Copula-Based Dependence Modeling with Deep Learning:

Neural Network-Based Estimation Approaches, Checkerboard Smoothing and Grid Frequency Modeling

Zur Erlangung des akademischen Grades eines
Doktors der Wirtschaftswissenschaften

(Dr. rer. pol.)

von der KIT-Fakultät für Wirtschaftswissenschaften
des Karlsruher Instituts für Technologie (KIT)

genehmigte

DISSERTATION

von

M.Sc. M.Sc. Bolin Liu

Tag der mündlichen Prüfung: 25.03.2026

Referent: Prof. Dr. Oliver Grothe

Korreferent: Prof. Dr. Maximilian Coblenz

Karlsruhe, 2026

Danksagung

Wissenschaft entsteht nie im Stillen. Auch diese Arbeit verdankt sich dem Vertrauen, der Unterstützung und der Begleitung vieler Menschen. Ihnen allen möchte ich an dieser Stelle herzlich danken. Mein besonderer Dank gilt Prof. Dr. Oliver Grothe. Ich danke Ihnen herzlich für die Möglichkeit zur Promotion und dafür, dass ich in dieser Zeit viel bei Ihnen lernen durfte. Besonders danke ich Ihnen für Ihre Impulse zum wissenschaftlichen Denken und Arbeiten, dafür, dass ich lernen durfte, Fragestellungen auf ihren Kern zurückzuführen und klar und effizient zu kommunizieren, sowie für die Freiheit, eigene Forschungsfragen zu verfolgen. Vor allem danke ich Ihnen für Ihre Unterstützung und dafür, dass Sie immer ein offenes Ohr hatten. Ebenso danke ich Prof. Dr. Maximilian Coblenz sowie der Carl-Zeiss-Stiftung für die Förderung im Rahmen des Projekts Copula and Machine Learning. Die Möglichkeit, fachliche Fragen mit Ihnen zu diskutieren und Gedanken gemeinsam weiterzuentwickeln, war für mich eine große Bereicherung. Unsere Gespräche haben diese Arbeit in vielerlei Hinsicht mitgeprägt. Mein herzlicher Dank gilt außerdem meiner Prüferin, Professorin Dr. Melanie Schienle, sowie dem Prüfungsvorsitzenden, Professor Dr. Martin Klarmann, für ihre wertvolle Zeit und ihr aufmerksames Interesse an meiner Arbeit. Ich möchte mich auch bei meinen Studierenden bedanken und insbesondere bei meinem Hiwi Linus Nuding, der mich stets engagiert und zuverlässig unterstützt hat. Ein besonderer Dank gilt zudem meinen Kolleginnen und Kollegen am Lehrstuhl für den spannenden Austausch, die vielen anregenden Gespräche und die angenehme Zusammenarbeit. Besonders danken möchte ich auch unserer Sekretärin Marion Rihm für die reibungslose und stets angenehme Zusammenarbeit bei allen Aufgaben rund um den Lehrstuhl. Von Herzen danke ich meiner Familie und meinen Freunden, die mich auf diesem Weg begleitet haben. Besonders meinen Eltern danke ich dafür, dass sie mir das Studium ermöglicht haben, und für ihre beständige Unterstützung, ihr Vertrauen und ihren Rückhalt auf dem gesamten Weg bis hierhin. Maxim Glyzhev danke ich für viele spannende Diskussionen und dafür, die Faszination für wissenschaftliche Fragen zu teilen. Mein persönlichster Dank gilt meiner Freundin Karin. Danke, dass du in dieser Zeit immer an meiner Seite warst. Dein Rückhalt, deine Geduld und dein Vertrauen haben mich durch viele Phasen dieser Dissertation getragen. Dass du immer an mich geglaubt hast, hat mir mehr bedeutet, als Worte ausdrücken können.

Abstract

Copulas and deep learning offer complementary strengths for probabilistic modeling: the former provides a rigorous separation of dependence and marginals, the latter enables flexible, data-driven learning of complex patterns. This thesis develops methods at their intersection, combining neural network-based copula estimation with dependence-aware predictive modeling.

Two approaches to nonparametric copula density estimation are introduced. A normalizing flow architecture learns copula-generating transformations, enabling flexible modeling of asymmetric dependence and tail behavior. An alternative estimator based on separable perturbations of the independence copula enforces marginal uniformity by construction and performs well for copulas with localized dependence patterns. Both methods are theoretically grounded, and empirical comparisons demonstrate improvements over existing nonparametric approaches.

A further contribution addresses the transition from discrete to continuous copula representations. A smoothing framework approximates checkerboard copulas by continuous densities via constrained optimization. Existence, consistency, and error bounds are proven, with applications to credit rating transitions and demographic data.

The final part of the thesis addresses the practical problem of predicting power grid frequency through two contributions. The first develops a hybrid framework combining Gaussian processes with sequence models for probabilistic short-term forecasts. The second introduces a copula-based error correction approach that captures nonlinear serial dependencies in forecast residuals, yielding improved uncertainty estimates compared to conventional Gaussian models.

Overall, the thesis helps bridge copula methods and deep learning, providing flexible tools for modeling complex dependence in data.

Contents

List of Figures

List of Tables

List of Abbreviations

AC-NFCM	Affine Coupling-based Normalizing Flow Copula Model
ACF	Autocorrelation Function
ALAE	Allocated Loss Adjustment Expenses
BS	Bernstein
CB	Checkerboard
CDF	Cumulative Distribution Function
CRPS	Continuous Ranked Probability Score
GRU	Gated Recurrent Unit
IAE	Integrated Absolute Error
KDE	Kernel Density Estimation
KS	Kolmogorov-Smirnov
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MISE	Mean Integrated Squared Error
MLP	Multilayer Perceptron
NFCM	Normalizing Flow Copula Model
NESP	Neural Estimator via Separable Perturbation
PCA	Principal Component Analysis
PDF	Probability Density Function
PIML	Physics-Informed Machine Learning
QP	Quadratic Programming
RFE	Relative Frobenius Error
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
SGD	Stochastic Gradient Descent
SVD	Singular Value Decomposition
WR	Withdrawn Ratings

1. Introduction

Copula theory and deep learning have developed largely independently, yet both address fundamental challenges in modeling complex multivariate distributions. Building on Sklar's theorem (Sklar, 1959), copulas provide a flexible framework for modeling dependence structures independently of marginal distributions, with applications ranging from financial risk management to hydrological modeling. However, parametric copula families impose restrictive assumptions about the form of dependence, while classical nonparametric estimators may struggle with complex patterns, higher dimensions, or pronounced tail behavior. Deep learning, on the other hand, has revolutionized probabilistic modeling through flexible architectures, such as normalizing flows and sequence models, that learn intricate distributional patterns directly from data. This thesis explores the intersection of these two fields, developing neural network-based methods for copula estimation and applying dependence-aware deep learning models to probabilistic prediction.

The contributions of this thesis build on five papers and are organized around three themes reflected in the subtitle. First, two different neural network architectures for nonparametric copula density estimation are developed: (i) a normalizing-flow approach that models copulas in a data-driven manner via transformations of a base distribution, and (ii) a model based on separable perturbations of the independence copula that performs particularly well for localized dependence patterns. Second, a smoothing framework is proposed that maps discrete checkerboard copulas (e.g., arising from contingency-table data) to continuous densities, thereby bridging checkerboard constructions and smooth copula models. Third, a methodology for probabilistic power-grid frequency modeling is developed. One contribution establishes a flexible forecasting framework based on Gaussian processes and sequence models for short-term frequency dynamics. A further contribution augments this model with copula-based error correction to capture remaining dependencies in the forecast residuals in a data-driven manner. Overall, the emphasis is on learning dependence structures from data without imposing restrictive parametric assumptions.

In the following, a detailed outline of the contributions and the structure of the dissertation is provided. Chapter 3 introduces a nonparametric copula estimator based on normalizing flows. Normalizing flows are a class of generative models that construct complex distributions by composing sequences of invertible, differentiable transformations. The

proposed method builds on the observation that any copula can be represented as the image of an arbitrary base distribution under a suitable copula-generating transformation. Based on this characterization, we develop a modified normalizing flow architecture using affine coupling layers that yields transformations mapping base distributions into the copula space. The resulting model constitutes a fully nonparametric copula estimator capable of representing a wide range of empirical dependence patterns, including asymmetry and strong tail dependence. In simulation studies comparing the method with existing nonparametric approaches such as kernel density estimation, the model performs comparably in simple settings and attains clear improvements in situations involving asymmetric dependence, pronounced tail behavior, or higher dimensions. Among all methods considered, the proposed approach is the only one that accurately captures tail dependence. Two empirical examples from insurance and engineering illustrate the practical use of the method. The chapter is based on joint work with Maximilian Coblenz and Oliver Grothe (Publ. I).

Chapter 4 develops an alternative neural network-based copula density estimator using separable perturbations of independence. The approach builds on a representation that expresses the copula density as the density of the independence copula plus a separable sum, where each term is the product of two univariate functions. This representation has been studied in the literature primarily for analytical purposes and the construction of parametric families, but the question of whether such decompositions can be learned directly from data has remained open. We address this question by developing a neural network architecture that learns the component functions from data while enforcing marginal uniformity by construction. We establish that any square-integrable copula density can be approximated arbitrarily well by such a decomposition, providing a universal approximation result. In simulations, the estimator performs competitively with kernel-based methods in standard settings and demonstrates substantial improvements for copulas exhibiting local structure. An application to vine copula modeling, where the proposed estimator is used for pair-copula construction, illustrates the practical utility of the approach. The chapter is based on joint work with Maximilian Coblenz and Oliver Grothe (Publ. II).

