
Your copula is a classifier in disguise: classification-based copula density estimation

David Huk

Mark Steel

Ritabrata Dutta

University of Warwick

Abstract

We propose reinterpreting copula density estimation as a discriminative task. Under this novel estimation scheme, we train a classifier to distinguish samples from the joint density from those of the product of independent marginals, recovering the copula density in the process. We derive equivalences between well-known copula classes and classification problems naturally arising in our interpretation. Furthermore, we show our estimator achieves theoretical guarantees akin to maximum likelihood estimation. By identifying a connection with density ratio estimation, we benefit from the rich literature and models available for such problems. Empirically, we demonstrate the applicability of our approach by estimating copulas of real and high-dimensional datasets, outperforming competing copula estimators in density evaluation as well as sampling.

1 INTRODUCTION

Copulas are a popular tool for modelling multivariate densities by decomposing the estimation procedure into two steps: first, learning the marginal density of each variable; and second, stitching those marginals together through a copula model (the word copula comes from Latin and means “a link, tie, bond”). The adoption of copulas is primarily due to their distinct advantages over other models as a consequence of this two-step procedure. Namely, marginal models can be purpose-built for each dimension, which is relevant for applications, *e.g.* in neuroscience (Berkes et al., 2008), climate research (Huk et al., 2023), econo-

metrics (Czado et al., 2022), and domain adaptation (Lopez-Paz et al., 2012). Further, copulas focus all optimisation efforts uniquely on modelling dependence, leading to higher quality densities and samples (Tagasovska et al., 2019), and allows for the estimation of marginal models in parallel or in real time as used in Bayesian density estimation (Huk et al., 2024), Bayesian Networks (Elidan, 2010), and reinforcement learning (Wang et al., 2021).

However, current copula models offer a limited selection of efficient methods depending on the use-case, with Gaussian and vine copulas being the two main candidates. The former, whilst fast, is often too simplistic to correctly capture dependence structures. The latter has an exponentially large model space scaling with the dimension, leading to greedy model searches used in practice, and further requires a simplifying assumption on the dependence structure itself to be computationally feasible. Unequivocally, the aforementioned application domains would benefit from enhanced copula models, able to scale to high dimensions, and simultaneously capable of capturing complex patterns while being computationally achievable.

To this aim, we leverage a novel connection between copula densities and classification to propose a new copula estimation framework. This involves training a classifier to distinguish observed dependent data from an independent version of itself, recovering the true copula density at optimality. After providing a background on relevant topics in Section 2, we elicit the connection between copulas and classification in Section 3 and develop our classification-based copula estimation scheme. We show equivalences between copulas classifications in Section 4. We empirically validate our approach in Section 5, and discuss related works in Section 6 with our conclusion in Section 7.

2 BACKGROUND

Notation. Let $p(\mathbf{x})$ be a multivariate density over $\mathcal{X} \subseteq \mathbb{R}^d$, from which we observe i.i.d. samples $\mathcal{D}_P =$

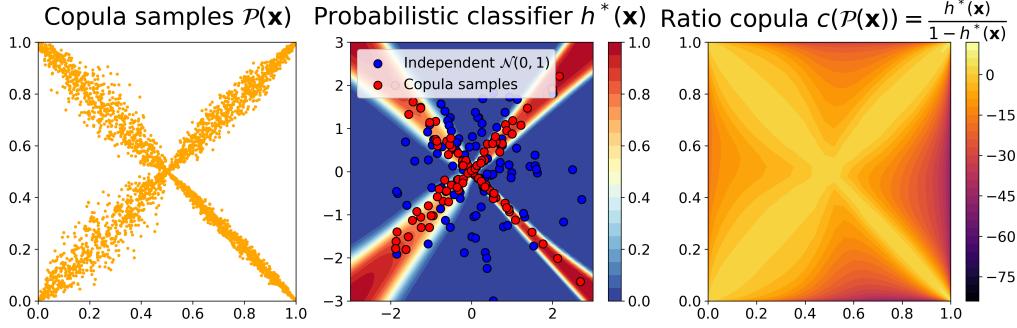


Figure 1: Our ratio copula model estimates copula densities through classification. First panel: dependent copula data. Second panel: we transform the dependent copula data to a Gaussian scale and train a classifier to distinguish between dependent data and an independent Gaussian noise. Third panel: this classifier learns to identify dependence and ultimately recovers the copula density, shown on the log scale above.

$\{\mathbf{x}^k\}_{k=1}^T \sim p(\mathbf{x})$. Similarly, let $p_1(x_1), \dots, p_d(x_d)$ be the marginal densities of $p(\mathbf{x})$, over $\mathcal{X}_i \subseteq \mathbb{R}$ with corresponding i.i.d. samples $\mathcal{D}_{P_i} = \{x_i^k\}_{k=1}^T \sim p_i(x_i)$ for $i = 1, \dots, d$, and assume that all distributions are continuous. Further, let P and P_1, \dots, P_d be the respective cumulative distribution functions (CDFs) of the previously mentioned densities, in order of appearance. We define applications of CDFs such as $\mathcal{P}(\mathbf{x}) \equiv (P_1(x_1), \dots, P_d(x_d))$ as shorthand for the dimension-wise application of marginal CDFs to a vector or $\mathcal{P}(\mathcal{D})$ to mean their application to every vector element of a set. Probability and cardinality of a set A are denoted by $\mathbb{P}(A)$ and $|A|$ correspondingly.

2.1 Copula-Based Density Estimation

Sklar's theorem (Sklar (1959) found in Appendix B) states that any joint probability density function with continuous marginal densities can always be uniquely factorised into the product of its marginals and a copula density $c : [0, 1]^d \mapsto (0, \infty)$ in the following way:

$$\begin{aligned} p(\mathbf{x}) &= p(x_1, \dots, x_d) \\ &= \prod_{i=1}^d p_i(x_i) \cdot c(P_1(x_1), \dots, P_d(x_d)) \\ &= \prod_{i=1}^d p_i(x_i) \cdot c(\mathcal{P}(\mathbf{x})) \end{aligned} \tag{1}$$

Consequently, one can decompose the estimation problem of learning $p(\mathbf{x})$ into first learning all marginals $\{p_i\}_{i=1}^d$, and then inferring an appropriate copula model for the dependence structure of $\mathcal{P}(\mathbf{x}) \in [0, 1]^d$.

For example, a scientist wishes to model daily temperature data over thousands of stations in different parts of a continent. With their domain expertise, they can design an effective model $p_i(x_i | S_i)$ for the temperature x_i of a given station i , taking into account rel-

evant information S_i such as the past days' observed temperature and other variables such as rainfall, cloud cover and humidity in the area. Such a marginal model p_i can be constructed for each station separately as it becomes a unidimensional time-series model with predictors. However, modelling all locations jointly while conditioning on every predictor across time and space might be unmanageable. In such cases, a copula is the appropriate tool to stitch the separate station-specific models together into a mathematically valid joint model over the whole space using (1).

Whereas taking the product of marginal models $\prod p_i(x_i)$ would also give a valid density model, the joint behaviour would be unaccounted for (represented as an independence copula giving value 1 to all possible observations $\mathcal{P}(\mathbf{x})$), effectively assuming independence across all stations. This is an issue since, for instance, the temperature at two nearby stations should be similar - more than that of two stations located further apart. This is exactly the joint behaviour that a copula serves to capture.

Current copula models: There exist many types of copula models, both parametric and non-parametric. Most parametric copulas are only really suited for modelling 2D dependences and greatly suffer from the curse of dimensionality (Hofert et al., 2013). The Gaussian copula (see Appendix B.1) is a popular parametric choice as it can be fitted quickly even to moderate dimensions. However, it lacks the desired flexibility to capture complex patterns or heavy tails; one of the argued reasons for the 2008 financial crash (Salmon, 2012). Archimedian and Archimax parametric copula families (Ng et al., 2022; Górecki and Okhrin, 2024) have similar drawbacks. Among non-parametric copulas, kernel density estimator (KDE) copulas (Geenens et al., 2017) are commonly used.

They transform the observed data via appropriate (often Gaussian) inverse CDFs to a latent space and perform regular KDE on the latent density. However, doing so suffers from drawbacks common to KDE methods in higher dimensions. Deep learning copula models are still a nascent line of research, with a handful of candidate solutions such as Kamthe et al. (2021); Janke et al. (2021); Ashok et al. (2024). The Implicit Generative Copula (IGC) of Janke et al. (2021) is a promising model, which works by generating data in a latent space and applying an approximate integral transformation to recover samples in $[0, 1]$. As an implicit model only able to generate samples, the IGC cannot evaluate densities, thus preventing many applications of copula models. As such, current copula models supporting density estimation and sampling are mostly limited to low or medium-dimensional modelling.

Vine copulas: The state-of-the-art approach is vine copula models, offering a divide-and-conquer solution to expand copula models to higher dimensions. Vine copulas build a hierarchical decomposition of the joint copula density by conditional factorisation into bivariate sub-components along dimensions (see Appendix B.2 and Aas et al. (2009); Czado (2019); Czado and Nagler (2022) for more detail). In practice, it is common to ignore part of the conditional relationships in an approach termed simplified vines (Nagler and Czado, 2016; Nagler et al., 2017). This results in a convergence rate that is independent of the dimension so long as the true data-generating model is a simplified vine, and its correct decomposition has been identified (Nagler and Czado, 2016). However, vines have an exponentially large model space with the dimension, meaning greedy search algorithms are employed in practice. Already for 5-dimensional data, itself simulated from a vine copula, recovering the correct decomposition with current vine models is extremely challenging (Janke et al., 2021), meaning these models are often misspecified in practice.

2.2 Probabilistic Classification for Density Ratio Estimation

The problem of density ratio estimation (DRE) is a crucial inference problem in machine learning and statistics (Sugiyama et al., 2010), most notably framed as a probabilistic classification task between samples from both densities involved in the ratio (Gutmann and Hyvärinen, 2010; Gutmann and Hirayama, 2011; Sugiyama et al., 2012; Menon and Ong, 2016).

Given a dataset $\mathcal{D}_p = \{\mathbf{x}_p^k\}_{k=1}^{T_p} \sim p(\mathbf{x})$ and $\mathcal{D}_q = \{\mathbf{x}_q^k\}_{k=1}^{T_q} \sim q(\mathbf{x})$ with $q(\mathbf{x}) > 0$ whenever $p(\mathbf{x}) > 0$, we are interested in estimating the ratio $r(\mathbf{x}) =$

$p(\mathbf{x})/q(\mathbf{x})$. Fictitious labels y are assigned so that $y = 1$ for \mathcal{D}_p and $y = 0$ for \mathcal{D}_q , with identical prior odds being assumed (although this can be relaxed). Then a probabilistic classifier $h(\mathbf{x}) : \mathcal{X} \rightarrow [0, 1]$ is trained to assign the probability of \mathbf{x} being simulated from $p(\mathbf{x})$. By Bayes' rule, the optimal value of $h(\mathbf{x})$ is $h^*(\mathbf{x}) = \mathbb{P}(y = 1|\mathbf{x})$, known as a *Bayes optimal classifier* (Hastie et al., 2009; Sugiyama et al., 2012; Gutmann and Hyvärinen, 2012). We denote $\frac{h(\mathbf{x})}{1-h(\mathbf{x})}$ as the *decision boundary* since if this quantity is above one, it indicates \mathbf{x} is more likely to belong to class $y = 1$ than $y = 0$. Crucially, the Bayes classifier can be equated to a density ratio via the decision boundary as follows:

$$r(\mathbf{x}) = \frac{p(\mathbf{x})}{q(\mathbf{x})} = \frac{p(\mathbf{x} | y = 1)}{q(\mathbf{x} | y = 0)} = \frac{h^*(\mathbf{x})}{1 - h^*(\mathbf{x})}. \quad (2)$$

Developing methods for such problems is an active area of research, with new models being proposed for increasingly complex tasks (Rhodes et al., 2020; Choi et al., 2021, 2022).

3 PROBABILISTIC CLASSIFICATION FOR COPULA ESTIMATION

In what follows, after establishing the connection between copula estimation and probabilistic classification, we derive a novel *ratio copula* estimator. Throughout, we assume that marginal densities are known and kept fixed, thus for $\mathbf{x} \sim p(\mathbf{x})$, we treat $\mathcal{P}(\mathbf{x})$ as samples from the true copula. This is a common approach in copula estimation (Fermanian, 2005; Genève, 2003; Scaillet, 2007), and ensures the validity of the resulting copula-based density (Ashok et al. (2024) Proposition 2) even in non-parametric settings.

We begin by noticing that, using Sklar's theorem (1), a copula density can be re-written as a ratio between the joint density and the product of the marginals,

$$c(\mathcal{P}(\mathbf{x})) = \frac{p(x_1, \dots, x_d)}{\prod_i p_i(x_i)}. \quad (3)$$

Then, following the identity of Equation 2, we can express our copula density $c(\mathcal{P}(\mathbf{x})) : [0, 1]^d \rightarrow \mathbb{R}^+$ as,

$$c(\mathcal{P}(\mathbf{x})) = \frac{h^*(\mathbf{x})}{1 - h^*(\mathbf{x})}, \quad (4)$$

where $h^*(x)$ is the optimal Bayes classifier for $\mathcal{D}_p \sim p(\mathbf{x})$ and $\mathcal{D}_{\prod_i p_i} \sim \prod_{i=1}^d p_i(x_i)$, assuming unitary prior odds. Under this interpretation, a copula function acts like a probabilistic classifier on \mathcal{X} , identifying regions of \mathcal{X} where samples are more likely to belong to the joint distribution than the product of marginals. In other words, copulas are classifiers trained to identify dependence.

