Inference and Sampling for Archimax Copulas

Yuting Ng*
Duke University
yuting.ng@duke.edu

Ali Hasan*
Duke University
ali.hasan@duke.edu

Vahid Tarokh
Duke University
vahid.tarokh@duke.edu

Abstract

Understanding multivariate dependencies in both the bulk and the tails of a distribution is an important problem for many applications, such as ensuring algorithms are robust to observations that are infrequent but have devastating effects. Archimax copulas are a family of distributions endowed with a precise representation that allows simultaneous modeling of the bulk and the tails of a distribution. Rather than separating the two as is typically done in practice, incorporating additional information from the bulk may improve inference of the tails, where observations are limited. Building on the stochastic representation of Archimax copulas, we develop a non-parametric inference method and sampling algorithm. Our proposed methods, to the best of our knowledge, are the first that allow for highly flexible and scalable inference and sampling algorithms, enabling the increased use of Archimax copulas in practical settings. We experimentally compare to state-of-the-art density modeling techniques, and the results suggest that the proposed method effectively extrapolates to the tails while scaling to higher dimensional data. Our findings suggest that the proposed algorithms can be used in a variety of applications where understanding the interplay between the bulk and the tails of a distribution is necessary, such as healthcare and safety.

1 Introduction

Modeling dependencies between random variables is a central task in statistics, and understanding the dependence between covariates throughout the distribution is important in characterizing a distribution outside of its areas of highest density. For example, in machine learning contexts, a major topic of interest lies in enforcing dependencies between covariates, such as spatial dependence in convolutional neural networks or temporal dependence in recurrent neural networks. Copulas are functions that model the joint dependence of random variables, and they have been successfully employed in a variety of practical modeling settings due to their ease of use and intuitive properties.

Moreover, since copulas are used to represent cumulative distribution functions (CDFs), they have been particularly useful in situations where *tail events* are important – events that have high impact but low probability. For example, in computer vision applications, local dependence modeled by convolutions may be sufficient for the bulk of the data, but for data in the tails of the distribution non-local dependencies may be present and need to be modeled. Current successful applications of copulas have largely been limited to a few basic parametric families and low dimensional settings,

*Contributed equally. This work was supported in part by the Air Force Office of Scientific Research under award number FA9550-20-1-0397.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

preventing their widespread adoption in settings where the dependencies are complicated or the dimension is large.

Archimax copulas are a class of copulas that merges a tractable form with sufficient expressiveness. Archimax copulas effectively balance the representation of data within the bulk of the distribution while extrapolating to the tails by combining the tractability of *Archimedean* copulas with the tail properties of *extreme-value copulas*.

Notably, they remove the simplified symmetry assumption among covariates that is present in Archimedean copulas and the max-stable property of extreme-value copulas. The use of Archimax copulas has resulted in better fit to data in applications such as healthcare [71] and hydrology [3, 16].

However, existing computational methods do not allow feasible inference and sampling for Archimax copulas, preventing their widespread use. Therein lies the motivation behind this work. We construct efficient and flexible inference and sampling methods using deep learning techniques and discuss how they compare to traditional means of density estimation that use existing copula methods and deep generative models. In addition, we provide numerical studies where the proposed Archimax techniques extrapolate to the tails better than existing methods.

1.1 Related work

Modeling distributions is a major task in machine learning, with techniques such as generative adversarial networks (GANs) [19], normalizing flows (NFs) [78], and variational autoencoders (VAEs) [54] being the major developments for representing complex distributions.

However, these techniques are largely used for modeling the bulk of a distribution and may not extrapolate well to out-of-distribution samples or samples within the tails, as discussed in [60]. Some methods have been proposed to represent only the tails, for example in [1, 7]. However, they disregard the information in the bulk and only focus on the tails. Recently, Bhatia et al. [6] considered combining GANs with extreme-value theory (EVT). However, this and the above methods are used only for sampling, and do not provide a way of quantifying the dependence of the observation.

Copulas are an important technique for representing distribution functions since they allow easy separation of the marginals and the joint dependence structure of a distribution. Copulas have been applied in machine learning wherein techniques from machine learning have been used in conjunction with traditional copula theory to model more general classes of densities with examples of such work found in [63, 75, 88], please see Appendix A.6 and A.7 for more background on copulas and application of copulas in machine learning. However, these have generally focused on simplified assumptions such as a symmetric dependence between covariates, or a hierarchy of bivariate dependencies. Moreover, these do not extrapolate to tail distributions and do not readily appear to generalize to high dimensions.

With regards to Archimax copulas, several theoretical works have been proposed analyzing the distributions [11, 14, 73]. In Chatelain et al. [16], the authors proposed a method for inferring a stable tail dependence function (stdf) when given an Archimedean generator. However the method assumes knowledge of the Archimedean generator or infers a one-parameter Archimedean generator from pairwise Kendall's taus. Past applications of Archimax copulas were of low dimensions, such as dimensions 2, 3 and 3 in the studies of river flow rates [3], rainfall [16] and nutrient intake [71].

Our contributions We propose methods for filling the gaps in the existing literature by:

1. Developing methods for inferring both the Archimedean generator and stdf;
2. Developing methods for sampling from Archimax copulas;
3. Providing flexible representations for both the radial and spectral components.

Specifically, deep generative models are used to represent the distributions of the radial and spectral components. By taking an expectation, these characterize the Archimedean generator and stdf. The code for this paper is available at `https://github.com/yutingng/gen-AX`.

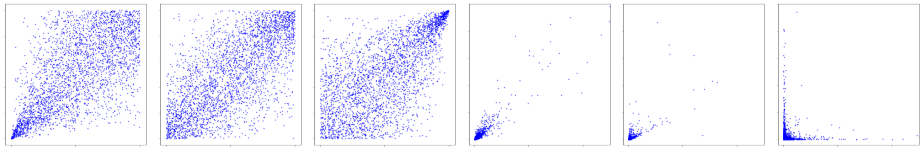

Figure 1: Examples of different radial envelopes and asymptotic dependencies.

2 Background

Copulas are given by separating the marginal distributions from the joint dependence of a random variable. Specifically, consider

$$F(\mathbf{x}) = C(F_1(x_1), \cdots, F_d(x_d)), \tag{1}$$

where F is a d-variate cumulative distribution function (CDF) of the random variable $\mathbf{X} = (X_1, \cdots, X_d) \in \mathbb{R}^d$, F_j is the jth univariate margin, and C is the copula describing the dependence between the uniform random variables $\mathbf{U} = (U_1, \cdots, U_d) = (F_1(X_1), \cdots, F_d(X_d)) \in [0,1]^d$. Moreover, if the marginals F_j are continuous, then the copula C is unique [86].[2]

As discussed in the introduction, Archimax copulas describe a generalization of Archimedean and extreme-value copulas. They are defined as:

Definition 1 (Archimax copula) *An Archimax copula is given by*

$$C(\mathbf{u}) = \varphi(\ell(\varphi^{-1}(u_1), \cdots, \varphi^{-1}(u_d))), \tag{2}$$

where $\varphi : [0,\infty) \to [0,1]$ *is an* Archimedean generator *and* $\ell : [0,\infty)^d \to [0,\infty)$ *is a* stable tail dependence function (stdf) *[10, 14, 73].*

From the definition, the Archimax copula is completely characterized by these two functions and the objective during inference lies in estimating both the *stdf* and the *Archimedean generator*. Both functions have specific properties that must be fulfilled in order to obtain a valid Archimax copula.

The *stdf* is defined as:

Definition 2 (stable tail dependence function (stdf)) *A* d*-variate stdf* $\ell : [0,\infty)^d \to [0,\infty)$ *is given by:*

$$\ell(\mathbf{x}) = d \int_{\Delta_{d-1}} \max_{j \in \{1,\cdots,d\}} \{x_j w_j\} \, \mathrm{d}F_{\mathbf{W}}(\mathbf{w}) = d\, \mathbb{E}_{\mathbf{W}} \left[\max_{j \in \{1,\cdots,d\}} \{x_j W_j\} \right], \tag{3}$$

with a spectral random variable $\mathbf{W} \in \Delta_{d-1}$ *satisfying moment constraints* $\mathbb{E}_{\mathbf{W}}[W_j] = 1/d$ *for* $j \in \{1, \cdots, d\}$ *[22, 47, 82].*

Intuitively, the *stdf* dictates the asymptotic dependence between covariates, with examples given in Figure 1. Notably, the definition completely relies on the distribution of $\mathbf{W}$, and the *stdf* is homogeneous of order one, i.e. $\ell(cx_1, \cdots, cx_d) = c\,\ell(x_1, \cdots, x_d)$ for $c > 0$.

On the other hand, the *Archimedean generator* is defined by:

Definition 3 (Archimedean generator) *An Archimedean generator* $\varphi : [0,\infty) \to [0,1]$ *is the Williamson* d*-transform of a random variable* $R > 0$*:*

$$\varphi(x) = \mathcal{W}_R(x) = \int_x^\infty \left(1 - \frac{x}{r}\right)^{d-1} dF_R(r) = \mathbb{E}_R \left[\left(1 - \frac{x}{R}\right)_+^{d-1} \right], \tag{4}$$

where $(y)_+ := \max(0, y)$ *for* $y \in \mathbb{R}$ *[72, 97].*

The Archimedean generator has a one-to-one correspondence with the distribution of R [72] and dictates the shape of the radial envelope applied across all covariates, with examples given in Figure 1.

[2] notation: random variables in uppercase, observations in lowercase, scalars are not bold, vectors are bold.

With these definitions in mind, we leverage Charpentier et al. [14, Theorem 3.3] for inference and sampling, which established that any random vector

$$\mathbf{X} \stackrel{d}{=} R \times (S_1, \cdots, S_d),\ \mathbf{X} \in [0, \infty)^d, \tag{5}$$

has dependence that follows an Archimax copula, where R and $\mathbf{S}$ are independent and known as the *radial* and *simplex* components, with supports $R > 0$, $\mathbf{S} \in [0, 1]^d$, $\ell(\mathbf{S}) \leq 1$. The marginals of $\mathbf{X}$ is given by φ such that the random vector

$$\mathbf{U} \stackrel{d}{=} (\varphi(X_1), \cdots, \varphi(X_d)),\ \mathbf{U} \in [0, 1]^d, \tag{6}$$

follows an Archimax copula. Moreover, every Archimax copula given in (2) has a decomposition given by (5) and (6). The *radial* component R is the same R in Definition 3, and the *simplex* component $\mathbf{S}$ has a one-to-one correspondence with the *spectral* component $\mathbf{W}$ from Definition 2. Therefore, representing the R and $\mathbf{W}$ in Definitions 3 and 2 respectively provides complete understanding of the distribution.

3 Method

As established in Section 2, the *stdf* and *Archimedean generator* are expectations of functions of the *spectral* component $\mathbf{W}$ and *radial* component R respectively. We first model these expectations where $\mathbf{W}$ and R are discrete random variables with finite support [31, 34]. We then let them be outputs of generative networks in the limit of infinite support. We begin by describing the inference algorithm for the stdf followed by the inference algorithm for the Archimedean generator. Through sampling the stdf we define the relationship between $\mathbf{S}$ and $\mathbf{W}$ which we make use of in the inference for the Archimedean generator. We finally show how the representations we use can be easily adapted for sampling from Archimax copulas. A flow chart describing the relationship between stdf and Archimedean generator parameter estimation is in Figure 2.

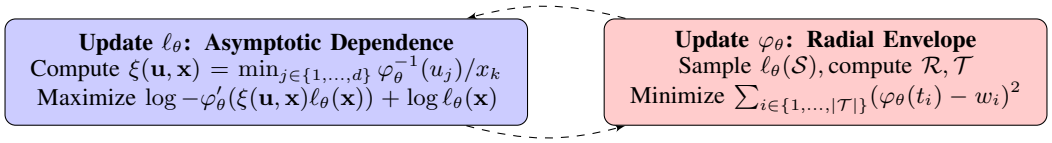

Figure 2: Flow chart describing relationship between stdf and Archimedean generator estimation.

3.1 Stable tail dependence function inference and sampling

For this section, we suppose that the Archimedean generator $\varphi(\cdot)$ is known and the goal is to infer or sample from the stdf.

3.1.1 Inference for stable tail dependence function

We consider a stdf following Definition 2, where ℓ is specified by the spectral decomposition with spectral component $\mathbf{W}$ [22, 47, 82]. To summarize the inference for the stdf: we first establish a representation of the stdf through the empirical expectation of samples $\mathbf{w}$ from a generative network $G_{\mathbf{W}}$. We then define the likelihood of transformed data observations and optimize the parameters of $G_{\mathbf{W}}$ to maximize this likelihood.

Following Chatelain et al. [16], when given the Archimedean generator φ we can define the transformation $\xi(\mathbf{u}, \mathbf{x})$ for an observation $\mathbf{u}$ and a pseudo-observation $\mathbf{x} = (x_1, \cdots, x_d) \in \Delta_{d-1}$:

$$\xi(\mathbf{u}, \mathbf{x}) = \min\{\varphi^{-1}(u_1)/x_1, \cdots, \varphi^{-1}(u_d)/x_d\}, \tag{7}$$

a transformation used for estimating extreme-value copulas [10, 79].

In the case of Archimax copulas, and by using the homogeneity property of the stdf (3), the CDF of the random variable $\xi(\mathbf{U}, \mathbf{x})$ is expressed as:

$$P(\xi(\mathbf{U}, \mathbf{x}) \leq x) = 1 - P(\xi(\mathbf{U}, \mathbf{x}) > x) = 1 - C(\varphi(x\mathbf{x})) = 1 - \varphi(x\ell(\mathbf{x})). \tag{8}$$

The full derivation is given in Appendix A.2.1.

Following Hasan et al. [38], we differentiate the CDF of $\xi(\mathbf{U}, \mathbf{x})$ with respect to x and obtain the log-likelihood of the transformed observation as:

$$\log \mathcal{L}(x; \varphi, \ell) = \log\left(-\varphi'(x\,\ell(\mathbf{x}))\right) + \log(\ell(\mathbf{x})). \tag{9}$$

Recalling Definition 2, let ℓ_θ be parameterized by a generative network $G_{\mathbf{W}}$ such that:

$$\ell_\theta(\mathbf{x}) = \frac{d}{l}\sum_{k=1}^{l}\left[\max_{j\in\{1,\cdots,d\}} x_j w_{kj}\right], \tag{10}$$

where $\mathbf{w}_k = (w_{k1}, \cdots, w_{kd})$ for $k = \{1, \cdots, l\}$ are l samples from $G_{\mathbf{W}}$ with dimension d and output activation given by the Softmax$(\cdot)$ function to respect the support $\mathbf{W} \in \Delta_{d-1}$. The moment constraints $\mathbb{E}_{\mathbf{W}}[W_j] = 1/d$ for $j = \{1, \cdots, d\}$ are scale conveniences and not necessities [31]. To approximate the moment constraints, we penalize the residual $\sum_{j=1}^{d}(\sum_{k=1}^{l} w_{kj}/l - 1/d)^2$.

We may then train the generative network $G_{\mathbf{W}}$ such that ℓ_θ approximates ℓ by minimizing the negative log-likelihood of transformed observations $\xi_(\mathbf{u}, \mathbf{x})$, with $\mathbf{x}$ uniformly sampled on the unit simplex Δ_{d-1}. The full technique is presented in Algorithm 1 in Appendix A.1.

The inverse φ^{-1} in the computation of $\xi(\mathbf{u}, \mathbf{x})$ in (7) can be computed numerically by Newton-Raphson. The derivative φ' in the computation of $\log \mathcal{L}(x; \varphi, \ell)$ in (9) can be calculated explicitly from Definition 3 as:

$$\varphi'(x) = \mathbb{E}_R\left[(d-1)\left(-\frac{1}{R}\right)\left(1 - \frac{x}{R}\right)_+^{d-2}\right]. \tag{11}$$

Putting these computations together results in the likelihood of the stdf given the generator.

3.1.2 Sampling the simplex component

We now consider sampling the simplex component $\mathbf{S}$. We noted in the introduction that there is a one-to-one relationship between $\mathbf{S}$ and $\mathbf{W}$. Using this relationship and the result of Charpentier et al. [14], the stdf may also be written in terms of the simplex component through the following definition:

Definition 4 (survival distribution of simplex component) *The survival distribution function (SDF) of the simplex component* $\mathbf{S}$ *is:*

$$P(\mathbf{S} > \mathbf{s}) = \max(0, 1 - \ell(s_1, \cdots, s_d))^{d-1}, \tag{12}$$

which relates to the so-called generalized Pareto copulas defined as:

Definition 5 (generalized Pareto copula) *A generalized Pareto copula is defined as the copula of a generalized Pareto distribution with support on* $(-\infty, 0]^d$ *and CDF specified by* ℓ *as:*

$$P(X_1 < x_1, \cdots, X_d < x_d) = \max(0, 1 - \ell(-x_1, \cdots, -x_d)). \tag{13}$$

Moreover, it has stochastic representation as:

$$\mathbf{X} \overset{d}{=} -\left(\frac{U}{W_1}, \cdots, \frac{U}{W_d}\right), \tag{14}$$

where U *is uniformly distributed on* $[0, 1]$*, independent of spectral component* $\mathbf{W} = (W_1, \cdots, W_d)$ *from the spectral decomposition of* ℓ *in Definition 2 [9, 29].*

Therefore, we can frame the sampling of the simplex component through the sampling from a generalized Pareto copula, such that, given samples $(x_{11}, \cdots, x_{1d}), \cdots, (x_{(d-1)1}, \cdots, x_{(d-1)d})$ of size $d-1$ from a generalized Pareto distribution with CDF in (13), we may obtain a sample of $\mathbf{s}$, where coordinates s_j for $j = \{1, \cdots, d\}$ are computed as [14]:

$$s_j = -\max(x_{1j}, \cdots, x_{(d-1)j}). \tag{15}$$

The coordinate-wise maxima is taken over $d-1$ samples to get the $d-1$ exponent in the SDF of $\mathbf{S}$, corresponding to the $d-1$ exponent in the expression of the Archimedean generator φ.

Due to our convenient representation of ℓ as an expectation of a function of $\mathbf{W}$ from generative network $G_{\mathbf{W}}$, we are able to sample a generalized Pareto distribution from its stochastic representation. The full technique is presented in Algorithm 2 in Appendix A.1.

3.2 Archimedean generator inference and sampling

We now assume that the stdf is known and the goal is to infer the Archimedean generator. We consider a general representation of the Archimedean generator following Definition 3, where φ is a $d-$monotone function specified by the Williamson $d-$transform of the radial component R [14, 72].

To provide an outline for the overall approach: we first consider the so-called Kendall distribution, which describes an integral transform of the copula. We then use the fact that the empirical Kendall distribution converges to the true Kendall distribution as the empirical copula converges to the true copula, providing a means of estimation. The Kendall distribution is formally defined as:

Definition 6 (Kendall distribution) *Let C be a copula, and let the CDF of the random variable $\mathbf{U} \in [0,1]^d$ be the copula C. Define the random variable $W := C(\mathbf{U})$. The Kendall distribution of C is the multivariate probability integral transform given by the CDF of W:*

$$K(w) = P(C(\mathbf{U}) \leq w), \quad w \in [0,1]. \tag{16}$$

In the case of Archimax copulas, using the stochastic representation (5) and the homogeneity property of the stdf (3), the Kendall distribution is expressed as:

$$K(w) = P(\varphi(R\,\ell(S_1, \cdots, S_d)) \leq w). \tag{17}$$

The full derivation is provided in Appendix A.3. The empirical Kendall distribution K_n for n given observations $(u_{11}, \cdots, u_{1d}), \cdots, (u_{n1}, \cdots, u_{nd})$ is defined as:

$$K_n(w) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\{w_i \leq w\}, \tag{18}$$

where

$$w_i = \frac{1}{n+1} \sum_{k=1}^{n} \mathbb{1}\{u_{k1} < u_{i1}, \cdots, u_{kd} < u_{id}\}. \tag{19}$$

In the case of symmetric dependence in the non-parametric inference of Archimedean copulas [34], $\mathbf{S}$ is uniformly distributed on the simplex Δ_{d-1} and $\ell(S_1, \cdots, S_d) = S_1 + \cdots + S_d \equiv 1$. Then, R is discrete with the cardinality of the support the same as W and K_n is the Kendall distribution of a unique Archimedean copula [34].

In the non-parametric inference of Archimax copulas, we define the random variables:

$$Z := \ell(S_1, \cdots, S_d) \quad \text{and} \quad T := RZ \tag{20}$$

where T is a discrete random variable with support the same size as W and K_n is the Kendall distribution of an Archimax copula. Note that R and Z are independent as R and $\mathbf{S}$ are independent.