Chapter 5 addresses the smoothing of checkerboard copulas. Checkerboard copulas arise naturally when modeling dependence for discrete or discretized data: they extend the empirical subcopula to a proper copula with piecewise constant density on a rectangular grid. While computationally convenient and nonparametric, their discontinuous density introduces block artifacts and zero regions that complicate downstream tasks requiring smoothness, such as simulation from conditional distributions, interpolation, or fine-resolution reconstruction. We propose a smoothing framework that approximates a given checkerboard copula by a continuous density via constrained L^2 minimization. The

optimization is formulated as a convex quadratic program in the coefficients of a finite basis expansion, where nonnegativity is enforced at collocation points and the mean-zero property of the basis functions automatically guarantees uniform margins. We prove existence and consistency of the resulting estimator and show that the error in cellwise probability masses is bounded by the L^2 density error, even though mass preservation is not explicitly enforced. Different basis families, including Legendre polynomials, trigonometric functions, and B-splines, are compared in a simulation study, demonstrating that basis selection affects local smoothness characteristics while maintaining comparable overall approximation quality. Applications to credit rating transitions and U.S. age-income data show that the method removes artificial discontinuities while accurately maintaining empirical mass distributions. The chapter is based on joint work with Maximilian Coblenz and Oliver Grothe (Publ. III).

Chapter 6 turns to the application of probabilistic modeling to power grid frequency. Grid frequency reflects the balance between power generation and consumption and serves as a key indicator of grid stability. With the increasing share of renewable energy sources, accurate modeling and prediction of grid frequency dynamics has become increasingly important yet challenging. While physics-based models incorporating swing equations and control mechanisms provide a solid foundation, they require detailed knowledge of system parameters that may not be available for all energy systems. We present a purely data-driven approach based on a combination of Gaussian processes and deep learning-based sequence models. Using architectures such as gated recurrent units and transformers, we extract information from static techno-economic features recorded at the beginning of each hour to predict the parameters of a Gaussian process governing the short-term frequency dynamics within that hour. The approach addresses the challenge that external features are typically recorded hourly while frequency dynamics evolve at the scale of seconds. Both for point prediction metrics and probabilistic evaluation measures, our models perform comparably to state-of-the-art physics-informed approaches and outperform various purely data-driven baselines. Moreover, synthetic time series generated by the models successfully reproduce the main statistical characteristics of observed grid frequency data. The chapter is based on joint work with Maximilian Coblenz and Oliver Grothe (Publ. IV).

Chapter 7 extends the grid frequency modeling framework in Chapter 6 by incorporating copula-based dependence structures. Common prediction models in the literature assume Gaussian distributions where either no temporal dependence is modeled or dependence is captured only through linear correlation. We develop a methodology that transforms an existing point predictor into a probabilistic estimator capable of flexibly modeling nonlinear serial dependencies. The construction analyzes forecast errors on historical data by partitioning the feature space into subspaces based on observed error

patterns, then models the error distribution within each subspace using nonparametric copula estimators. This allows us to capture localized error patterns that may be missed in a global analysis. Models corrected using the copula-based error distribution perform better on probabilistic evaluation measures than baselines assuming independence, and our best copula-based model also outperforms Gaussian prediction models that account for dependence only through correlation. Especially, we demonstrate that conventional models significantly underestimate the uncertainty in grid frequency forecasts, while our copula-based predictors provide a more accurate representation of this uncertainty. The code is publicly available. The chapter is based on joint work with Maximilian Coblenz and Oliver Grothe (Publ. V).

As a preliminary foundation for the remainder of the thesis, Chapter 2 reviews the mathematical and methodological background used throughout. Section 2.1 provides a concise overview of copula theory, while Section 2.2 covers the deep learning concepts employed in this work, including normalizing flows and sequence models. To keep the main chapters largely self-contained, Chapters 3 through 7 briefly revisit the most important concepts and notation as needed, motivate their research questions within the respective field, and review the relevant literature. Chapter 8 concludes the thesis. The appendix provides supplementary material organized by chapter, including additional technical details, extended proofs, and further results; the main text includes references to the relevant sections. The author's publications and conference presentations are listed on pp. 176–177 and are referenced within the thesis by "Publ." and "Conf.".

2. Preliminaries on Copulas and Neural Network Models

This chapter provides the mathematical and methodological background for the remainder of the thesis. We begin in Section 2.1 with a brief review of copula theory, which offers a flexible framework for modeling dependence structures separately from marginal distributions. For comprehensive introductions, see Nelsen (2006), Joe (1997), Durante and Sempi (2015), and Hofert et al. (2018); we therefore only provide a concise overview. Section 2.2 then reviews the neural network architectures used throughout this thesis, including feedforward networks, normalizing flows, and sequence models for temporal data. For comprehensive introductions to machine learning and deep learning, we refer to Goodfellow et al. (2016), Bishop (2006), and Murphy (2012).

2.1. Copulas

A d-dimensional copula is defined as a multivariate distribution function on $[0,1]^d$ with uniform margins. Intuitively, it describes the dependence structure of a random vector independently of its marginal behavior. Sklar's theorem (Sklar, 1959) formalizes this separation: for any d-dimensional distribution function F on $\mathbb{R}^d$ with continuous univariate marginal distribution functions $F_1, \dots, F_d$, there exists a unique copula C such that

$$F(x_1, \dots, x_d) = C\big(F_1(x_1), \dots, F_d(x_d)\big), \qquad (x_1, \dots, x_d)^\top \in \mathbb{R}^d. \tag{2.1}$$

Conversely, for any copula C and any collection of univariate distribution functions $F_1, \dots, F_d$, the right-hand side of (2.1) defines a valid d-dimensional distribution function with margins $F_1, \dots, F_d$. The copula C associated with a given F can be constructed explicitly via the inversion method (see Nelsen, 2006, Section 3.1),

$$C(u_1, \dots, u_d) = F\big(F_1^{-1}(u_1), \dots, F_d^{-1}(u_d)\big), \qquad (u_1, \dots, u_d)^\top \in [0,1]^d, \tag{2.2}$$

where F_i^{-1} denotes the quantile function of the marginal distribution F_i, $i = 1, \dots, d$. In Chapter 3, we exploit this representation to estimate copula densities by modeling the

underlying multivariate distribution flexibly via neural networks. If F is absolutely continuous with joint density f, then the margins $F_1, \dots, F_d$ are also absolutely continuous (with marginal densities $f_1, \dots, f_d$), and (2.1) carries over to densities. The copula density c is defined by

$$c(u_1, \dots, u_d) = \frac{\partial^d}{\partial u_1 \cdots \partial u_d} C(u_1, \dots, u_d), \qquad (u_1, \dots, u_d)^\top \in (0,1)^d,$$

and the joint density admits the factorization

$$f(x_1, \dots, x_d) = c\big(F_1(x_1), \dots, F_d(x_d)\big) \prod_{i=1}^{d} f_i(x_i), \qquad (x_1, \dots, x_d)^\top \in \prod_{i=1}^{d} \{x \in \mathbb{R} : f_i(x) > 0\}.$$

Copulas are invariant under strictly increasing transformations: if $g_1, \dots, g_d$ are strictly increasing functions, then the copula of $(g_1(X_1), \dots, g_d(X_d))^\top$ coincides with that of $(X_1, \dots, X_d)^\top$. This follows from (2.2), since strictly increasing transformations preserve the ordering of observations.

Since the copula captures the entire dependence structure, many classical dependence measures can be expressed directly as functionals of the copula. Consequently, these measures are also invariant under strictly increasing transformations.

Two prominent examples are Kendall's τ and Spearman's ρ; for a comprehensive treatment of measures of association in bivariate settings, see Chapter 5 of Nelsen (2006). Let $(X_1, X_2)^\top$ be a bivariate random vector with continuous marginals and copula C. Kendall's τ is defined as the difference between the probabilities of concordance and discordance of two independent copies $(X_1, X_2)^\top$ and $(X_1', X_2')^\top$

$$\tau = \mathbb{P}\big[(X_1 - X_1')(X_2 - X_2') > 0\big] - \mathbb{P}\big[(X_1 - X_1')(X_2 - X_2') < 0\big] = 4 \int_{[0,1]^2} C(u,v)\, \mathrm{d}C(u,v) - 1. \tag{2.3}$$

Spearman's ρ is the Pearson correlation between the transformed variables $U = F_1(X_1)$ and $V = F_2(X_2)$

$$\rho = \operatorname{corr}(U, V) = 12 \int_0^1 \int_0^1 \big(C(u,v) - uv\big)\, \mathrm{d}u\, \mathrm{d}v. \tag{2.4}$$

Beyond such global measures, extremal dependence is captured by tail dependence coefficients. The upper and lower tail dependence coefficients are defined by

$$\lambda_U := \lim_{q \to 1^-} \mathbb{P}\big(U_2 > q \,\big|\, U_1 > q\big), \qquad \lambda_L := \lim_{q \to 0^+} \mathbb{P}\big(U_2 \le q \,\big|\, U_1 \le q\big), \tag{2.5}$$

provided the limits exist, where $(U_1, U_2)^\top \sim C$.