3.1 Ratio Copula Estimator

We propose to train a *ratio copula* estimator c^{ratio} for the true copula density function c using probabilistic classification. However, estimating the Bayes optimal classifier on \mathcal{X} might not be straightforward depending on the constraints of the space, and estimating it on $[0, 1]^d$ directly is even more cumbersome due to the constrained support. For this reason, we use the probit transformation $\Phi^{-1} : [0, 1] \mapsto \mathbb{R}$ applied dimension-wise, commonly used for copula models (Geenens et al., 2017)):

$$\Phi^{-1} \circ \mathcal{P}(\mathbf{x}) : \mathcal{X} \rightarrow \mathbb{R}^d. \quad (5)$$

By the probability integral transformation¹, if $\mathbf{x} \sim p(\mathbf{x})$, then $\Phi^{-1} \circ \mathcal{P}(\mathbf{x})$ marginally follows a standard Gaussian. Therefore, the product of their marginal densities is $\tilde{q}(\cdot) = \prod_{i=1}^d \mathcal{N}(\cdot; 0, 1)$. Nevertheless, their joint density will be some other unknown function $\tilde{p}(\Phi^{-1} \circ \mathcal{P}(\mathbf{x}))$. By Sklar's theorem, \tilde{p} admits a copula decomposition with a corresponding copula \tilde{c} . Since the marginals of \tilde{p} follow a standard Gaussian, its marginal CDFs are exactly Φ . Consequently, the copula expression simplifies, having the marginal CDFs Φ of \tilde{p} cancelling with the Φ^{-1} part of our transformation in (5):

$$\tilde{c}(\Phi(\Phi^{-1} \circ \mathcal{P}(\mathbf{x}))) = \tilde{c}(\mathcal{P}(\mathbf{x})). \quad (6)$$

By the uniqueness of copula densities, as \tilde{c} and c describe the same random variables, we recover our original copula $c(\mathcal{P}(\mathbf{x}))$ with an expression in terms of the transformed densities:

$$c(\mathcal{P}(\mathbf{x})) = \frac{\tilde{p}(\Phi^{-1} \circ \mathcal{P}(\mathbf{x}))}{\tilde{q}(\Phi^{-1} \circ \mathcal{P}(\mathbf{x}))} = \frac{\tilde{p}(\Phi^{-1} \circ \mathcal{P}(\mathbf{x}))}{\prod_{i=1}^d \mathcal{N}(\Phi^{-1} \circ P_i(x_i); 0, 1)}. \quad (7)$$

Thus, we can now focus on the ratio of the RHS in (7). Namely, our aim becomes training a classifier to distinguish dependent data $\mathcal{D}_{\tilde{p}} = \Phi^{-1} \circ \mathcal{P}(\mathcal{D}_p)$ from independent data $\mathcal{D}_{\tilde{q}} = \Phi^{-1} \circ \mathcal{P}(\mathcal{D}_{\prod_i p_i})$, with respective labels $y = 1$ and $y = 0$, all conveniently on \mathbb{R}^d .

Following the literature on probabilistic classification, we assume c belongs to a parametric class of functions $\mathcal{C} \equiv \{c^{\text{ratio}}(\cdot; \boldsymbol{\theta}) | c^{\text{ratio}}(\cdot; \boldsymbol{\theta}) : \mathbb{R}^d \rightarrow \mathbb{R}^+, \boldsymbol{\theta} \in \Theta\}$, which can consist of simple functions (*e.g.* linear) to more complex functions (*e.g.* non-linear functions parametrized by suitable neural network architectures). Then, we estimate the parameters of

$c^{\text{ratio}}(\Phi^{-1} \circ \mathcal{P}(\mathbf{x}); \boldsymbol{\theta})$ by minimizing the logistic loss:

$$\begin{aligned} \hat{\boldsymbol{\theta}} &= \arg \min_{\boldsymbol{\theta} \in \Theta} \mathcal{L}(\boldsymbol{\theta}) \\ &= \arg \min_{\boldsymbol{\theta} \in \Theta} -\mathbb{E}_{\tilde{p}} \ln \left(\frac{c^{\text{ratio}}(\Phi^{-1} \circ \mathcal{P}(\mathbf{x}); \boldsymbol{\theta})}{\nu + c^{\text{ratio}}(\Phi^{-1} \circ \mathcal{P}(\mathbf{x}); \boldsymbol{\theta})} \right) \\ &\quad - \nu \cdot \mathbb{E}_{\tilde{q}} \ln \left(\frac{\nu}{\nu + c^{\text{ratio}}(\Phi^{-1} \circ \mathcal{P}(\mathbf{x}); \boldsymbol{\theta})} \right) \end{aligned} \quad (8)$$

where $\nu = \mathbb{P}(y = 0)/\mathbb{P}(y = 1)$ to compensate for unequal prior odds. The ratio copula $c^{\text{ratio}}(\cdot; \boldsymbol{\theta})$ implicitly defines a classifier between $\mathcal{D}_{\tilde{p}}$ and $\mathcal{D}_{\tilde{q}}$. As minimising $\mathcal{L}(\boldsymbol{\theta})$ is equivalent to minimising a Bregman divergence between \mathcal{C} and $c(\cdot)$ (Menon and Ong, 2016), $\exists \boldsymbol{\theta}^*$ such that $c^{\text{ratio}}(\cdot; \boldsymbol{\theta}^*)$ is the projection of $c(\cdot)$ on our model class \mathcal{C} . If our model class is well specified, then $c^{\text{ratio}}(\cdot; \boldsymbol{\theta}^*) = c(\cdot)$. In such a case, our estimator will asymptotically recover $c^{\text{ratio}}(\cdot; \boldsymbol{\theta}^*)$, without needing any normalising constant (Gutmann and Hyvärinen, 2012). However, as this only holds at optimality, we simultaneously estimate a normalising constant as an additional parameter Z included in $\boldsymbol{\theta}$, which pre-multiplies the ratio copula. The estimation of Z is done jointly with $\boldsymbol{\theta}$ by minimising the same loss, ensuring that the resulting ratio copula is a valid density.

Our ratio copula estimator inherits the theoretical guarantees associated with the loss in (12) from Gutmann and Hyvärinen (2012), which we concretise here with a justification of its applicability given in Appendix A.1.

Theorem 3.1. Estimator properties; Gutmann and Hyvärinen (2012) Assuming continuity of all marginal densities, $c^{\text{ratio}}(\Phi^{-1} \circ \mathcal{P}(\mathbf{x}); \boldsymbol{\theta}^*)$ obtained by minimising Equation (12) is unique. Further, under certain regularity conditions detailed in Appendix A.1, the copula estimator is consistent with $\hat{\boldsymbol{\theta}} \xrightarrow{\mathbb{P}} \boldsymbol{\theta}^*$, and $\sqrt{|\mathcal{D}_p|}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*)$ is asymptotically normal.

Following the analysis from Gutmann and Hyvärinen (2012) (Corollaries 4-7) of the loss function (12), our copula estimator's accuracy is influenced by two main parts: the factor of independent to dependent observations ν and the form of \tilde{q} . Specifically, we want as many samples from \tilde{q} as possible in order to inflate ν , and we want \tilde{q} to be similar to \tilde{p} . However, in our case, only dependent observations $\mathcal{D}_{\tilde{p}}$ are initially available and the form of \tilde{q} is necessarily fixed by virtue of estimating a copula.

3.2 Learning ratio copulas in practice

In the following, we demonstrate how to leverage our construction of (7) and our formulation in terms of density ratio estimation to satisfy our desiderata on the sampling and density of \tilde{q} .

¹If $x_i \sim p_i$ is continuous, then $P_i(x_i) \sim \text{Unif}[0, 1]$.

Efficient sampling of \tilde{q} : During training, all observations are labelled as dependent with $y = 1$, whereas we have no independent samples from \tilde{q} . To obtain such samples, one could use the heuristic of recreating independent samples by dimension-wise shuffling the values of $\mathcal{D}_{\tilde{p}}$, but this would lead to a misrepresentation of the statistical properties. Instead, we can take advantage of our transformation to \mathbb{R}^d in (5) and exploit the fact that $\tilde{q}(\Phi^{-1} \circ \mathcal{P}(\mathbf{x})) = \mathcal{N}(\Phi^{-1} \circ \mathcal{P}(\mathbf{x}); \mathbf{0}, \mathbf{I}_d)$, which we can sample from *ad nauseam*. This satisfies our first desideratum, namely, access to a large amount of samples from \tilde{q} .

Narrowing the KL gap in a copula: For the second desideratum, we want the density of the independent data to be close to that of the dependent data. We can rewrite the density ratio in (7) by introducing an intermediate density:

$$\begin{aligned} c(\mathcal{P}(\mathbf{x})) &= \frac{\tilde{p}(\Phi^{-1} \circ \mathcal{P}(\mathbf{x}))}{\mathcal{N}_d(\Phi^{-1} \circ \mathcal{P}(\mathbf{x}); \mathbf{0}, \mathbf{I}_d)} \quad (9) \\ &= \underbrace{\frac{\tilde{p}(\Phi^{-1} \circ \mathcal{P}(\mathbf{x}))}{\mathcal{N}_d(\Phi^{-1} \circ \mathcal{P}(\mathbf{x}) | \mathbf{0}, \Sigma)}}_{\text{ratio copula model}} \cdot \underbrace{\frac{\mathcal{N}_d(\Phi^{-1} \circ \mathcal{P}(\mathbf{x}) | \mathbf{0}, \Sigma)}{\mathcal{N}_d(\Phi^{-1} \circ \mathcal{P}(\mathbf{x}) | \mathbf{0}, \mathbf{I}_d)}}_{\text{known Gaussian ratio}} \end{aligned}$$

which is a centred correlated Gaussian with covariance matrix Σ , a good choice being the sample covariance of $\Phi^{-1} \circ \mathcal{P}(\mathcal{D}_p)$ to mimic the inherent dependence in the density \tilde{p} . This intermediate Gaussian guides the classifier closer to the data density to improve the accuracy of the final ratio copula. As the analytical expression of the second ratio between the Gaussians is known, we instead, we focus our estimation efforts on the ratio $\frac{\tilde{p}(\Phi^{-1} \circ \mathcal{P}(\mathbf{x}))}{\mathcal{N}_d(\Phi^{-1} \circ \mathcal{P}(\mathbf{x}) | \mathbf{0}, \Sigma)}$ by classifying between the dependent data $\mathcal{D}_{\tilde{p}}$ and $\mathbf{z} \sim \mathcal{N}_d(\mathbf{0}, \Sigma)$. Once the ratio copula model is trained on this task, we recover the correct copula by multiplying the trained model with the ratio of Gaussians. We call the resulting copula estimator a *Gaussian-Guided (GG) ratio copula*.

We give an algorithmic description of the proposed approach in Algorithm 1 in Appendix C where orange parts are optional for producing a Gaussian-Guided ratio copula.

Furthermore, it is of interest to measure how difficult a classification task is for a ratio copula model. In particular, knowing when estimating a Gaussian copula might be too hard a task for a non-parametric (or over-parameterised) estimator, would indicate that we should use at least a Gaussian-Guided ratio copula model, to match the gap between densities.

Bound on Kullback-Leibler (KL) divergence for Gaussian copulas. As a Gaussian copula is the ratio between an independent Gaussian and a Gaussian

with a nontrivial covariance matrix Σ , the difficulty of the classification-based estimation revolves around how different these two densities are. Fortunately, we can recover a simple analytical expression for the KL divergence between the two Gaussians. This is useful to show that as long as the ellipsoid formed by the Gaussian \tilde{p} is not too narrow (ie. the eigenvalues of its covariance are lower bounded), then the KL is upper bounded by a constant depending on d .

Lemma 3.2 (Bound on KL of a Gaussian copula classifier). Consider a d -dimensional Gaussian copula with covariance matrix Σ . Then, the KL divergence between the copula numerator and denominator is upper-bounded with

$$KL(\mathcal{N}_d(\cdot; \mathbf{0}, \Sigma) || \mathcal{N}_d(\cdot; \mathbf{0}, \mathbf{I}_d)) \leq -\frac{d}{2} \log(e_{\min}) \quad (10)$$

where e_1, \dots, e_d are the eigenvalues of Σ with e_{\min} the (possibly joint) smallest eigenvalue of Σ .

We prove this statement in Appendix A.2. The result implies that under restrictions on the smallest width of the ellipsoid of \tilde{p} , the KL cannot be infinite for a fixed dimension d , and in fact scales linearly. As expected, if Σ is close to nonsingular, the KL divergence can still be very large indeed; it is linear in d but Σ is likely to become more ill-conditioned (closer to nonsingularity) as d grows. Consequently, we can rely on the empirical findings of Rhodes et al. (2020), who show classical DRE methods fail for KL differences of over 20 natural logarithms between the two densities. This result provides an informed way of deciding which DRE method to employ for a Gaussian copula based on the sample covariance of $\mathcal{D}_{\tilde{p}}$; whether to use a simple ratio copula model, empower it with a Gaussian-Guided ratio copula, or employ one of the many advanced DRE frameworks (Rhodes et al., 2020; Choi et al., 2021, 2022; Srivastava et al., 2023).

4 EQUIVALENCES BETWEEN CLASSIFIERS AND COPULAS

In this Section, we reason about copulas in terms of the classification rules they implicitly define in light of Equation 4, providing new insights into their modelling capabilities. Throughout, we work on the Gaussian space with $\Phi^{-1} \circ \mathcal{P}(\mathbf{x}) \in \mathbb{R}^d$ for any $\mathbf{x} \in \mathcal{X}$.

Further, in Figure 4 of Appendix A.7, we make use of the following identity

$$c(\mathbf{P}(\mathbf{x})) = \frac{h(\mathbf{x})}{1 - h(\mathbf{x})} \iff h(\mathbf{x}) = \frac{c(\mathbf{P}(\mathbf{x}))}{1 + c(\mathbf{P}(\mathbf{x}))}$$

to illustrate our theory by showing the implied classifiers $h(\mathbf{x})$ for six bivariate parametric copulas.

4.1 Gaussian Copulas and QDA

In the case of a Gaussian copula, the density in the numerator is a correlated Gaussian and the denominator is an uncorrelated Gaussian, both centred at $\mathbf{0}$. From Equation 4, it then follows that the classification problem occurs between a multivariate Gaussian \tilde{q} with identity covariance and a fully dependent Gaussian \tilde{p} ; our aim becomes the classification of two Gaussians with different covariance structures. This is exactly the goal of Quadratic Discriminant Analysis (QDA) (Hastie et al., 2009), which fits a function $h^k(\mathbf{z}; \mu_1, \mu_0, \Sigma_1, \Sigma_0) : \mathbb{R}^d \mapsto [0, 1]$ to classify between $\mathcal{N}(\cdot; \mu_1, \Sigma_1)$ with labels $y = 1$ and $\mathcal{N}(\cdot; \mu_0, \Sigma_0)$ with labels $y = 0$.

Proposition 4.1 (Gaussian copulas and QDA). *Consider a Gaussian copula parameterised by a covariance matrix Σ . The Bayes optimal classifier associated with this copula is found by Quadratic Discriminant Analysis between a standard Gaussian and a centred Gaussian with covariance matrix Σ .*

We provide a proof in Appendix A.3. This result opens up a new way of estimating Gaussian copulas by fitting a QDA classifier to $\Phi^{-1} \circ \mathcal{P}(\mathbf{x})$. Further, this result provides a new perspective on Gaussian copulas by linking them to the quadratic decision boundaries of QDA, characterising the complexity of Gaussian copulas. We show an equivalent result for Student's t copulas in Appendix A.4.

