bohlke-Schneider, M., Callot, L., Medico, R., and Gasthaus, J. (2019). High-dimensional multivariate forecasting with low-rank Gaussian copula processes. In: *Advances in neural information processing systems*.

Salinas, D., Shen, H., and Perrone, V. (2020). A quantile-based approach for hyperparameter transfer learning. In: *International conference on machine learning*, pp. 8407—8417.

Salvadori, G. (2004). Bivariate return periods via 2-Copulas. *Stat. Methodol.* 1: 129—144.

Salvadori, G. and De Michele, C. (2004). Frequency analysis via copulas: theoretical aspects and applications to hydrological events. *Water Resour. Res.* 40: 1—17.

Salvadori, G., Durante, F., De Michele, C., Bernardi, M., and Petrella, L. (2016). A multivariate copula-based framework for dealing with hazard scenarios and failure probabilities. *Water Resour. Res.* 52: 3701—3721.

Salvadori, G., Durante, F., Tomasicchio, G. R., and D'Alessandro, F. (2015). Practical guidelines for the multivariate assessment of the structural risk in coastal and off-shore engineering. *Coast. Eng.* 95: 77—83.

Salvadori, G., Tomasicchio, G. R., and D'Alessandro, F. (2014). Practical guidelines for multivariate analysis and design in coastal and off-shore engineering. *Coast. Eng.* 88: 1−14.

Sattari, A., Jafarzadegan, K., and Moradkhani, H. (2024). Enhancing streamflow predictions with machine learning and Copula-Embedded Bayesian model averaging. *J. Hydrol.* 643: 131986.

Schepsmeier, U. and Czado, C. (2016). Dependence modelling with regular Vine copula models: a case-study for car crash simulation data. *J. R. Stat. Soc. Series C: Appl. Stat.* 65: 415−429.

Schmidhuber, J. (2015). Deep Learning in neural networks: an overview. *Neural Netw.* 61: 85−117.

Schölkopf, B. and Smola, A. J. (2002). *Learning with Kernels*. MIT Press.

Shafer, G. and Vovk, V. (2008). A tutorial on conformal prediction. *J. Mach. Learn. Res.* 9.

Silva, R., Blundell, C., and Teh, Y. W. (2011). Mixed cumulative distribution networks. In: *International conference on artificial intelligence and statistics*, pp. 670−678.

Silva, R. and Gramacy, R. (2009). MCMC methods for Bayesian mixtures of copulas. In: *International conference on artificial intelligence and statistics*, pp. 512−519.

Silva, D., Leonhardt, S., and Antink, C. H. (2021). Copula-based data augmentation on a deep learning architecture for cardiac sensor fusion. *IEEE J. Biomed. Health Inform.* 25: 2521−2532.

Simchoni, G. and Rosset, S. (2025). Flexible copula-based mixed models in deep learning: a scalable approach to arbitrary marginals. In: *The 28th international conference on artificial intelligence and statistics*, pp. 1−11.

Sklar, A. (1959). Fonctions de répartition à n dimensions et leurs marges. *Publ. Inst. Statist. Univ. Paris* 8: 229−231.

Smith, M. S. and Loaiza-Maya, R. (2023). Implicit Copula variational inference. *J. Comput. Graph Stat.* 32: 769−781.

Sohl-Dickstein, J., Weiss, E. A., Maheswaranathan, N., and Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. In: *32nd International conference on machine learning*, pp. 1−10.

Song, Y. and Ermon, S. (2019). Generative modeling by estimating gradients of the data distribution. In: *Advances in neural information processing systems*.

Song, Y. and Ermon, S. (2020). Improved techniques for training score-based generative models. In: *Advances in neural information processing systems*.

Song, K., Jung, Y., Kim, D., and Moon, I. C. (2021). Implicit kernel attention. In: *AAAI Conference on artificial intelligence*, pp. 9713−9721.

Song, Y. and Kingma, D. P. (2021). How to train your energy-based models. Working paper.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2020). Score-based generative modeling through stochastic differential equations. Working paper.

Su, C. L., Nešlehová, J. G., and Wang, W. (2019). Modelling hierarchical clustered censored data with the hierarchical Kendall copula. *Can. J. Stat.* 47: 182−203.