Using K_n, we reconstruct the support of R, given ℓ and the support of $\mathbf{S}$, which in turn provides the support of Z. With the final objective of letting R and $\mathbf{S}$ be independent and identically distributed (iid) outputs of generative networks, we first (linearly) interpolate K_n to be equispaced, such that each w_i has the same probability $k_i = 1/(n_r n_z)$, where n_r and n_z are the chosen sizes of supports for R and Z. We describe the reconstruction procedure for the general case with non-iid random variables including cases where $r_j z_l = r_{j'} z_{l'}$ in Appendix A.3. The sizes of supports n_r and n_z are chosen empirically with examples given in Appendix B.1.

Suppose the supports of the distributions of R, Z, W, T are finite and respectively denoted by: $\mathcal{W} = \{w_1, w_2, \cdots, w_{n_r n_z}\}$, $\mathcal{R} = \{r_1, r_2, \cdots, r_{n_r}\}$, $\mathcal{Z} = \{z_1, z_2, \cdots, z_{n_z}\}$, $\mathcal{T} = \{r_j z_l : r_j \in \mathcal{R}, z_l \in \mathcal{Z}\} = \{t_1, t_2, \cdots, t_{n_r n_z}\}$, where the $n_r n_z$ elements of $\mathcal{W}$ are sorted in decreasing (non-increasing) order, and the $n_r n_z$ elements of $\mathcal{T}$ are sorted in increasing (non-decreasing) order. This reverse ordering is due to φ being a decreasing function.

We minimize the mean sum of square residuals motivated by the uniform convergence of the empirical process $\sqrt{n}(K_n - K)$ as $n \to \infty$ established by Barbe et al. [4]:

$$\frac{1}{n_r n_z} \sum_{i=1}^{n_r n_z} (w_i - \varphi_\theta(t_i))^2 \tag{21}$$

where, following Definition 3,

$$\varphi_\theta(t_i) = \mathcal{W}_{\mathcal{R}}(t_i) = \frac{1}{n_r}\sum_{j=1}^{n_r}\left(1 - \frac{t_i}{r_j}\right)_+^{d-1}. \tag{22}$$

The finite support assumption of R and Z is not necessary, and we consider a modification where the supports $\mathcal{R}, \mathcal{Z}$ are specified by samples from generative networks. The main objective is to learn the parameters of generative network G_R given the empirical Kendall distribution K_n and samples of Z. Since scaling the support $\mathcal{R}$ by a constant $c > 0$ does not change the copula, we add a regularization term for $\mathbb{E}_R[R] = 1$. The algorithm can be understood as an alternating minimization algorithm, where the map between $\mathcal{R}$ and $\mathcal{Z}$ to $\mathcal{W}$ via $\mathcal{T}$, and the support $\mathcal{R}$ are updated in an alternating fashion. The full technique is presented in Algorithm 3 in Appendix A.1.

The learned G_R also provides a source of samples for the radial component R, which we use to generate full samples from the Archimax copula.

3.3 Inference and sampling for Archimax copulas

We finally summarize the combination of inference and sampling for both the Archimedean and simplex component to obtain the full algorithm for inference and sampling for Archimax copulas. This culminates into an iterative technique that successively updates each component.

3.3.1 Inference for Archimax copulas

We initialize with $\varphi(x) = \exp(-x)$ and infer ℓ to learn $G_{\mathbf{W}}$, following Section 3.1. This special combination of φ and ℓ corresponds to extreme-value dependence with the max-stable property. To aid inference for this initialization step, we pre-process our data to have extreme-value dependence. We do so by computing the *block maxima*, a technique from extreme-value copulas where we group observations and take the coordinate-wise maximas within each group. Specifically, given observations $(x_{11}, \cdots, x_{1d}), \cdots, (x_{n1}, \cdots, x_{nd})$, the block-maximas $(m_{11}, \cdots, m_{1d}), \cdots, (m_{k1}, \cdots, m_{kd})$ for k blocks of size n/k are computed as:

$$m_{ij} = \max(x_{(n/k)(i-1)+1\,j}, \cdots, x_{(n/k)(i)\,j}), \tag{23}$$

for $i = \{1, \cdots, k\}$ and $j = \{1, \cdots, d\}$. To determine the block size, with larger block sizes n/k better approximating extreme-value dependence and more blocks k for more observations, we test the block maximas for extreme-value dependence via the max-stable property using the test by Kojadinovic et al. [55], and select the first k where the null hypothesis of extreme-value dependence is not rejected, starting from $k = n$, with details given in Appendix A.4.1.[3]

Given an estimate of ℓ and a learned $G_{\mathbf{W}}$, we are able to generate many samples of $\mathbf{S}$, thereby providing a way to compute $Z = \ell(\mathbf{S})$. We then infer φ, learning G_R, following Section 3.2. At this point, we repeat the estimation for $G_{\mathbf{W}}$ with the updated φ to improve our estimate for ℓ. The algorithms may be iterated as needed with suggested convergence criteria such as Cramér–von Mises (CvM) distance between successively estimated copulas.

3.3.2 Sampling for Archimax copulas

Given samples of the radial and simplex components, the stochastic representation of Archimax copulas (5) gives a straightforward method for sampling Archimax copulas [14]. Specifically, we sample $\mathbf{S}$ and R, then multiply and normalize by φ; see Algorithm 4 in Appendix A.1 for details.

4 Experiments

To understand the modeling and sampling capabilities of the proposed algorithms, we conduct a number of empirical studies. We compare the proposed method to a number of existing copula based and deep learning based methods for density estimation. Our experiments relate to the main focus

[3] An alternative initialization scheme based on testing different one-parameter families of Archimedean generators with the log-likelihood of transformed observations ξ is also given in Appendix A.4.1.

of the proposed method where we are interested in understanding the dependencies between the variables in both the bulk and the tail. In that sense, we conduct a number of experiments where we wish to extrapolate to the tail. Further experimental details may be found in Appendix B.

The metric we use to compare the methods is based on the Cramér-von Mises (CvM) statistic, which computes the L^2 distance between the empirical copula and the estimated copula. This statistic is commonly used to determine differences between distributions in goodness-of-fit tests [81]. We use a version where we compare the empirical copula of true samples to the empirical copula of generated samples. Specifically, it can be written as: $CvM = \int (C_{*,n}(\mathbf{u}) - C_{\theta,n}(\mathbf{u}))^2 \mathrm{d}\mathbf{u}$, where $C_{*,n}$ is the empirical copula of true samples and $C_{\theta,n}$ is the empirical copula of generated samples [81].

Inference for Archimedean generator Our first set of experiments involve inference for the Archimedean generator following the proposed method in Section 3.2. We consider data with dimension $d = 10$ and sample size of $n = 1000$ given the true stdf ℓ. To the best of our knowledge, our proposed method is the first method for non-parametric inference of flexible Archimedean generators in Archimax copulas. As such, there are no baselines for comparison. Instead, we compute the results in terms of the map $\lambda(w) = \varphi^{-1}(w)/(\varphi^{-1}(w))', w \in (0, 1)$ due to its scale invariant property and known asymptotic variance, which was described as a useful metric for how well φ is fit in [33, 34]. We evaluated our proposed method on the Clayton (C), Frank (F), Joe (J) and Gumbel (G) generators, representing different radial envelopes, for Kendall's tau of $\tau = \{0.2, 0.5\}$, representing different associations. The stdf comes from the family of negative scaled extremal Dirichlet (NSD), a flexible class of stdf [5]. All estimates were within the asymptotic variance of $\lambda(w), w \in (0, 1)$. The next best method is comparing to a Clayton generator estimated by pairwise Kendall's tau [16]. We give the results in terms of MSE to λ in Table 1, and plot estimates of $\lambda(w)$ in Figure 5 in Appendix B.1. Additional experiment results, including small sample performance $n = 200$, and choices of support sizes n_r and n_z, are in Appendix B.1.

Inference for stable tail dependence function and sampling for simplex component We now consider the reverse scenario where we wish to estimate the stdf ℓ given the true Archimedean generator φ. We compute the integrated relative absolute error (IRAE) between the estimated ℓ_θ and the true ℓ. The IRAE is given by $IRAE(\ell, \ell_\theta) = \frac{1}{|\Delta_{d-1}|} \int_{\Delta_{d-1}} |\ell(\mathbf{x}) - \ell_\theta(\mathbf{x})|/\ell(\mathbf{x}) \, \mathrm{d}\mathbf{x}$ [16]. For the same experiment settings as above, results are in Table 1. Additional experiment results, including time taken, are in Appendix B.2.

Table 1: Inference of φ given true ℓ and inference of ℓ given true φ

	C 0.2	C 0.5	F 0.2	F 0.5	J 0.2	J 0.5	G 0.2	G 0.5
MSE $\times 10^{-3}$ [16]	**0.01**	**0.04**	0.9	9	1	9	0.7	8
MSE $\times 10^{-3}$ (OURS)	0.2	0.2	**0.1**	**0.1**	**0.3**	**0.1**	**0.2**	**0.1**
IRAE ± 0.01 [16]	0.05	0.11	0.04	0.04	0.05	1.00	0.06	0.15
IRAE ± 0.01 (OURS)	0.06	0.12	0.04	0.05	0.06	**0.07**	0.08	0.15

Modeling nutrient intake The USDA studied the nutrient intake of women [92]. One particular task is understanding the dependencies between the intake of different nutrients. We can model this using an Archimax copula to understand the dependencies between the nutrients in the bulk and the tail. To assess this, we fit and compare models from the literature on representing distributions and compute the CvM goodness-of-fit statistic for each model. Specifically, we break our comparison into two different types of models: copula based models and deep network based models. For the copula models, for methods marked with *, we use the φ described in Section 3.2 and for methods marked with † we use the ℓ described in Section 3.1.2. For the deep generative models, we use standard methods based on the Wasserstein GAN [2], masked autoregressive flow (MAF) [77], and variational autoencoders (VAE) [54]. The results are presented in Table 2. The proposed method (Gen-AX) with the Archimax has the lowest CvM distance among all the competing methods, suggesting that the proposed method is recovering the true dependency structure. The state of the art Clayton Archimax (C-AX) did not perform well possibly due to the difficulty in scaling the single parameter Clayton generator to higher dimension. The Archimedean copula (AC *) possibly benefited from the use of our proposed φ. We additionally provide examples of samples versus the ground truth in Appendix B.3 for the different methods as well as an explanation of the different abbreviations.

Table 2: Goodness-of-fit to nutrient intake data and 100d NSD copula

	COPULAS								DEEP NETS			OURS
	GC	RV	CV	DV	AC *	HAC	EV †	C-AX †	GAN	MAF	VAE	Gen-AX *†
Nutrient CvM $\times 10^{-3}$	0.081	0.3	0.2	0.6	0.030	0.2	0.4	0.6	0.033	0.036	0.053	**0.026**
C-NSD CvM $\times 10^{-5}$	-	-	-	-	-	-	-	-	16	**3**	5	**3**

Extrapolating to extreme rainfall Archimax copulas were initially developed as a tool to study the behaviour of methods that estimate the joint distribution of extreme events [11]. The extreme-value copula that arises in the limit can be understood from the stdf ℓ and the index of regular variation of the Archimedean generator φ. Unlike extreme-value copulas which emerge from the limiting distribution of extreme events, the motivation for the use of Archimax copulas is to model extreme data, where observations are rare, from a mix of moderately less extreme data, where observations are relatively more abundant.

In this experiment, we consider another realistic dataset which models the monthly rainfall in French Britanny as studied by Chatelain et al. [16]. We are interested in testing how well the proposed method can extrapolate to the extremes from non-extreme data. Specifically, we analyze the monthly rainfall data, which did not pass the test of extreme-value dependence [55, 16].

We first train the models on the full dataset. We then generate many samples from the trained model, compute the block maxima, then estimate the extremal dependence from the block maxima using the CFG estimator [10]. The results are presented in Table 3 where we compare the proposed method to deep generative models. Additional experiment details and results, including plots of samples from the bulk and the extremes, are in Appendix B.4. We did not compare to copula based models since classical copulas are generally Gumbel or independent in the extremes and thus not suitable for this application. However, for the purpose of modeling monthly rainfall without extrapolating to extremes, we compared to a variety of copula based models, including skew-t copulas which is a class of flexible asymmetric copulas [23, 56, 99, 87]. The full details and results are given in Appendix B.4.1.

Out-of-distribution detection Using the same realistic dataset as above, we added outliers generated uniformly at random on the unit cube. The AUC and F1 scores for outlier detection based on likelihoods are given in Table 4. Additional experiment details and figures are in Appendix B.5.

Table 3: Goodness-of-fit to dependence in the extremes

IRAE $\pm$ 0.01	GAN	MAF	VAE	C-AX	OURS
C-NSD	0.52	0.42	0.16	0.12	**0.03**
F-NSD	0.10	0.15	**0.04**	**0.04**	**0.04**
J-NSD	**0.03**	0.48	**0.03**	0.08	**0.03**
G-NSD	0.07	0.38	0.08	0.16	**0.04**

Table 4: Out-of-distribution detection

	MAF	VAE	OURS
AUC	0.82	0.37	**0.92**
F1	0.48	0.04	**0.72**

High dimensional modeling We finally consider an experiment where we infer and sample data from a 100-dimensional Clayton-NSD Archimax copula. Scaling to high dimensions is an important property of the proposed method since many existing copula methods fail to scale beyond lower dimensions. As such, we only compare with deep generative models since the existing copula models resulted in numerical errors during optimization. We report the CvM statistic in the second line of Table 2. We additionally provide examples of the samples in Appendix B.6.

5 Conclusion

We developed highly flexible and scalable inference and sampling algorithms, facilitating the use of Archimax copulas in practical settings. We experimentally compare to state-of-the-art density modeling techniques, and the results suggest that the proposed method effectively extrapolates to tails while scaling to higher dimensional data. The methods are especially useful in scenarios requiring extrapolations to the tails while also incorporating data from the bulk.

Limitations and future work A single Archimedean generator φ to describe the radial envelope across all coordinates may not be sufficiently expressive for certain datasets. For these cases, hierarchical Archimax copulas may be more appropriate and a direction of future work [42]. Other directions for future work include modifying the generator architectures to allow modeling of temporal dependence and also application of Archimax copulas to describe dependencies of non-tabular data, such as via graph neural networks [68].

Potential negative societal impacts Model misspecification may lead to misspecification of risks, leading to potentially catastrophic outcomes in areas such as healthcare, safety and finance. Risks may be mitigated by confirming a reasonable fit between the observations generated by the model and the data.

References

[1] Michaël Allouche, Stéphane Girard, and Emmanuel Gobet. EV-GAN: Simulation of extreme events with relu neural networks. *Journal of Machine Learning Research*, 23(150):1–39, 2022. URL `http://jmlr.org/papers/v23/21-0663.html`.

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

[3] Tomáš Bacigál, Vladimír Jágr, and Radko Mesiar. Non-exchangeable random variables, Archimax copulas and their fitting to real data. *Kybernetika*, 47(4):519–531, 2011.

[4] Philippe Barbe, Christian Genest, Kilani Ghoudi, and Bruno Rémillard. On Kendall's process. *Journal of Multivariate Analysis*, 58(2):197–229, 1996.

[5] Léo R. Belzile and Johanna G. Nešlehová. Extremal attractors of Liouville copulas. *Journal of Multivariate Analysis*, 160:68–92, 2017. ISSN 0047-259X. doi: https://doi.org/10.1016/j.jmva.2017.05.008. URL `https://www.sciencedirect.com/science/article/pii/S0047259X17300453`.

[6] Siddharth Bhatia, Arjit Jain, and Bryan Hooi. Exgan: Adversarial generation of extreme samples. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6750–6758, 2021.

[7] Younes Boulaguiem, Jakob Zscheischler, Edoardo Vignotto, Karin van der Wiel, and Sebastian Engelke. Modeling and simulating spatial extremes by combining extreme value theory with generative adversarial networks. *Environmental Data Science*, 1:e5, 2022. doi: 10.1017/eds.2022.4.

[8] Mary Ann Branch, Thomas F. Coleman, and Yuying Li. A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems. *SIAM Journal on Scientific Computing*, 21(1):1–23, 1999. URL `https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.least_squares.html`.

[9] T. A. Buishand, L. de Haan, and C. Zhou. On spatial extremes: With application to a rainfall problem. *The Annals of Applied Statistics*, 2(2):624–642, 2008. ISSN 19326157. URL `http://www.jstor.org/stable/30244220`.

[10] P. Capéraà, A.-L. Fougères, and C. Genest. A nonparametric estimation procedure for bivariate extreme value copulas. *Biometrika*, 84(3):567–577, 09 1997.

[11] Philippe Capéraà, Anne-Laure Fougères, and Christian Genest. Bivariate distributions with given extreme value attractor. *Journal of Multivariate Analysis*, 72(1):30–49, 2000.

[12] Bo Chang, Shenyi Pan, and Harry Joe. Vine copula structure learning via monte carlo tree search. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 353–361. PMLR, 16–18 Apr 2019. URL `https://proceedings.mlr.press/v89/chang19a.html`.

[13] Yale Chang, Yi Li, Adam Ding, and Jennifer Dy. A robust-equitable copula dependence measure for feature selection. In Arthur Gretton and Christian C. Robert, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 84–92, Cadiz, Spain, 09–11 May 2016. PMLR. URL `https://proceedings.mlr.press/v51/chang16.html`.

[14] A. Charpentier, A.-L. Fougères, C. Genest, and J.G. Nešlehová. Multivariate Archimax copulas. *Journal of Multivariate Analysis*, 126:118–136, 2014.

[15] Arthur Charpentier, Jean-David Fermanian, and Olivier Scaillet. The estimation of copulas: Theory and practice. *Copulas: From theory to application in finance*, pages 35–64, 2007.

[16] Simon Chatelain, Anne-Laure Fougères, and Johanna G. Nešlehová. Inference for Archimax copulas. *The Annals of Statistics*, 48(2):1025 – 1051, 2020.

[17] Pawel Chilinski and Ricardo Silva. Neural likelihoods via cumulative distribution functions. In Jonas Peters and David Sontag, editors, *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 124 of *Proceedings of Machine Learning Research*, pages 420–429. PMLR, 03–06 Aug 2020. URL `http://proceedings.mlr.press/v124/chilinski20a.html`.

[18] Stuart G. Coles and Jonathan A. Tawn. Modelling extreme multivariate events. *Journal of the Royal Statistical Society. Series B (Methodological)*, 53(2):377–392, 1991. ISSN 00359246. URL `http://www.jstor.org/stable/2345748`.

[19] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A Bharath. Generative adversarial networks: An overview. *IEEE Signal Processing Magazine*, 35(1):53–65, 2018.

[20] Ruifei Cui, Perry Groot, Moritz Schauer, and Tom Heskes. Learning the causal structure of copula models with latent variables. In *UAI*. Corvallis: AUAI Press, 2018.

[21] Data to AI Lab at MIT. Copulas. URL `https://github.com/sdv-dev/Copulas`.

[22] L. De Haan. A Spectral Representation for Max-stable Processes. *The Annals of Probability*, 12(4):1194 – 1204, 1984.

[23] Stefano Demarta and Alexander J. McNeil. The t copula and related copulas. *International Statistical Review / Revue Internationale de Statistique*, 73(1):111–129, 2005. ISSN 03067734, 17515823. URL `http://www.jstor.org/stable/25472643`.

[24] Alexandre Drouin, Étienne Marcotte, and Nicolas Chapados. TACTiS: Transformer-attentional copulas for time series. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 5447–5493. PMLR, 17–23 Jul 2022. URL `https://proceedings.mlr.press/v162/drouin22a.html`.

[25] Elad Eban, Gideon Rothschild, Adi Mizrahi, Israel Nelken, and Gal Elidan. Dynamic copula networks for modeling real-valued time series. In *Artificial Intelligence and Statistics*, pages 247–255. PMLR, 2013.

[26] Gal Elidan. Copula bayesian networks. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 23, pages 559–567. Curran Associates, Inc., 2010. URL `https://proceedings.neurips.cc/paper/2010/file/2a79ea27c279e471f4d180b08d62b00a-Paper.pdf`.

[27] Gal Elidan. Inference-less density estimation using copula bayesian networks. In *UAI*, 2010.

[28] Gal Elidan. Copula network classifiers (cncs). In *Artificial intelligence and statistics*, pages 346–354. PMLR, 2012.

[29] Michael Falk, Simone A. Padoan, and Florian Wisheckel. Generalized pareto copulas: A key to multivariate extremes. *Journal of Multivariate Analysis*, 174:104538, 2019. ISSN 0047-259X. doi: https://doi.org/10.1016/j.jmva.2019.104538. URL `https://www.sciencedirect.com/science/article/pii/S0047259X19300296`.