4.2 In the case of a KDE copula

Next, we look at KDE copulas regarding their implicit classification problems to establish their relative complexity. In particular, for a Gaussian KDE copula, we can obtain the following:

Proposition 4.2 (Gaussian KDE copulas and averages of QDA). *Consider a KDE copula with a Gaussian kernel having identity variance scaled by b , based on T_p kernel means at observations $\{(Z_1^k, \dots, Z_d^k)\}_{k=1}^{T_p}$ for $Z_i^k = \Phi^{-1} \circ P_i(x_i^k)$, of the form:*

$$\hat{c}_{T_p}(\Phi^{-1} \circ \mathcal{P}(\mathbf{x})) = \frac{\sum_{k=1}^{T_p} \prod_{i=1}^d \{\mathcal{N}(\Phi^{-1} \circ P_i(x_i) - Z_i^k | 0, b)\}}{T_p \cdot \mathcal{N}_d(\Phi^{-1} \circ \mathcal{P}(\mathbf{x}) | \mathbf{0}, \mathbf{I}_d)}. \quad (11)$$

Then, the associated Bayes optimal classifier is found by the average of T_p decision boundaries from QDA classifiers $h^k(\mathbf{z}; \mu_1^k, \mu_0^k, \Sigma_1, \Sigma_0)$, with component-specific means μ_1^k, μ_0^k but shared variances Σ_1, Σ_0 across components.

The proof is included in Appendix A.5. This can be extended to other types of kernels for similar results. Proposition 4.2 describes a KDE copula as an average of QDA decision boundaries over the T_p observed sam-

ples. This results in a more complex decision boundary with more nonlinearity introduced linearly in the number of samples. However, due to the curse of dimensionality, an exponential number of samples will be required to maintain a constant level of complexity, thus elucidating KDE copulas' difficulty in scaling to high dimensions.

4.3 For a simplified vine copula

Vine copulas are decompositions of copula densities using only bivariate copulas as building blocks and in simplified vine copulas, the bivariate copulas are assumed to have no conditioning (see Appendix B.2 and Czado et al. (2019) for an introduction). Here, we consider two cases, namely, using Gaussian copulas or Gaussian KDE copulas as those building blocks within simplified vine copula models. We give the statement here with a proof found in Appendix A.6.

Corollary 4.3. *The density of a simplified vine copula constructed using bivariate Gaussian copulas as building blocks coincides with the product of $\frac{d(d-1)}{2}$ QDA decision boundaries between two bivariate centred Gaussians with different covariances.*

Moreover, the density of a simplified vine copula constructed using Gaussian KDE copulas as building blocks coincides with the product of $\frac{d(d-1)}{2}$ different averages of T_p QDA decision boundaries between two bivariate Gaussians with unequal means and variances.

This result is instructive in describing the complexity of simplified vines through their use of multiple decision boundaries. They function much like hierarchical classifiers (examples of which are classification and regression trees, Loh (2011), or random forests, Breiman (2001)) by breaking the modelling task into smaller subparts on subgroups of variables. One can also draw a parallel with multilayered perceptrons that have parts of the model, such as their different layers, specialised in capturing specific features of the data relayed to them. Vine copulas instead have many separate models which capture different features, where components further up the decomposition have inputs with more conditioning - encoding the information extracted by previous classifiers. This can then be seen as a mixture of experts model, a common framework in machine learning (Yuksel et al., 2012; Jiang et al., 2024).

5 EXPERIMENTS

Section 4 described the limitations of popular copulas. To achieve better performance, designing more flexible copula models is essential. In this Section, we experimentally verify the validity of our proposed estimator.

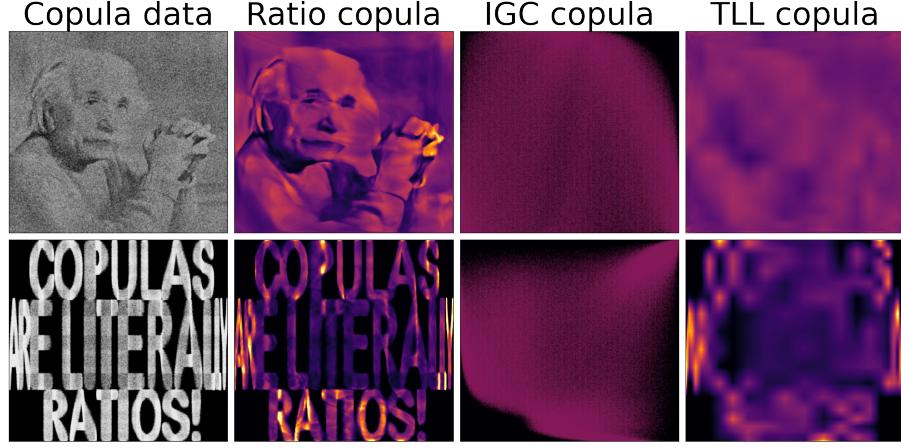


Figure 2: Bivariate nonparametric copula models trained on the dependence in the light of pixels from 2D images. From left to right, columns show on $[0, 1]^2$: the observed data, the LL of our ratio copula, samples from the IGC, and the LL of the TLL. Only the ratio copula manages to correctly capture the sharp transitions in the LLs.

Implementation details are discussed in Appendix C with additional experiments on alternate losses and samples produced by our ratio copula models. Our code is available at <https://github.com/Huk-David/Ratio-Copula>. We investigate the following:

- Are ratio copulas competitive models in two dimensions, encouraging their use as non-parametric models in vine copula decompositions?
- Are ratio copulas effective in high dimensions and on typically challenging data for copula models?

Baselines: We compare our ratio copula to commonly used models for copula estimation. Namely, the Gaussian copula, the TLL² constant, linear or quadratic copulas in the pyvinecopublib package of Vatter and Nagler (2022) as well as a simplified vine copula using TLL constant bivariate copulas. Finally, we compare with the implicit generative copula (IGC) of Janke et al. (2021), as a comparison with an effective deep copula, which is only able to sample.

Metrics: As a metric for success we choose the *average* copula log-likelihood (denoted LL), a well-used metric on its own, but in the context of copula models, once marginal densities have been estimated, only the copula LL is relevant to the LL of the full model. We also compare samples from each copula model using an empirical version of the Wasserstein-2 metric, as done in other probabilistic modelling contributions (*e.g.* Tagasovska et al. (2019)).

²Transformation local likelihood (TLL), a class of non-parametric copula estimators using (5) with polynomials.

Sampling: Our ratio copula offers an analytical density. Hence we use Hamiltonian Monte Carlo (Neal, 2011), a gradient-informed MCMC sampler, to output samples from the estimated model.

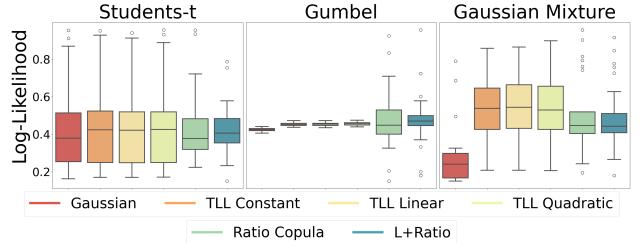


Figure 3: Box plots showing the average LL across 25 fits of bivariate copula models based on samples from the Student’s t, Gumbel and Gaussian Mixture parametric copulas.

Estimating 2D copulas: Modelling bivariate copulas is particularly valuable, as such models construct vine copula decompositions.

In Figure 3 we fit parametric copulas based on 5000 samples, namely the Student-t, the Gumbell and a mixture of two Gaussian copulas with random covariances. We then test the model LL by evaluating 5000 held-out test samples, repeating this experiment 25 times with different seeds. We show the LL values for the Gaussian, TLL copulas, as well as two versions of our ratio copula: Ratio Copula (a multilayered perceptron) and L+Ratio (a polynomial logistic classifier of order 5 with all interaction terms). The Gaussian copula performs the worst overall. The TLL and our models are similar for the Student-t and on the Gumbel but

Model	Digits		MNIST	
	LL (Mean \pm Std)	W2 (Mean \pm Std)	LL (Mean \pm Std)	W2 (Mean \pm Std)
GG+ratio	12.97 \pm 0.67	8.03 \pm 0.05	198.39 \pm 19.72	36.44 \pm 0.51
GG+Ratio <i>large</i>	13.69 \pm 0.77	7.99 \pm 0.07	334.42 \pm 45.91	37.40 \pm 0.79
Vine	12.79 \pm 0.07	8.50 \pm 0.02	183.00 \pm 0.39	37.57 \pm 0.18
Gaussian	11.78 \pm 0.13	8.34 \pm 0.01	126.41 \pm 0.09	37.00 \pm 0.15
IGC	-	8.14 \pm 0.03	-	36.02 \pm 0.06

Table 1: Average LL and W2 scores for the Digits and MNIST datasets with one standard deviation margins over 10 runs, the best scores are in bold. Our method (GG+Ratio) performs best and improves further with a larger network architecture for the classifier (GG+Ratio *large*), showcasing its adaptability.

perform slightly worse on the Gaussian mixture. This is a non-trivial achievement since current methods have been extensively optimized for low-dimensional data. Our estimator’s variance is higher than the TLL’s variance for the Gumbel, likely due to it being an Archimedean copula. Notably, the L+Ratio copula marginally outperforms the Ratio Copula version, which we credit to its simple structure acting as an implicit regularisation. Both of our models perform consistently well, making them suitable for vine copulas.

In Figure 2, we show estimates of toy examples of bivariate copulas generated from the dependence found in monochromatic images when data is sampled according to the light of each pixel (more descriptions of this dataset can be found in C.2). The training size is one million, meant to emulate the big data setting. We show observed data in the first column, with the following columns showing the ratio copula density, samples from the IGC, and the TLL quad density, in that order. The ratio copula captures the sharp transitions of the densities accurately, while the TLL is much more diffuse. The IGC struggles to converge to a correct representation.

High-dimensional examples: We model the copula density obtained with a 50/50 train/test split, from the digits dataset ($n = 1797, d = 64$) and the MNIST dataset ($n = 60000, d = 784$), both showing handwritten digits. This dataset assesses the ability of our model to pick up complex and multimodal dependencies in the data and is typically difficult for copula models. In Table 1, we show the results of the Gaussian-guided ratio copula (GG+Ratio with a simple network and GG+Ratio *large* having a larger network architecture for the classifier, see Appendix C.3), a simplified vine, a Gaussian copula, and the IGC. Unlike the vine copula, whose building blocks scale exclusively with sample size (Corollary 4.3), our method easily integrates more complex architectures as done with the GG+Ratio *large*.

For digits, our method performs best in LL and sampling. For MNIST, we achieve a better LL, and are

second in W2 to the IGC. We show generated samples for both digits and MNIST from all methods in Appendix C.3. Notably, samples from the vine on MNIST are of worse quality, and IGC MNIST samples appear over-inflated or unfocused. Only the ratio copula captures the right thickness of the characters.

To conclude, we demonstrated that ratio copulas are extremely flexible models if given enough data (Figure 2), while in smaller data regimes, the L+ratio might be preferred. Thus, their use is warranted in vine constructions, broadening the class of non-parametric bivariate models. Further, though the application on Digits and MNIST, we demonstrate the ability of ratio copulas to scale to high dimensions with complex dependencies, thereby establishing ratio copulas as a useful non-parametric copula model.

6 RELATED RESEARCH

Our research elucidates connections in three ways. Firstly, this paper bridges copulas with DRE and the related Mutual information (MI) estimation. Secondly, in the taxonomy of implicit learning losses from Mohamed and Lakshminarayanan (2016) applied to copulas, our work stands alone, not only through the use of a Bregman divergence, but also as the only deep copula using non-likelihood-based learning with the bespoke goal of density estimation. Lastly, we establish a connection with energy-based models (EBMs) and point out some existing *implicit* applications of copulas in the field.

The DRE - MI - copula triangle: One of the many applications of DRE is to estimate MI, a measure of mutual dependence, computed as $KL(p \parallel \prod_i p_i)$. This is identified as the negative copula entropy by Davy and Doucet (2003) and by Ma and Sun (2011) (the canonical example of MI, *although not stated as such*, is the estimation of a Gaussian copula, e.g. Section 6.2 in Choi et al. (2022)). The community has taken notice, in turn estimating MI using copulas: Zeng and Durrani (2011); Ince et al. (2017); Lall et al. (2021). The reverse direction is used by Samo (2021)

to estimate copulas with a constrained maximisation problem of MI. Thus, existing research has explored two edges of this triangle; DRE→MI and MI↔copula. We connect the last edge, by using DRE to estimate copulas; DRE→copula. In particular, our approach and Choi et al. (2021) share the use of an invertible mapping to facilitate the ratio estimation.

Learning methods for copulas: In Mohamed and Lakshminarayanan (2016), three likelihood-free learning methods are discussed, with a shared goal of density estimation by comparison: [1] moment matching methods, [2] the maximum mean discrepancy (MMD), and [3] probabilistic classification (including Bregman and f -divergences). In the copula estimation community, likelihood-based training is prevalent. An exception is Hofert et al. (2021), who use a generative moment matching network, a purely generative model, to learn a deep copula by matching the moments of samples against observations - exemplifying an application of [1] to copulas. Similarly, Janke et al. (2021) use the MMD as a learning metric for a generative network to replicate observed copula samples - showcasing the use of [2]. Huk et al. (2023) use the MMD as a learning metric for when MLE-based training is unavailable but still obtain a likelihood model. Relatedly, Alquier et al. (2023) estimate parametric copulas with the MMD, instead motivated by its robustness properties. Finally, the current paper is the first copula estimation method implementing [3]: the use of probabilistic classification. Our work is unique in constructing a deep copula model trained using a likelihood-free objective while still obtaining a density estimator - in contrast to the contributions of Hofert et al. (2021) and Janke et al. (2021).

EBMs and other hidden copulas: EBMs (Ranzato et al., 2006) are a class of unnormalised models giving unconstrained values to data on \mathcal{X} , to maximise the values given to observations from p and minimise the values anywhere else. They are often trained with contrastive objectives (Hinton, 2002), and in particular also employ probabilistic DRE (Gutmann and Hyvärinen, 2012; Rhodes et al., 2020). They can be interpreted as an extreme version of ratio copulas, aimed at capturing dependence in data, not only discriminating against $\prod_i p_i$ but the whole space of \mathcal{X} . Conversely, our ratio copula model is an EBM trained on the ratio of $p/\prod_i p_i$. A notable use case of EBMs is modelling the latent space of variational autoencoders (VAE) Kingma (2013), which is equipped with independent marginal densities. Xiao and Han (2022) train an EBM with DRE to learn $p/\prod_i p_i$ for better performance, indirectly exemplifying the performance of ratio copulas! In fact, early classification DRE works

advocate the use of $\prod_i p_i$ to learn p ; Hastie et al. (2009) p.479 and Shi and Horvath (2006) both unknowingly use a ratio copula model. Our paper thus consolidates their empirical work with a grounded derivation of this procedure, bridging the rich field of DRE with copula estimation.