Having established how dependence can be quantified via copulas, we now turn to specific copula structures. The simplest case is the independence copula

$$\Pi(u_1, \dots, u_d) = \prod_{i=1}^{d} u_i, \tag{2.6}$$

which arises when $(X_1, \dots, X_d)^\top$ has independent components; in this case $F(x_1, \dots, x_d) = \prod_{i=1}^{d} F_i(x_i)$, and applying (2.2) yields precisely Π. In the bivariate case, every copula C is bounded by the Fréchet–Hoeffding bounds W and M. For all $u, v \in [0, 1]$,

$$W(u, v) := \max\{u + v - 1, 0\} \;\le\; C(u, v) \;\le\; \min\{u, v\} =: M(u, v).$$

The copula M corresponds to perfect positive dependence (comonotonicity), and W to perfect negative dependence (countermonotonicity). The independence copula Π lies between these extremes. For $d \ge 3$, the lower Fréchet–Hoeffding bound is no longer a copula; see Section 2.10 of Nelsen (2006) for details.

To model dependence structures between these boundary cases, a variety of parametric copula families have been developed; we provide an overview here and refer to Chapter 4 of Joe (2014) for detailed definitions and properties. Elliptical copulas such as the Gaussian and t-copula are obtained by applying the inversion method (2.2) to the multivariate standard normal distribution and the multivariate t-distribution, respectively. Both are parameterized by a correlation matrix; the t-copula additionally depends on the degrees of freedom. Archimedean copulas form another important class. In the bivariate case, they are defined by

$$C(u, v) = \varphi^{[-1]}\big(\varphi(u) + \varphi(v)\big), \tag{2.7}$$

where $\varphi : [0, 1] \to [0, \infty]$ is a generator function satisfying $\varphi(1) = 0$ and $\varphi^{[-1]}$ denotes its pseudo-inverse; see Chapter 4 of Nelsen (2006) for regularity conditions and multivariate extensions. The particular dependence structure is determined by the choice of φ; well-known examples include the Clayton, Gumbel, and Frank copulas.

The parametric families introduced above differ markedly in their tail behavior: the Gaussian copula exhibits no tail dependence, while the t-copula features symmetric tail dependence that increases as the degrees of freedom decrease. Among Archimedean copulas, the Clayton copula exhibits lower but no upper tail dependence, the Gumbel copula upper but no lower tail dependence, and the Frank copula no tail dependence.

While parametric copula families provide flexible models in bivariate settings, high-dimensional applications often require more elaborate constructions to capture complex dependence patterns. Vine copulas are based on the pair copula construction, which de-

composes a multivariate density into a product of bivariate copula densities by successive conditioning (Aas et al., 2009). These bivariate building blocks are arranged in a graph structure called a vine, allowing each pair of variables to be modeled by a different copula family; see Czado (2019) for a comprehensive treatment. Factor copula models take a different approach, assuming that dependence among the observed variables is driven by a small number of common latent factors; conditional on these factors, the observed variables are independent or follow a much simpler dependence structure. This substantially reduces modeling complexity; see Krupskii and Joe (2013) for construction details and further specifications.

Having introduced various copula families, we now discuss how their parameters can be estimated from observed data. Suppose we observe a sample $\mathbf{x}_1, \dots, \mathbf{x}_n$ with $\mathbf{x}_i = (x_{i1}, \dots, x_{id})^\top$. If the marginal distribution functions $F_1, \dots, F_d$ are known, one can transform the data to uniform observations via

$$u_{ij} = F_j(x_{ij}), \qquad i = 1, \dots, n,\ j = 1, \dots, d,$$

and maximize the log-likelihood

$$\ell_n(\theta) = \sum_{i=1}^{n} \log c_\theta\big(u_{i1}, \dots, u_{id}\big) \tag{2.8}$$

over $\theta \in \Theta$, where c_θ denotes the density of a parametric copula family $\{C_\theta : \theta \in \Theta\}$.

In practice, the marginals are typically unknown. A common approach is to work with rank-based pseudo-observations. Let R_{ij} denote the rank of x_{ij} among $x_{1j}, \dots, x_{nj}$ and define

$$\hat{u}_{ij} := \frac{R_{ij}}{n+1}, \qquad i = 1, \dots, n,\ j = 1, \dots, d. \tag{2.9}$$

The pseudo log-likelihood is then

$$\tilde{\ell}_n(\theta) = \sum_{i=1}^{n} \log c_\theta\big(\hat{u}_{i1}, \dots, \hat{u}_{id}\big), \tag{2.10}$$

and the pseudo-maximum likelihood estimator maximizes $\tilde{\ell}_n(\theta)$ over $\theta \in \Theta$. For details, see Chapter 4 of Hofert et al. (2018); asymptotic properties are established in Genest et al. (1995).

When no parametric assumption on the copula is imposed, nonparametric methods offer a flexible alternative. One widely used estimator is the empirical copula, which is

defined as the empirical distribution function of the pseudo-observations

$$\hat{C}_n(u_1, \dots, u_d) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}\{\hat{u}_{i1} \le u_1, \dots, \hat{u}_{id} \le u_d\}, \qquad \mathbf{u} = (u_1, \dots, u_d)^\top \in [0,1]^d. \quad (2.11)$$

It is a consistent estimator of C, and its asymptotic properties follow from the empirical copula process

$$\sqrt{n}\big(\hat{C}_n(\mathbf{u}) - C(\mathbf{u})\big), \qquad \mathbf{u} \in [0,1]^d \quad (2.12)$$

(see Rüschendorf, 1976; Deheuvels, 1979 for details).

The empirical copula is a step function. Several approaches have been proposed to obtain smoother estimators. One such approach is to use empirical checkerboard copulas, which partition the unit cube into a regular grid and distribute the empirical mass uniformly within each cell, yielding a piecewise uniform distribution; see (Li et al., 1997; Genest et al., 2017). The resulting copula density is piecewise constant and therefore remains non-smooth. In Chapter 4, we develop a smoothing procedure that transforms such checkerboard copulas into continuous copula densities while closely approximating the underlying mass distribution. Another common approach is to use Bernstein copulas, which replace the indicator functions in the empirical copula with Bernstein polynomials, yielding a polynomial approximation that preserves the copula properties (see Sancetta and Satchell, 2004). In Chapter 7, we employ Bernstein copulas to model the temporal dependence of forecast errors in grid frequency prediction. Kernel-based methods adapt kernel density estimation techniques (Silverman, 1986) to the copula setting; see (Nagler, 2018) for an overview. In Chapters 3 and 4, we develop neural network-based estimators for the copula density and compare them with these approaches.

2.2. Neural Network Models

In this thesis, we use neural networks primarily as flexible function approximators. The basic building block is the feedforward network, also known as a multilayer perceptron (MLP), in which an input vector $\mathbf{x}$ is transformed through a sequence of layers. Each layer applies an affine transformation followed by a componentwise nonlinear activation function $\mathbf{x} \mapsto \mathsf{act}(A\mathbf{x} + \mathbf{b})$. Here, A is a weight matrix, $\mathbf{b}$ a bias vector, and act a fixed nonlinear activation function such as the hyperbolic tangent, $\mathsf{act}(z) = \tanh(z)$, or the rectified linear unit (ReLU), $\mathsf{act}(z) = \max\{0, z\}$. Stacking L such layers yields the recursion

$$\mathbf{h}^{(0)} = \mathbf{x}, \quad \mathbf{h}^{(\ell)} = \mathsf{act}\big(A^{(\ell)}\mathbf{h}^{(\ell-1)} + \mathbf{b}^{(\ell)}\big), \quad \ell = 1, \dots, L.$$

The final representation $\mathbf{h}^{(L)}$ is mapped to an output, e.g., through an additional affine transformation and an output function appropriate to the task (such as a softmax for classification). While this basic structure suffices for many tasks, more complex architectures such as normalizing flows or recurrent networks extend it to capture richer dependencies.

Having specified the basic network architecture, we next describe how its parameters are fitted to data. Training is the process of choosing the parameters by minimizing a loss function that quantifies the discrepancy between the model and the observed data. Given observations $\{\mathbf{x}_i\}_{i=1}^n$, a common choice is the empirical loss

$$\mathcal{L}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell(\theta; \mathbf{x}_i),$$

where $\ell(\cdot)$ measures the contribution of each observation. In density estimation, the natural loss is the negative log-likelihood under the model, so that minimizing $\mathcal{L}(\theta)$ corresponds to maximum likelihood estimation. We adopt this approach in Chapters 3 and 4. The layered structure of feedforward networks allows gradients to be computed efficiently by recursive application of the chain rule, a procedure known as backpropagation (Rumelhart et al., 1986). These gradients are then used to update the parameters iteratively with stochastic gradient methods such as stochastic gradient descent (SGD) (Robbins and Monro, 1951) or Adam (Kingma and Ba, 2015).

From a statistical standpoint, neural networks constitute parametric models characterized by a finite but typically very large number of parameters. As a consequence, the model class is highly expressive and can behave similarly to nonparametric methods. Universal approximation theorems (Cybenko, 1989; Hornik et al., 1989) provide the theoretical foundation: under mild conditions, feedforward networks can approximate any continuous function on a compact domain arbitrarily well as network width or depth increases.

Beyond standard feedforward networks, we make use of two specialized model families later in this thesis: normalizing flows for flexible density modeling and neural sequence modeling approaches for sequential data. In the following, we briefly review the main ideas behind these architectures.