[30] Amélie Fils-Villetard, Armelle Guillou, and Johan Segers. Projection estimators of pickands dependence functions. *The Canadian Journal of Statistics / La Revue Canadienne de Statistique*, 36(3):369–382, 2008. ISSN 03195724. URL `http://www.jstor.org/stable/41219865`.

[31] Anne-Laure Fougères, Cécile Mercadier, and John Nolan. Dense classes of multivariate extreme value distributions. *Journal of Multivariate Analysis*, 116:109–129, 2013. ISSN 0047-259X. doi: https://doi.org/10.1016/j.jmva.2012.11.015. URL `https://www.sciencedirect.com/science/article/pii/S0047259X12002746`.

[32] Christian Genest and Anne-Catherine Favre. Everything you always wanted to know about copula modeling but were afraid to ask. *Journal of Hydrologic Engineering*, 12(4):347–368, 2007. doi: 10.1061/(ASCE)1084-0699(2007)12:4(347). URL `https://ascelibrary.org/doi/abs/10.1061/%28ASCE%291084-0699%282007%2912%3A4%28347%29`.

[33] Christian Genest and Louis-Paul Rivest. Statistical inference procedures for bivariate archimedean copulas. *Journal of the American Statistical Association*, 88(423):1034–1043, 1993.

[34] Christian Genest, Johanna Nešlehová, and Johanna Ziegel. Inference in multivariate archimedean copula models. *TEST: An Official Journal of the Spanish Society of Statistics and Operations Research*, 20(2):223, 2011.

[35] Jan Górecki and Malte S. Kurz. The HACopula toolbox. URL `https://github.com/gorecki/HACopula`.

[36] Jan Górecki, Marius Hofert, and Martin Holeňa. On structure, family and parameter estimation of hierarchical archimedean copulas. *Journal of Statistical Computation and Simulation*, 87 (17):3261–3324, 2017. doi: 10.1080/00949655.2017.1365148. URL `https://doi.org/10.1080/00949655.2017.1365148`.

[37] Shaobo Han, Xuejun Liao, David Dunson, and Lawrence Carin. Variational Gaussian copula inference. In Arthur Gretton and Christian C. Robert, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 829–838, Cadiz, Spain, 09–11 May 2016. PMLR. URL `https://proceedings.mlr.press/v51/han16.html`.

[38] Ali Hasan, Khalil Elkhalil, Yuting Ng, João M Pereira, Sina Farsiu, Jose Blanchet, and Vahid Tarokh. Modeling extremes with d-max-decreasing neural networks. In *The 38th Conference on Uncertainty in Artificial Intelligence*, 2022.

[39] José Miguel Hernández-Lobato, James R Lloyd, and Daniel Hernández-Lobato. Gaussian process conditional copulas with applications to financial time series. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL `https://proceedings.neurips.cc/paper/2013/file/67d16d00201083a2b118dd5128dd6f59-Paper.pdf`.

[40] Marcel Hirt, Petros Dellaportas, and Alain Durmus. Copula-like variational inference. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper/2019/file/e721a54a8cf18c8543d44782d9ef681f-Paper.pdf`.

[41] Marius Hofert, Martin Mächler, and Alexander J. McNeil. Archimedean Copulas in High Dimensions: Estimators and Numerical Challenges Motivated by Financial Applications. *Journal de la société française de statistique*, 154(1):25–63, 2013. URL `http://www.numdam.org/item/JSFS_2013__154_1_25_0/`.

[42] Marius Hofert, Raphaël Huser, and Avinash Prasad. Hierarchical Archimax copulas. *Journal of Multivariate Analysis*, 167:195–211, 2018.

[43] Jim Huang and Brendan J Frey. Structured ranking learning using cumulative distribution networks. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc., 2008. URL `https://proceedings.neurips.cc/paper/2008/file/03c6b06952c750899bb03d998e631860-Paper.pdf`.

[44] Jim Huang and Nebojsa Jojic. Maximum-likelihood learning of cumulative distribution functions on graphs. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 342–349, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL `https://proceedings.mlr.press/v9/huang10b.html`.

[45] Jim C. Huang and Brendan J. Frey. Cumulative distribution networks and the derivative-sum-product algorithm. In *Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence*, UAI'08, page 290–297, Arlington, Virginia, USA, 2008. AUAI Press. ISBN 0974903949.

[46] Jim C. Huang and Brendan J. Frey. Cumulative distribution networks and the derivative-sum-product algorithm: Models and inference for cumulative distribution functions on graphs. *Journal of Machine Learning Research*, 12(10):301–348, 2011.

[47] Xu Huang. *Statistics of bivariate extreme values*. PhD, Tinbergen Institute, 1992.

[48] Tim Janke, Mohamed Ghanmi, and Florian Steinke. Implicit generative copulas. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 26028–26039. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/dac4a67bdc4a800113b0f1ad67ed696f-Paper.pdf.

[49] Piotr Jaworski, Fabrizio Durante, Wolfgang Karl Hardle, and Tomasz Rychlik, editors. *Copula theory and its applications*. Lecture notes in statistics. Springer, Berlin, Germany, 2010 edition, July 2010.

[50] Piotr Jaworski, Fabrizio Durante, and Wolfgang Karl Hardle, editors. *Copulae in mathematical and quantitative finance*. Lecture notes in statistics. Springer, Berlin, Germany, 2013 edition, June 2013.

[51] Harry Joe. *Dependence modeling with copulas*. Chapman and Hall/CRC, June 2014. ISBN 9781466583238. doi: 10.1201/b17116. URL https://www.taylorfrancis.com/books/9781466583238.

[52] Nebojsa Jojic, Chris Meek, and Jim Huang. Exact inference and learning for cumulative distribution functions on loopy graphs. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 23, pages 874–882. Curran Associates, Inc., 2010. URL https://proceedings.neurips.cc/paper/2010/file/705f2172834666788607efbfca35afb3-Paper.pdf.

[53] Sanket Kamthe, Samuel Assefa, and Marc Deisenroth. Copula flows for synthetic data generation, 2021. URL https://arxiv.org/abs/2101.00598.

[54] Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

[55] Ivan Kojadinovic, Johan Segers, and Jun Yan. Large-sample tests of extreme-value dependence for multivariate copulas. LIDAM Reprints ISBA 2011025, Université catholique de Louvain, Institute of Statistics, Biostatistics and Actuarial Sciences (ISBA), 2011. URL https://EconPapers.repec.org/RePEc:aiz:louvar:2011025.

[56] Tõnu Kollo and Gaida Pettere. Parameter estimation and application of the multivariate skew t-copula. In Piotr Jaworski, Fabrizio Durante, Wolfgang Karl Härdle, and Tomasz Rychlik, editors, *Copula Theory and Its Applications*, pages 289–298, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[57] Yves-Laurent Kom Samo. Inductive mutual information estimation: A convex maximum-entropy copula approach. In Arindam Banerjee and Kenji Fukumizu, editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 2242–2250. PMLR, 13–15 Apr 2021. URL https://proceedings.mlr.press/v130/kom-samo21a.html.

[58] Eric Landgrebe, Madeleine Udell, et al. Online mixed missing value imputation using Gaussian copula. In *Workshop on the Art of Learning with Missing Values (Artemiss) hosted by the 37th International Conference on Machine Learning (ICML)*, 2020.

[59] Mike Laszkiewicz, Johannes Lederer, and Asja Fischer. Copula-based normalizing flows. In *ICML Workshop on Invertible Neural Networks, Normalizing Flows, and Explicit Likelihood Models*, 2021. URL https://openreview.net/forum?id=T4Wf0w2jcz.

[60] Charline Le Lan and Laurent Dinh. Perfect density models cannot guarantee anomaly detection. *Entropy*, 23(12):1690, 2021.

[61] Benjamin Letham, Wei Sun, and Anshul Sheopuri. Latent variable copula inference for bundle pricing from retail transaction data. In *ICML*, pages 217–225, 2014. URL http://proceedings.mlr.press/v32/letham14.html.

[62] Zheng Li, Yue Zhao, Nicola Botta, Cezar Ionescu, and Xiyang Hu. Copod: copula-based outlier detection. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 1118–1123. IEEE, 2020.

[63] Chun Kai Ling, Fei Fang, and J. Zico Kolter. Deep archimedean copulas. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1535–1545. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/10eb6500bd1e4a3704818012a1593cc3-Paper.pdf.

[64] Weiwei Liu. Copula multi-label learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/5d2c2cee8ab0b9a36bd1ed7196bd6c4a-Paper.pdf.

[65] David Lopez-paz, Jose Hernández-lobato, and Bernhard Schölkopf. Semi-supervised domain adaptation with non-parametric copulas. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper/2012/file/8e98d81f8217304975ccb23337bb5761-Paper.pdf.

[66] David Lopez-Paz, Jose Miguel Hernández-Lobato, and Ghahramani Zoubin. Gaussian process vine copulas for multivariate dependence. In *International Conference on Machine Learning*, pages 10–18. PMLR, 2013.

[67] Jian Ma and Zengqi Sun. Mutual information is copula entropy. *Tsinghua Science and Technology*, 16(1):51–54, 2011. doi: 10.1016/S1007-0214(11)70008-6.

[68] Jiaqi Ma, Bo Chang, Xuefei Zhang, and Qiaozhu Mei. Copulagnn: Towards integrating representational and correlational roles of graphs in graph neural networks. In *International Conference on Learning Representations*, 2021.

[69] Jan-Frederik Mai and Matthias Scherer. *Simulating Copulas*. WORLD SCIENTIFIC, 2nd edition, 2017. doi: 10.1142/10265. URL https://www.worldscientific.com/doi/abs/10.1142/10265.

[70] Bijan Mazaheri, Siddharth Jain, and Jehoshua Bruck. Robust correction of sampling bias using cumulative distribution functions. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 3546–3556. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/24368c745de15b3d2d6279667debcba3-Paper.pdf.

[71] Alexander J McNeil and Johanna Nešlehová. From Archimedean to Liouville copulas. *Journal of Multivariate Analysis*, 101(8):1772–1790, 2010.

[72] Alexander J. McNeil and Johanna Nešlehová. Multivariate Archimedean copulas, d-monotone functions and $l1$-norm symmetric distributions. *The Annals of Statistics*, 37(5B):3059 – 3097, 2009. doi: 10.1214/07-AOS556. URL https://doi.org/10.1214/07-AOS556.

[73] Radko Mesiar and Vladimír Jágr. d-dimensional dependence functions and Archimax copulas. *Fuzzy Sets and Systems*, 228:78–87, 2013.

[74] Roger B. Nelsen. *An introduction to Copulas*. Springer series in statistics. Springer New York, New York, NY, 2. ed. 2006. corr. 2. pr. softcover version of original hardcover edition 2006 edition, 2010. ISBN 9780387286785 9781441921093. OCLC: 700190717.

[75] Yuting Ng, Ali Hasan, Khalil Elkhalil, and Vahid Tarokh. Generative Archimedean copulas. In *37th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2021.

[76] Georg Ostrovski, Will Dabney, and Rémi Munos. Autoregressive quantile networks for generative modeling. In *ICML*, pages 3933–3942, 2018. URL http://proceedings.mlr.press/v80/ostrovski18a.html.

[77] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. *Advances in neural information processing systems*, 30, 2017.

[78] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.

[79] James Pickands. Multivariate extreme value distribution. In *43rd Session of the International Statistical Institution*, volume 2, pages 859–878,894–902, Buenos Aires, 1981.

[80] Barnabás Póczos, Zoubin Ghahramani, and Jeff Schneider. Copula-based kernel dependency measures. In *Proceedings of the 29th International Conference on International Conference on Machine Learning*, ICML'12, page 1635–1642, Madison, WI, USA, 2012. Omnipress. ISBN 9781450312851.

[81] Bruno Remillard and Olivier Scaillet. Testing for equality between two copulas. 8:229–231, june 2007. doi: http://dx.doi.org/10.2139/ssrn.1014550. URL https://ssrn.com/abstract=1014550.

[82] Paul Ressel. Homogeneous distributions—and a spectral representation of classical mean values and stable tail dependence functions. *Journal of Multivariate Analysis*, 117:246–256, 2013.

[83] David Salinas, Michael Bohlke-Schneider, Laurent Callot, Roberto Medico, and Jan Gasthaus. High-dimensional multivariate forecasting with low-rank Gaussian copula processes. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/0b105cf1504c4e241fcc6d519ea962fb-Paper.pdf.

[84] David Salinas, Huibin Shen, and Valerio Perrone. A quantile-based approach for hyperparameter transfer learning. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 8438–8448. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/salinas20a.html.

[85] Ricardo Silva, Charles Blundell, and Yee Whye Teh. Mixed cumulative distribution networks. In Geoffrey Gordon, David Dunson, and Miroslav Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 670–678, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR. URL https://proceedings.mlr.press/v15/silva11a.html.

[86] Abe Sklar. Fonctions de répartition à n dimensions et leurs marges. 8:229–231, 1959.

[87] Michael S. Smith, Quan Gan, and Robert J. Kohn. Modelling dependence using skew t copulas: Bayesian inference and applications. *Journal of Applied Econometrics*, 27(3):500–522, 2012. doi: https://doi.org/10.1002/jae.1215. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/jae.1215.

[88] Natasa Tagasovska, Damien Ackerer, and Thibault Vatter. Copulas as high-dimensional generative models: Vine copula autoencoders. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/15e122e839dfdaa7ce969536f94aecf6-Paper.pdf.

[89] Yaniv Tenzer and Gal Elidan. Speedy model selection (sms) for copula models. In *UAI*, 2013.

[90] Yaniv Tenzer and Gal Elidan. Helm: Highly efficient learning of mixed copula networks. In *UAI*, pages 790–799, 2014.

[91] Dustin Tran, David Blei, and Edo M Airoldi. Copula variational inference. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper/2015/file/e4dd5528f7596dcdf871aa55cfccc53c-Paper.pdf.

[92] USDA. Dataset: CSFII 1985, continuing survey of food intakes by individuals, women 19-50 years of age and their children 1-5 years of age, 6 waves, 1985. Dataset, U.S. Department of Agriculture, Agriculture Research Service, 1985. https://www.ars.usda.gov/ARSUserFiles/80400530/pdf/8586/csfii85_6waves_doc.pdf, https://www.ars.usda.gov/northeast-area/beltsville-md-bhnrc/beltsville-human-nutrition-research-center/food-surveys-research-group/docs/csfii-1985-1986/.

[93] Hongwei Wang, Lantao Yu, Zhangjie Cao, and Stefano Ermon. Multi-agent imitation learning with copulas. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 139–156. Springer, 2021.

[94] Huahua Wang, Farideh Fazayeli, Soumyadeep Chatterjee, and Arindam Banerjee. Gaussian Copula Precision Estimation with Missing Values. In Samuel Kaski and Jukka Corander, editors, *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, volume 33 of *Proceedings of Machine Learning Research*, pages 978–986, Reykjavik, Iceland, 22–25 Apr 2014. PMLR. URL https://proceedings.mlr.press/v33/wang14a.html.

[95] Ruofeng Wen and Kari Torkkola. Deep generative quantile-copula models for probabilistic forecasting. In *ICML*, 2019.

[96] Mario Wieser, Aleksander Wieczorek, Damian Murezzan, and Volker Roth. Learning sparse latent representations with the deep copula information bottleneck. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=HkOwHx-RW.

[97] R. E. Williamson. Multiply monotone functions and their Laplace transforms. *Duke Mathematical Journal*, 23(2):189 – 207, 1956.

[98] Andrew G Wilson and Zoubin Ghahramani. Copula processes. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 23, pages 2460–2468. Curran Associates, Inc., 2010. URL https://proceedings.neurips.cc/paper/2010/file/fc8001f834f6a5f0561080d134d53d29-Paper.pdf.

[99] Toshinao Yoshiba. Maximum likelihood estimation of skew-t copulas with its applications to stock returns. *Journal of Statistical Computation and Simulation*, 88(13):2489–2506, 2018. doi: 10.1080/00949655.2018.1469631. URL https://doi.org/10.1080/00949655.2018.1469631.

[100] Yuxuan Zhao and Madeleine Udell. Matrix completion with quantified uncertainty through low rank Gaussian copula. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 20977–20988. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/f076073b2082f8741a9cd07b789c77a0-Paper.pdf.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix A.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See Appendix B, and the code is available at `https://github.com/yutingng/gen-AX`.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix B.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Appendix B.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix B.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] Nutrient intake data [92], Copulas `https://github.com/sdv-dev/Copulas`, HAC toolbox `https://github.com/gorecki/HACopula`, PyTorch, SciPy, NumPy, Python.
   (b) Did you mention the license of the assets? [No] Most information presented on the USDA Web site is considered public domain information. Public domain information may be freely distributed or copied, but use of appropriate byline/photo/image credits is requested. Attribution may be cited as follows: "U.S. Department of Agriculture."
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] The code in the supplemental material is to be made available online.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No] Data was collected only after consent was given.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] Personally identifiable information has been removed.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [No] The instructions given to participants is available at [92].
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [No] The description of potential participant risks is available at [92].
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [No] The description of participant compensation is available at [92].

A Algorithms and Background

A.1 Algorithms

We provide algorithms to describe the algorithmic contributions in the main paper. Algorithm 1 describes the method for learning the stable tail dependence function (stdf) given the Archimedean generator. Algorithm 2 describes sampling the simplex component with a learned spectral measure. Algorithm 3 describes learning the Archimedean generator with a given stdf. Finally, a sampling algorithm for the full Archimax copula is provided in Algorithm 4 assuming a learned generator and stdf. The code for this paper is available at `https://github.com/yutingng/gen-AX`.

Algorithm 1 Learn stable tail dependence function (stdf)

input observations $\{\mathbf{u}_i : i = 1, ..., m\}$.
input Archimedean generator φ_θ.
initialize $G_{\mathbf{W}}$.
do while loss $=$ NLL $+$ reg **not converged**:
  sample $\{\mathbf{x}_j : j = 1, ..., n\}$ from Unif(Δ_{d-1}).
  compute $\{\xi(\mathbf{u}_i, \mathbf{x}_j) : i = 1, ..., m,\ j = 1, ..., n\}$ from (7).
  sample $\{\mathbf{w}_k : k = 1, ..., l\}$ from $G_{\mathbf{W}}$.
  compute $\{\ell_\theta(\mathbf{x}_j) : j = 1, ..., n\}$ from (10).
  compute NLL $= \frac{1}{mn} \sum_{i=1,j=1}^{i=m,j=n} \log \mathcal{L}(\xi(\mathbf{u}_i, \mathbf{x}_j); \varphi_\theta, \ell_\theta)$ from (9).
  compute reg $= \sum_{j=1}^{d} (\sum_{k=1}^{l} w_{kj}/l - 1/d)^2$.
  descent argmin$_{G_{\mathbf{W}}}$ NLL $+$ reg.
end while
return learned $G_{\mathbf{W}}$.

Algorithm 2 Sample simplex component

input learned $G_{\mathbf{W}}$.
do while $\ell_\theta(\mathbf{s}) > 1$:
  sample $\{u_i : i = 1, ..., d-1\}$ from Unif$(0, 1)$.
  sample $\{(w_{i1}, ..., w_{id}) : i = 1, ..., d-1\}$ from $G_{\mathbf{W}}$.
  compute $\{(x_{i1}, ..., x_{id}) : i = 1, ..., d-1\}$ from (14).
  compute $\mathbf{s} = (s_1, ..., s_d)$ from (15).
  compute $\ell_\theta(\mathbf{s})$ from (10).
end while
return a sample $\mathbf{s} = (s_1, ..., s_d)$.

Algorithm 3 Learn Archimedean generator

input Kendall observations $\mathcal{W} = \{w_i : i = 1, ..., n_r n_z\}$.
input learned G_Z such that $Z := \ell_\theta(\mathbf{S})$, $\mathbf{S}$ sampled from learned $G_{\mathbf{W}}$ with Algorithm 2.
initialize G_R.
sort $\mathcal{W}$ in decreasing (non-increasing) order.
do while MSE $= \sum_{i=1}^{n_r n_z} (w_i - \varphi_\theta(t_i))^2/(n_r n_z) > \epsilon$:
  sample $\mathcal{R} = \{r_1, ..., r_{n_r}\}$ from G_R.
  sample $\mathcal{Z} = \{z_1, ..., z_{n_z}\}$ from G_Z.
  compute $\mathcal{T} = \{r_j z_l,\ r_j \in \mathcal{R}, z_l \in \mathcal{Z}\}$.
  sort $\mathcal{T}$ in increasing (non-decreasing) order.
  compute $(\varphi_\theta(t_1), ..., \varphi_\theta(t_{n_r n_z}))$ from (22).
  descent argmin$_{G_R} \sum_{i=1}^{n_r n_z} (w_i - \varphi_\theta(t_i))^2/(n_r n_z) + (\sum_{j=1}^{n_r} r_j/n_r - 1)^2$.
end while
return learned G_R.