7 DISCUSSION

In this work, we introduce the ratio copula as a novel framework for copula estimation. Our method fits a classifier between transformed copula samples and independent data on a Gaussian space, identifying dependence. We establish equivalences between popular copulas and classification tasks, providing new insights into their models. Empirically, our model is on par with the best copula methods for density evaluation as well as sampling.

Future directions of our work can incorporate different losses (Menon and Ong, 2016) to fine-tune ratio copulas to focus on traits such as extreme dependence in the tails. An exciting direction is utilising other DRE methods for copula estimation, most notably linking copula density estimation with (time) score matching as in Choi et al. (2022). Furthermore, some applications of copulas such as Bayesian Copula Networks Elian (2010), and conformal prediction (Messoudi et al., 2021; Park et al., 2024) desire to work with the copula distribution rather than the density. Extending our ratio copula formulation to obtain a model that parameterises CDFs is another meaningful direction.

Crucially, given the success copulas have enjoyed in related fields (Tagasovska et al., 2019; Czado et al., 2022; Huk et al., 2024) as methodological tools, further improvements would yield dividends across all of their applications.

Author Contributions: David Huk originated the idea, designed the project, wrote the code, ran the experiments, wrote the paper, and derived the proofs. Ritabrata Dutta and Mark Steel jointly supervised the project, provided valuable feedback, and helped edit the paper.

Acknowledgements: We thank all the reviewers for their helpful feedback. We also thank Alexis Derumigny, and Lorenzo Pacchiardi for helpful comments before the camera-ready version. David Huk is funded by the Center for Doctoral Training in Mathematical Sciences at Warwick. Ritabrata Dutta is funded by EPSRC (grant nos. EP/V025899/1 and EP/T017112/1) and NERC (grant no. NE/T00973X/1).

References

- Aas, K., Czado, C., Frigessi, A., and Bakken, H. (2009). Pair-copula constructions of multiple dependence. *Insurance: Mathematics and economics*, 44(2):182–198.
- Alimoglu, F. and Alpaydin, E. (1997). Combining multiple representations and classifiers for pen-based handwritten digit recognition. In *Proceedings of the Fourth International Conference on Document Analysis and Recognition*, volume 2, pages 637–640. IEEE.
- Alquier, P., Chérif-Abdellatif, B.-E., Derumigny, A., and Fermanian, J.-D. (2023). Estimation of copulas via maximum mean discrepancy. *Journal of the American Statistical Association*, 118(543):1997–2012.
- Ashok, A., Marcotte, É., Zantedeschi, V., Chapados, N., and Drouin, A. (2024). TACTis-2: Better, faster, simpler attentional copulas for multivariate time series. In *The Twelfth International Conference on Learning Representations*.
- Bedford, T. and Cooke, R. M. (2001). Probability density decomposition for conditionally dependent random variables modeled by vines. *Annals of Mathematics and Artificial intelligence*, 32:245–268.
- Berkes, P., Wood, F., and Pillow, J. (2008). Characterizing neural dependencies with copula models. *Advances in Neural Information Processing Systems*, 21.
- Breiman, L. (2001). Random forests. *Machine learning*, 45:5–32.
- Choi, K., Liao, M., and Ermon, S. (2021). Featurized density ratio estimation. In *Uncertainty in Artificial Intelligence*, pages 172–182. PMLR.
- Choi, K., Meng, C., Song, Y., and Ermon, S. (2022). Density ratio estimation via infinitesimal classification. In *International Conference on Artificial Intelligence and Statistics*, pages 2552–2573. PMLR.
- Czado, C. (2019). Analyzing dependent data with vine copulas. *Lecture Notes in Statistics, Springer*, 222.
- Czado, C., Bax, K., Sahin, Ö., Nagler, T., Min, A., and Paterlini, S. (2022). Vine copula based dependence modeling in sustainable finance. *The Journal of Finance and Data Science*, 8:309–330.
- Czado, C., Ivanov, E., and Okhrin, Y. (2019). Modelling temporal dependence of realized variances with vines. *Econometrics and Statistics*, 12:198–216.
- Czado, C. and Nagler, T. (2022). Vine copula based modeling. *Annual Review of Statistics and Its Application*, 9:453–477.
- Davy, M. and Doucet, A. (2003). Copulas: a new insight into positive time-frequency distributions. *IEEE signal processing letters*, 10(7):215–218.
- Diederik, P. K. (2014). Adam: A method for stochastic optimization. (*No Title*).
- Elidan, G. (2010). Copula bayesian networks. *Advances in neural information processing systems*, 23.
- Fermanian, J.-D. (2005). Goodness-of-fit tests for copulas. *Journal of multivariate analysis*, 95(1):119–152.
- Flamary, R., Courty, N., Gramfort, A., Alaya, M. Z., Boisbunon, A., Chambon, S., Chapel, L., Corenflos, A., Fatras, K., Fournier, N., Gautheron, L., Gayraud, N. T., Janati, H., Rakotomamonjy, A., Redko, I., Rolet, A., Schutz, A., Seguy, V., Sutherland, D. J., Tavenard, R., Tong, A., and Vayer, T. (2021). Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8.
- Geenens, G., Charpentier, A., and Paindaveine, D. (2017). Probit transformation for nonparametric kernel estimation of the copula density. *Bernoulli*, 23(3):1848 – 1873.
- Genève, O. S.-H. (2003). Research paper number 57 nonparametric estimation of copulas for time series. *Journal of Risk*, 5:25–54.
- Górecki, J. and Okhrin, O. (2024). *Hierarchical Archimedean Copulas*. Springer Nature.
- Gutmann, M. and Hyvärinen, A. (2010). Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings.
- Gutmann, M. U. and Hirayama, J.-i. (2011). Bregman divergence as general framework to estimate unnormalized statistical models. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, pages 283–290.
- Gutmann, M. U. and Hyvärinen, A. (2012). Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of machine learning research*, 13(2).
- Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.
- Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800.
- Hofert, M., Mächler, M., and McNeil, A. J. (2013). Archimedean copulas in high dimensions: Estima-

- tors and numerical challenges motivated by financial applications. *Journal de la Société Française de Statistique*, 154(1):25–63.
- Hofert, M., Prasad, A., and Zhu, M. (2021). Quasi-random sampling for multivariate distributions via generative neural networks. *Journal of Computational and Graphical Statistics*, 30(3):647–670.
- Huk, D., Adewoyin, R. A., and Dutta, R. (2023). Probabilistic rainfall downscaling: Joint generalized neural models with censored spatial gaussian copula. *arXiv preprint arXiv:2308.09827*.
- Huk, D., Zhang, Y., Steel, M., and Dutta, R. (2024). Quasi-Bayes meets vines. *arXiv preprint arXiv:2406.12764*.
- Ince, R. A., Giordano, B. L., Kayser, C., Rousselet, G. A., Gross, J., and Schyns, P. G. (2017). A statistical framework for neuroimaging data analysis based on mutual information estimated via a gaussian copula. *Human brain mapping*, 38(3):1541–1573.
- Janke, T., Ghanmi, M., and Steinke, F. (2021). Implicit generative copulas. *Advances in Neural Information Processing Systems*, 34:26028–26039.
- Jiang, A. Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., Chaplot, D. S., de las Casas, D., Hanna, E. B., Bressand, F., Lengyel, G., Bour, G., Lample, G., Lavaud, L. R., Saulnier, L., Lachaux, M.-A., Stock, P., Subramanian, S., Yang, S., Antoniak, S., Scao, T. L., Gervet, T., Lavril, T., Wang, T., Lacroix, T., and Sayed, W. E. (2024). Mixtral of experts.
- Kamthe, S., Assefa, S., and Deisenroth, M. (2021). Copula flows for synthetic data generation. *arXiv preprint arXiv:2101.00598*.
- Kanamori, T., Hido, S., and Sugiyama, M. (2009). A least-squares approach to direct importance estimation. *Journal of Machine Learning Research*, 10(48):1391–1445.
- Kingma, D. P. (2013). Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*.
- Lall, S., Sinha, D., Ghosh, A., Sengupta, D., and Bandyopadhyay, S. (2021). Stable feature selection using copula based mutual information. *Pattern Recognition*, 112:107697.
- LeCun, Y. (1998). The mnist database of handwritten digits. <http://yann.lecun.com/exdb/mnist/>.
- Loh, W.-Y. (2011). Classification and regression trees. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, 1(1):14–23.
- Lopez-Paz, D., Hernández-lobato, J., and Schölkopf, B. (2012). Semi-supervised domain adaptation with non-parametric copulas. *Advances in Neural Information Processing Systems*, 25.
- Ma, J. and Sun, Z. (2011). Mutual information is copula entropy. *Tsinghua Science & Technology*, 16(1):51–54.
- McGibbon, R. T. (2015). pyhmc: Hamiltonian monte carlo in python.
- Menon, A. and Ong, C. S. (2016). Linking losses for density ratio and class-probability estimation. In *International Conference on Machine Learning*, pages 304–313. PMLR.
- Messoudi, S., Destercke, S., and Rousseau, S. (2021). Copula-based conformal prediction for multi-target regression. *Pattern Recognition*, 120:108101.
- Mohamed, S. and Lakshminarayanan, B. (2016). Learning in implicit generative models. *arXiv preprint arXiv:1610.03483*.
- Nagler, T. and Czado, C. (2016). Evading the curse of dimensionality in nonparametric density estimation with simplified vine copulas. *Journal of Multivariate Analysis*, 151:69–89.
- Nagler, T., Schellhase, C., and Czado, C. (2017). Nonparametric estimation of simplified vine copula models: comparison of methods. *Dependence Modeling*, 5(1):99–120.
- Neal, R. M. (2011). MCMC using Hamiltonian dynamics. In *Handbook of Markov Chain Monte Carlo*, pages 113–162. Chapman and Hall/CRC.
- Ng, Y., Hasan, A., and Tarokh, V. (2022). Inference and sampling for archimax copulas. *Advances in Neural Information Processing Systems*, 35:17099–17116.
- Ogasawara, H. (2024). The multivariate t-distribution with multiple degrees of freedom. *Communications in Statistics-Theory and Methods*, 53(1):144–169.
- Park, J. W., Tibshirani, R., and Cho, K. (2024). Semiparametric conformal prediction. *arXiv preprint arXiv:2411.02114*.
- Ranzato, M. a., Poultney, C., Chopra, S., and Cun, Y. (2006). Efficient learning of sparse representations with an energy-based model. In Schölkopf, B., Platt, J., and Hoffman, T., editors, *Advances in Neural Information Processing Systems*, volume 19. MIT Press.
- Rhodes, B., Xu, K., and Gutmann, M. U. (2020). Telescoping density-ratio estimation. *Advances in Neural Information Processing Systems*, 33:4905–4916.
- Salmon, F. (2012). The formula that killed wall street. *Significance*, 9(1):16–20.
- Samo, Y.-L. K. (2021). Inductive mutual information estimation: A convex maximum-entropy copula approach. In *International Conference on Artificial Intelligence and Statistics*, pages 2242–2250. PMLR.

- Scaillet, O. (2007). Kernel-based goodness-of-fit tests for copulas with fixed smoothing parameters. *Journal of Multivariate Analysis*, 98(3):533–543.
- Shi, T. and Horvath, S. (2006). Unsupervised learning with random forest predictors. *Journal of Computational and Graphical Statistics*, 15(1):118–138.
- Silverman, B. W. (2018). *Density estimation for statistics and data analysis*. Routledge.
- Sklar, M. (1959). Fonctions de répartition à n dimensions et leurs marges. In *Annales de l'ISUP*, volume 8, pages 229–231.
- Srivastava, A., Han, S., Xu, K., Rhodes, B., and Guttman, M. U. (2023). Estimating the density ratio between distributions with high discrepancy using multinomial logistic regression. *Transactions on Machine Learning Research*.
- Sugiyama, M., Suzuki, T., and Kanamori, T. (2010). Density ratio estimation: A comprehensive review (statistical experiment and its related topics). , 1703:10–31.
- Sugiyama, M., Suzuki, T., and Kanamori, T. (2012). *Density ratio estimation in machine learning*. Cambridge University Press.
- Tagasovska, N., Ackerer, D., and Vatter, T. (2019). Copulas as high-dimensional generative models: Vine copula autoencoders. *Advances in Neural Information Processing Systems*, 32.
- Vatter, T. and Nagler, T. P. (2022). 0.6. 1.
- Wang, H., Yu, L., Cao, Z., and Ermon, S. (2021). Multi-agent imitation learning with copulas. In *Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part I* 21, pages 139–156. Springer.
- Wasserman, L. (2006). *All of nonparametric statistics*. Springer Science & Business Media.
- Xiao, Z. and Han, T. (2022). Adaptive multi-stage density ratio estimation for learning latent space energy-based model. *Advances in Neural Information Processing Systems*, 35:21590–21601.
- Yuksel, S. E., Wilson, J. N., and Gader, P. D. (2012). Twenty years of mixture of experts. *IEEE Transactions on Neural Networks and Learning systems*, 23(8):1177–1193.
- Zeng, X. and Durrani, T. (2011). Estimation of mutual information using copula density function. *Electronics letters*, 47(8):493–494.

Checklist

1. For all models and algorithms presented, check if you include:
 - (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes] We include a notation part in Section 2, and state assumptions at the start of relevant sections of the text.
 - (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes] We provide Lemma 3.2 to quantify the complexity of our task.
 - (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes] Will be included in the additional material and published upon acceptance.
2. For any theoretical claim, check if you include:
 - (a) Statements of the full set of assumptions of all theoretical results. [Yes]
 - (b) Complete proofs of all theoretical results. [Yes] In the Appendix.
 - (c) Clear explanations of any assumptions. [Yes]
3. For all figures and tables that present empirical results, check if you include:
 - (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
 - (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
 - (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
 - (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
 - (a) Citations of the creator If your work uses existing assets. [Yes]
 - (b) The license information of the assets, if applicable. [Yes]
 - (c) New assets either in the supplemental material or as a URL, if applicable. [Yes]
 - (d) Information about consent from data providers/curators. [Not Applicable]

- (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]
5. If you used crowdsourcing or conducted research with human subjects, check if you include:
- (a) The full text of instructions given to participants and screenshots. [Not Applicable]
 - (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
 - (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Supplementary Materials

Your copula is a classifier in disguise: classification-based copula density estimation

A PROOFS

A.1 Theorem 3.1

We begin by stating assumptions needed for each part of the proof, followed by proving each of the three statements, namely [1] uniqueness, [2] consistency in the limit of infinite data from p , [3] asymptotic normality of the parameter estimate. This result is largely an application of Gutmann and Hyvärinen (2012) Theorems 1,2 and 3.