Suh, S. and Choi, S. (2016). Gaussian copula variational autoencoders for mixed data. Working paper.

Sun, Y., Cuesta-Infante, A., and Veeramachaneni, K. (2019). Learning Vine copula models for synthetic data generation. In: *AAAI Conference on artificial intelligence*, pp. 5049−5057.

Sun, M., Konstantelos, I., and Strbac, G. (2018). A deep learning-based feature extraction framework for system security assessment. *IEEE Trans. Smart Grid* 10: 5007−5020.

Sun, S. and Yu, R. (2024). Copula conformal prediction for multi-step time series forecasting. In: *International conference on learning representations*, pp. 1−17.

Tagasovska, N., Ackerer, D., and Vatter, T. (2019). Copulas as high-dimensional generative models: vine copula autoencoders. In: *Advances in neural information processing systems*.

Tagasovska, N., Ozdemir, F., and Brando, A. (2023). Retrospective uncertainties for deep models using Vine copulas. In: *International conference on artificial intelligence and statistics*, pp. 7528−7539.

Tenzer, Y. and Elidan, G. (2013). Speedy model selection (SMS) for copula models. In: *Conference on uncertainty in artificial intelligence*, pp. 625−634.

Tenzer, Y. and Elidan, G. (2014). HELM: highly efficient learning of mixed copula networks. In: *Conference on uncertainty in artificial intelligence*, pp. 790−799.

Tian, J., Xue, H., Xue, Y., and Fang, P. (2024). Copula-nested spectral kernel network. In: *International conference on machine learning*, pp. 1−15.

Toubeau, J. F., Bottieau, J., Vallee, F., and De Greve, Z. (2019). Deep learning-based multivariate probabilistic forecasting for short-term scheduling in power markets. *IEEE Trans. Power Syst.* 34: 1203−1215.

Tran, D., Blei, D. M., and Airoldi, E. M. (2015). Copula variational inference. In: *Advances in neural information processing systems*, pp. 295−310.

van den Oord, A., Li, Y., and Vinyals, O. (2018). Representation learning with contrastive predictive coding. Working paper.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In: *Advances in neural information processing systems*.

Veeramachaneni, K., Cuesta-Infante, A., and O'Reilly, U. M. (2015). Copula graphical models for wind resource estimation. In: *International joint conference on artificial intelligence*, pp. 2646−2654.

Walecki, R., Rudovic, O., Schuller, B., Pavlovic, V., and Pantic, M. (2017). Deep structured learning for facial expression intensity estimation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3405−3414.

Wang, G., Cao, L., Zhao, H., Liu, Q., and Chen, E. (2021a). Coupling macro-sector-micro financial indicators for learning stock representations with less uncertainty. In: *AAAI Conference on artificial intelligence*, pp. 4418−4426.

Wang, P. Z. and Wang, W. Y. (2019). Neural Gaussian copula for variational autoencoder. In: *Conference on Empirical methods in natural language processing and 9th international joint conference on natural language processing*, pp. 4333−4343.

Wang, H., Yu, L., Cao, Z., and Ermon, S. (2021b). Multi-agent imitation learning with copulas. In: *Machine learning and knowledge discovery in databases*, pp. 139−156.

Wen, R. and Torkkola, K. (2019). Deep generative quantile-copula models for probabilistic forecasting. In: *Proceedings of the 36th international conference on machine learning*.

Wen, R., Torkkola, K., Narayanaswamy, B., and Madeka, D. (2017). A multi-horizon quantile recurrent forecaster. In: *NIPS time series workshop*, pp. 1−9.

Wieczorek, A., Wieser, M., Murezzan, D., and Roth, V. (2018). Learning sparse latent representations with the deep copula information bottleneck. In: *International conference on learning representations*.

Wiese, M., Knobloch, R., and Korn, R. (2019). Copula & Marginal Flows: Disentangling the Marginal from its Joint. Working paper.