Algorithm 4 Sample Archimax copulas

input learned G_R, $G_{\mathbf{W}}$
sample $r, r_1, \cdots, r_{n_r}$ from G_R.
sample $\mathbf{s} = (s_1, \cdots, s_d)$ from $G_{\mathbf{W}}$ with Algorithm 2.
compute $\mathbf{u} = (\varphi_\theta(rs_1), \cdots, \varphi_\theta(rs_d))$ from (22).
return a sample $\mathbf{u} = (u_1, \cdots, u_d)$.

A.2 Stable tail dependence function inference and sampling

A.2.1 Inference for stable tail dependence function

Pickands transformation The Pickands transformation in (7) for a particular test point $\mathbf{x} \in \Delta_{d-1}$ results in the transformed observation $\xi(\mathbf{U}, \mathbf{x})$ with survival distribution function (SDF):

$$\begin{aligned}
P(\xi(\mathbf{U}, \mathbf{x}) > x) &= P(\min\{\varphi^{-1}(U_1)/x_1, \cdots, \varphi^{-1}(U_d)/x_d\} > x), \\
&= P(\varphi^{-1}(U_1) > xx_1, \cdots, \varphi^{-1}(U_d) > xx_d), \\
&= P(U_1 < \varphi(xx_1), \cdots, U_d < \varphi(xx_1)), && \varphi \text{ is decreasing}, \\
&= \varphi(\ell(xx_1, \cdots, xx_d)), && C(\mathbf{u}) := \varphi(\ell(\varphi^{-1}(\mathbf{u}))), \\
&= \varphi(x\ell(x_1, \cdots, x_d)), && \ell(c\mathbf{x}) = c\ell(\mathbf{x}), c > 0, \\
&= \varphi(x\ell(\mathbf{x})).
\end{aligned}$$

In the case of extreme-value copulas, $\varphi(x) := \exp(-x)$, $\varphi^{-1}(u) := -\log(u)$ and $C(\mathbf{u}) := \exp(-\ell(-\log(u_1), \cdots, -\log(u_d)))$, such that the $d-$dimensionsal observations distributed according to the extreme-value copula are transformed into $1-$dimensional observations distributed according to an exponential with rate $\ell(\mathbf{x})$. The stdf at a particular test point $\ell(\mathbf{x})$ is then estimated from the mean of the transformed observations, with endpoint corrections, as [79]:

$$\hat{\ell}(x) = \left(\sum_{i=1}^{n} -\log\left(\frac{i}{n+1}\right) \right) / \left(\sum_{i=1}^{n} \xi(\mathbf{u}_i, \mathbf{x}) \right). \tag{24}$$

A modification to the Pickands estimator, using the transformation $\log(\xi(\mathbf{U}, \mathbf{x}))$, leads to the CFG estimator [10]:

$$\log(\hat{l}(\mathbf{x})) = \frac{1}{n} \sum_{i=1}^{n} \log\left(-\log\left(\frac{i}{n+1}\right)\right) - \frac{1}{n} \sum_{i=1}^{n} \log(\xi(\mathbf{u}_i, \mathbf{x})). \tag{25}$$

A modification to the CFG estimator was made by Chatelain et al. [16] for the case of Archimax copulas:

$$\log(\hat{l}(\mathbf{x})) = \frac{1}{n} \sum_{i=1}^{n} \log\left(\varphi^{-1}\left(\frac{i}{n+1}\right)\right) - \frac{1}{n} \sum_{i=1}^{n} \log(\xi(\mathbf{u}_i, \mathbf{x})). \tag{26}$$

In the main paper, we provide the likelihood of a transformed observation $\xi(\mathbf{u}, \mathbf{x})$ in (9) and directly train the generative network to maximize this likelihood. We do not have an additional endpoint correction step since the stochastic form of ℓ_θ in (10) is a valid stdf. On the other hand, the Pickands and CFG estimators above have corrected endpoints but may not be valid stdfs [38]. In the additional experimental results in Appendix B.2 our estimator, although based on the Pickands estimator, performs better than the Pickands estimator. This may be due to our estimator representing the class of valid stdfs, a phenomenon noted in [30] where projecting to the class of valid stdfs reduced estimation error. This direct method of training may perform better than the alternative method that first estimates $\hat{\ell}$ then trains the generative network to match $\hat{\ell}$. A future direction may be to improve the robustness of our MLE estimator with the CFG modification.

Architecture of generative network for spectral component $G_{\mathbf{W}}$ is a generic multilayer perceptron with layers: (Linear(d, d_h), BatchNorm(d_h), ReLU(), Linear(d_h, d_h), BatchNorm(d_h), ReLU(), Linear(d_h, d), Softmax()), where d is the dimension of the observations and d_h is the

number of nodes in the hidden layers. The number of layers and number of hidden nodes may be modified as needed. Batch normalization, i.e. BatchNorm$(\cdot)$, greatly helps in preventing samples from being static during training.

Details of Algorithm 1 To reduce computational complexity, we use mini-batch gradient descent and a smaller number of samples in the empirical expectations of φ_θ and ℓ_θ during training.

A.2.2 Sampling the simplex component

Details of Algorithm 2 From Definition 4, the marginals of the simplex component are distributed as Beta$(1, d-1)$, where d is the dimension. To enforce the marginals, we compute the samples of the empirical copula $\mathbf{u}$, then apply the quantile function, i.e. inverse cumulative distribution function (CDF), to obtain $\mathbf{s}$ such that:

$$s_j = 1 - (1 - u_j)^{\frac{1}{d-1}} \text{ for } j = \{1, \cdots, d\}. \tag{27}$$

Given n observations $(x_{11}, \cdots, x_{1d}), \cdots, (x_{n1}, \cdots, x_{nd})$ the samples of the empirical copula $(u_{11}, \cdots, u_{1d}), \cdots, (u_{n1}, \cdots, u_{nd})$ are coordinate-wise rank-normalized such that:

$$u_{ij} = \frac{1}{n} \sum_{k=1}^{n} \mathbb{1}\{x_{kj} \leq x_{ij}\} \text{ for } i = \{1, \cdots, n\}, j = \{1, \cdots, d\}. \tag{28}$$

A.3 Archimedean generator inference and sampling

Kendall distribution function The expression of $K(w)$ for Archimax copulas is:

$$\begin{aligned}
K(w) &= P(C(U_1, \cdots, U_d) \leq w), \\
&= P(\varphi(\ell(\varphi^{-1}(U_1), \cdots, \varphi^{-1}(U_d))) \leq w), && C(\mathbf{u}) := \varphi(\ell(\varphi^{-1}(\mathbf{u})), \\
&= P(\varphi(\ell(RS_1, \cdots, RS_d)) \leq w), && \mathbf{U} \stackrel{d}{=} \varphi(R\mathbf{S}), \\
&= P(\varphi(R\,\ell(S_1, \cdots, S_d)) \leq w), && \ell(c\,\mathbf{s}) = c\,\ell(\mathbf{s}), c > 0, \\
&= P(\varphi(RZ) \leq w), && Z := \ell(\mathbf{S}), \\
&= P(\varphi(T) \leq w), && T := RZ.
\end{aligned}$$

Reconstruction of radial distribution with non-iid random variables and repetition of elements To provide an outline for the overall approach, given the current estimate of $\mathcal{R}$, we compute the mapping between $\mathcal{R}, \mathcal{Z}$ and $\mathcal{T}$, solve for the probabilities $P_\mathcal{R}$, update the support $\mathcal{R}$, and iterate as needed. The algorithm can be understood as an alternating minimization algorithm, where the map between $\mathcal{R}, \mathcal{Z}$ to $\mathcal{W}$ via $\mathcal{T}$, and the support $\mathcal{R}$ are updated in an alternating fashion.

We initialize by computing $\mathcal{W} = \{w_1, \cdots, w_m\}$ and $P_\mathcal{W} = \{p(w_1), \cdots, p(w_m)\}$ of the empirical Kendall distribution function in (18) and (19), where $m \leq n_r n_z$. We do not perform the additional (linear) interpolation step of the main paper. We then sort $\mathcal{W}$ in decreasing order. We also initialize $\mathcal{R} = \{r_{n_r} = 1,\, r_j = r_{j+1}\alpha_j \text{ for } j = \{1, ..., n_r - 1\}\}$, where we select $\alpha = (0.9, ..., 0.9)$. The support $\mathcal{Z}$ and probabilities $P_\mathcal{Z}$ are assumed to be given.

Given $\mathcal{Z}$ and the current estimate of $\mathcal{R}$, we compute

$$\mathcal{T} = \{r_j z_l : r_j \in \mathcal{R}, z_l \in \mathcal{Z}\}. \tag{29}$$

We then sort $\mathcal{T}$ in increasing order and compute the ordering

$$(\sigma_r, \sigma_z)(i) : \{1, \cdots, m\} \to \{1, \cdots, n_r\} \times \{1, \cdots, n_z\} \tag{30}$$

defined as a surjective function such that

$$t_i = r_{\sigma_r(i)} z_{\sigma_r(i)}. \tag{31}$$

We solve for the probabilities $P_\mathcal{R} = \{p(r_1), \cdots, p(r_{n_r})\}$ by minimizing the residuals

$$\sum_{i=1}^{m} \left(\left(\sum_{(\sigma_r^{-1}(j), \sigma_z^{-1}(l)) = i} p(r_j) p(z_l) \right) - p(w_i) \right)^2. \tag{32}$$

We solve for the support $\mathcal{R} = \{r_1, \cdots, r_{n_r}\}$, by minimizing the residuals

$$\sum_{i=1}^{m} (\varphi_\theta(t_i) - w_i)^2 \tag{33}$$

where, following Definition 3,

$$\varphi_\theta(t_i) = \sum_{j=1}^{n_r} p(r_j) \left(1 - \frac{t_i}{r_j}\right)_+^{d-1}. \tag{34}$$

Numerical illustration We give the details of the algorithm with a simple numerical example.

Consider the supports $\mathcal{R} = (r_1, r_2, r_3) = (1, 2, 3)$, $\mathcal{Z} = (z_1, z_2) = (0.5, 0.75)$ with probabilities $P_\mathcal{R} = (0.4, 0.4, 0.2)$, $P_\mathcal{Z} = (0.25, 0.75)$ and noisy observations $\mathcal{W} = (0.07, 0.19, 0.34, 0.49, 0.67)$, $P_\mathcal{W} = (0.07, 0.33, 0.14, 0.31, 0.15)$.

In this case, $(t_1, t_2, t_3, t_4, t_5) = (0.5, 0.75, 1.0, 1.5, 2.25)$ corresponds to $(r_1 z_1, r_1 z_2, r_2 z_1, r_2 z_2 \cup r_3 z_1, r_3 z_2)$ with probabilities $(0.1, 0.3, 0.1, 0.3 + 0.05, 0.15)$.

For $P_\mathcal{R}$, we solve the overdetermined system of linear equations:

$$\begin{pmatrix} 0.25 & & \\ 0.75 & & \\ & 0.25 & \\ & 0.75 & 0.25 \\ & & 0.75 \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} = \begin{pmatrix} 0.07 \\ 0.33 \\ 0.14 \\ 0.31 \\ 0.15 \end{pmatrix} \tag{35}$$

and obtain the solution as $(0.43, 0.37, 0.20)$, where the solution has been normalized to sum to 1.

For $\mathcal{R}$, we minimize the residuals in (33). Since scaling such that $\mathcal{R} = \{cr_1, \cdots, cr_{n_r}, c > 0\}$ does not change the copula, we solve for $\mathcal{R}$ in terms of ratios $(\alpha_1, \cdots, \alpha_{n_r - 1})$, recursively defined such that $r_{n_r} = 1$ and $r_j = r_{j+1}\alpha_j$ for $j = \{1, \cdots, n_r - 1\}$.

The full technique is presented in Algorithm 5, with code attached in the supplementary material.

Algorithm 5 Estimate radial component

input $\mathcal{W} = \{w_i : i = 1, ..., m\}$, $P_\mathcal{W} = \{p(w_i) : i = 1, ..., m\}$.
input $\mathcal{Z} = \{z_l : l = 1, ..., n_z\}$, $P_\mathcal{Z} = \{p(z_l) : l = 1, ..., n_z\}$.
initialize $(\alpha_1, ..., \alpha_{n_r - 1})$ for instance $(0.9, ..., 0.9)$.
sort $\mathcal{W}$ in decreasing order.
do while MSE $= \frac{1}{m}\sum_{i=1}^{m}(\varphi_\theta(t_i) - w_i)^2 > \epsilon$:
  compute $\mathcal{R} = \{r_{n_r} = 1,\ r_j = r_{j+1}\alpha_j$ for $j = 1, ..., n_r - 1\}$.
  compute $\mathcal{T} = \{r_j z_l,\ r_j \in \mathcal{R}, z_l \in \mathcal{Z}\}$.
  sort $\mathcal{T}$ in increasing order.
  compute $\{(\sigma_r, \sigma_z)(i) : i = 1, ..., m\}$ from (31).
  solve $P_\mathcal{R}$ from (32).
  compute $(\varphi_\theta(t_1), ..., \varphi_\theta(t_m))$ from (34).
  solve $\text{argmin}_\alpha \sum_{i=1}^{m}(w_i - \varphi_\theta(t_i))^2$ such that $\alpha \in (\alpha_l, \alpha_u)$ for instance (0.01,1).
end while
return Estimated support $\mathcal{R}$ and probabilities $P_\mathcal{R}$.

The ratios α may be solved iteratively for a unique solution. In our case, motivated by the uniform convergence of the empirical process $\sqrt{n}(K_n - K)$ as $n \to \infty$ [4], we optimize for a least-squares solution with bounds $\alpha \in (0.01, 1)$ using `scipy.optimize.least_squares` [8].

The disadvantage of the above general approach compared to the approach presented in the main paper is the direct relationship between the computation cost and n_r, n_z, the sizes of supports for R, Z.

Architecture of generative network for radial component G_R is a generic multilayer perceptron with layers: (Linear$(1, d_h)$, BatchNorm(d_h), ReLU(), Linear(d_h, d_h), BatchNorm(d_h), ReLU(), Linear$(d_h, 1)$, Exp()), where d_h is the number of nodes in the hidden layers. The number of layers and number of hidden nodes may be modified as needed. Unlike in $G_\mathbf{W}$, the use of batch normalization is not essential in G_R.

Details of Algorithm 3 To speed up training, we initially resample $\mathcal{Z}$ only once every k mini-batch iterations, decreasing k until $k = 1$ as we approach convergence.

A.4 Inference and sampling for Archimax copulas

A.4.1 Inference for Archimax copulas

Pre-process for extreme-value dependence for initial estimate of stdf To determine the block size for the block maximas in (23), we use the test for extreme-value dependence via the max-stable property by Kojadinovic et al. [55], where the max-stable property is defined as:

$$C(u_1, \cdots, u_d) = C^r(u_1^{1/r}, \cdots, u_1^{1/r}), \text{ for } r = 1, 2, \cdots, \mathbf{u} \in [0, 1]^d. \tag{36}$$

The Cramér–von Mises (CvM) distance in (38) between $C(u_1, \cdots, u_d)$ and $C^r(u_1^{1/r}, \cdots, u_1^{1/r})$ is computed using Monte Carlo integration with samples $\mathbf{u}$ drawn uniformly at random from $[0, 1]^d$.

Details of Algorithm 1 for initial estimate of stdf Randomizing the order of observations may help to create different block-maximas in each mini-batch iteration.

Alternative initialization scheme We also considered initialization with different one-parameter families of Archimedean generators, with choice of generator based on the highest log-likelihood of transformed observation ξ, from equations (7) and (9). The parameter for each family may be computed from an average of inversion of pairwise Kendall tau, as per the following equation [11], for each pair:

$$\tau_{\varphi,\ell} = \tau_\ell + (1 - \tau_\ell)\tau_\varphi, \tag{37}$$

where $\tau_{\varphi,\ell}$ is the Kendall's tau of the Archimax bivariate marginal, τ_ℓ is the Kendall's tau of the extreme-value component and τ_φ is the Kendall's tau of the Archimedean component. An average of inversion of pairwise Kendall tau was employed in [16], with emphasis on the Clayton generator.

Initialization with specific families of Archimedean generators might bias initialization, and thus we suggested initializing via the Archimedean generator first set to $\varphi(x) = \exp\{-x\}$ representing extreme-value copulas and pre-processing the initial data to have extreme-value dependence via block-maximas. This was also motivated by the experiment on extrapolating to extremes.

Identifiability It follows from a result of Chatelain et al. [16] that the sources of non-identifiability in modeling Archimax copulas are only in: (i) power transformation of φ and ℓ, and (ii) the scale ambiguity of φ. The power transformation of φ and ℓ can be illustrated through the following example: Consider both pairs of generators and stdf ($\varphi_a(x) = \exp(-x^{1/\theta})$ and $\ell_a(\mathbf{x}) = \|\mathbf{x}\|_1$) and ($\varphi_b(x) = \exp(-x)$ and $\ell_b(\mathbf{x}) = \|\mathbf{x}\|_\theta = (x_1^\theta + \cdots + x^\theta)^{1/\theta}$). Both (φ_a, ℓ_a) and (φ_b, ℓ_b) lead to the same Archimax copula. The scale ambiguity of φ comes from the fact that $\varphi(cx), c > 0$ leads to the same Archimax copula.

We note that these sources of non-identifiability are non-issues in our methods. For the power transformation, there is no ambiguity of φ and ℓ since the class of $1 - \varphi(1/\cdot)$ where $\varphi(\cdot)$ is calculated as the Williamson d-transform of R with a finite support is regularly varying with index -1 [16, 5]. In addition, we include a regularization term such that $\mathbb{E}_R[R] = 1$.

A.4.2 Sampling for Archimax copulas

Details of Algorithm 4 Given learned generative networks $G_R, G_{\mathbf{W}}$, we can generate many samples from the Archimax copula.

A.5 Background on Archimax copulas

Archimax copulas generalize *Archimedean* and *extreme-value* copulas. They allow asymmetry and arbitrary tail dependence. They were initially developed as a tool to study the behaviour of methods used to estimate the joint distribution of extreme events [10]. The main motivation for Archimax copulas is to model extreme data (e.g. very strong and rare earthquakes) from a mix of moderately

less extreme data (e.g. strong earthquakes) and extreme data. This in turn can be used to generate samples for further studies and simulations.

Archimax copulas were applied to applications such as nutrient intake [71], river flow rates [3] and rainfall [16], where the dependence is asymmetric and sub-asymptotic. In these applications, the authors noted a better fit when using Archimax copulas over Archimedean and extreme-value copulas.

We provide a few connections between Archimax, Archimedean and extreme-value copulas:

- When the *Archimedean generator* $\varphi(x) = \exp(-x)$, and the *radial component* $R \sim \text{Erlang}(d)$, Archimax copulas reduce to extreme-value copulas.
- When the *stable tail dependence function (stdf)* $\ell(\mathbf{x}) = (x_1 + \cdots + x_d) = \|\mathbf{x}\|_1$ and the simplex component $\mathbf{S} \sim \text{Unif}(\Delta_{d-1})$, Archimax copulas reduce to Archimedean copulas.

Archimax copulas have intuitive interpretations, such as scale mixture of extremes, dependent frailties and resource sharing. In the case of resource sharing, $R > 0$ is a resource to be distributed randomly among d agents in a way specified by $\mathbf{S}$, where both R and $\mathbf{S}$ are themselves results of independent random processes. For example, R may be profits, and $\mathbf{S}$ may be the way profits is to be divided between stakeholders.

A.6 Background on multivariate copulas

Copulas are cumulative distribution functions (CDFs) of dependent uniform random variables. They summarize the dependence described by an arbitrary joint CDF after the marginals have been normalized to be uniform. They provide easy marginalization and calculation of tails. In addition, when used in a graphical model, some conditional independence that cannot be easily represented with Markov random fields or Bayesian networks, can be easily represented with cumulative distribution networks [45]. They are also particularly convenient in some applications, such as ranking, where the likelihood is a CDF [43].

For an introduction to copulas, the following textbooks and collection of works are great resources [32, 74, 51, 49, 50].

We also summarize the common multivariate copulas in Table 5.