Normalizing flows are generative models that represent a complex target distribution by transforming a simple base distribution (such as a standard Gaussian) through a sequence of invertible, differentiable mappings. Concretely, we write the overall transformation as a composition

$$T_\theta = T_K \circ T_{K-1} \circ \ldots \circ T_1,$$

where each T_k is an invertible transformation parameterized by a neural network. Let $\mathbf{Z} \sim p_{\mathbf{Z}}$ be a base random vector and define the transformed random vector $\mathbf{X} = T_\theta(\mathbf{Z})$.

For an observation $\mathbf{x}$, the model density is obtained via the change-of-variables formula

$$\log p_\theta(\mathbf{x}) = \log p_{\mathbf{Z}}\big(T_\theta^{-1}(\mathbf{x})\big) + \log \left| \det \frac{\partial T_\theta^{-1}(\mathbf{x})}{\partial \mathbf{x}} \right|.$$

Many flow architectures used for density estimation are constructed such that for each T_k the inverse mapping and the log-determinant of the Jacobian can be computed efficiently, which makes likelihood evaluation and gradient-based learning tractable. To generate new samples, one draws $\mathbf{z} \sim p_{\mathbf{Z}}$ and applies the forward transformation $\mathbf{x} = T_\theta(\mathbf{z})$. By composing many such transformations, one obtains highly flexible density models with tractable likelihoods; see, e.g., the reviews by Papamakarios et al. (2021) and Kobyzev et al. (2021) for an overview of normalizing flow architectures and applications. In Chapter 3, we use normalizing flow architectures to learn flexible multivariate distributions whose copulas are well adapted to the given data.

Sequence models are a further important class of architectures that we use in this thesis. These aim to capture statistical dependencies within ordered data, such as time series, text, or sequences of events. Given a sequence $\mathbf{x}_{1:T} = (\mathbf{x}_1, \dots, \mathbf{x}_T)$, the goal is typically to model the joint distribution or to predict future elements based on the past, for example through conditional densities of the form $p(\mathbf{x}_{t+1} \mid \mathbf{x}_{1:t})$. A common neural architecture for this purpose is the recurrent neural network (RNN); see, e.g., Goodfellow et al. (2016, Chapter 10) for an introduction. Conceptually, an RNN processes the sequence element by element while maintaining a state vector $\mathbf{s}_t$ that serves as a summary of the past inputs $(\mathbf{x}_1, \dots, \mathbf{x}_t)$. A simple RNN layer updates

$$\mathbf{s}_t = \mathsf{act}\big(A_s \mathbf{s}_{t-1} + A_x \mathbf{x}_t + \mathbf{b}\big), \quad t = 1, \dots, T,$$

with some initial state $\mathbf{s}_0$. Here, $\mathbf{x}_t \in \mathbb{R}^{d_x}$ denotes the input at time t, where d_x is the input dimension, and $\mathbf{s}_t \in \mathbb{R}^{d_s}$ is the corresponding state, where d_s is the dimension of the state representation. The matrix $A_s \in \mathbb{R}^{d_s \times d_s}$ maps the previous state to the new representation, $A_x \in \mathbb{R}^{d_s \times d_x}$ maps the current input to the state space, and $\mathbf{b} \in \mathbb{R}^{d_s}$ is a bias vector. In this way, the state $\mathbf{s}_t$ recursively encodes information about the past inputs and can be used as a basis for predictions or further processing.

In later chapters, we rely on more advanced sequence architectures, in particular gated recurrent units (GRUs) (Cho et al., 2014) and transformer models (Vaswani et al., 2017), which better capture long-range dependencies than simple RNNs and improve trainability. Their specific formulations are described in Chapter 6, where they serve as flexible components in predictive models for grid frequency dynamics.

3. Copula Estimation with Normalizing Flows

The chapter is based on joint work with Maximilian Coblenz und Oliver Grothe (Publ. I) and was presented at the COMPSTAT 2024 (Conf. II) and CMStatistics 2024 (Conf. III). In this chapter, we propose normalizing flow copula models (NFCMs), a flexible nonparametric approach to copula modeling and density estimation. Our method constructs the copula as the dependence structure generated by a learned composition of invertible normalizing-flow transformations of a base distribution.

3.1. Introduction

Since Sklar (1959) introduced copulas, copula theory has become a core framework for modeling multivariate dependence. By separating marginal distributions from the dependence structure, copulas can be used to model more flexible distribution families than classical models. For a detailed introduction to copula theory, see, e.g., Nelsen (2006), Durante and Sempi (2015), Joe (2014), and Hofert et al. (2018). While parametric copula families usually make specific assumptions about the form of the dependence, e.g., a certain asymmetry structure or tail dependence behavior, in practice there are cases where this type of prior knowledge is missing or not easily identifiable. In such cases, a nonparametric model that estimates a copula in a data-driven manner without making specific assumptions about the dependence structure can be helpful.

Many methods for nonparametric copula estimation have been proposed in the literature. One class of methods is based on empirical distribution functions of given pseudo-observations (Deheuvels, 1979) and their smoothing versions such as checkerboard copulas (Genest et al., 2017; Cuberos et al., 2020), Bernstein copulas (Sancetta and Satchell, 2004; Bouezmarni et al., 2013), and beta copulas (Segers et al., 2017). Another important class of nonparametric copula density estimators is obtained by applying kernel density estimators (KDEs) (Parzen, 1962; Chen, 2017) to pseudo-observations (Gijbels and Mielniczuk, 1990; Charpentier et al., 2007; Geenens et al., 2017; Wen and Wu, 2020). A list and implementation of different methods for KDE-based copula density estimation can

be found in Nagler (2018). A multivariate extension of KDE-based copula estimators (i.e., dimension greater than two) is presented in Nagler and Czado (2016) and Nagler et al. (2017), which propose a nonparametric density estimator based on simplified vine copulas. The literature on nonparametric estimation of copulas and their densities is growing; other references include wavelet-based estimators (Genest et al., 2009; Autin et al., 2010; Chatrabgoun and Parham, 2016), estimation based on Legendre orthogonal polynomials (Ngounou Bakam and Pommeret, 2025), and a nonparametric approach for Archimedean copulas (Ling et al., 2020). While these approaches achieve good results in many settings, they can face challenges in capturing complex dependence structures, especially in higher dimensions and in the presence of pronounced tail dependence or asymmetry.

Recently, in the field of machine learning, the normalizing flow has been proposed as an appropriate technique of probabilistic modeling and inference (Papamakarios et al., 2021; Kobyzev et al., 2021). Here, neural networks are used to construct and combine invertible and differentiable building blocks that can transform a simple probability distribution into a complex distribution. In practice, normalizing flow models are often used to create generative models that generate new data points from the same distribution as the training data (Kingma and Dhariwal, 2018), and to estimate the probability density function of complex data samples (Tabak and Turner, 2013; Dinh et al., 2016; Papamakarios et al., 2017). These works demonstrate that transformation-based models can flexibly represent complex multivariate distributions while retaining tractable likelihoods and efficient sampling.

In this paper, we combine the need for flexible nonparametric copula models with the idea of distributional modeling by transformations and normalizing flow models. We propose a class of normalizing flow copula models (NFCMs) based on a copula representation that expresses a copula as the dependence structure induced by a suitable transformation of a base distribution, enabling estimation from data. In particular, we design an affine coupling-based normalizing flow copula model (AC-NFCM). The AC-NFCM chains together relatively simple, invertible building blocks to obtain a flexible yet tractable copula model that can learn the copula density from data with unknown marginal distributions, and can efficiently generate synthetic samples. We demonstrate strong performance on simulated data, measured by integrated absolute error (IAE), including scenarios with different dimensions, dependence structures, and extreme values copula families. Among all models considered, AC-NFCM is the only one that accurately captures tail dependence. On real data, AC-NFCM generates synthetic samples that closely match the pseudo-observations.

Our approach is related to several strands of prior work. There are already approaches that merge the copula concept and normalizing flows. In Wiese et al. (2019), the decomposition principle of copula modeling is integrated into a normalizing flow structure to

better model the tails of the unknown distributions. In Laszkiewicz et al. (2021) and Laszkiewicz et al. (2022), the effects of replacing the base distribution in a normalizing flow model with a general copula distribution are discussed; an empirical study shows that such a replacement can make the modified normalizing flows more stable for data with fat tails. In Kamthe et al. (2021) and McDonald et al. (2022), motivated by Sklar's theorem, the process of modeling a joint distribution is decomposed into modeling its marginal distributions and modeling the underlying copula distribution. Although these works provide impressive results, the question remains how to ensure that the learned distribution is a bona fide copula, i.e., whether the learned marginal distributions are uniformly distributed. Our work is also related to the concept of the implicit copula (Nelsen, 2006; Smith, 2023) and the conditional sampling method (Hofert et al., 2018), which is based on the multivariate distributional transformation (Rosenblatt, 1952; Rüschendorf, 1976; Rüschendorf, 2009). An implicit copula is the copula associated with a multivariate distribution and can be obtained by inverting Sklar's theorem (Smith, 2023; McNeil et al., 2015, p. 190). Smith and Klein (2021) derive an implicit copula from a heteroskedastic semiparametric regression, and Smith and Maneesoonthorn (2018) develop an implicit-copula framework for time series. We extend the implicit-copula framework by modeling the random vector so that its copula arises from a triangular transformation of a base distribution, enabling nonparametric estimation of the copula density from data. Finally, there are copula construction methods where the copula function is directly transformed into another copula function (Durante et al., 2010; Morillas, 2005; Kolesárová et al., 2015; Saminger-Platz et al., 2024). Our method differs from these approaches in that we transform the densities of random vectors into copula densities and use neural networks to estimate the transformation function.