Xiao, Q., Liu, Y., Chen, C., Zhou, L., Liu, Z., Jiang, Z., Yang, B., and Tang, L. (2024). Time-variant reliability assessment for bridge structures based on deep learning and regular Vine copula models. *Meas.: J. Int. Meas. Confed.* 237: 115253.

Xu, J. and Cao, L. (2023). Copula variational LSTM for high-dimensional cross-market multivariate dependence modeling. *IEEE Transact. Neural Networks Learn. Syst.*: 1−15.

Xu, Q., Fan, Z., Jia, W., and Jiang, C. (2020). Fault detection of wind turbines via multivariate process monitoring based on vine copulas. *Renew. Energy* 161: 939−955.

Xu, Y., Shi, L., and Ni, Y. (2017). Deep-learning-based scenario generation strategy considering correlation between multiple wind farms. *J. Eng.* 2017: 2207−2210.

Yang, H., Xu, W., Zhao, J., Wang, D., and Dong, Z. (2011). Predicting the probability of ice storm damages to electricity transmission facilities based on ELM and Copula function. *Neurocomputing* 74: 2573−2581.

Yang, Y., Zhao, Z., and Cao, L. (2023). Deep spectral Copula mechanisms modeling coupled and volatile multivariate time series. In: *IEEE 10th international conference on data science and advanced analytics*, pp. 1−10.

Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., and Leskovec, J. (2018). Graph convolutional neural networks for web-scale recommender systems. In: *Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 974−983.

Zeng, Z. and Wang, T. (2022). Neural Copula: a unified framework for estimating generic high-dimensional copula functions. Working paper.

Zhang, B., Li, Q., Pang, H., Xu, J., and Huang, Y. (2021). Wind speed prediction for wind farm based on Clayton Copula function and deep learning fusion. In: *Proceedings of 2021 IEEE 4th international electrical and energy conference*, pp. 1−6.

Zhang, S., Geng, B., Varshney, P. K., and Rangaswamy, M. (2019). Fusion of deep neural networks for activity recognition: a regular vine copula based approach. In: *International conference on information fusion*, pp. 1−7.

Zhang, J., Han, S., Li, M., Li, H., Zhao, W., Wang, J., and Liang, H. (2024a). CasMDN: a deep learning-based multivariate distribution modelling approach and its application in geotechnical engineering. *Comput. Geotech.* 168: 106164.

Zhang, W., Ling, C. K., and Zhang, X. (2024b). Deep copula-based survival analysis for dependent censoring with identifiability guarantees. In: *Proceedings of the AAAI conference on artificial intelligence*, pp. 20613−20621.

Zhang, B., Wang, S., Qing, Y., Zhu, J., Wang, D., and Liu, J. (2022). A vine copula-based polynomial chaos framework for improving multi-model hydroclimatic projections at a multi-decadal convection-permitting scale. *Water Resour. Res.* 58: e2022WR031954.

Zhao, X., Bai, M., Yang, X., Liu, J., Yu, D., and Chang, J. (2021). Short-term probabilistic predictions of wind multi-parameter based on one-dimensional convolutional neural network with attention mechanism and multivariate copula distribution estimation. *Energy* 234: 121306.

Zhong, C., Li, Y., Yang, D., Li, M., Zhou, X., Fu, B., Liu, C. C., and Welsh, A. H. (2025). Cecnn: copula-enhanced convolutional neural networks in joint prediction of refraction error and axial length based on ultra-widefield fundus images. *Ann. Appl. Stat.* 19: 1292−1313.

Zhong, T., Wang, G., Walker, J., Zhang, K., and Zhou, F. (2021). Variational autoencoder with copula for collaborative filtering. In: *Workshop on deep learning practice for high dimensional sparse data with KDD*, pp. 1−4.

Zhou, Y., Ren, X., and Li, S. (2020). Nonlinear non-Gaussian and multimode process monitoring-based multi-subspace vine copula and deep neural network. *Ind. Eng. Chem. Res.* 59: 14385−14397.

Zhou, N., Xu, X., Yan, Z., and Shahidehpour, M. (2022). Spatio-temporal probabilistic forecasting of photovoltaic power based on monotone broad learning system and copula theory. *IEEE Trans. Sustain. Energy* 13: 1874−1885.