Table 5: Multivariate copulas

GAUSSIAN (GC)	$\Phi_{\mathbf{R}}(\Phi^{-1}(u_1), \cdots, \Phi^{-1}(u_d))$
VINE (RV, CV, DV)	$\prod_{e \in \mathcal{E}(\mathcal{V})} c_{U_{e_1}, U_{e_2} \mid \{U_{e_d}\}}(F_{U_{e_1} \mid \{U_{e_d}\}}(u_{e_1}), F_{U_{e_2} \mid \{U_{e_d}\}}(u_{e_2}))$
ARCHIMEDEAN (AC)	$\varphi(\varphi^{-1}(u_1) + \cdots + \varphi^{-1}(u_d))$
H. ARCHIMEDEAN (HAC)	$C_0(C_1(\cdots), \cdots, C_J(\cdots))$, $C_i \in$ AC
EXTREME-VALUE (EV)	$\exp\{-\ell(-\log(u_1), \cdots, -\log(u_d))\}$

The Gaussian copula (GC) has a tractable expression for both the CDF and the density. However, it is independent in the tails, a significant reason why Gaussian copula is not suitable for modeling financial risks. Vine copulas, such as R-vines (RV), C-vines (CV) and D-vines (DV), are computationally intensive and hard to interpret due to repeated conditioning with pair copulas. Archimedean copulas are symmetric in all coordinates, which is an assumption that is usually not held in practice. Hierarchical Archimedean copulas aim to break this symmetry but are difficult to construct due to nesting conditions that are hard to satisfy. Extreme-value copulas are max-stable copulas which results in lower tail independence, an assumption that is sometimes not held in practice. Many copulas are not flexible in the extremes, which leads to independence, except in the case of Archimdean copulas which only Gumbel copulas satisfy the tail dependence. In high dimensions, none of the existing copulas typically fit data well. Model misspecification is often accepted in return for tractability, and some dependence is better than independence [41].

Inferring the parameters of a copula is usually done via maximum likelihood estimation if a density can be computed, or by using minimum distance estimator and goodness-of-fit tests if a density cannot be computed. In both cases, expectations are usually replaced by their empirical versions. For more background on estimating copulas, see [15].

Sampling from a copula using the conditional sampling method with Rosenblatt transform is usually not possible in high dimensions, due to repeated differentiation. In our experiments, the conditional sampling method, using automatic differentiation in `PyTorch` breaks down at dimension $d = 4$. In general, only models with stochastic representation may be easy to sample [69].

A.7 Background on copulas in machine learning

Copulas is a rising topic in machine learning, as evident from numerous publications, including but not limited to:

- Cumulative distribution networks, modeled as a product of copulas [45, 44, 46]. They can represent some conditional independencies not represented by Markov random fields and Bayesian networks, allow loops [52] and mixed graphs [85], with application to ranking [43] and heavy-tailed distributions [52].
- Copula Bayesian networks, modeled as a product of conditional copulas [26], with application to missing data [27], classification [28], time-series [25] and fast structure learning [89, 90].
- Copula processes [98] and application of copulas to time-series [66, 39, 83, 95].
- Copula based dependence measures and distances [67, 57, 80, 13, 70].
- Copula variational inference, allowing dependencies between latent variables [91, 37, 40].
- Generative modeling [88, 17, 59, 48, 1, 7].
- Applications of copula in areas including graph neural networks [68], multi-label learning [64], multi-agent interactions [93], bundle pricing [61], missing value [94, 58, 100], sparse representation [96], outlier detection [62], causal discovery [20], domain adaptation and transfer learning [65, 84] and structure learning [12].
- Recent work on deep network based copulas, including Archimedean [63, 75], extreme-value [38], autoregressive [76, 53] and transformer-attentional copulas [24].

B Experiments

The metric we use to compare the methods is based on the Cramér-von Mises (CvM) statistic [81] which is defined as:

$$\text{CvM} = \int (C_{*,n}(\mathbf{u}) - C_{\theta,n}(\mathbf{u}))^2 \, \mathrm{d}\mathbf{u}, \tag{38}$$

where $C_{*,n}$ is the empirical copula of true samples and $C_{\theta,n}$ is the empirical copula of generated samples and the integral is computed using Monte Carlo integration with 10,000 samples of $\mathbf{u}$ drawn uniformly at random from $[0, 1]^{d-1}$.

The empirical copula for n given observations $(u_{11}, \cdots, u_{1d}), \cdots, (u_{n1}, \cdots, u_{nd})$ is defined as:

$$C_n(\mathbf{u}) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\{u_{i1} \leq u_1, \cdots, u_{id} \leq u_d\}. \tag{39}$$

All timings are with a 2.7 GHz Intel Core i7, 16GB 2133MHz LPDDR3.

B.1 Inference for Archimedean generator

We summarize the common Archimedean generators in Table 6.

The Clayton (C) generator is lower tail dependent, upper tail independent, the Frank (F) generator is symmetric in both lower and upper tails, the Joe (J) and Gumbel (G) generators are lower tail independent and upper tail dependent. Thus the above generators represent different radial envelopes.

The map λ is commonly used to estimate and evaluate estimates of φ [33, 34]. The map λ is defined as:

$$\lambda(w) = \varphi^{-1}(w)/(\varphi^{-1}(w))' = \{\varphi' \circ \varphi^{-1}(w)\}\varphi^{-1}(w), \tag{40}$$

Table 6: Archimedean generators

	$\varphi_\theta(x)$	$\theta_{\tau=0.2}$	$\theta_{\tau=0.5}$
CLAYTON (C)	$(1+x)^{-1/\theta}$	0.5	2
FRANK (F)	$-\log(1-(1-\exp(-\theta))\exp(-x))/\theta$	1.86	5.74
JOE (J)	$1-(1-\exp(-t))^{1/\theta}$	1.44	2.86
GUMBEL (G)	$\exp(-t^{1/\theta})$	1.25	2

such that an estimate of φ can be recovered from λ as

$$\varphi^{-1}(w) = \exp\left\{\int_{w_0}^{w} 1/\lambda(t)dt\right\}. \tag{41}$$

It is more convenient to present results in terms of λ since it is scale invariant, unlike φ, where $\varphi(cx)$ for any $c > 0$ lead to the same copula. In dimension $d = 2$, λ is directly related to the Kendall distribution function as $\lambda(w) = w - K(w)$. As such, the asymptotic variance for λ_n may be computed from the asymptotic variance for K_n, where n is the number of observations. This relationship is more complicated in dimension $d > 2$ but often used as an approximate, useful for drawing confidence bands around λ_n to reject models whose λ_θ fall outside the band. In addition, the asymptotic variance of the independence copula may be easily computed as:

$$\sigma^2_{\lambda_n}(x) = \frac{x(x - \log(x) - 1)}{n}. \tag{42}$$

Selection of support sizes n_r, n_z For $n_r = 100$ and $n_z \in \{20, 30, \cdots, 100\}$, we plot the estimates of λ in Figure 3. Results from different runs are in blue. The ground truth is a Clayton generator with $\tau = 0.2$ and a negative scaled extremal Dirichlet stdf with $\alpha = (1, 1, 1, 1, 2, 2, 2, 3, 3, 4), \rho = 0.69$. The ground truth is plotted in black and approximate confidence bands around the ground truth is in dotted black. The computed the mean squared error (MSE) in fitting K and λ and the time taken are given in Figure 4 and Table 7, where the standard deviation is given in parenthesis.

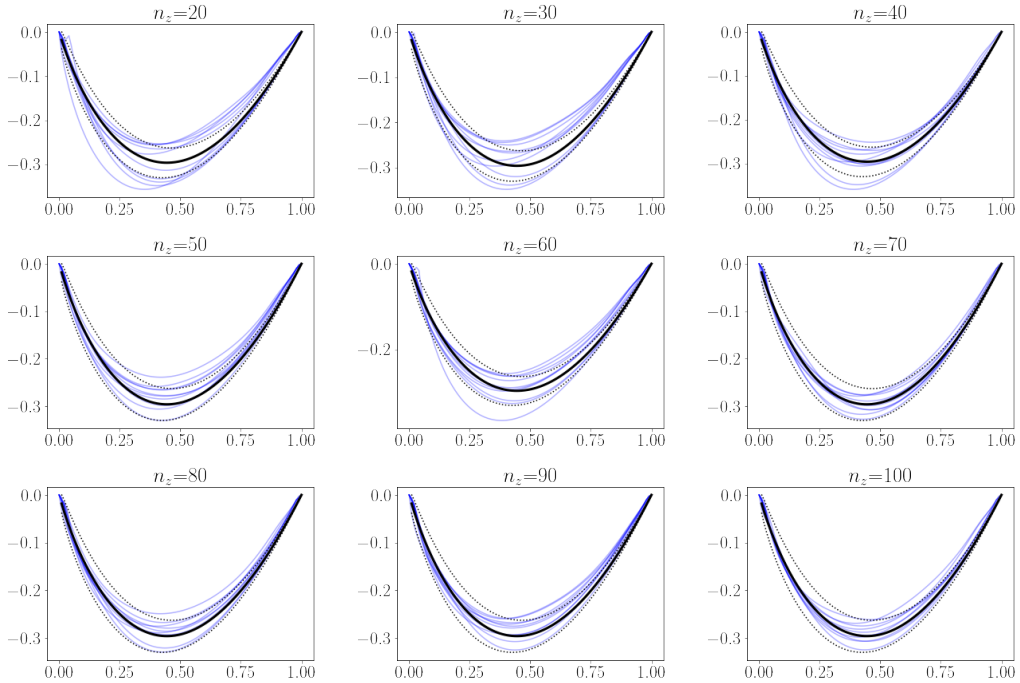

Figure 3: Estimates of λ with $n_r = 100$ and $n_z \in \{20, \cdots, 100\}$. Ground truth in black, approximate confidence bands in dotted black. Estimates from different runs in blue.

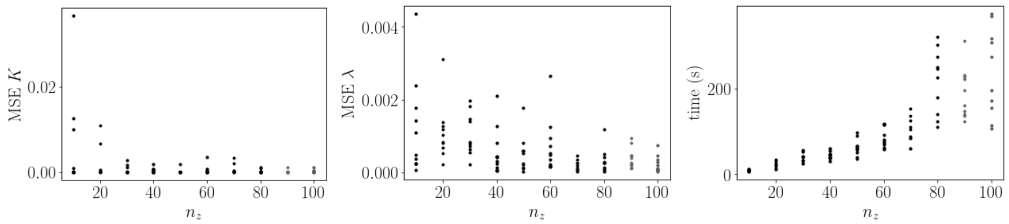

Figure 4: Mean squared error (MSE) in fitting K and λ and the time taken.

Table 7: Mean squared error (MSE) in fitting K and λ and the time taken.

$n_r = 100, n_z$	10	20	30	40	50	60	70	80	90	100
MSE K $\times 10^{-3}$	5.99(11.08)	1.84(3.60)	0.55(0.91)	0.34(0.56)	0.21(0.57)	0.46(1.02)	0.60(1.06)	0.23(0.39)	**0.12(0.31)**	0.16(0.31)
MSE λ $\times 10^{-3}$	1.24(1.27)	1.10(0.75)	1.04(0.56)	0.57(0.62)	0.49(0.49)	0.74(0.72)	**0.18(0.13)**	0.32(0.32)	0.39(0.27)	0.23(0.21)
TIME (SEC)	8.59(1.78)	21.30(7.10)	40.06(11.11)	43.46(8.23)	62.46(18.45)	86.08(21.91)	100.62(29.75)	217.68(71.45)	190.10(56.29)	239.04(96.97)

From the results, for $n_r = 100$, it would be appropriate to use $n_z \in \{70, \cdots, 100\}$, with a tradeoff between accuracy and computation time.

Plots of λ for sample size $n = 1000$ While our method infers an arbitrary Archimedean generator and takes the joint dependence across covariates into account, the method in [16] infers a Clayton generator from pairwise Kendall taus. Thus the performance gap between our method and the method in [16] is expected to increase as the generator differs from the Clayton generator and as the observations become less symmetric.

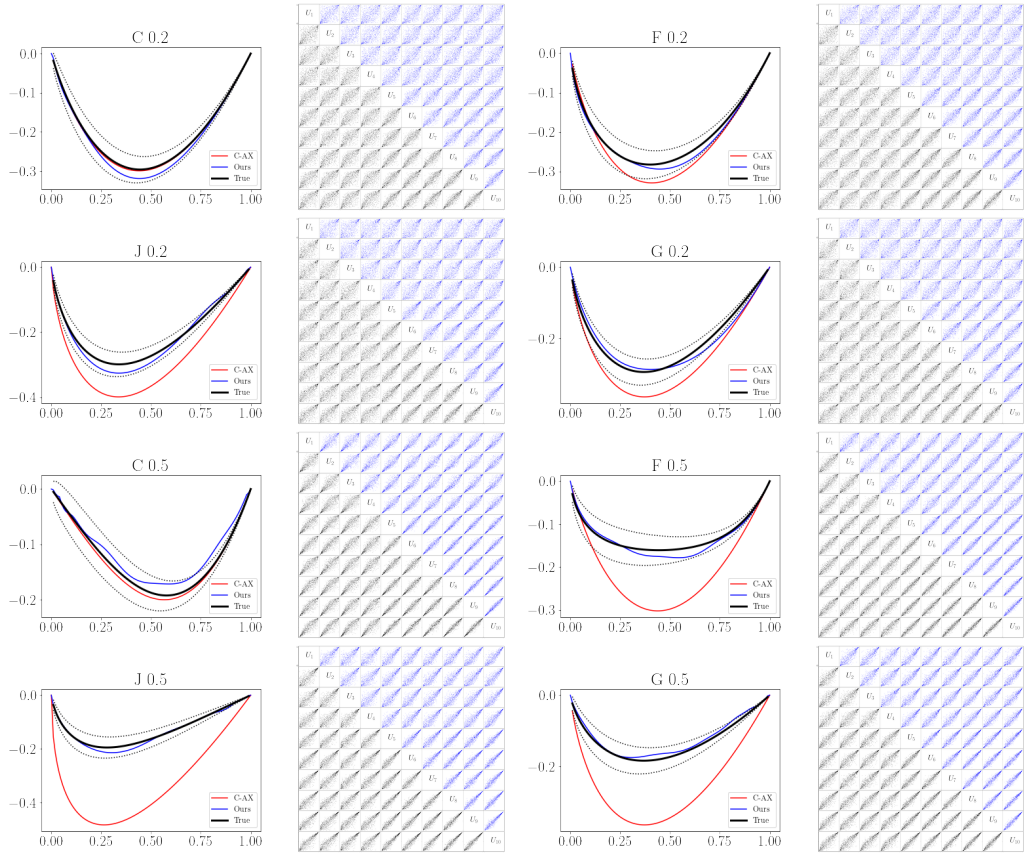

Figure 5: Estimates of λ given samples of $\mathbf{S}$ from true ℓ for $n = 1000$. Ground truth in black, approximate confidence bands in dotted black, the method from [16] in red, our method in blue. On the right of each plot of λ is a plot of samples from the copula, ground truth below the diagonal in black, our method above the diagonal in blue.

Plots of λ for sample size $n = 200$ Increasing the number of observations n improved estimation accuracy, hinting at consistency.

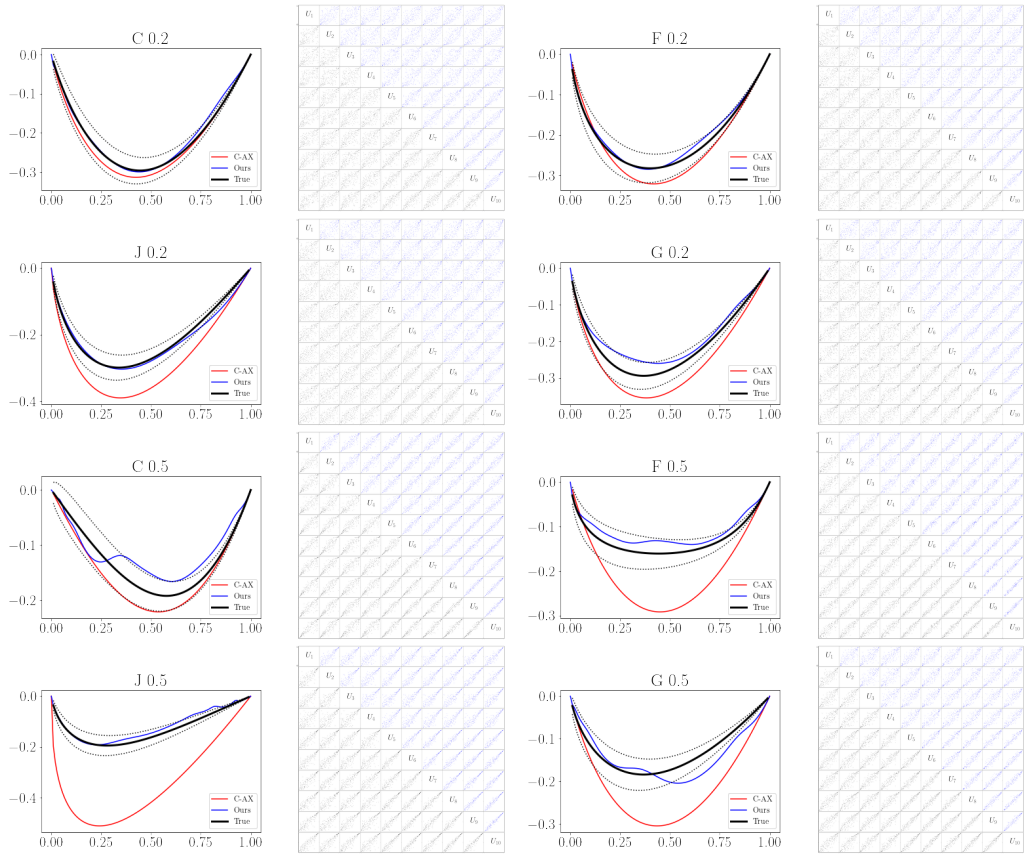

Figure 6: Estimates of λ given samples of $\mathbf{S}$ from true ℓ for $n = 1000$. Ground truth in black, approximate confidence bands in dotted black, the method from [16] in red, our method in blue. On the right of each plot of λ is a plot of samples from the copula, ground truth below the diagonal in black, our method above the diagonal in blue.

B.2 Inference for stable tail dependence function and sampling for simplex component

The negative scaled extremal Dirichlet (NSD) [5] is a rich class encompassing many parametric models of the stdf and spectral component, including the logistic, asymmetric logistic, negative logistic, and extremal Dirichlet models [18].

It is specified by:

$$\ell(\mathbf{x}) = \frac{\Gamma(\alpha_1 + \cdots + \alpha_d - \rho)}{\Gamma(\alpha_1 + \cdots + \alpha_d)} \mathbb{E}_{\mathbf{D}} \left[\max_{j=1,\cdots,d} \left(\frac{x_j D_j^{-\rho} \Gamma(\alpha_j)}{\Gamma(\alpha_j - \rho)} \right) \right], \tag{43}$$

where $\mathbf{D} = (D_1, ..., D_d)$ is distributed as a Dirichlet$(\alpha_1, ..., \alpha_d)$ with $\alpha_1, ..., \alpha_d > 0$ and $\rho \in (0, \min(\alpha_1, \cdots, \alpha_d))$.

The integrated relative absolute error (IRAE) is commonly used to evaluate estimates of ℓ, and is given by [16]:

$$\text{IRAE}(\ell, \ell_\theta) = \frac{1}{|\Delta_{d-1}|} \int_{\Delta_{d-1}} |\ell(\mathbf{x}) - \ell_\theta(\mathbf{x})| / \ell(\mathbf{x}) \, \mathrm{d}\mathbf{x}. \tag{44}$$

The IRAE is computed using Monte Carlo integration with 10,000 samples $\mathbf{x}$ drawn uniformly at random from the simplex Δ_{d-1}.

For given true φ, we report the IRAE of the modified Pickands estimator from [16], the modified CFG estimator from [16], and our method in Table 8. The stdf is a NSD with $\alpha = (1, 1, 1, 1, 2, 2, 2, 3, 3, 4)$, $\rho = 0.69$.

Table 8: Inference of ℓ given true φ

	C 0.2	C 0.5	F 0.2	F 0.5	J 0.2	J 0.5	G 0.2	G 0.5
IRAE ± 0.01 P [16]	0.16	1.00	0.05	0.06	0.06	0.07	0.08	0.17
IRAE ± 0.01 CFG [16]	0.05	0.11	0.04	0.04	0.05	1.00	0.06	0.15
IRAE ± 0.01 (OURS)	0.06	0.12	0.04	0.05	0.06	**0.07**	0.08	0.15

As mentioned in Appendix A.2.1, though our estimator is based on the Pickands estimator, it performs better than the Pickands estimator. This suggests that our direct method of training with the generative network generating the class of valid stdfs would perform better than the alternative method that first estimates $\hat{\ell}$ then train the generative network to match $\hat{\ell}$. As noted in [16], the modified CFG estimator performs better than the modified Pickands estimator for Archimax copulas. A future direction may be to improve the robustness of our estimator with the CFG modification.

We also provide the IRAE and time taken versus number of minibatch iterations in Figure 7. The plots suggest that our algorithm converges and is not computationally intensive.