The remainder of the chapter is organized as follows. In Section 3.2, we introduce normalizing flow copula models and establish their theoretical properties. In Section 3.3, we propose the affine coupling-based normalizing flow copula model and present the density estimation process. In Section 3.4, we compare the modeling performance of the affine coupling-based normalizing flow copula model with other nonparametric estimation methods through simulation studies for data of different dimensions. We examine the modeling performance with respect to tail dependence, asymmetry, and extreme values copula families. In Section 3.5, we apply the proposed methodology to two real-world datasets from the fields of insurance and engineering and compare the synthetic samples with the corresponding pseudo-observations. Section 3.6 concludes the chapter. Supplementary code for this chapter is available at Liu et al. (2025b).

3.2. Copula Construction via Transformation

In this section, we revisit copulas and the fact that any copula can be represented as a transformation of an arbitrary basis distribution. This perspective motivates constructing copula models by directly learning such transformations. Building on this idea, we introduce the normalizing flow copula model, in which the transformation is designed to have a triangular and self-normalizing structure. This structure ensures that the resulting distribution is a valid copula and facilitates its implementation using flow-based neural networks in the following section.

3.2.1. Preliminaries

According to Sklar's Theorem (Sklar, 1959), any d-dimensional distribution function F with marginal distributions $F_1, \dots, F_d$ admits a copula representation

$$F(\mathbf{x}) = C\big(F_1(x_1), \dots, F_d(x_d)\big), \qquad \mathbf{x} \in \mathbb{R}^d,$$

where the copula C is uniquely defined on $\prod_{j=1}^{d} \operatorname{ran} F_j$ by

$$C(\mathbf{u}) = F\big(F_1^{-1}(u_1), \dots, F_d^{-1}(u_d)\big) \tag{3.1}$$

(see Nelsen, 2006 and Joe, 2014 for a comprehensive exposition).

The probability integral transformation provides a direct interpretation of this representation: if X is a continuous random variable with CDF F, then $F(X) \sim \text{Unif}(0, 1)$. Consequently, for a continuous random vector $\mathbf{X} = (X_1, \dots, X_d)$ with marginal CDFs $F_1, \dots, F_d$, the transformed vector

$$\big(F_1(X_1), \dots, F_d(X_d)\big) \tag{3.2}$$

has uniform margins and distribution function C. Thus, a continuous random vector $\mathbf{X}$ implicitly defines a copula. This holds either in the distributional sense, by applying the probability integral transform to its margins as in (3.2), or by explicitly computing its functional form as in (3.1), i.e., by inserting the marginal quantile functions F_i^{-1} into the joint distribution F of $\mathbf{X}$; this procedure is called the inversion method (see Nelsen, 2006, p. 51). Copulas following from random vectors are sometimes referred to as implicit copulas of $\mathbf{X}$ (Smith, 2023).

Note that the copula C of a random vector contains the full dependence structure of the joint distribution; in particular, it can encode features such as tail dependence or asymmetry (see Nelsen, 2006 and Appendix A.1).

If F is absolutely continuous with density f and marginal densities $f_1, \dots, f_d$, then the copula is also absolutely continuous with density

$$c(u_1, \dots, u_d) = \frac{f\big(F_1^{-1}(u_1), \dots, F_d^{-1}(u_d)\big)}{f_1\big(F_1^{-1}(u_1)\big) \cdots f_d\big(F_d^{-1}(u_d)\big)}, \qquad \text{for a.e. } \mathbf{u} \in (0,1)^d. \tag{3.3}$$

In practice, copulas are typically estimated from data by first transforming the observations to the unit hypercube. Let $\mathbf{x}^1, \dots, \mathbf{x}^N \in \mathbb{R}^d$ denote the observed sample and let R_{ij} be the rank of the j-th component of $\mathbf{x}^i$ among $\{x_j^1, \dots, x_j^N\}$. One further defines the pseudo-observations

$$\hat{\mathbf{u}}^i = \frac{1}{N+1}\big(R_{i1}, \dots, R_{id}\big)^\top, \qquad i = 1, \dots, N,$$

with $\hat{\mathbf{u}}^1, \dots, \hat{\mathbf{u}}^N \in (0,1)^d$ (see Hofert et al., 2018, p. 139). Given a parametric copula family $\{C(\cdot;\theta) : \theta \in \Theta\}$ with density $c(\cdot;\theta)$, inference is then usually based on the copula log-likelihood (Genest et al., 1995)

$$\ell(\theta) = \sum_{i=1}^{N} \log c(\hat{\mathbf{u}}^i; \theta),$$

possibly combined with a separate estimation step for the original observations' margins, as in the inference functions for margins approach (Joe and Xu, 1996).

We now recall the standard density transformation formula (see, e.g., Klenke, 2014; Casella and Berger, 2002), which is the starting point for the transformation-based representation used in our model.

Lemma 1 (Density Transformation Formula)**.** *Let $\mathbf{X}$ be a d-dimensional random vector with the associated measure μ and a continuous density f. Suppose A is an open or closed subset of $\mathbb{R}^d$ with $\mu(\mathbb{R}^d \setminus A) = 0$. Furthermore, let $\psi : \mathbb{R}^d \to \mathbb{R}^d$ be a measurable mapping whose restriction*

$$\psi|_A : \mathbb{R}^d \supset A \to \psi(A) := V \subset \mathbb{R}^d$$

is a C^1-diffeomorphism with Jacobian matrix $\psi'(x) \neq 0$ for all $x \in A$. Then

$$g(y) := \begin{cases} \dfrac{f(\psi^{-1}(y))}{|\det \psi'(\psi^{-1}(y))|}, & \text{if } y \in \psi(A), \\ 0, & \text{otherwise} \end{cases} \tag{3.4}$$

is the density function of the random vector $\mathbf{Y} = \psi(\mathbf{X})$.

Now we have the tools to define the NFCM class, which is presented in the next section.

3.2.2. Normalizing Flow Copula Model

We introduce a convenient notation for copulas obtained from transformations of a base distribution.

Definition 3.2.1 (Transformation Notation)**.** *Let $d \geq 2$. Let $\mathbf{X}$ be an absolutely continuous $\mathbb{R}^d$-valued random vector with density $f_{\mathbf{X}} : \mathbb{R}^d \to \mathbb{R}_+$. Let $\psi : \mathbb{R}^d \to \mathbb{R}^d$ be an invertible, measurable map for which the density transformation formula holds. If the distribution function of $\psi(\mathbf{X})$ is a copula, we write $C_{\mathbf{X},\psi}$ for its distribution function and $c_{\mathbf{X},\psi}$ for its density. When $C_{\mathbf{X},\psi}$ equals a given copula C, we say that C admits a copula via transformation representation with base distribution $\mathbf{X}$ and transformation ψ.*

To make the probability integral transformation explicit in our notation, we introduce an operator that collects the marginal CDFs of a transformed random vector.

Definition 3.2.2 (CDF-Generator)**.** *Let $\mathcal{C}(\mathbb{R}^d, \mathbb{R}^d)$ be the space of continuous mappings from $\mathbb{R}^d$ to $\mathbb{R}^d$. Let $\mathbf{X}$ be a d-dimensional continuous random vector, and $g : \mathbb{R}^d \to \mathbb{R}^d$ a continuous function. We define the CDF-Generator*

$$\mathcal{S}_{\mathbf{X}} : \mathcal{C}(\mathbb{R}^d, \mathbb{R}^d) \to \mathcal{C}(\mathbb{R}^d, \mathbb{R}^d), \quad g \mapsto (F_1, \dots, F_d)^\top,$$

where F_i, $i = 1, \dots, d$, represent the cumulative distribution functions of the margins of the continuous random vector $g(\mathbf{X})$.

By construction, the mapping $\mathcal{S}_{\mathbf{X}}\{g\} \circ g$ applies the probability integral transformation componentwise to $g(\mathbf{X})$ and therefore has the distribution function of a copula whenever $g(\mathbf{X})$ is absolutely continuous. The following lemma shows that every transformation ψ in Definition 3.2.1 that yields an absolutely continuous copula can be written in this form.

Lemma 2 (Transformation Decomposition)**.** *With the assumptions in Definition 3.2.1, the distribution function of $\psi(\mathbf{X})$ is an absolutely continuous copula if and only if there exists $g \in \mathcal{C}(\mathbb{R}^d, \mathbb{R}^d)$ for which the density transformation formula holds, such that $\psi = \mathcal{S}_{\mathbf{X}}\{g\} \circ g$.*

Proof. Let $g \in \mathcal{C}(\mathbb{R}^d, \mathbb{R}^d)$ satisfy the density transformation formula. In particular, $g(\mathbf{X})$ is absolutely continuous. By definition of the CDF-Generator, $\mathcal{S}_{\mathbf{X}}\{g\} \circ g(\mathbf{X})$ has the distribution function of a copula, since the probability integral transformation is applied to each dimension of the transformed random vector $g(\mathbf{X})$. The copula is absolutely continuous, since $g(\mathbf{X})$ is absolutely continuous.