Assumption A1: We begin by assuming that all marginals are known and, that they are continuous on the space $\mathcal{X} \in \mathbb{R}$ where $p(\mathbf{x}) > 0$ so that the copula exists by Sklar's theorem (Sklar, 1959). We consider the loss $\mathcal{L}(\theta)$ from (12), as in Gutmann and Hyvärinen (2012).

This is similar to the assumption taken for Sklar's theorem, where assuming continuity lets us derive the uniqueness of the copula. Positivity is required for the copula to exist in the first place.

Assumption A2: Our ratio copula model is well-specified, meaning $c(\mathbf{x}) \in \mathcal{C}$.

This is a classic assumption for deriving statistical results in model estimation, as done in the proof for the rate of convergence for simplified vine copula models Nagler and Czado (2016). In our case, using MLP-based copula models gives us access to a much larger model class compared to simplified vines, making the discrepancy between the true model and the best model within our model class smaller, thereby reducing misspecification concerns.

Proof. As our marginals are positive whenever $p(\mathbf{x}) > 0$, the same goes for $q(\mathbf{x}) = \prod_{i=1}^d p_i(x_i)$. This then applies to \tilde{p} and \tilde{q} too. Therefore, we satisfy the assumptions of Theorem 1 in Gutmann and Hyvärinen (2012), which gives us the result namely that $c^{\text{ratio}}(\Phi^{-1} \circ \mathcal{P}(\mathbf{x}); \boldsymbol{\theta}^*)$ is the only minimised of our loss. Further, as the model is well specified due to A2, we recover $c(\mathbf{x}) = \tilde{p}(\mathbf{x})/\tilde{q}(\mathbf{x}) = p(\mathbf{x})/q(\mathbf{x})$. \square

Assumption A3: Keeping ν fixed, and increasing T_p (meaning T_q will also increase at a fixed constant rate), the supremum of the empirical approximation $\widehat{\mathcal{L}}(\boldsymbol{\theta})$ of Loss 12 is such that, with the empirical loss defined as:

$$\begin{aligned}\widehat{\mathcal{L}}(\boldsymbol{\theta}) &= \arg \min_{\boldsymbol{\theta} \in \Theta} - \sum_{k=1}^{T_p} \ln \left(\frac{c^{\text{ratio}}(\Phi^{-1} \circ \mathcal{P}(\mathbf{x}_p^k); \boldsymbol{\theta})}{\nu + c^{\text{ratio}}(\Phi^{-1} \circ \mathcal{P}(\mathbf{x}_p^k); \boldsymbol{\theta})} \right) \\ &\quad - \nu \cdot \sum_{k=1}^{T_q} \ln \left(\frac{\nu}{\nu + c^{\text{ratio}}(\Phi^{-1} \circ \mathcal{P}(\mathbf{x}_q^k); \boldsymbol{\theta})} \right),\end{aligned}\tag{12}$$

for $\mathbf{x}_p^k \sim \tilde{p}$ and $\mathbf{x}_q^k \sim \tilde{q}$, we have:

$$\sup_{\boldsymbol{\theta}} |\widehat{\mathcal{L}}(\boldsymbol{\theta}) - \mathcal{L}(\boldsymbol{\theta})| \rightarrow^{\mathbb{P}} 0, \quad \text{when } T_p \rightarrow \infty.\tag{13}$$

Assumption A4: Defining the matrix \mathcal{I}_ν as:

$$\mathcal{I}_\nu = \int \mathbf{g}(\mathbf{z})\mathbf{g}(\mathbf{z})^T F_\nu(\mathbf{z})\tilde{p}(\mathbf{z}) d\mathbf{z}, \quad (14)$$

this matrix has full rank, for the quantities defined as below:

$$\mathbf{g}(\mathbf{z}) = \nabla_{\boldsymbol{\theta}} \ln c^{\text{ratio}}(\mathbf{z}; \boldsymbol{\theta})|_{\boldsymbol{\theta}^*} \cdot \tilde{q}(\mathbf{z}), \quad \text{and} \quad F_\nu(\mathbf{z}) = \frac{\nu \tilde{q}(\mathbf{z})}{\tilde{p}(\mathbf{z}) + \nu \cdot \tilde{p}(\mathbf{z})}. \quad (15)$$

Assumptions A3 and A4 have their counterparts in maximum likelihood estimation, see *e.g.* Wasserman (2006).

Proof. As assumptions A1, A2, A3 and A4 imply that we satisfy assumption (a,b,c) of theorem 2 in Gutmann and Hyvärinen (2012), we obtain its result, namely that $\hat{\boldsymbol{\theta}} \rightarrow^{\mathbb{P}} \boldsymbol{\theta}^*$. \square

Lastly, we obtain asymptotic normality of our parameter estimate.

Proof. As we satisfy assumption A1-4, theorem 3 in Gutmann and Hyvärinen (2012) applies, and we obtain that $\sqrt{T_p}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*)$ is asymptotically normal with mean $\mathbf{0}$ and covariance matrix $\boldsymbol{\Sigma}$ with expression:

$$\boldsymbol{\Sigma} = \mathcal{I}_\nu^{-1} - \left(1 + \frac{1}{\nu}\right) \mathcal{I}_\nu^{-1} \mathbb{E}(F_\nu \mathbf{g}) \mathbb{E}(F_\nu \mathbf{g})^T \mathcal{I}_\nu^{-1} \quad (16)$$

where $\mathbb{E}(F_\nu \mathbf{g}) = \int F_\nu(\mathbf{z})\mathbf{g}(\mathbf{z})\tilde{p}(\mathbf{z}) d\mathbf{z}$.

\square

In particular, as $\nu \rightarrow \infty$, $\boldsymbol{\Sigma}$ becomes independent of the form of \tilde{q} , meaning that no matter the complexity of $c(\cdot)$ relative to a simple independence copula, with infinite noise samples, estimating it is only dependent on the shape of p alone. Furthermore, under some mild assumptions detailed in Geenens et al. (2017), the resulting \tilde{p} after applying transformation (5) has unconstrained support and it, as well as its partial derivatives, will be bounded on \mathbb{R}^d . This is useful to ensure a well-behaved $\boldsymbol{\Sigma}$ for asymptotic normality, since $\boldsymbol{\Sigma}$ is defined in terms of the gradients of \tilde{p} .

A.2 Lemma 3.2

Proof. We first derive the KL divergence between the two Gaussians of the numerator and denominator in a Gaussian copula with covariance $\boldsymbol{\Sigma}$.

$$\begin{aligned} & KL(\mathcal{N}_d(\cdot; \mathbf{0}, \boldsymbol{\Sigma}) || \mathcal{N}_d(\cdot; \mathbf{0}, \mathbf{I}_d)) \\ &= \int p(\mathbf{x}) \cdot \left[\frac{1}{2} \log \frac{|\mathbf{I}_d|}{|\boldsymbol{\Sigma}|} - \frac{1}{2} (\mathbf{x} - \mathbf{0})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \mathbf{0}) + \frac{1}{2} (\mathbf{x} - \mathbf{0})^T \mathbf{I}_d^{-1} (\mathbf{x} - \mathbf{0}) \right] d\mathbf{x} \\ &= \frac{1}{2} \log \frac{1}{|\boldsymbol{\Sigma}|} - \frac{1}{2} \mathbb{E}[\mathbf{x}^T \boldsymbol{\Sigma}^{-1} \mathbf{x}] + \frac{1}{2} \mathbb{E}[\mathbf{x}^T \mathbf{x}] \\ &= \frac{1}{2} \log \frac{1}{|\boldsymbol{\Sigma}|} - \frac{1}{2} \text{tr}\{\boldsymbol{\Sigma}^{-1} \boldsymbol{\Sigma}\} + \frac{1}{2} \text{tr}\{\boldsymbol{\Sigma}\} = \frac{1}{2} \log \frac{1}{|\boldsymbol{\Sigma}|} \end{aligned}$$

where the final equality follows from the fact that in a Gaussian copula, all dimensions marginally have unit variance, meaning $\boldsymbol{\Sigma}$ has ones on the diagonal. Next, for e_1, \dots, e_d the eigenvalues of $\boldsymbol{\Sigma}$ with e_{\min} the lowest of them, we get the desired bound

$$\begin{aligned} & KL(\mathcal{N}_d(\cdot; \mathbf{0}, \boldsymbol{\Sigma}) || \mathcal{N}_d(\cdot; \mathbf{0}, \mathbf{I}_d)) = \frac{1}{2} \log \frac{1}{|\boldsymbol{\Sigma}|} \\ &= \frac{1}{2} \log \frac{1}{\prod_{i=1}^d e_i} \leq -\frac{d}{2} \log e_{\min}. \end{aligned}$$

\square

A.3 Proposition 4.1

Proof. A binary QDA classifier $h(\mathbf{z} : \mathbb{R}^d \mapsto \{0, 1\})$ assigns class $y = 1$ to \mathbf{z} with a probability given by

$$\mathbb{P}(y = 1|\mathbf{z}) = \frac{\sqrt{(2\pi)^d |\Sigma_1|}^{-1} \exp\left(-\frac{1}{2} (\mathbf{z} - \boldsymbol{\mu}_1)^T \Sigma_1^{-1} (\mathbf{z} - \boldsymbol{\mu}_1)\right)}{\sqrt{(2\pi)^d |\Sigma_1|}^{-1} \exp\left(-\frac{1}{2} (\mathbf{z} - \boldsymbol{\mu}_1)^T \Sigma_1^{-1} (\mathbf{z} - \boldsymbol{\mu}_1)\right) + \sqrt{(2\pi)^d |\Sigma_0|}^{-1} \exp\left(-\frac{1}{2} (\mathbf{z} - \boldsymbol{\mu}_0)^T \Sigma_0^{-1} (\mathbf{z} - \boldsymbol{\mu}_0)\right)} \quad (17)$$

and class $y = 0$ otherwise, where $\boldsymbol{\mu}_1, \boldsymbol{\mu}_0$ are the means and Σ_1, Σ_0 are the covariances of the Gaussian for classes 1 and 0 respectively. Next, we consider the following expression for this classifier

$$\frac{\mathbb{P}(y = 1|\mathbf{z})}{1 - \mathbb{P}(y = 1|\mathbf{z})} = \frac{\sqrt{(2\pi)^d |\Sigma_1|}^{-1} \exp\left(-\frac{1}{2} (\mathbf{z} - \boldsymbol{\mu}_1)^T \Sigma_1^{-1} (\mathbf{z} - \boldsymbol{\mu}_1)\right)}{\sqrt{(2\pi)^d |\Sigma_0|}^{-1} \exp\left(-\frac{1}{2} (\mathbf{z} - \boldsymbol{\mu}_0)^T \Sigma_0^{-1} (\mathbf{z} - \boldsymbol{\mu}_0)\right)} \quad (18)$$

By assuming class 1 follows a centred Gaussian with covariance matrix Σ and further assuming class 0 follows a standard Gaussian, that is setting $\boldsymbol{\mu}_1 = \mathbf{0}, \boldsymbol{\mu}_0 = \mathbf{0}$ and $\Sigma_1 = \Sigma, \Sigma_0 = \mathbf{I}_d$, we obtain

$$\frac{\mathbb{P}(y = 1|\mathbf{z})}{1 - \mathbb{P}(y = 1|\mathbf{z})} = \frac{\sqrt{(2\pi)^d |\Sigma|}^{-1} \exp\left(-\frac{1}{2} \mathbf{z}^T \Sigma^{-1} \mathbf{z}\right)}{\sqrt{(2\pi)^d}^{-1} \exp\left(-\frac{1}{2} \mathbf{z}^T \mathbf{I}_d \mathbf{z}\right)}.$$

This can now be seen to be the ratio between the likelihood of a dependent Gaussian with mean $\mathbf{0}$ and covariance Σ and a standard Gaussian, which is exactly equal to the expression of a Gaussian copula parameterised by Σ when taking $\mathbf{z} = \Phi^{-1} \circ \mathcal{P}(\mathbf{x}) \in \mathbb{R}^d$ as inputs to both:

$$c_{\Sigma}(\Phi^{-1} \circ \mathcal{P}(\mathbf{x})) = \frac{\mathcal{N}_d(\Phi^{-1} \circ \mathcal{P}(\mathbf{x}); \mathbf{0}, \Sigma)}{\prod_{i=1}^d \mathcal{N}(\Phi^{-1} \circ P_i(x_i); 0, 1)}.$$

□

A.4 Student's t copula

We provide here a result on non-Gaussian copulas, where we use the definition of a student's t-distribution with multiple degrees of freedom from Ogasawara (2024) (note that this is not the "usual" multivariate Student with one degree of freedom parameter, where the common latent mixing variable induces dependence even if correlations are zero):

Proposition A.1. Consider a Student's t copula with density

$$c_{\nu, P}^t(\mathbf{u}) = \frac{f_{\nu, P}\left(F_{\nu}^{-1}(P_1(x_1)), \dots, F_{\nu}^{-1}(P_d(x_d))\right)}{\prod_{i=1}^d f_{\nu}\left(F_{\nu}^{-1}(P_i(x_i))\right)},$$

where for ν the degrees of freedom, $f_{\nu, P}$ is the d -dimensional student's t pdf with mean $\mathbf{0}$ and correlation matrix P , F_{ν}^{-1} is the inverse of a univariate standard student's t cdf, and f_{ν} is the density of the univariate standard t-distribution. The associated Bayes optimal classifier of this copula is found by the decision boundary between a student's t distribution and a student's t distribution with multiple degrees of freedom when all its degrees of freedom are equal to ν , both densities with mean $\mathbf{0}$ but correlation matrices P and $\text{diag}(P)$ respectively.