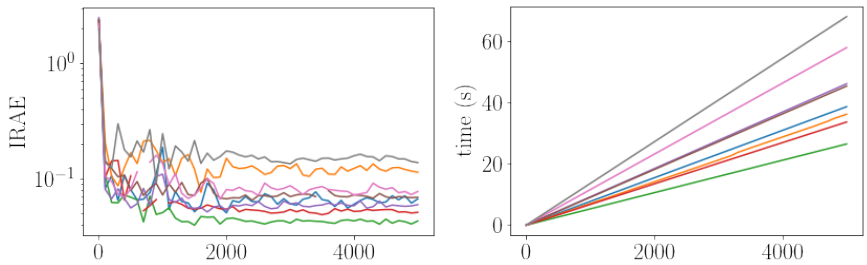

Figure 7: IRAE in fitting ℓ and the time taken versus number of mini-batch iterations. Each line is for a different copula setting in Table 8.

B.3 Modeling nutrient intake

The data and documentation from the study of nutrient intake in women is made available by the U.S. Department of Agriculture (USDA) [92]. There are $n = 1459$ observations of dimension $d = 17$, corresponding to the variables: Energy, Protein, Vitamin A (IU), Vitamin A (RE), Vitamin E, Vitamin C, Thiamin, Riboflavin, Niacin, Vitamin B6, Folate, Vitamin B12, Calcium, Phosphorus, Magnesium, Iron, Zinc. This dataset was previously studied in [71] where using a Clayton-NSD Archimax copula improved fit over a Clayton Archimedean copula. However, in [71], the experiment was limited to only $n = 737$ and $d = 3$, corresponding to Calcium, Iron and Protein.

We compared our method to copula based models and deep network based models from literature using the CvM distance in (38). The abbreviations for the copula based models in Table 2 of the main paper correspond to the abbreviations in Table 5 summarizing the common multivariate copulas. They are Gaussian (GC), R-vine (RV), C-vine (CV), D-vine (DV), Archimedean (AC *), hierarchical Archimedean (HAC) [36] and extreme-value (EV †) copulas. The abbreviation (C-AX †) corresponds to the state-of-the-art in inferring Archimax copulas with a Clayton generator for φ and the modified CFG estimator for ℓ [16]. The abbreviations for the deep network based models in Table 2 are Wasserstein GAN with gradient penalty (WGAN), masked autoregressive flow (MAF) and variation autoencoders (VAE). Lastly, the abbreviation (Gen-AX *†) corresponds to our method. For methods marked with *, we use φ described in Algorithm 3 and for methods marked with † we use ℓ and the sampling methods described in Algorithms 2 and 4.

Table 2 shows our method outperforming the above methods using the CvM distance.

We additionally provide plots of generated samples versus true samples in Figure 9 (a-l). The generated samples are plotted in blue above the diagonal while the true samples are plotted in black below the diagonal. From the plot of our generated samples in Figure 9 (l), an improvement can be made with hierarchical Archimax copulas [42] to have different Archimedean generators φ and thus different radial envelopes for covariates.

We briefly describe the training process using our proposed method. We first initialize ℓ_θ with Algorithm 1 on block-maximas. Using the test for extreme-value dependence [55], we chose the block size $n/k = 5$. The plots for block sizes $n/k \in \{1, 2, 5, 10\}$ and exponents $r \in \{2, 3, \cdots, 10\}$ are given in Figure 8.

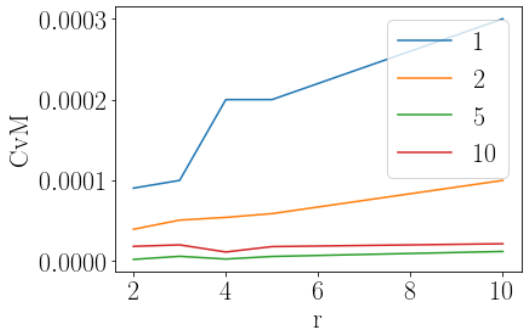

Figure 8: Selection of block size $n/k = 5$ using the test for extreme-value dependence [55].

We compared our initial estimate ℓ_θ to the state-of-the-art CFG estimator applied on the block maximas $\hat{\ell}$ [10]. The plot of IRAE against mini-batch iterations show convergence in 2000 mini-batch iterations, with a duration of 20s, and an IRAE of 0.08. We then learn φ_θ with Algorithm 3 on the full dataset, with samples of $\mathbf{S}$ from Algorithm 2. The plot of MSE in fitting the empirical Kendall distribution show convergence in 2000 mini-batch iterations, with a duration of 25s, and an MSE of 0.0008. We note that the initialization scheme seems to be performing well since the learnt φ_θ focused on modifying only the lower tail. We then update ℓ_θ with Algorithm 1 on the full dataset, given φ_θ. The NLL of transformed observations ξ went from 1.67 in the initialization to -1.12 with the use of φ_θ. The IRAE to the state-of-the-art modified CFG estimator on the full dataset was 0.078.

The estimation of the copula based models was done using the `Copulas` library [21] in Python and the `HACopula` toolbox [35] in MATLAB.

The architectures of the deep networks are:

- WGAN: 3 layers for generator, 3 layers for discriminator, hidden size 128.
- MAF: 2 flows, hidden size 128 for each flow.
- VAE: 3 layers for encoder, 2 layers for decoder, hidden size 128, latent size 16.
- Gen-AX: 3 layers hidden size 30 for $G_{\mathbf{W}}$, 3 layers hidden size 10 for G_R.

The implementation was with `PyTorch`, the Adam optimizer was used with learning rate 1e-3.

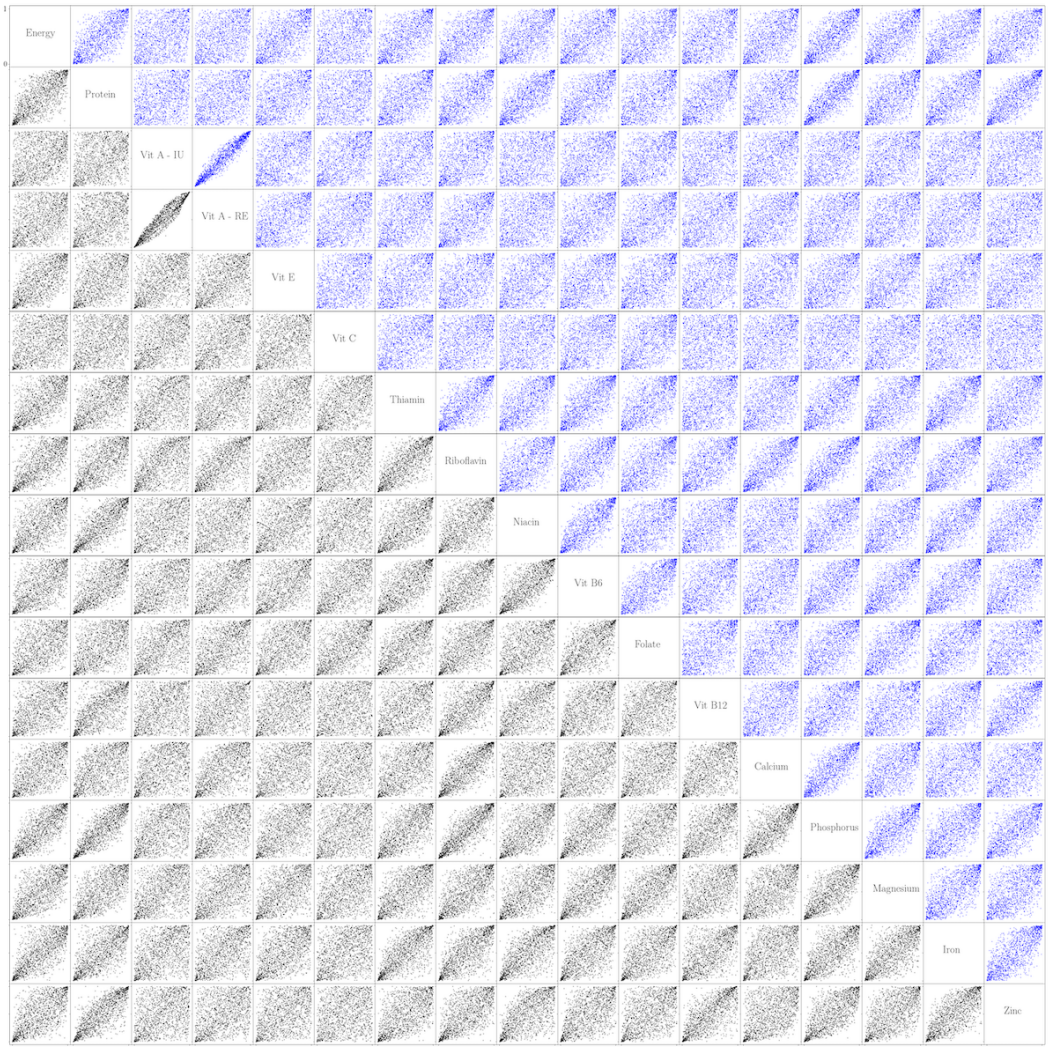

Figure 9: (a) Gaussian copula (GC).

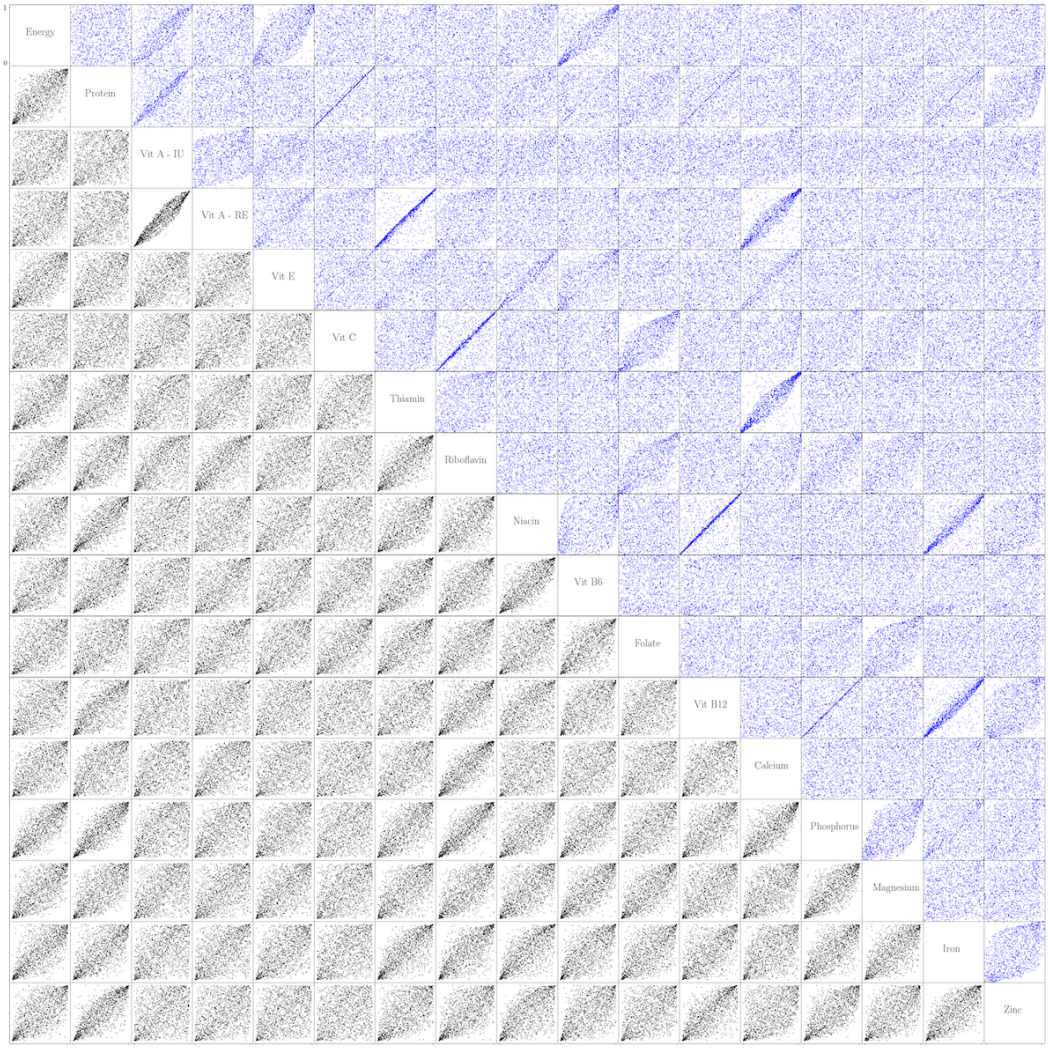

Figure 9: (b) R-vine (RV).

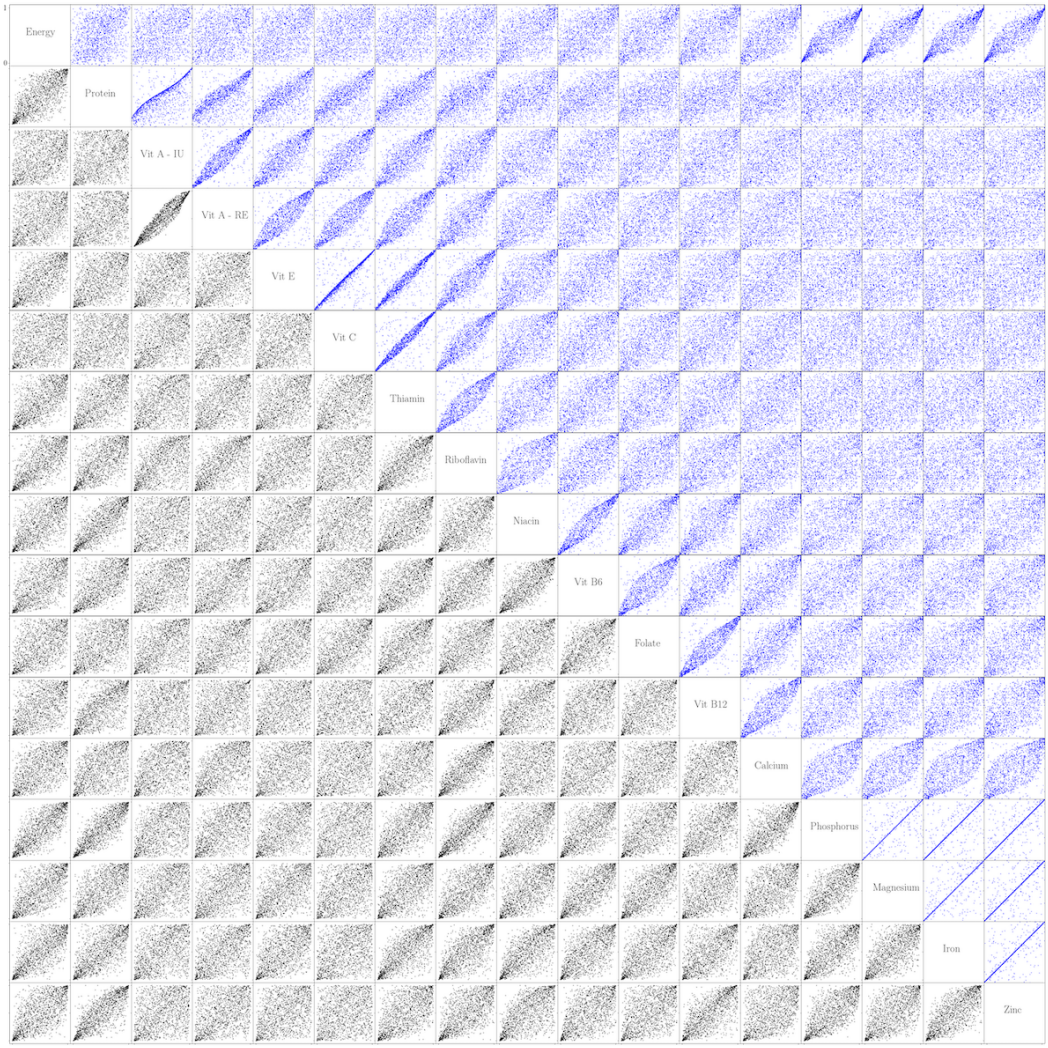

Figure 9: (c) C-vine (CV).

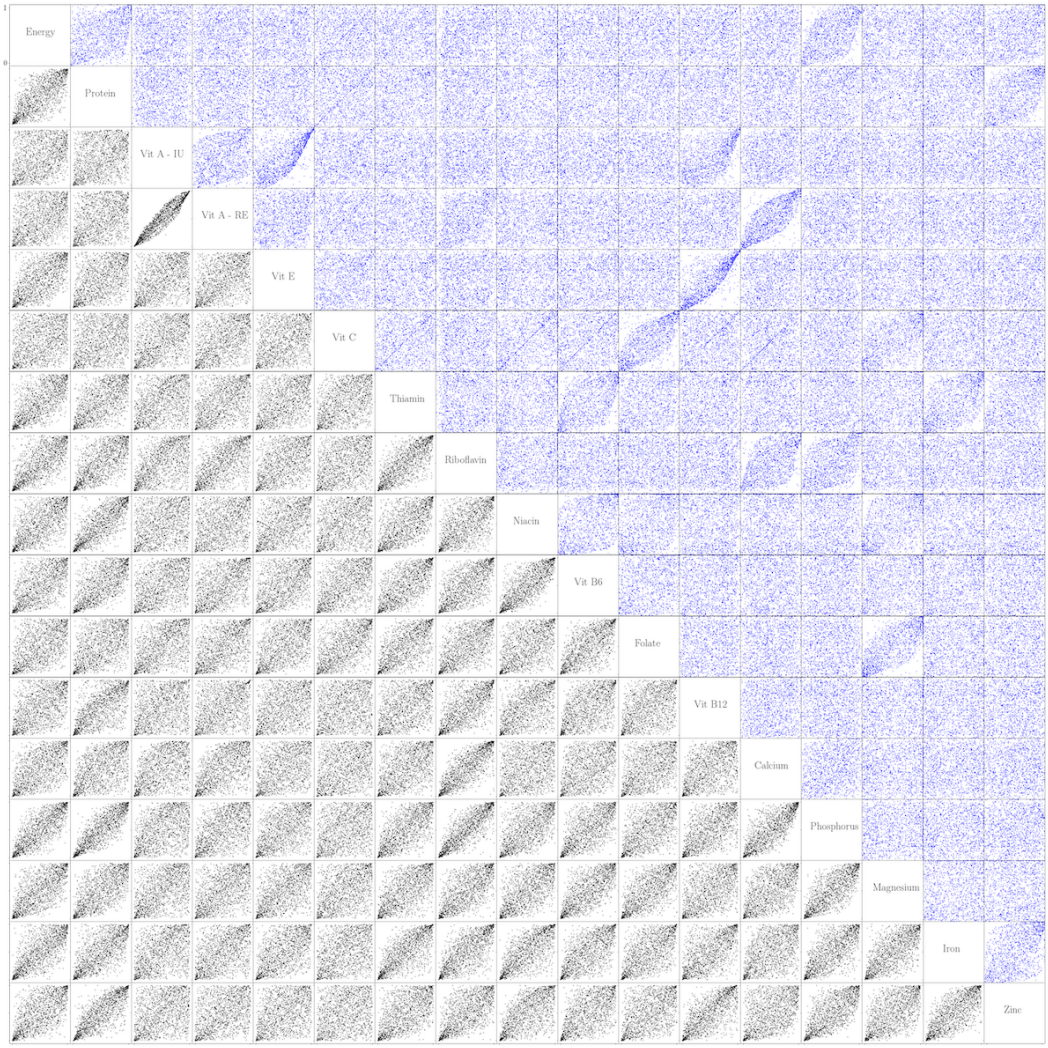

Figure 9: (d) D-vine (DV).