Now let the distribution function of $\psi(\mathbf{X})$ be an absolutely continuous copula and the resulting random vector $\mathbf{U} := \psi(\mathbf{X})$. Let F^{-1} be the quantile function of an absolutely continuous distribution. Then for $g := (F^{-1}, \dots, F^{-1})^\top \circ \psi$, we obtain

$$g(\mathbf{X}) = (F^{-1}, \dots, F^{-1})^\top \circ \psi(\mathbf{X}) = \left(F^{-1}(U_1), \dots, F^{-1}(U_d)\right)^\top.$$

This means that the margins of $g(\mathbf{X})$ all have the distribution function F, thus $\mathcal{S}_{\mathbf{X}}\{g\} = (F, \dots, F)^\top$. Then

$$\psi = (F, \dots, F)^\top \circ (F^{-1}, \dots, F^{-1})^\top \circ \psi = \mathcal{S}_{\mathbf{X}}\{g\} \circ g.$$

Furthermore, $(F^{-1}, \dots, F^{-1})^\top$ as a transformation also satisfies the density transformation formula, since F is absolutely continuous and in particular also has a quantile density (the derivative of the quantile function). Therefore g as a composition of $(F^{-1}, \dots, F^{-1})^\top$ and ψ also satisfies the density transformation formula. □

Motivated by Lemma 2, we henceforth restrict attention to transformations of the form $\psi = \mathcal{S}_{\mathbf{X}}\{g\} \circ g$ and collect them in the normalizing flow copula model defined below.

Definition 3.2.3 (Normalizing Flow Copula Model)**.** *A normalizing flow copula model is a pair* $(\mathbf{X}, \mathcal{T})$ *consisting of one d-dimensional absolutely continuous distributed random vector* $\mathbf{X}$ *and a set* $\mathcal{T}$ *of measurable mappings on* $\mathbb{R}^d$ *that fulfill the requirements in Definition 3.2.1 and can be described by a parameter space* Θ*, that is,*

$$\mathcal{T} = \Psi_\Theta := \{\mathcal{S}_{\mathbf{X}}\{g_\theta\} \circ g_\theta : \theta \in \Theta\}.$$

We denote the normalizing flow copula model as $(\mathbf{X}, \Psi_\Theta)$*. In the following,* $\mathcal{T}$ *is referred to as the transformation class.*

To simplify the notation, we write the transformed random vector $g(\mathbf{X})$ as $\mathbf{Y}$, unless otherwise specified. The marginal CDFs of $\mathbf{Y}$ are denoted by $F_{Y_i}, i = 1, ..., d$, the marginal PDFs by $f_{Y_i}, i = 1, ..., d$. We note that $C_{\mathbf{X}, \mathcal{S}_{\mathbf{X}}\{g_\theta\} \circ g_\theta}$ is the implicit copula of $\mathbf{Y} = g_\theta(\mathbf{X})$.

Sampling from $C_{\mathbf{X},\psi}$ is straightforward if the sampling procedure of the base distribution $\mathbf{X}$ is known. To generate a sample from the copula $C_{\mathbf{X},\psi}$ of size N, first draw observations $\{\mathbf{x}^{(k)}, k = 1, \dots, N\}$ from $\mathbf{X}$ and transform them using g. A sample from $C_{\mathbf{X},\psi}$ is then obtained by applying the probability integral transformation marginally to each transformed point $g(\mathbf{x}^{(k)})$. That is,

$$\mathbf{v}^{(k)} = \big(\mathcal{S}_{\mathbf{X}}\{g\} \circ g\big)(\mathbf{x}^{(k)}) = \big(F_{Y_1}(y_1^{(k)}), \dots, F_{Y_d}(y_d^{(k)})\big), \qquad k = 1, \dots, N,$$

which yields a sample of size N from the copula $C_{\mathbf{X},\psi}$. The entire sampling procedure in the NFCM is shown in Algorithm 1.

Algorithm 1: Sampling from an NFCM

Input: d-dimensional NFCM $(\mathbf{X}, \Psi_\Theta)$ with parameter θ, sample size N, marginal CDFs of $g_\theta(\mathbf{X})$: $\{F_{Y_i}, i = 1, \dots, d\}$

Output: N sample points $\{\mathbf{u}^k, k = 1, \dots, N\}$ of copula $C_{\mathbf{X},\psi_\theta}$

```
Generate N sample points {x^k, k = 1, ..., N} of base distribution X;
for k = 1, 2, ..., N do
    Calculate y^k = g_θ(x^k);
    for i = 1, 2, ..., d do
        Calculate u_i^k = F_{Y_i}(y_i^k)
```

The copula density $c_{\mathbf{X},\psi}(\mathbf{u})$, $\mathbf{u} \in (0,1)^d$, can be calculated using (3.3) and (3.4) and will later be required for maximum likelihood estimation. It follows:

$$\begin{aligned} c_{\mathbf{X},\psi}(u_1, \dots, u_d) &= \frac{f_{\mathbf{Y}}(F_{Y_1}^{-1}(u_1), \dots, F_{Y_d}^{-1}(u_d))}{f_{Y_1}(F_{Y_1}^{-1}(u_1)) \cdots f_{Y_d}(F_{Y_d}^{-1}(u_d))} \\ &= \frac{|\det J_{g^{-1}}(F_{Y_1}^{-1}(u_1), \dots, F_{Y_d}^{-1}(u_d))| f_{\mathbf{X}}(g^{-1}(F_{Y_1}^{-1}(u_1), \dots, F_{Y_d}^{-1}(u_d)))}{f_{Y_1}(F_{Y_1}^{-1}(u_1)) \cdots f_{Y_d}(F_{Y_d}^{-1}(u_d))}. \end{aligned} \tag{3.5}$$

We illustrate the NFCM with a Gaussian copula example. Let the base distribution be the multivariate standard normal distribution $\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_2)$. We consider the following linear transformation $\mathbf{Y} := g(\mathbf{X}) = \mathbf{A}\mathbf{X}$, $\mathbf{A} \in \mathbb{R}^{2\times 2}$. The transformed random vector $g(\mathbf{X})$ is again normally distributed with the covariance matrix $= \mathbf{A}\mathbf{A}^\top$, provided that $\mathbf{A}\mathbf{A}^\top$ is not singular. Let the transformation matrix $\mathbf{A}$ be

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}.$$

Then, the new covariance matrix is

$$= \begin{pmatrix} a_{11}^2 + a_{12}^2 & a_{11}a_{21} + a_{12}a_{22} \\ a_{21}a_{11} + a_{22}a_{12} & a_{21}^2 + a_{22}^2 \end{pmatrix} =: \begin{pmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{pmatrix}.$$

The marginal distributions of $g(\mathbf{X}) \sim \mathcal{N}(0, \mathbf{A}\mathbf{A}^\top)$ are univariate normal, i.e.:

$$Y_1 \sim \mathcal{N}(0, \sigma_{11}), \quad Y_2 \sim \mathcal{N}(0, \sigma_{22}).$$

That is, $\mathcal{S}_{\mathbf{X}}\{g\}(x_1, x_2)$ is given by

$$\mathcal{S}_{\mathbf{X}}\{g\}(x_1, x_2) = \begin{pmatrix} \Phi\left(\frac{x_1}{\sqrt{\sigma_{11}}}\right) \\ \Phi\left(\frac{x_2}{\sqrt{\sigma_{22}}}\right) \end{pmatrix},$$

where Φ denotes the standard normal cumulative distribution function. The resulting NFCM $C_{\mathbf{X},\mathcal{S}_{\mathbf{X}}\{g\}\circ g}$ is the Gaussian copula with correlation parameter

$$\rho = \frac{\sigma_{12}}{\sqrt{\sigma_{11}}\sqrt{\sigma_{22}}} = \frac{a_{11}a_{21} + a_{12}a_{22}}{\sqrt{a_{11}^2 + a_{12}^2}\sqrt{a_{21}^2 + a_{22}^2}}.$$

To model a Gaussian copula with a certain correlation as an NFCM representation for a given dataset, only the transformation matrix $\mathbf{A}$ needs to be determined. By appropriately choosing a_{ij} above, any value of ρ in the interval $(-1, 1)$ can be generated.

While the NFCM can be defined with arbitrary invertible transformations g_θ, in what follows we focus on the case where g_θ is triangular. This choice yields computationally convenient Jacobians and, as we show below, still allows us to represent arbitrary absolutely continuous copulas. In probabilistic modeling, triangular transformations are a standard device for parameterizing joint distributions via invertible maps. Building on this idea, Klein et al. (2022) propose a tailored triangular map to develop a likelihood-based regression framework that flexibly models the full conditional joint distribution of multivariate responses, including covariate-dependent dependence structures. For a broader discussion of the role of triangular transformations in normalizing flows, see, e.g., Kobyzev et al. (2021).

Definition 3.2.4 (Triangular Transformation)**.** *A mapping $T : \mathbb{R}^d \to \mathbb{R}^d$ is defined as triangular iff T has the form*

$$T(\mathbf{x}) = \begin{pmatrix} T_1(x_1) \\ T_2(x_1, x_2) \\ \vdots \\ T_d(x_1, x_2, \dots, x_d) \end{pmatrix}.$$

T is called increasing iff T_i is an increasing function with respect to x_i for every $i \in \{1, \dots, d\}$.

Remark 3.2.1. *The Jacobian matrix of a triangular transformation (if it exists) is always a lower triangular matrix and the Jacobian determinant is then the product of the diagonal elements. In particular, a composition of triangular transformations is still a triangular transformation.*

The next result shows that increasing triangular maps are expressive enough for our copula construction.