Proof. The decision boundary of the above-mentioned classification problem will take the form of a ratio between the two densities. The numerator is simply $f_{\nu, P}$, matching the numerator of the copula density. The denominator of the copula is a product of independent marginal student-t densities with common degrees of freedom ν . We note that a multivariate student's t does not factorise into d marginals when its correlation matrix is diagonal. This can be understood through the equivalent representation of a student's t with $\mathbf{Z} \sim f_{\nu, P} \iff \mathbf{Z} = \mathbf{X}/\sqrt{Y/\nu}$ where $\mathbf{X} \sim \mathcal{N}(\mathbf{0}, P)$ and Y follows a Chi-squared distribution with ν degrees of freedom. As the same Y is used

across all dimensions, the resulting \mathbf{Z} variables are not marginally independent. However, this is the case in a multivariate t-distribution with multiple degrees of freedom with a diagonal correlation matrix, as different Chi-squared variables are used for each dimension. When all Chi-squared variables share the same degree of freedom, the factorised form becomes that of the denominator of a student's t copula, concluding our proof. \square

A.5 Proposition 4.2

The proof consists of first identifying that a KDE copula is a sum of ratios between Gaussians. Then, we use a similar reasoning to Proposition 4.1 for each summand to conclude the proof.

Proof. We begin with the expression of a Gaussian KDE copula $\hat{c}_{T_p}(\Phi^{-1} \circ \mathcal{P}(\mathbf{x}))$ based on T_p kernel means at observations $\{(Z_1^k, \dots, Z_d^k)\}_{k=1}^{T_p}$ for $Z_i^k = \Phi^{-1} \circ P_i(x_i^k)$, $x_i^k \sim p_i$:

$$\hat{c}_{T_p}(\Phi^{-1} \circ \mathcal{P}(\mathbf{x})) = \frac{\sum_{k=1}^{T_p} \prod_{i=1}^d \{\mathcal{N}(\Phi^{-1} \circ P_i(x_i) - Z_i^k | 0, b)\}}{T_p \cdot \mathcal{N}_d(\Phi^{-1} \circ \mathcal{P}(\mathbf{x}) | \mathbf{0}, \mathbf{I}_d)}. \quad (19)$$

Notice that the expression in the numerator is a d -dimensional Gaussian density, as

$$\prod_{i=1}^d \{\mathcal{N}(\Phi^{-1} \circ P_i(x_i) - Z_i^k | 0, b)\} = \mathcal{N}(\mathbf{z}; \mathbf{Z}^k, b \cdot \mathbf{I}_d) \quad (20)$$

where we write $\mathbf{z} = \Phi^{-1} \circ \mathcal{P}(\mathbf{x})$ and $\mathbf{Z}^k = (Z_1^k, \dots, Z_d^k)$. It then ensues that

$$\hat{c}_{T_p}(\Phi^{-1} \circ \mathcal{P}(\mathbf{x})) = \frac{\sum_{k=1}^{T_p} \mathcal{N}(\mathbf{z}; \mathbf{Z}^k, b \cdot \mathbf{I}_d)}{T_p \cdot \mathcal{N}_d(\mathbf{z} | \mathbf{0}, \mathbf{I}_d)} \quad (21)$$

showing a sum of ratios of Gaussians with means \mathbf{Z}^k and $\mathbf{0}$ and variances $b \cdot \mathbf{I}_d$ and \mathbf{I}_d for the numerator and denominator respectively.

Following the same reasoning as in Proposition 4.1, each such term coincides with the ratio of posterior class probabilities of a binary QDA classifier for two Gaussian classes with the aforementioned means and variances, having labels $y = 1$ for the numerator and $y = 0$ for the denominator class. Concretely, denoting each such QDA classifier as $h^k(\mathbf{z}; \mu_1^k, \mu_0^k, \Sigma_1, \Sigma_0)$, shortening the notation to $h^k(\mathbf{z})$, with means $\mu_1^k = \mathbf{Z}^k, \mu_0^k = \mathbf{0}$ but shared variances $\Sigma_1 = b \cdot \mathbf{I}_d, \Sigma_0 = \mathbf{I}_d$, we obtain the following ratio:

$$\delta^k(\mathbf{z}) := \frac{h^k(\mathbf{z})}{1 - h^k(\mathbf{z})} = \frac{\sqrt{(2\pi)^d |b \cdot \mathbf{I}_d|}^{-1} \exp\left(-\frac{1}{2} (\mathbf{z} - \mathbf{Z}^k)^T b \cdot \mathbf{I}_d^{-1} (\mathbf{z} - \mathbf{Z}^k)\right)}{\sqrt{(2\pi)^d |\mathbf{I}_d|}^{-1} \exp\left(-\frac{1}{2} (\mathbf{z} - \mathbf{0})^T \mathbf{I}_d^{-1} (\mathbf{z} - \mathbf{0})\right)}. \quad (22)$$

Now, considering an average of such QDA decision boundaries, denoted as $\delta_{T_p}(\mathbf{z}) := \frac{\sum_{k=1}^{T_p} \delta^k(\mathbf{z})}{T_p}$, we get:

$$\delta_{T_p}(\mathbf{z}) = \frac{\sum_{k=1}^{T_p} \sqrt{(2\pi)^d |b \cdot \mathbf{I}_d|}^{-1} \exp\left(-\frac{1}{2} (\mathbf{z} - \mathbf{Z}^k)^T b \cdot \mathbf{I}_d^{-1} (\mathbf{z} - \mathbf{Z}^k)\right)}{\sqrt{(2\pi)^d |\mathbf{I}_d|}^{-1} \exp\left(-\frac{1}{2} (\mathbf{z} - \mathbf{0})^T \mathbf{I}_d^{-1} (\mathbf{z} - \mathbf{0})\right)} = \hat{c}_{T_p}(\Phi^{-1} \circ \mathcal{P}(\mathbf{x})). \quad (23)$$

Thus, $\delta_{T_p}(\mathbf{z})$ and $\hat{c}_{T_p}(\Phi^{-1} \circ \mathcal{P}(\mathbf{x}))$ coincide, showing that a KDE copula with Gaussian kernels is equivalent to a mixture of QDA decision boundaries with mixture specific means per class, but same variances per class across mixtures, concluding our proof. \square

A.6 Corollary 4.3

Proof. We begin by stating the simplified vine copula density for a d -dimensional vector $(u_1, \dots, u_d) \in [0, 1]^d$ and $i, j \in \{1, \dots, d\}$. For more details on vine copulas, the reader is referred to Appendix B.

$$\mathbf{c}(u_1, \dots, u_d) = \prod_{i \neq j}^{d(d-1)/2} c_{ij}(P_{i|\mathcal{S}_{ij}}(x_i|\mathcal{S}_{ij}), P_{j|\mathcal{S}_{ij}}(x_j|\mathcal{S}_{ij})), \quad (24)$$

The conditioning sets \mathcal{S}_{ij} are determined by the specific vine structure, and are a possibly empty subset of dimensions excluding i, j . Each of the $d(d - 1)/2$ copulas in the decomposition takes two arguments $P_{i|\mathcal{S}_{ij}}(x_{i|\mathcal{S}_{ij}}), P_{j|\mathcal{S}_{ij}}(x_{j|\mathcal{S}_{ij}})$ as inputs, both marginally uniform in $[0, 1]$, and outputs their bivariate copula density. Based on the modelling assumption chosen for those bivariate densities $c_{i,j}$, we follow our previous results to deduce separate equivalences for the final vine model.

Part 1: Simplified vine with bivariate Gaussian copulas. When using bivariate copula building blocks, the vine equation becomes:

$$c(u_1, \dots, u_d) = \prod_{i \neq j}^{d(d-1)/2} \frac{\mathcal{N}_2(\Phi^{-1}(P_{i|\mathcal{S}_{ij}}(x_{i|\mathcal{S}_{ij}})), \Phi^{-1}(P_{j|\mathcal{S}_{ij}}(x_{j|\mathcal{S}_{ij}})); \mathbf{0}, \Sigma)}{\mathcal{N}(\Phi^{-1}(P_{i|\mathcal{S}_{ij}}(x_{i|\mathcal{S}_{ij}})); 0, 1) \cdot \mathcal{N}(\Phi^{-1}(P_{j|\mathcal{S}_{ij}}(x_{j|\mathcal{S}_{ij}})); 0, 1)}. \quad (25)$$

As $P_{i|\mathcal{S}_{ij}}(x_{i|\mathcal{S}_{ij}}), P_{j|\mathcal{S}_{ij}}(x_{j|\mathcal{S}_{ij}})$ are marginally uniform, by the probability integral transformation, this expression is a ratio of two bivariate Gaussians with means $\mathbf{0}$ and covariances Σ, I_2 for the numerator and denominator respectively. From Proposition 4.1, it follows that each term in the vine decomposition is a QDA decision boundary, giving the result.

Part 2: Simplified vines with Gaussian KDE bivariate copulas. Similarly, each bivariate KDE copula is a ratio between a mixture of bivariate kernels and a bivariate standard Gaussian, with some inputs on the \mathbb{R}^2 Gaussian latent space that are marginally Gaussian due to the probability integral transformation. Due to the form of a Gaussian KDE copula, Proposition 4.2 applies to each of the $d(d - 1)/2$ bivariate copulas involved in the vine decomposition (up to replacing the observations $\{(Z_1^k, \dots, Z_d^k)\}_{k=1}^{T_p}$ by new $Z_1^k := \Phi^{-1} \circ P_{i|\mathcal{S}_{ij}}(x_{i|\mathcal{S}_{ij}})$ which include the conditioning implied by vines), meaning that the simplified vine density is then composed as a product of $d(d - 1)/2$ bivariate classifiers, each being an average of T_p decision boundaries obtained by QDA on a 2-dimensional Gaussian problem. \square

A.7 Equivalences for parametric copulas in terms of classifiers

We can use our framework to obtain the classifiers implied by known copulas. Indeed,

$$c(\mathbf{P}(\mathbf{x})) = \frac{h(\mathbf{x})}{1 - h(\mathbf{x})} \iff h(\mathbf{x}) = \frac{c(\mathbf{P}(\mathbf{x}))}{1 + c(\mathbf{P}(\mathbf{x}))}$$

which we use to empirically illustrate the classifiers implied by various copula families by using their densities.

Figure 4 shows six different bivariate copulas and their classifiers on the copula $[0, 1]^2$ space as well as the Gaussian space via a dimension-wise inverse Gaussian transform. A notable benefit of representing copulas by their classifier counterparts is an improved clarity of the shape; the LL values $\log c(U, V)$ in the plots of all but the Frank copula explode at the corners which makes them hard to evaluate, whereas the classifier versions $h(u, v)$ depict clear shapes in accordance with those of copulas. This can be seen on the Gaussian space plots, where the classifier identifies the diagonal as being high-likelihood regions for the dependent data and places increasingly high probability on the diagonal as the distance from the centre (and so from the independent factorised density) increases. The classifier's increasingly high probability is expected, as for instance, the Gaussian copula density is unbounded in the corners on the diagonal. Using our classifier interpretation also exemplifies the theoretical results from Section 4. Further, the similarity of the classifier to the shape of the copula is expected as the copula LL can be understood as a difference of the LL of the numerator and denominator of a copula i.e. $\log c(P_1(c_1), P_2(x_2)) = \log p(x_1, x_1) - \log(p_1(x_1) \cdot p_2(x_2))$, which is what the classifier is trained to distinguish.

B COPULA MODELS

Copulas are widely used in statistics and machine learning for modelling joint probability distributions. They enable the construction of joint densities through a two-step estimation process. First, the marginal densities are estimated as if they were independent. Then, the copula is modelled to capture the dependence structure between the variables. This approach is grounded in Sklar's theorem:

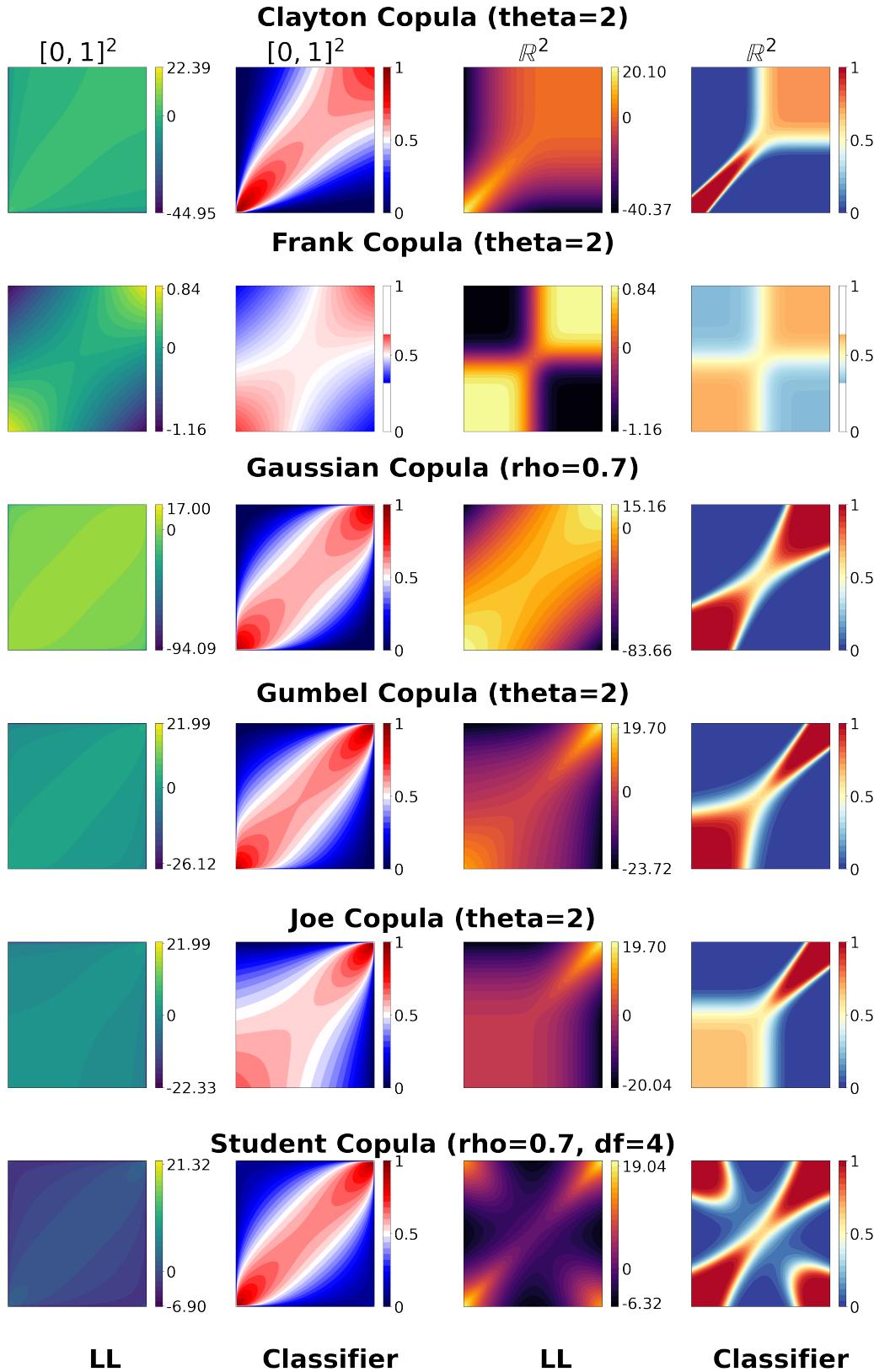


Figure 4: Example of six parametric bivariate copulas and their equivalent classifiers $h^*(\mathbf{x})$. The first two columns show the copula log-likelihoods (LL) and their implied classifiers on the copula $[0, 1]^2$ scale, while the last two columns show the same quantities but on the Gaussian \mathbb{R}^2 scale.