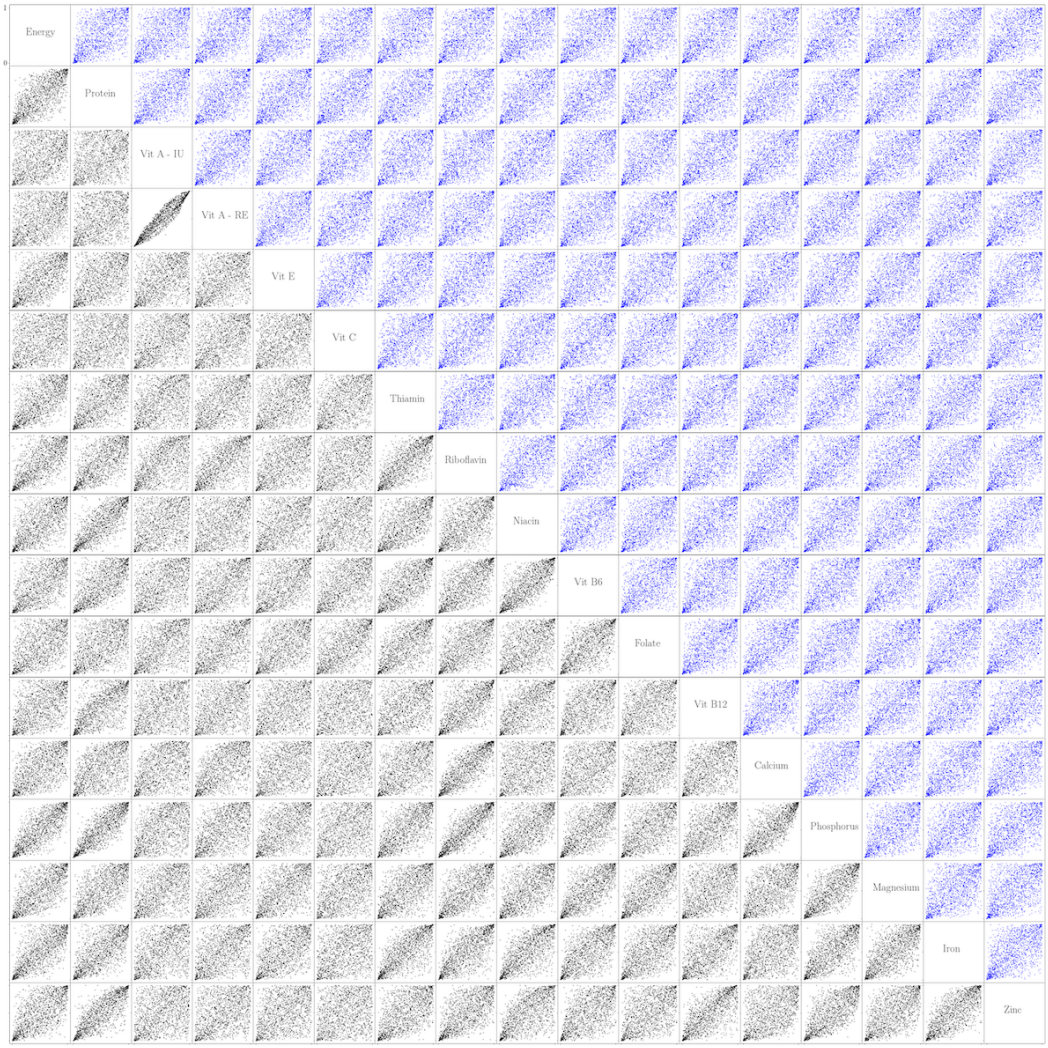

Figure 9: (e) Archimedean copula (AC *) inferred with Algorithm 3.

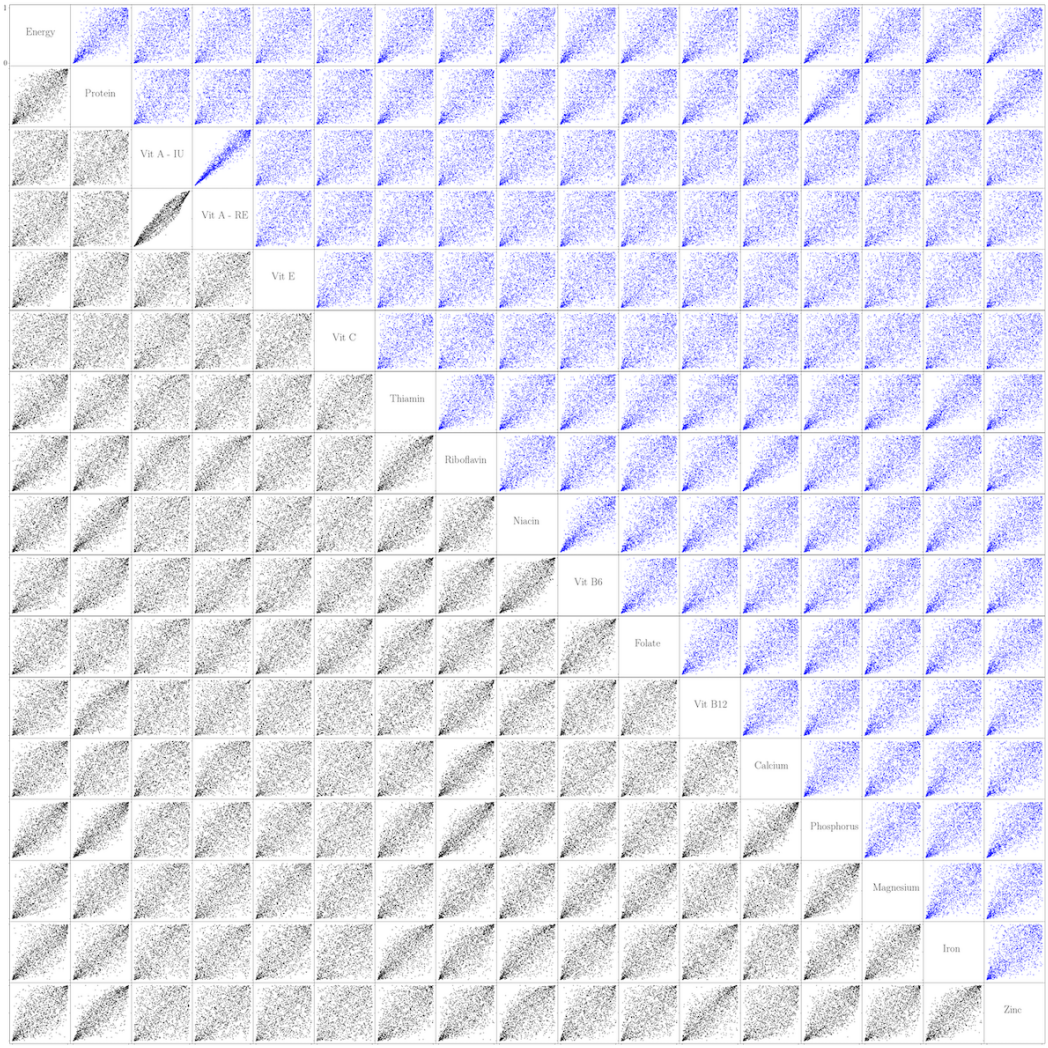

Figure 9: (f) Hierarchical Archimedean copula (HAC) [36].

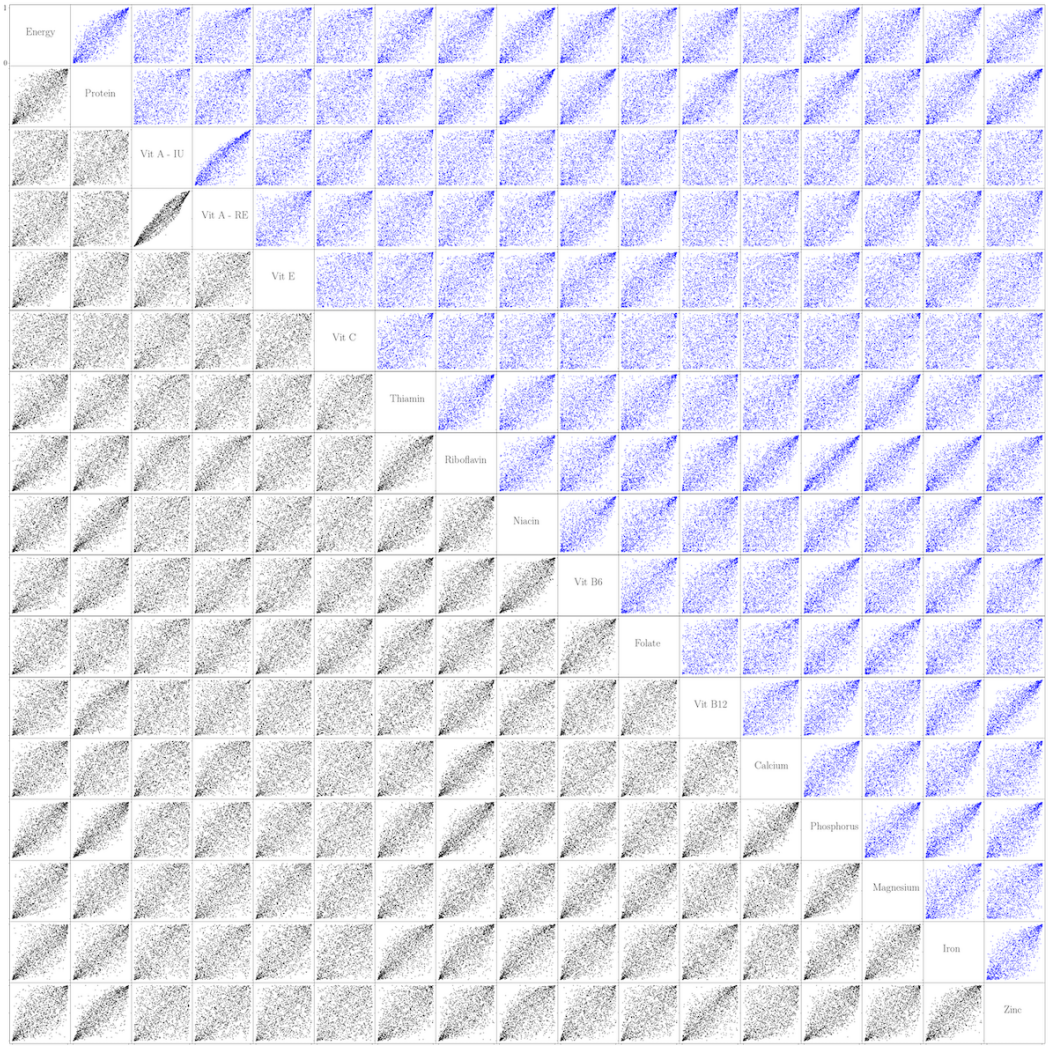

Figure 9: (g) Extreme-value copula (EV †) sampled with Algorithms 2 and 4.

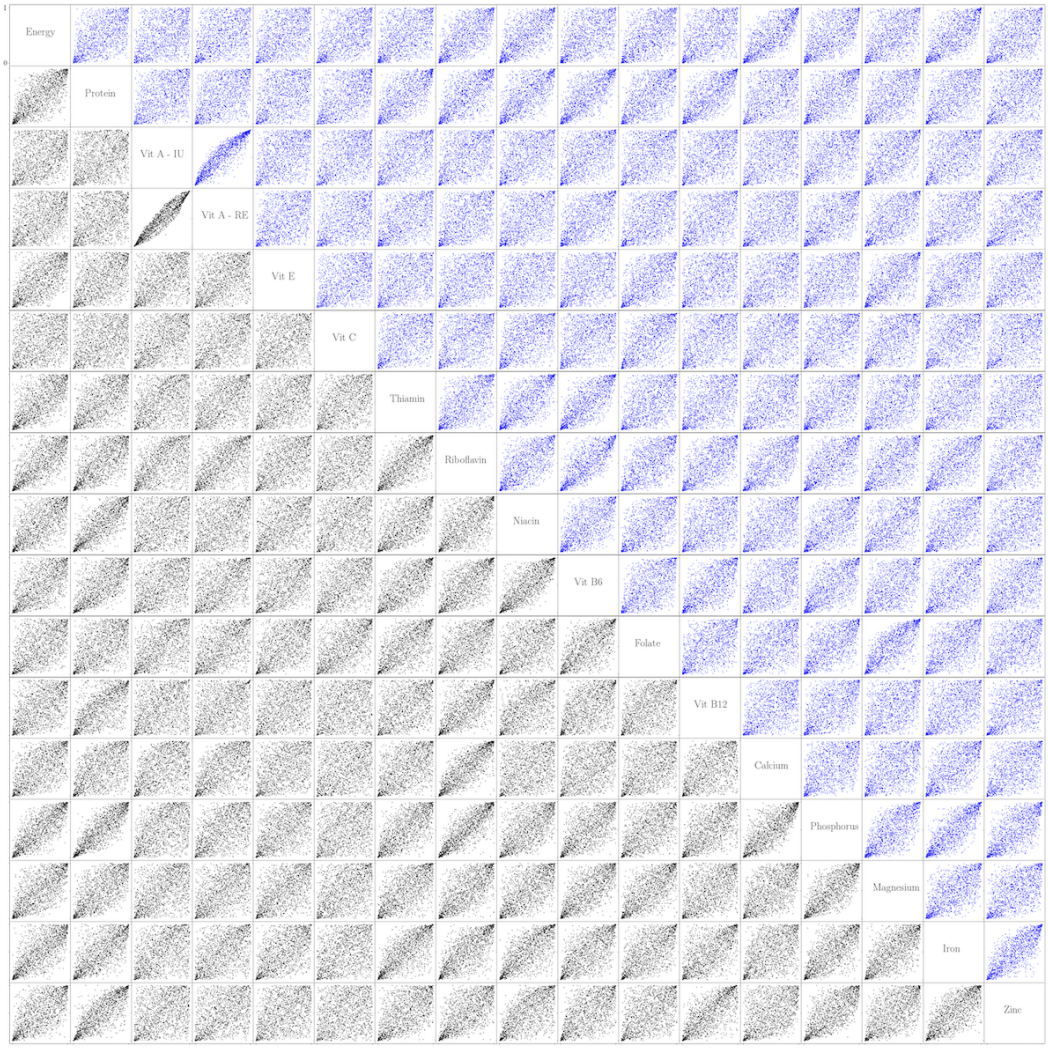

Figure 9: (h) Archimax copula (C-AX †) inferred with the state of the art [16], sampled with Algorithms 2 and 4.

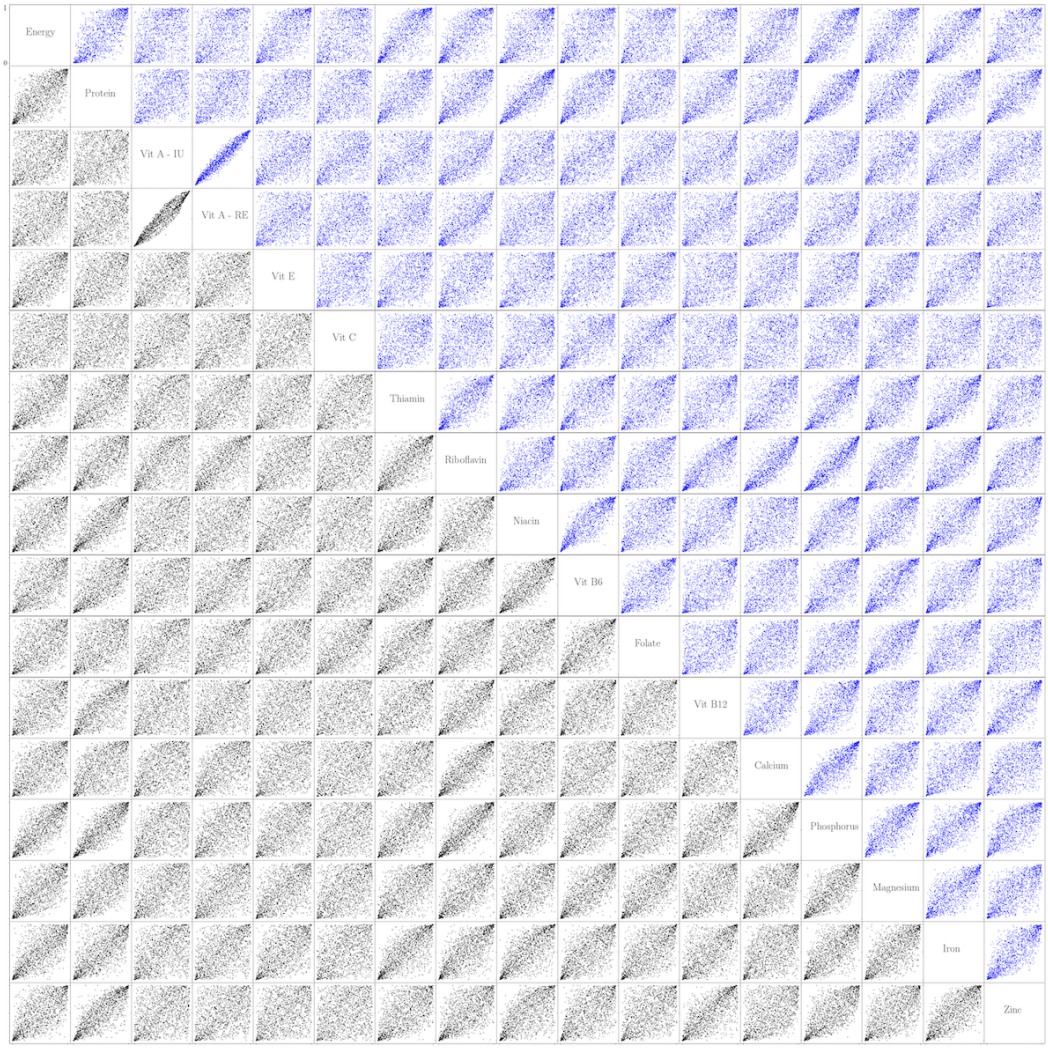

Figure 9: (i) Wasserstein GAN with gradient penalty (WGAN).

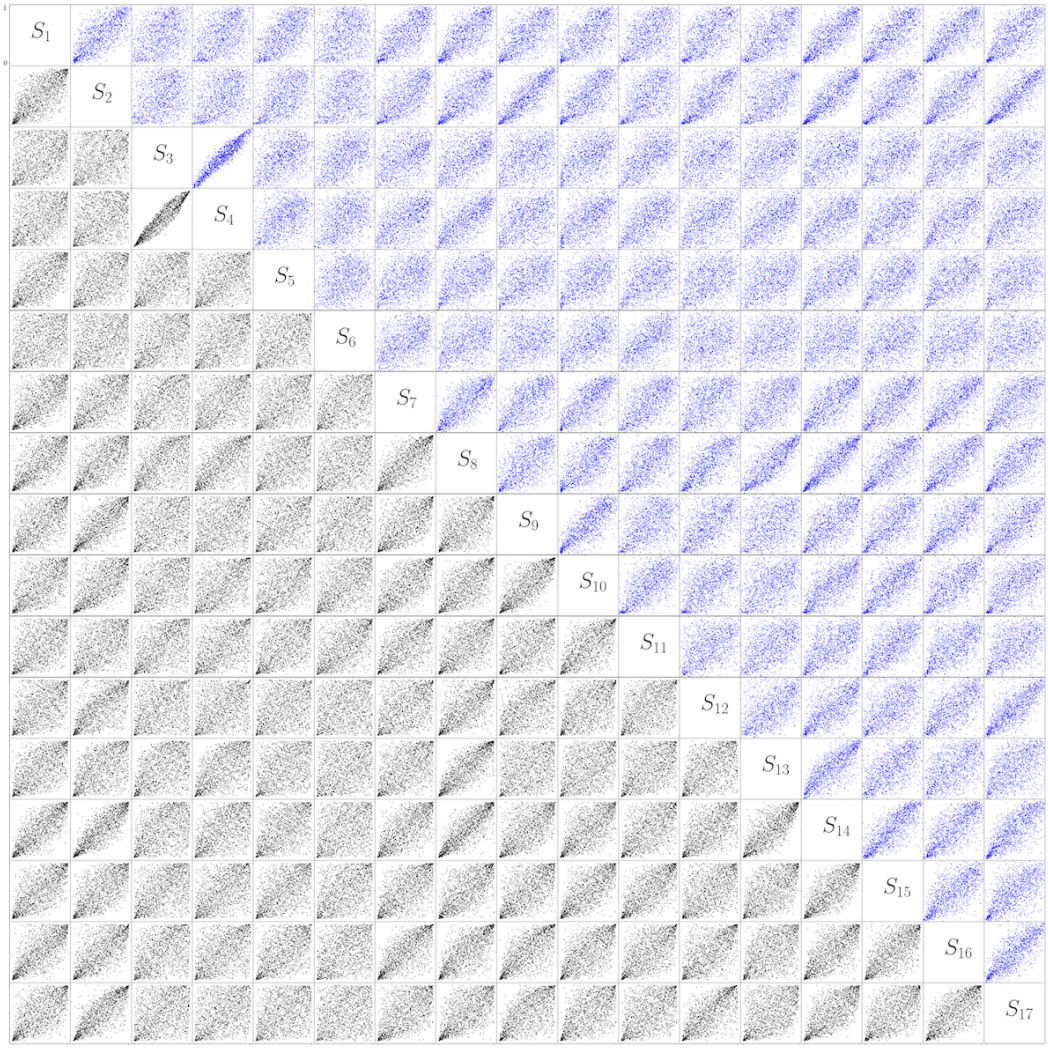

Figure 9: (j) Masked autoregressive flow (MAF).