Lemma 3 (Bogachev et al. (2005))**.** *For two absolutely continuous increasing probability measures on $\mathbb{R}^d$, μ and ν, there is a unique increasing triangular mapping T up to zero sets of μ, so that $\nu = T_*\mu$, where $T_*\mu$ denotes the pushforward measure of μ under T.*

Corollary 3.2.1. *Let C be a d-dimensional absolutely continuous copula and $\mathbf{X}$ a d-dimensional absolutely continuous base distribution. Then there exists an increasing triangular mapping $g : \mathbb{R}^d \to \mathbb{R}^d$ such that*

$$C = C_{\mathbf{X},\mathcal{S}_{\mathbf{X}}\{g\}\circ g}.$$

Proof. We choose d arbitrary absolutely continuous univariate distributions $F_1, \dots, F_d$. According to Sklar's Theorem, $F : \mathbf{y} \mapsto C(F_1(y_1), ..., F_d(y_d)), \mathbf{y} \in \mathbb{R}^d$, is a d-dimensional CDF with univariate marginal distributions $F_1, \dots, F_d$. $F(\cdot)$ is also absolutely continuous. Let $\mathbf{Y}$ be a random vector with CDF F. We write down the probability measure for $\mathbf{Y}$ with ν and the probability measure for the random vector $\mathbf{X}$ with μ. Then, according to Lemma 3, there is an increasing triangular transformation g such that $\nu = g_*\mu$, i.e. $\mathbf{Y} \sim g(\mathbf{X})$. In particular, the random vector $\mathcal{S}_{\mathbf{X}}\{g\} \circ g(\mathbf{X})$ then has a distribution function C, that is, $C = C_{\mathbf{X},\mathcal{S}_{\mathbf{X}}\{g\}\circ g}$. □

Corollary 3.2.1 guarantees that every absolutely continuous copula can be modeled by an NFCM with an increasing triangular transformation. Notably, this result ensures that a triangular transformation can generate a certain dependence structure from a simple base distribution such as a multivariate standard normal distribution, which is symmetric and has no tail dependence. For the generation of asymmetric (i.e., radially asymmetric and/or non-exchangeable) copulas, such a transformation can often be easily constructed by applying an asymmetric and/or non-linear transformation. Whether the resulting copulas are asymmetric can also be verified using the density formula (3.5), for example by checking whether $c(v_1, v_2) = c(v_2, v_1)$. In the context of tail dependence generation, the situation is fundamentally more complex. Verifying the existence of tail dependence already requires careful analysis of the corresponding limit condition. This complexity increases further when considering the impact of transformations on the dependence structure.

In the following, we consider a special bivariate NFCM based on an increasing triangular transformation g_σ, so that the resulting copula $C_{\mathbf{X},\mathcal{S}_{\mathbf{X}}\{g_\sigma\}\circ g_\sigma}$ possesses the property of tail dependence, even though the base distribution $\mathbf{X}$ has no tail dependence.

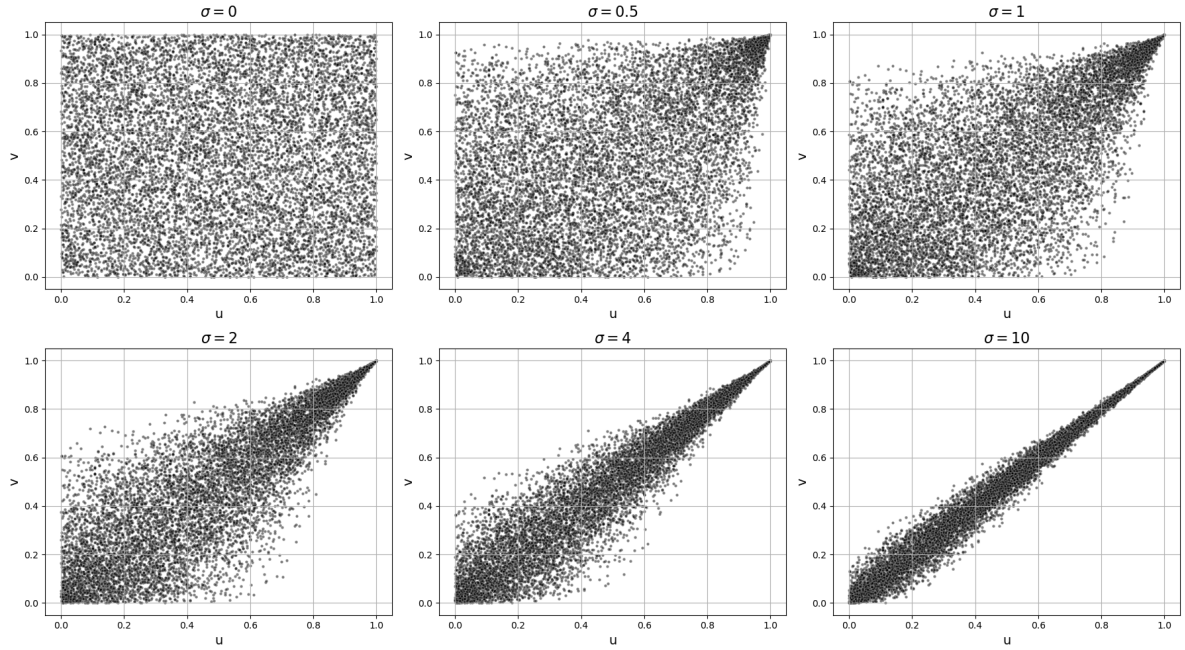


Figure 3.1.: Synthetic samples from the NFCM in Example 3.2.1 with increasing parameter σ from 0 to 10.

Example 3.2.1 (Tail Dependence)**.** *Consider the NFCM* $(\mathbf{X}, \{\mathcal{S}_{\mathbf{X}}\{g_\sigma\} \circ g_\sigma : \sigma > 0\})$ *with*

$$\mathbf{Y} = g_\sigma(\mathbf{X}) = \begin{pmatrix} T_1(X_1) \\ T(X_1, X_2) \end{pmatrix} = \begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} := \begin{pmatrix} X_1 \\ \exp(\sigma X_1) + X_2 \end{pmatrix}.$$

For $\sigma \to 0$, $\exp(\sigma X_1) \to 1$ *and the dependence between* Y_1 *and* Y_2 *vanishes; the induced NFCM tends to the independence copula. As* σ *increases, large values of* Y_1 *dominate* Y_2 *via the term* $\exp(\sigma X_1)$*, creating near-monotone behavior in the upper tail (see Figure 3.1).*

Remark 3.2.2. *For* $\sigma > 0$, g_σ *is a strictly increasing triangular transformation. Let* (h_1, h_2) *denote the inverse mapping of* g_σ*. Let* f_{X_1}, f_{X_2} *be the marginal PDFs of the base distribution* $\mathbf{X}$ *and* f_{Y_1}, f_{Y_2} *the marginal PDFs of* $\mathbf{Y} = g_\sigma(\mathbf{X})$*. For* $q \in (0, 1)$ *define*

$$K_q(y_1) := \int_{F_{Y_2}^{-1}(q)}^{\infty} f_{X_2}\big(h_2(y_1, y_2)\big)\, \frac{\partial h_2}{\partial y_2}(y_1, y_2)\, dy_2 \in [0, 1].$$

By the density transformation formula,

$$\mathbb{P}\Big(Y_2 > F_{Y_2}^{-1}(q) \mid Y_1 > F_{Y_1}^{-1}(q)\Big) = \frac{\int_{F_{Y_1}^{-1}(q)}^{\infty} f_{X_1}(h_1(y_1))\, h_1'(y_1)\, K_q(y_1)\, dy_1}{\int_{F_{Y_1}^{-1}(q)}^{\infty} f_{X_1}(h_1(y_1))\, h_1'(y_1)\, dy_1}.$$

Let μ_q be the conditional law of Y_1 given $\{Y_1 > F_{Y_1}^{-1}(q)\}$. The ratio above equals

$$\mathbb{E}_{\mu_q}\big[K_q(Y_1)\big].$$

By the definition of the essential infimum with respect to μ_q,

$$\mathbb{E}_{\mu_q}\big[K_q(Y_1)\big] \;\geq\; \operatorname*{ess\,inf}_{y_1 \geq F_{Y_1}^{-1}(q)} K_q(y_1).$$

Note that the above chain of reasoning holds for any increasing triangular transformation.

Now for our example, the distribution of $\exp(\sigma X_1)$ is heavy-tailed and convolving with a Gaussian kernel (X_2) flattens its extreme quantiles. Consequently,

$$\exists\, q^* \in (0,1)\ \forall\, q \geq q^* :\quad F_{Y_2}^{-1}(q) - F_{e^{\sigma Y_1}}^{-1}(q) = F_{Y_2}^{-1}(q) - e^{\sigma F_{Y_1}^{-1}(q)} \leq 0.$$

For this model one computes, for all $y_1 \in \mathbb{R}$,

$$K_q(y_1) = 1 - \Phi\big(F_{Y_2}^{-1}(q) - e^{\sigma y_1}\big),$$

hence for all $q \geq q^*$ and all $y_1 \geq F_{Y_1}^{-1}(q)$,

$$K_q(y_1) \;\geq\; 1 - \Phi(0) = \tfrac{1}{2}.$$

Therefore,

$$\mathbb{P}\Big(Y_2 > F_{Y_2}^{-1}(q) \mid Y_1 > F_{Y_1}^{-1}(q)\Big) \geq \operatorname*{ess\,inf}_{y_1 \geq F_{Y_1}^{-1}(q)} K_q(y_1) \;\geq\; \tfrac{1}{2} \quad \textit{for all } q \geq q^*,$$

which means upper tail dependence.

Taken together, these results show that triangular transformations provide the NFCM with sufficient flexibility to capture complex dependence, thereby motivating the affine-coupling-based implementation developed in the next section.