Theorem B.1 (Sklar (1959)). *Let \mathbf{P} be a d -dimensional distribution function with continuous marginal distributions P_1, P_2, \dots, P_d . Then there exists a copula distribution \mathbf{C} such that for all $\mathbf{x} = (x_1, x_2, \dots, x_d) \in \mathbb{R}^d$:*

$$\mathbf{P}(x_1, \dots, x_d) = \mathbf{C}(P_1(x_1), \dots, P_d(x_d)). \quad (26)$$

And if a probability density function (pdf) is available:

$$\mathbf{p}(x_1, \dots, x_d) = p_1(x_1) \cdot \dots \cdot p_d(x_d) \cdot \mathbf{c}(P_1(x_1), \dots, P_d(x_d)), \quad (27)$$

where $p_1(x_1), \dots, p_d(x_d)$ are the marginal pdfs, and $\mathbf{c}(P_1(x_1), \dots, P_d(x_d))$ is the copula pdf.

If the marginal distributions are continuous, the copula is unique. This allows the estimation of the joint distribution $p(\mathbf{x})$ to be broken down into two steps: first, estimating the marginal distributions $\{p_i\}_{i=1}^d$, and then modeling the copula $c(u_1, \dots, u_d)$, where $u_i := P_i(x_i)$ for $i \in \{1, \dots, d\}$, are the images of the x_i under the CDF of each dimension. By applying CDF transformations to the marginals, the copula becomes agnostic to differences between dimensions, such as axis scaling, and focuses solely on capturing the dependence structure among them.

B.1 Gaussian copula

A widely used parametric copula model is the *Gaussian copula*, which assumes that the dependence between dimensions follows that of a multivariate Gaussian distribution with mean $\mathbf{0}$ and covariance matrix Σ . It transforms uniform marginals into Gaussian variables and then models their dependence using the Gaussian distribution. Therefore, the diagonal of Σ only has ones by construction as the uniform variables of the copula are transformed with an inverse standard Gaussian CDF. The copula density is given by:

$$c(u_1, \dots, u_d) = \frac{\mathcal{N}_d(\Phi^{-1}(u_1), \dots, \Phi^{-1}(u_d); \mathbf{0}, \Sigma)}{\prod_{i=1}^d \mathcal{N}(\Phi^{-1}(u_i); 0, 1)}.$$

Here, Φ^{-1} is the inverse cumulative distribution function (CDF) of the standard normal distribution, and \mathcal{N}_d denotes the multivariate normal density. The parameters of the Gaussian copula are the off-diagonal entries of the covariance matrix Σ , which capture the dependence between dimensions. For $d = 2$, there is a single parameter to estimate, representing the correlation between the two variables.

B.2 Vine copulas

A *vine copula* is a flexible method for modelling dependencies in high-dimensional data by decomposing a d -dimensional copula into a sequence of $d(d - 1)/2$ bivariate copulas through structured conditioning (Bedford and Cooke, 2001). This allows for efficient estimation by modelling pairs of variables instead of the full joint distribution (Nagler and Czado, 2016). The decomposition is based on the following identity, which follows from Sklar's theorem:

$$p_{a|b}(x_a|x_b) = c_{a,b}(P_a(x_a), P_b(x_b)) \cdot p_a(x_a) \quad (28)$$

where a and b are subsets of the dimensions $\{1, \dots, d\}$. Vine copulas rely on a conditional factorization $\mathbf{p}(x_1, \dots, x_d) = \prod_{i=1}^d p_{i|< i}(x_i|x_{<i})$, repeatedly applying (28) to split the joint density into d marginal densities and $\frac{d(d-1)}{2}$ bivariate pair copulas.

A full vine copula model for the joint copula density can be written as:

$$\mathbf{c}(u_1, \dots, u_d) = \prod_{i \neq j}^{d(d-1)/2} c_{ij}(P_{i|\mathcal{S}_{ij}}(x_i|\mathcal{S}_{ij}), P_{j|\mathcal{S}_{ij}}(x_j|\mathcal{S}_{ij})|\mathcal{S}_{ij}), \quad (29)$$

where each pair copula depends on the conditional distributions $P_{i|\mathcal{S}_{ij}}$ and $P_{j|\mathcal{S}_{ij}}$, and \mathcal{S}_{ij} is a subset of dimensions excluding i and j . The set \mathcal{S} conditions the copulas to capture higher-order dependencies.

In a simplified vine copula, the conditioning on \mathcal{S} is ignored, leading to a model that is easier to work with:

$$\mathbf{c}(u_1, \dots, u_d) = \prod_{i \neq j}^{d(d-1)/2} c_{ij}(P_{i|\mathcal{S}_{ij}}(x_i|\mathcal{S}_{ij}), P_{j|\mathcal{S}_{ij}}(x_j|\mathcal{S}_{ij})), \quad (30)$$

where the pair copulas are now modelled as unconditional bivariate copulas, simplifying the dependency structure. For instance, in the case of a 3-dimensional random vector (U, V, W) , the joint copula density can be factorized as:

$$c_{U,V,W}(u, v, w) = c_{U,V}(u, v) \cdot c_{V,W}(v, w) \cdot c_{U,W|V}\left(C_{U|V}(u|v), C_{W|V}(w|v) \middle| v\right), \quad (31)$$

where $c_{U,V}(u, v)$ and $c_{V,W}(v, w)$ are the bivariate copulas for pairs (U, V) and (V, W) , and $c_{U,W|V}$ captures the conditional dependence between U and W given V . The conditional copula depends on the conditional CDFs $C_{U|V}$ and $C_{W|V}$, making the model more complex but allowing greater flexibility in capturing dependencies. With a simplified vine, which assumes that the pairwise copulas do not change with conditioning, the factorization becomes:

$$c_{U,V,W}(u, v, w) = c_{U,V}(u, v) \cdot c_{V,W}(v, w) \cdot c_{U,W|V}(C_{U|V}(u|v), C_{W|V}(w|v)). \quad (32)$$

C PRACTICAL DETAILS

Here, we provide more details on running experiments and using the ratio copula in practice. We show our algorithm for fitting the ratio copula model in Algorithm 1, where orange parts are to be included optionally to produce a guided Gaussian ratio copula model instead of a simple ratio copula.

Algorithm 1 Training a Ratio Copula

Optionally, with **Gaussian-Guided** ratio copula.

Require: Dataset $\mathcal{D}_p \sim p(\mathbf{x})$ and parameterised ratio copula $c^{\text{ratio}}(\mathbf{z}; \boldsymbol{\theta}) : \mathbb{R}^d \mapsto \mathbb{R}$;

1. Dataset $\mathcal{D}_{\tilde{p}} = \{\Phi^{-1} \circ \mathcal{P}(\mathcal{D}_p)\}$ on Gaussian scale;

Estimate sample covariance $\Sigma \leftarrow \text{Cov}(\mathcal{D}_{\tilde{p}})$;

- 2.

While $\boldsymbol{\theta}$ not converged **do**:

- Sample $\mathbf{z}_{\tilde{p}} \in \mathcal{D}_{\tilde{p}}$ and $\mathbf{z}_q \sim \mathcal{N}_d(\cdot; \mathbf{0}, \mathbf{I}_d)$;

Instead, sample $\mathbf{z}_{\tilde{p}} \in \mathcal{D}_{\tilde{p}}$ and $\mathbf{z}_{\tilde{q}} \sim \mathcal{N}_d(\cdot; \mathbf{0}, \Sigma)$;

- Compute $c^{\text{ratio}}(\mathbf{z}_{\tilde{p}}; \boldsymbol{\theta})$ and $c^{\text{ratio}}(\mathbf{z}_{\tilde{q}}; \boldsymbol{\theta})$;
- Calculate loss $\mathcal{L}(\boldsymbol{\theta})$ using Equation (12);
- Update parameters $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})$;

Return: Ratio copula model $c^{\text{ratio}}(\mathbf{z}; \hat{\boldsymbol{\theta}})$.

Return: Gaussian-Guided ratio copula model $c^{\text{GG}}(\mathbf{z}; \hat{\boldsymbol{\theta}}) := c^{\text{ratio}}(\mathbf{z}; \hat{\boldsymbol{\theta}}) \cdot \frac{\mathcal{N}_d(\mathbf{z} | \mathbf{0}, \Sigma)}{\mathcal{N}_d(\mathbf{z} | \mathbf{0}, \mathbf{I}_d)}$.

Preprocessing: For each experiment, we start with data on \mathcal{X} and need to apply an estimate of the CDFs to obtain data on a copula $[0, 1]^d$ scale. We use the empirical CDF as a consistent approximator of the true CDF for all experiments. While this leads to an approximate version of samples on $[0, 1]^d$, we treat these samples as samples from the true copula density, using the same empirical CDF per dataset across all methods fitted to make results comparable. Since only the copula LL matters for showing a copula model is performant, using the same CDF across methods ensures that the comparison is fair and focuses on the modelling capabilities of each model, not on the CDF approximation.

Hyperparameters and network architectures: In our experiments, for the simplified vine copula, we use a TLL constant copula as the bivariate building block, corresponding to a bivariate KDE copula. The TLL constant copula requires a bandwidth b for the kernel. We followed standard literature on KDE estimation by choosing Silverman’s rule of thumb for the form of the bandwidth (Silverman, 2018), based on the sample size n in the training data, and the dimension of the data d :

$$b = \left(\frac{n(d+2)}{4} \right)^{(-1/(d+4))}. \quad (33)$$

However, in the Digits experiment, this choice resulted in numeric overflow in the LL evaluation (likely due to the low number of sampled compared to the dimension), leading us to choose the bandwidth b with a grid search using cross-validation on training data. Further, simplified vine copulas can be *truncated* at a given level $t \leq d$, meaning that the number of bivariate copulas is limited at the first $t(t-1)/2$ such copulas in the decomposition. This is equivalent to setting all bivariate copulas past a certain level of conditioning as independent copulas (*i.e* equal to 1). This is a common approach to reduce the computational burden of fitting a vine copula (Czado and Nagler, 2022). We use a full set of $d(d-1)/2$ bivariate copulas for the Digits experiment, but use a truncation level of 50 for the MNIST experiment, in order to keep computational times of the vine copula similar to other models.

For the IGC, we follow the experimental setup of Janke et al. (2021) in their original paper. We use a fully connected MLP, with an initial layer taking as input Gaussian noise, with dimension $3d$, followed by two fully connected layers with width 100, with a final connection from the second fully connected layer to the d -dimensional output. This architecture is used for all experiments with the exception of MNIST where a third fully connected hidden layer was used. We show the parameter number per experiment in Table 2.

Experiment Name	Number of Parameters
2D Image	11,002
Digits	35,864
MNIST	324,584

Table 2: Number of Parameters in the IGC copula for Different Experiments.

Finally, our ratio copula model follows a similarly simple architecture to the IGC, to show its general applicability. We specifically chose architectures with a similar parameter count to make comparisons fair. In the Digits and MNIST examples, we use convolutional layers following a standard classifier architecture for image data. We provide a table describing our ratio copula architecture in Tables 3, 4, 5, 6, 7, and 8. We use a different number of independent noise samples in the optimisation procedure of Loss (12) based on the complexity of the task and the desired trade-off between computational time and statistical efficiency. In the 2D parametric copula example, we use the whole 5000 samples and 50,000 noise samples (but these are fixed across epochs). In the 2D image experiment, we use 10,000 true samples and 100,000 noise samples per epoch. In the Digits experiment, we use the whole 898 training samples and use 89,800 noise samples per batch. In the MNIST example, we use the whole data of size 30,000 and 300,000 independent samples.

Layer (type)	Output Shape	Param #
Linear	[-1, 1, 100]	300
ReLU + Residual	[-1, 1, 100]	0
Linear	[-1, 1, 100]	10,100
ReLU + Residual	[-1, 1, 100]	0
Linear	[-1, 1, 100]	10,100
ReLU + Residual	[-1, 1, 1]	0
Linear	[-1, 1, 1]	101
Total params:		20,601

Table 3: Summary of the ratio copula architecture used in the 2D parametric copula examples. Here, a simple ratio copula model is used.

Layer (type)	Output Shape	Param #
Linear	[-1, 1, 100]	300
<i>ReLU</i> + Residual	[-1, 1, 100]	0
Linear	[-1, 1, 100]	10,100
<i>ReLU</i> + Residual	[-1, 1, 100]	0
Linear	[-1, 1, 100]	10,100
<i>ReLU</i> + Residual	[-1, 1, 100]	0
Linear	[-1, 1, 100]	10,100
<i>ReLU</i> + Residual	[-1, 1, 100]	0
Linear	[-1, 1, 100]	10,100
<i>ReLU</i> + Residual	[-1, 1, 100]	0
Linear	[-1, 1, 100]	10,100
<i>ReLU</i> + Residual	[-1, 1, 100]	0
Linear	[-1, 1, 1]	101
Total params:	50,901	

Table 4: Summary of the ratio copula architecture used for the 2D image data. Here, a simple ratio copula model is used.

Layer (type)	Output Shape	Param #
Conv2d	[-1, 64, 4, 4]	640
LeakyReLU	[-1, 64, 4, 4]	0
Conv2d	[-1, 64, 2, 2]	36,928
LeakyReLU	[-1, 64, 2, 2]	0
Flatten	[-1, 256]	0
Linear	[-1, 1]	257
Total params:	37,825	

Table 5: Summary of the ratio copula architecture used in Digits experiments. We use a GG+ratio model for Digits.

Layer (type)	Output Shape	Param #
Conv2d	[-1, 64, 4, 4]	640
LeakyReLU	[-1, 64, 4, 4]	0
Conv2d	[-1, 64, 2, 2]	36,928
LeakyReLU	[-1, 64, 2, 2]	0
Flatten	[-1, 256]	0
Linear	[-1, 256]	65,792
LeakyReLU	[-1, 256]	0
Linear	[-1, 1]	257
Total params:	103,617	

Table 6: Summary of the GG+Ratio *large* copula architecture used in Digits experiments.