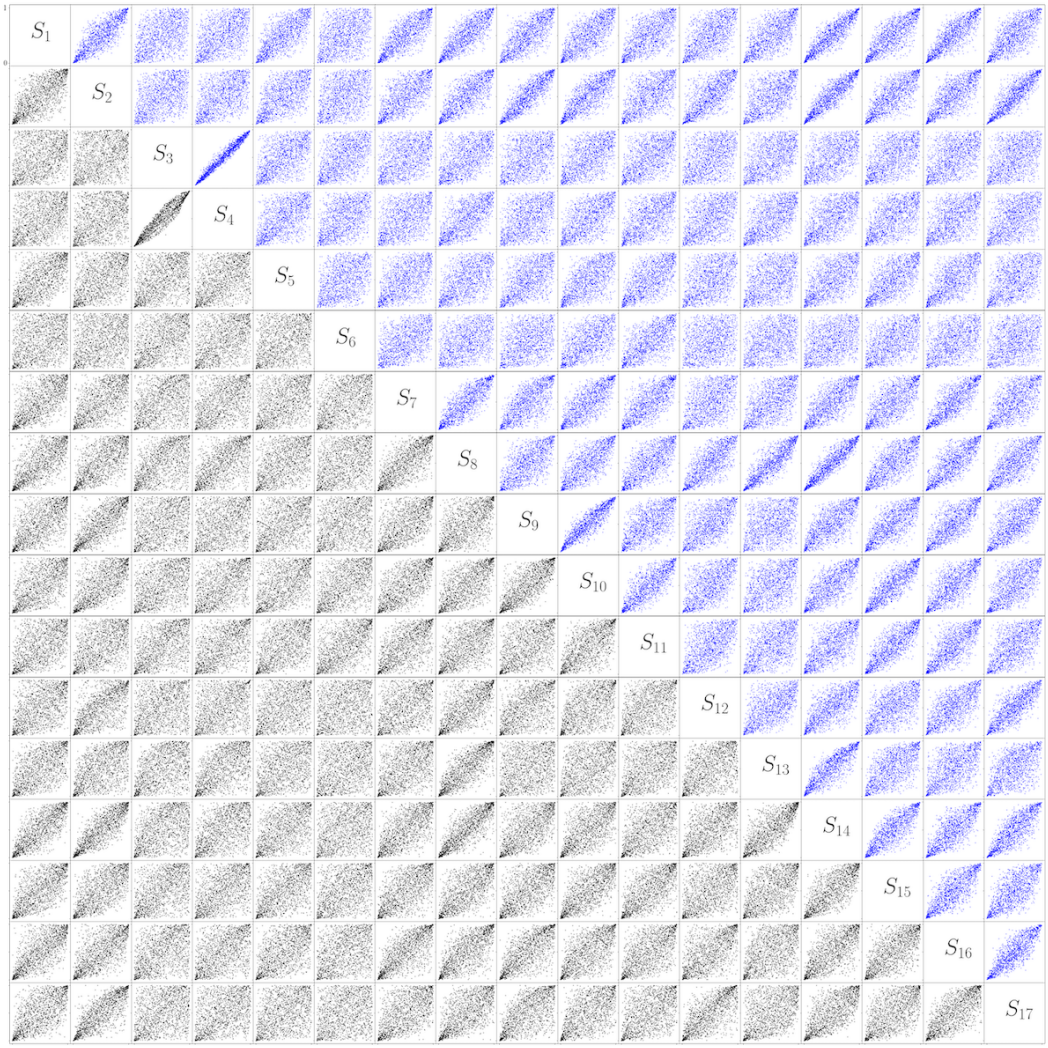

Figure 9: (k) Variational autoencoder (VAE).

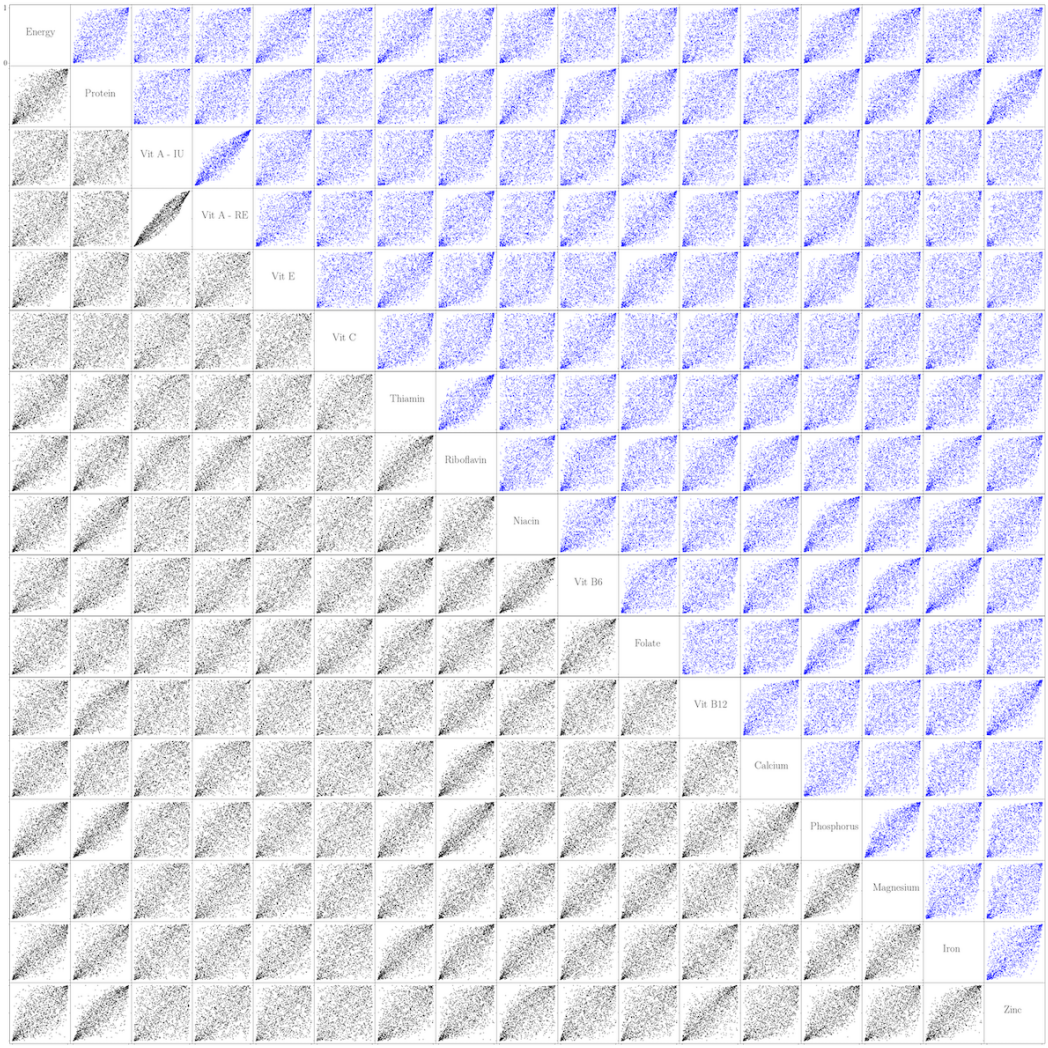

Figure 9: (l) Archimax copula (Gen-AX *†) inferred and sampled with our methods.

B.4 Extrapolating to extreme rainfall

This data simulates the monthly rainfall for 3 locations (Belle-Ile, Groix, and Lorient) in French Brittany, where the dependence is asymmetric and non-extreme. We follow the Archimax model in [16] and extend the experiment with different Archimedean generators of the same Kendall tau τ. The methods used for comparison are Wasserstein GAN with gradient penalty (WGAN), masked autoregressive flow (MAF), variational autoencoder (VAE) and the state-of-the-art Clayton-Archimax copula (C-AX). All methods were first trained on all observations $n = 240$ with dimension $d = 3$. Many samples were then generated from the trained model to estimate $\hat{\ell}$ from block maximas using the state-of-the-art CFG estimator for extreme-value copulas [10]. As mentioned in the main paper, for extrapolating to extremes, we did not compare to other copula based models as many classical copulas are independent or Gumbel in the extremes.

Table 3 of the main paper show our method performs consistently across the different Archimedean generators. In addition, our method always performs the best and in the case of ties, always among the best. Plots of the generated samples and generated samples in the extremes are given in Figure 10, with the Clayton-NSD Archimax copula experiment setting.

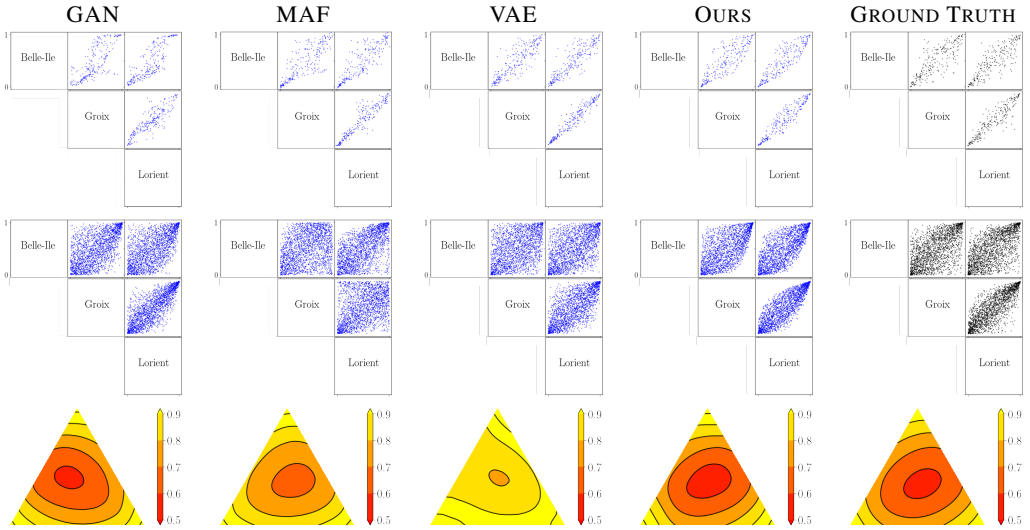

Figure 10: Extrapolating to dependence in the extremes. Plots of generated samples (top), generated samples in the extremes (middle) and $\ell(\mathbf{w}), \mathbf{w} \in \Delta_{3-1}$ (bottom).

B.4.1 Modeling monthly rainfall

Although we did not compare to other copula based models for extrapolating to extremes, we compared to other copula based models for modeling monthly rainfall. In particular, the skew-t copula is a good addition to one's arsenal for flexible asymmetrical copulas [23, 56, 99, 87].

For the Clayton-NSD Archimax copula experiment setting, the CvM distance for Archimax, Gaussian, extreme-value, t and skew-t copulas are: **0.0003** (lower is better), 0.0005, 0.0006, 0.0008, 0.0027.

For this scenario, the Archimax and Gaussian copula performed better than the extreme-value, t and skew-t copulas. This may be because the Clayton-NSD copula does not exhibit extreme-value dependence, i.e. it fails the test of extreme-value dependence [55]. The skew-t copula might have performed not as well as the t copula due to over-parameterization. In addition, maximum likelihood estimation for the skew-t copula was extremely time consuming even for three dimensions and thus intractable for higher dimensions.

B.5 Out-of-distribution detection

The inliers were generated from the Clayton-NSD copula representing the monthly rainfall of French Britanny and the outliers were generated uniformly at random on the unit cube. The number of inlier observations was $n_{in} = 225$, the number of outlier observations was $n_{out} = 25$, corresponding to 10% data contamination. A visual comparison of the results is given in Figure 11.

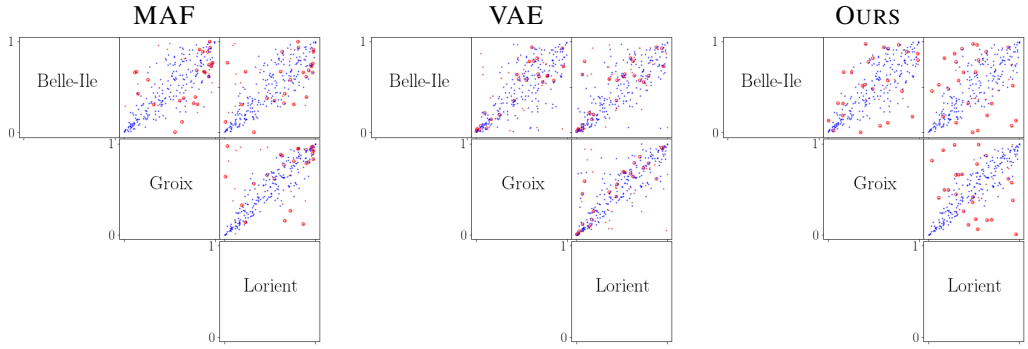

Figure 11: Out-of-distribution detection based on likelihoods. Inliers are represented with blue dots, outliers with red dots, and detected points are circled in red.

The masked autoregressive flow (MAF) provided an explicit likelihood. The likelihood of the variational autoencoder (VAE) was approximated from the reconstruction error. Including the KL divergence to the latent prior made results worse. The likelihood of the Archimax copula was approximated from the inclusion-exclusion scheme, checked to converge using various interval sizes.

B.6 High dimensional modeling

We infer and sample a 100-dimensional Clayton-NSD Archimax copula, with parameters $\theta = 2,\ \alpha = (\alpha_0, \cdots, \alpha_0),\ \alpha_0 = (1, 1, 1, 1, 2, 2, 2, 3, 3, 4), \rho = 0.69$ using our method. The inference results are given in Table 2 of the main paper. We plot the samples in Figure 12, 20 coordinates at a time, with coordinates $(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29)$ in Figure 12 (a) and coordinates $(0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 9, 19, 29, 39, 49, 59, 69, 79, 89, 99)$ in Figure 12 (b). The generated samples are in blue above the diagonal, the true samples are in black below the diagonal.

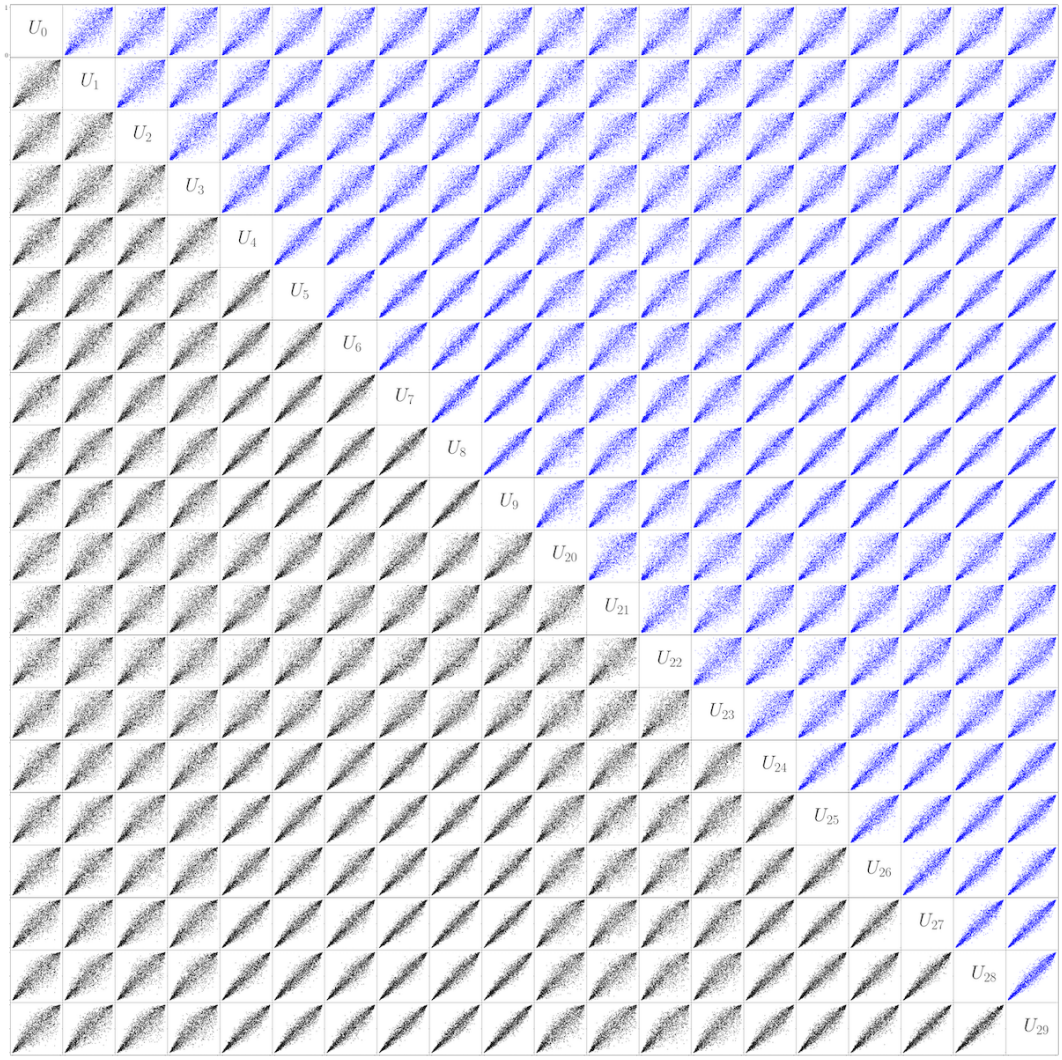

Figure 12: (a) Samples from 100-dimensional Clayton-NSD copula, showing coordinates $(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29)$.

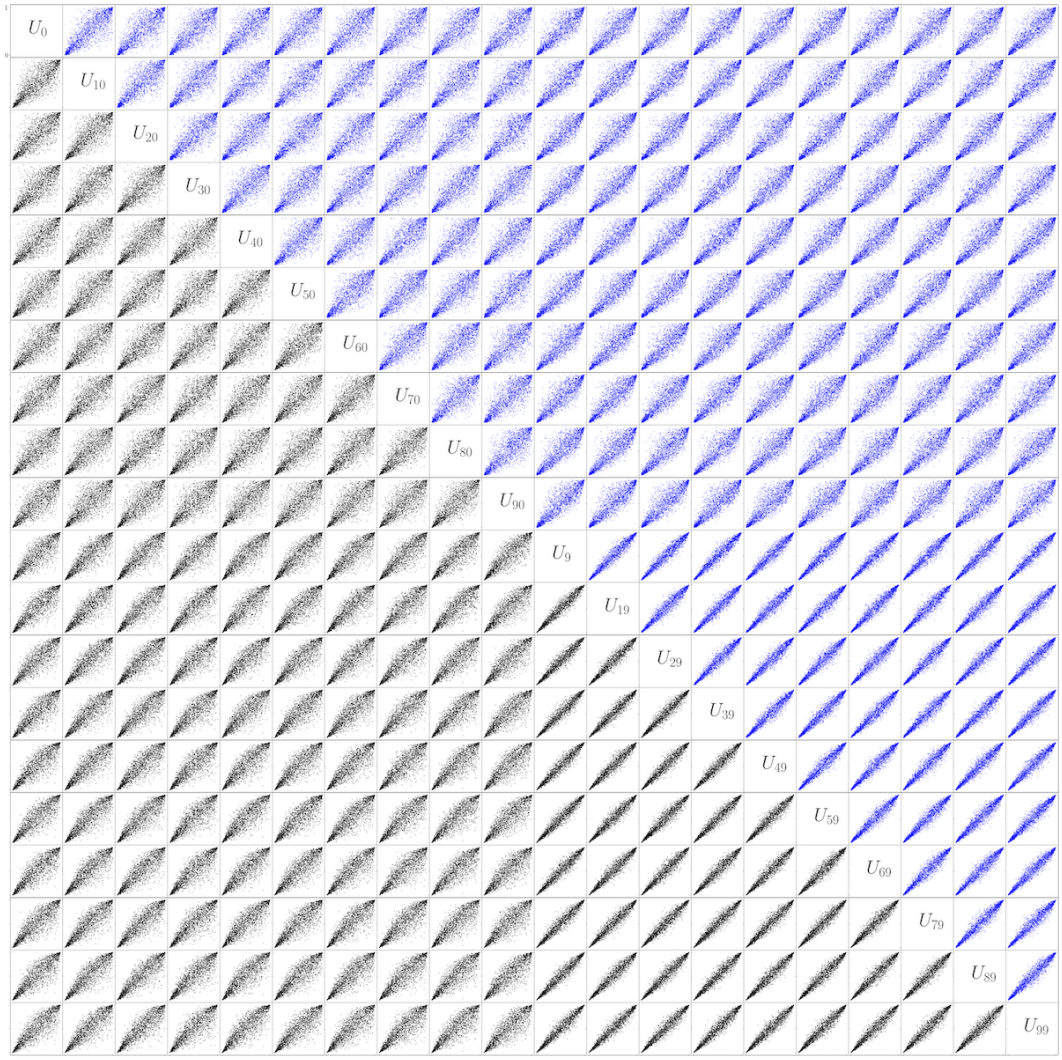

Figure 12: (b) Samples from 100-dimensional Clayton-NSD copula, showing coordinates $(0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 9, 19, 29, 39, 49, 59, 69, 79, 89, 99)$.