Layer (type)	Output Shape	Param #
Conv2d	[1, 64, 14, 14]	640
LeakyReLU	[1, 64, 14, 14]	0
Conv2d	[1, 64, 7, 7]	36,928
LeakyReLU	[1, 64, 7, 7]	0
Flatten	[1, 3136]	0
Linear	[1, 64]	200,768
LeakyReLU	[1, 64]	0
Linear	[1, 32]	2,080
LeakyReLU	[1, 32]	0
Linear	[1, 1]	33
Total params:	240,449	

Table 7: Summary of the ratio copula architecture used in MNIST experiments. We use a GG+ratio model for MNIST.

Layer (type)	Output Shape	Param #
Conv2d	[1, 64, 14, 14]	640
LeakyReLU	[1, 64, 14, 14]	0
Conv2d	[1, 64, 7, 7]	36,928
LeakyReLU	[1, 64, 7, 7]	0
Flatten	[1, 3136]	0
Linear	[1, 128]	401,536
LeakyReLU	[1, 128]	0
Linear	[1, 64]	8,256
LeakyReLU	[1, 64]	0
Linear	[1, 64]	4,160
Linear	[1, 32]	2,080
Linear	[1, 1]	33
Total params:	453,633	

Table 8: Summary of the GG+ratio *large* model architecture used in MNIST experiments.

Optimisation: For the IGC and ratio copula models, we employ the Adam optimiser (Diederik, 2014) in Pytorch with the base settings, to demonstrate the ease of use of our model, as in Janke et al. (2021). For the 2D parametric copula experiments, we optimise these models for 500 steps, for the 2D image experiments we use 100,000 optimisation steps, for the Digits dataset we use 500 steps and we use 1000 steps for MNIST to reflect the complexity of the task. For the ratio copula, at each epoch, we resample a new set of independent noise (either from $\mathcal{N}(\cdot; \mathbf{0}, \mathbf{I}_d)$ or from a correlated Gaussian $\mathcal{N}(\cdot; \mathbf{0}, \Sigma)$ in the Gaussian guided ratio copula). Therefore, for the ratio copula, following the analysis in Theorem 3.1 and Gutmann and Hyvärinen (2012), there is a trade-off available for our method between computation time and statistical performance; as we optimise the model for more epochs and sample more independent noise, the better our estimate of the true ratio copula. Therefore, while we limit the number of epochs to the same number as the IGC, our method could benefit from longer training times.

Diagnostics for overfitting Other than using Lemma 3.2, an easily available diagnostic is to look at the loss during training. Since the copula naturally overlaps independent with dependent samples, we should always observe a positive loss. If it is too close to 0, then the classifier is overfitting and the two densities are too far apart (Rhodes et al., 2020). Furthermore, using a normalising constant estimator also provides a diagnostic of fit: copula densities are already normalised by definition, hence if correctly estimated, the normalising constant will be 1. If any of these symptoms arise, it signals we should employ a better DRE method for estimating the ratio copula. For instance, switching from a ratio copula to a Gaussian-Guided ratio copula.

Sampling from the ratio copula: As our ratio copula model has an analytical density, we employ Markov chain Monte Carlo methods to obtain samples. Specifically, we use the Hamiltonian Monte Carlo scheme (Neal, 2011), from the implementation in the pyhmc python package (McGibbon, 2015). For each sample, we initiate the chain at a random sample from a standard independent Gaussian on the latent Gaussian space, and implement 100 gradient-informed MCMC steps to ensure the chain has reached the main mass of the density. We then select the next step after that as the sample from our model. As in *e.g.* KDE or Gaussian copula models, this sampling is done from p on the latent Gaussian space, meaning that we multiply our final ratio copula by $\mathcal{N}(\cdot; \mathbf{0}, \mathbf{I}_d)$, and target the resulting density with our MCMC sampling scheme. The samples obtained by the MCMC scheme are consequently on a Gaussian scale, meaning we apply a Gaussian CDF to map them to the copula $[0, 1]^d$ space.

Metrics: We use the LL, as measured by the output of our ratio model’s log-density on held-out test data. Similarly, we use the python implementation of Vatter and Nagler (2022) of simplified vine copula models as well as bivariate copulas to output LL values. For the Wasserstein 2 metric, we make use of the implementation of Flamary et al. (2021). For all experiments, we measure the W2 using 500 samples from models, except for MNIST where we use 25 samples per model.

Computation: All experiments were run on a single Intel Xeon Platinum 8260 (Cascade Lake) CPU. We show a table of runtimes for a single run of each model on experiments in Table 9. In the 2D image experiment of Figure 2 of the main text (2D IMG), vine denotes a TLL quadratic copula, and in the 2D parametric copulas (2D copulas) the vine denotes TLL copula runtimes (constant, linear and quadratic TLL take about the same time in those settings). For the 2D copulas, the first time for the ratio copula is a simple ratio copula with an MLP model (2:30 mins), while the second time (20s) is for fitting a logistic classifier as the ratio copula model. With 2 runs for 2D IMG, $25 \cdot 3$ runs for 2D copulas, and 10 runs for Digits and MNIST, our total computation time for the experiments is 65 hours on a single CPU.

Model\Dataset	2D IMG	2D copulas	Digits	MNIST
Ratio	4h 30 mins	2:30 mins/20 s	10 mins	105 mins
IGC	3h	-	2 mins	10 mins
Vine	5 mins *	15 s *	4 mins	140 mins

Table 9: Time taken for fitting and sampling for a single run of different models on datasets from Section 5. The asterisks (*) signify that the TLL bivariate copula from the vine package was used instead of a multivariate vine.

C.1 Additional experiments on losses for tail modelling

In Menon and Ong (2016), authors propose losses that emphasize high density-ratio values. In copulas, under our ratio copula perspective, this corresponds to extreme tail dependence.

We exemplify the effect of different losses which place more or less emphasis on the tails for estimating ratio copulas. The Noise Contrastive Estimation (NCE) loss of Gutmann and Hyvärinen (2012) we use emphasizes training according to a weight in terms of the copula value $c(\mathcal{P}(\mathbf{x}))$ as $w(c) = 1/(c^2 + c)$, while the exponential loss has $w(c) = 1/c^{(3/2)}$ and the least squares importance fitting (LSIF) loss (Kanamori et al., 2009) has $w(c) = 1$, both latter losses placing a greater focus on tails. We illustrate this behaviour on a two-dimensional example shown in Figure 5. For observed data on the first row, we fit the same ratio copula model using different loss functions, showing the resulting classifiers and log-likelihoods on the left and right columns respectively. The exponential loss better captures tail events (boxed in blue), with sharper densities and better separation of observation "rays", while NCE conflates them. The higher LL values given to data-regions highlight the advantage of the exponential loss. The LSIF loss is harder to optimize which makes its density more diffuse. We hereby demonstrate another advantage of connecting copulas to DRE, allowing practitioners to make use of already developed tools, in this instance to improve tail modelling.

C.2 Experiment on bivariate copulas from pictures in Figure 2

The data to fit was obtained by normalising the pixel values of a monochromatic picture, to sum up to 1 across the image. We then sample with replacement following the probabilities given by the brightness of each pixel and add minute Gaussian noise to dequantize samples. This constitutes our dataset on \mathbb{R}^2 , for which we estimate marginals (with empirical CDFs) and use them to finally obtain copula samples (the plots from the first column, with white dots for each sample) on $[0, 1]^2$ used to fit copula models. Notice how the letters of the bottom row have different shades of white: this is because of the transformation to the copula space. Our model recovers these nuances, see the horizontal bars of each "T" which have a larger density than the vertical bars. In Figure 2, we display on $[0, 1]^2$, going from left to right: the observed dataset, the LL estimated by the ratio copula, samples from the IGC, and the LL estimated from the TLL.

C.3 High dimensional experiments

The two high-dimensional datasets used are the Digits dataset (Alimoglu and Alpaydin, 1997) and MNIST (Le-Cun, 1998). They both show handwritten digits in $8 \cdot 8$ or $28 \cdot 28$ format. As both these datasets represent complex dependencies, as measured using Lemma 3.2 from the main text, we use a GG+ratio copula model. We show results for a simpler (GG+Ratio) and larger version (GG+Ratio *large* see Tables 6 and 8), to showcase the potential for scalability of ratio copulas. In Figure 6, we show random samples from the simpler ratio copula, the simplified vine copula, the IGC and the Gaussian copula as well as true test data. We show samples from the same copula classes fitted on the MNIST data in Figure 7. Samples shown are obtained from the copula model and then transformed back to the data space using the empirical CDF as described in the preprocessing paragraph.

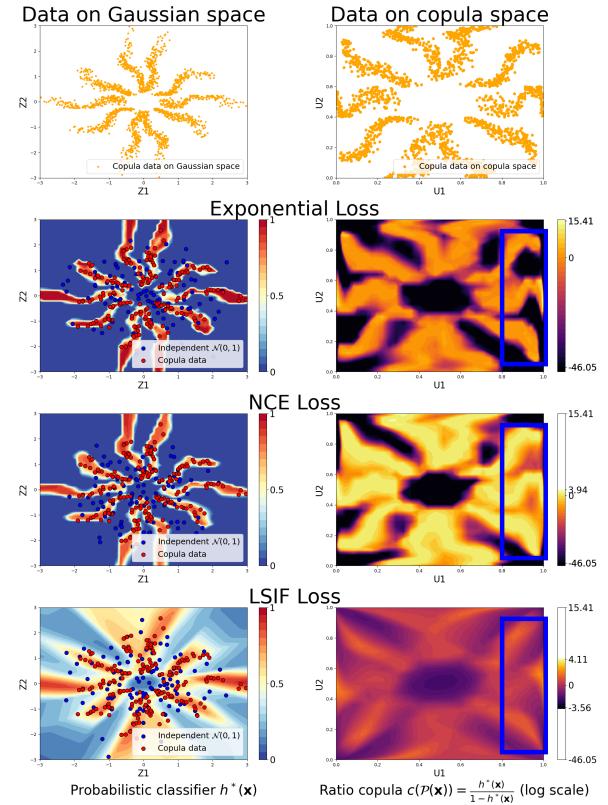


Figure 5: Same Ratio copula model trained with losses that emphasise tail behaviours. The exponential loss obtains a better separation of likelihood regions in the tails (boxed in blue) while the NCE loss conflates tail events and the LSIF loss is too diffuse.

We note that the samples shown are random across tiles, meaning that we are not showing how well a model imitates a given target digit. We are exploring the learned distribution implied by the model after training on the data samples, doing so by showing a collection of uncurated random samples from any given model.

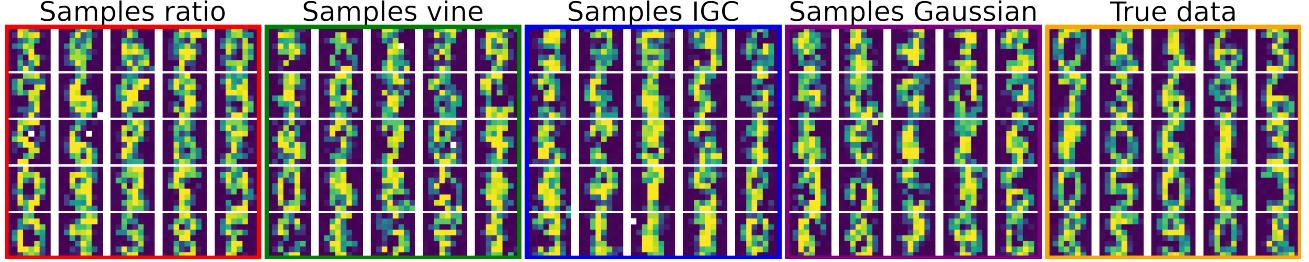


Figure 6: Random samples from all models from the Digits experiment. Samples are obtained by random sampling from the copula models and transformed back to the data space. The copula samples shown are not related to the true data, meaning the digits shown are not meant to match between models and true data.

For the Digits dataset, all models perform well. Arguably, the Gaussian copula sometimes fails to correctly capture the dependence to the same level as what is present in the true data. Other models perform similarly, with differences being hard to notice. The vine copula does have some samples where the dependence on the image follows certain *lines* along pixels, likely due to the bivariate decomposition used in vines causing pixels nearby to be correlated by ignoring longer dependencies in space due to the simplifying assumption. In particular, the ratio copula produces samples with high fidelity to the true data, consistently across all depicted samples.

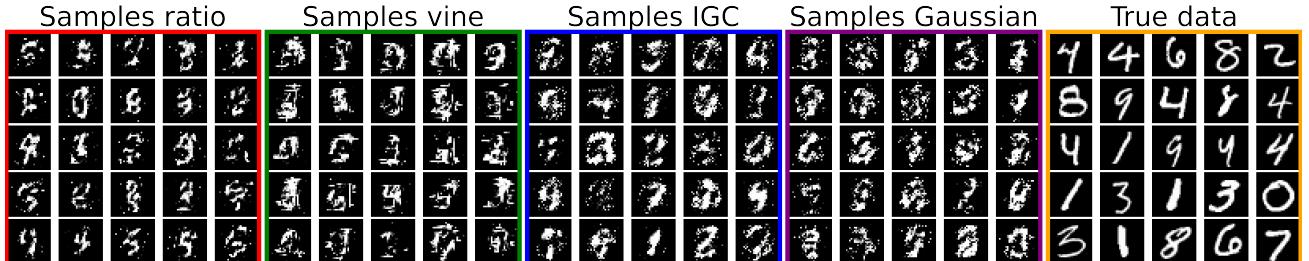


Figure 7: Random samples from all models from the MNIST experiment. Samples are obtained randomly from the copula models and transformed back to the data space. The Ratio copula obtains visually compelling samples while other copula models are more dispersed.

For MNIST, the IGC and Gaussian copula samples are more blurry and imprecise, while the vine copula has samples with distinct *lines* forming the picture. The reason for the blurriness of the Gaussian copula is believed to be its less precise dependence structure, being strictly diagonal for the Gaussian. The reason for the blurriness in the IGC is not clear, but we believe the non-convexity of the MMD as a loss might make it harder for the model to recover the optimal dependence structure, instead settling at a local optimum. The reason for the apparent lines in the vine is likely associated with the recursive decomposition of a simplified vine copula, where bivariate copulas propagate dependence in a chain-like fashion, from neighbour to neighbour. Given that exact model selection is extremely hard for simplified vines, this dependence structure will likely be only approximately correct, causing some parts of the numbers to be misrepresented. This might be further amplified by the simplifying assumption, causing the model to not necessarily capture all the dependence present in the data. Again, the ratio copula consistently produces high quality samples, on par with the best of competing methods.