3.3. Implementation of NFCM with Affine Coupling Transformations and Estimation

In this section, we provide a neural network implementation of an NFCM, where the transformation g is represented by a trainable neural network with affine couplings with parameters $\theta, \theta \in \mathbf{\Theta}$. Estimating the copula from data means then to fix a base distribution beforehand and to estimate the transformation $g_{\tilde{\theta}}$ from the transformation class $\{g_\theta, \theta \in \mathbf{\Theta}\}$ (this step is further refined in Section 3.3.1) and provide the CDF-Generator $\mathcal{S}_{\mathbf{X}}\{g_{\tilde{\theta}}\}$, which is discussed in Section 3.3.2. The overall training process is presented in Section 3.3.3.

3.3.1. Construction of g with Affine Coupling Transformations: AC-NFCM

The main idea to design a flexible yet well computable transformation is to chain several elementary transformations consisting of certain types of invertible neural networks. The chaining of such transformation blocks has two main advantages. First, a function composed of invertible transformations, i.e. $g_\theta := T_n \circ T_{n-1} \circ ... \circ T_2 \circ T_1$, is also invertible, and the inversion can be evaluated step by step using the inverses of the individual blocks

$$g_\theta^{-1} = (T_n \circ T_{n-1} \circ ... \circ T_2 \circ T_1)^{-1} = T_1^{-1} \circ T_2^{-1} \circ ... \circ T_{n-1}^{-1} \circ T_n^{-1}.$$

Second, the logarithm of the Jacobian determinant of the inverse transformation decomposes into the sum of the logarithms of the Jacobian determinants of the individual inverse blocks,

$$\log\det J_{g_\theta^{-1}} = \sum_{i=1}^{n} \log\det J_{T_i^{-1}}.$$

Hence, if the Jacobian determinants of the individual transformation blocks can be computed efficiently, the overall computational complexity remains manageable.

Several neural network architectures provide transformation blocks with easily computable inverses and Jacobian determinants. A detailed overview of these methodologies is given, for example, in Kobyzev et al. (2021) and Papamakarios et al. (2021). One of the most widely used approaches in the normalizing-flow literature is the class of affine coupling transformations, due to their computational efficiency and universal approximation properties. The recent work Teshima et al. (2020) demonstrates that coupling-based invertible neural networks are universal approximators of diffeomorphisms. Moreover, we show below that an affine coupling transformation is a monotone (increasing) triangular

map. Combined with the universal approximation capability of the NFCM for such triangular transformations, established in Corollary 3.2.1, this makes affine coupling blocks particularly well suited for our framework.

In the following, we make this connection explicit by choosing each block T_i in the composition of g_θ to be an affine coupling transformation. To define these building blocks, we first introduce the corresponding notation.

In the following, $\mathbf{x}_{i:j}$ denotes the sub-vector $(x_i, x_{i+1}, ..., x_{j-1}, x_j)^\top$ of a vector $\mathbf{x} \in \mathbb{R}^d$. We introduce the affine coupling transformation $\psi_{k,s,t}$ (Dinh et al., 2016; Teshima et al., 2020), which serves as the fundamental building block for the composed transformation g_θ:

$$\begin{aligned}\psi_{k,s,t}(\mathbf{x})_{1:k} &= \mathbf{x}_{1:k},\\ \psi_{k,s,t}(\mathbf{x})_{k+1:d} &= \mathbf{x}_{k+1:d} \odot \exp\left[s(\mathbf{x}_{1:k})\right] + t(\mathbf{x}_{1:k}),\end{aligned}$$

where $s, t : \mathbb{R}^k \to \mathbb{R}^{d-k}$ are implemented in practice using neural networks. Here, $\odot$ denotes the Hadamard product and $\exp(\cdot)$ is applied componentwise. In the AC-NFCM, the overall transformation is then given by

$$g_\theta = T_n \circ ... \circ T_1, \qquad T_i = \psi_{k_i,s_i,t_i},$$

where the parameters θ collect all network parameters of the functions s_i and t_i (and the permutations introduced below). For ease of presentation, we suppress the subindex i in the following when possible.

Note that $\psi_{k,s,t}$ is an increasing triangular transformation, since $\exp(\cdot)$ is always positive, which means that the Jacobian matrix has a lower triangular shape (see Remark 3.2.1)

$$\mathbf{J}_{\psi_{k,s,t}} = \begin{bmatrix}\mathbf{I}_k & \mathbf{0}\\ \mathbf{L} & \operatorname{diag}\left(\exp\left[s\left(\mathbf{x}_{1:k}\right)\right]\right)\end{bmatrix}.$$

The Jacobian determinant $\det \mathbf{J}_{\psi_{k,s,t}}$ is independent of $\mathbf{L}$ and can be easily computed as

$$\det \mathbf{J}_{\psi_{k,s,t}} = \prod_{i=1}^{d-k} \exp\left[s_i\left(\mathbf{x}_{1:k}\right)\right]. \tag{3.6}$$

The inverse of $\psi_{k,s,t}$ and its Jacobian determinant are also easy to compute:

$$\begin{aligned}\psi^{-1}_{k,s,t}(\mathbf{y})_{1:k} &= \mathbf{y}_{1:k},\\ \psi^{-1}_{k,s,t}(\mathbf{y})_{k+1:d} &= \left(\mathbf{y}_{k+1:d} - t(\mathbf{y}_{1:k})\right) \odot \exp\left(-s(\mathbf{y}_{1:k})\right),\end{aligned}$$

and

$$\det \mathbf{J}_{\psi_{k,s,t}^{-1}} = \prod_{i=1}^{d-k} \exp\left[-s_i\left(\mathbf{y}_{1:k}\right)\right]. \tag{3.7}$$

If s and t are identity mappings, we recover the transformation in Example 3.2.1, which shows the ability to model tail dependence.

When chaining multiple coupling transformations, one technical detail requires attention. As (3.6) shows, the first k dimensions remain unchanged within a single coupling transformation. To ensure that dependencies among all dimensions can be captured, each dimension must be able to influence every other dimension. This is achieved by inserting permutation matrices that reorder the dimensions before applying the next coupling transformation (Dinh et al., 2016). Note that this imposes no significant additional computational cost, since the absolute determinant of any square permutation matrix is 1, and the permutation matrices can be randomly initialized once prior to training.

In the following, we refer to the NFCM whose g_θ is constructed using the coupling transformation method mentioned above as the Affine Coupling based Normalizing Flow Copula Model (AC-NFCM).

3.3.2. Adaptation of the CDF-Generator $\mathcal{S}_{\mathbf{X}}$

Once the base distribution and the transformation g_θ have been fixed, the corresponding CDF-generator can be obtained from the marginal distributions of the transformed base distribution. Recall that the CDF-generator $\mathcal{S}_{\mathbf{X}}$ maps the margins of $\mathbf{Y} := g_\theta(\mathbf{X})$ to the uniform distribution on $[0,1]$ and therefore consists elementwise of the marginal distribution functions

$$\mathcal{S}_{\mathbf{X}}\{g_\theta\}(y_1,\dots,y_d) = \left(F_{1,\theta}(y_1),\dots,F_{d,\theta}(y_d)\right)^\top,$$

where $F_{i,\theta}$ denotes the CDF of the i-th component of $g_\theta(\mathbf{X})$.

A straightforward and flexible approach is to represent these marginal distributions nonparametrically by drawing a large sample from the base distribution, transforming it with g_θ, and then using empirical estimates of the marginal CDFs and densities. Nonparametric estimation of marginal distributions is well established in copula modeling; see, e.g., Hofert et al. (2018, Section 4.1.2) and Genest et al. (1995).

We generate a synthetic sample

$$\{\tilde{\mathbf{x}}^{(1)},\dots,\tilde{\mathbf{x}}^{(N_{\text{syn}})}\} \subset \mathbb{R}^d,$$

from the base distribution, where $\tilde{\mathbf{x}}^{(m)} = (\tilde{x}_1^{(m)},\dots,\tilde{x}_d^{(m)})^\top$. Transforming the sample with g_θ yields

$$\tilde{y}_i^{(m)} := g_{\theta,i}(\tilde{\mathbf{x}}^{(m)}), \qquad m = 1,\dots,N_{\text{syn}},\ i = 1,\dots,d.$$

For each component i, we denote the corresponding order statistics by

$$\tilde{y}_{i,[1]} \leq \tilde{y}_{i,[2]} \leq \cdots \leq \tilde{y}_{i,[N_{\text{syn}}]},$$

where the subscript $[j]$ indicates the j-th order statistic of the sample

$$\{\tilde{y}_i^{(m)} \,:\, m = 1, \ldots, N_{\text{syn}}\}.$$

The marginal distribution function of the i-th component of $g_\theta(\mathbf{X})$ is then approximated by the empirical distribution function

$$\hat{F}_{i,\theta}(y) = \frac{1}{N_{\text{syn}}} \sum_{m=1}^{N_{\text{syn}}} \mathbf{1}_{\{\tilde{y}_i^{(m)} \leq y\}}.$$

For evaluating the AC-NFCM density, we also require the marginal densities and quantile functions. In copula modeling, when no parametric assumptions on the marginal distributions are made, the marginal densities are commonly estimated via kernel density estimation (see, e.g., Liebscher, 2005). Following this approach, we use a Gaussian kernel K_G with bandwidth b (see Silverman, 1986, Chapter 3):

$$\hat{g}_{i,\theta}(y) = \frac{1}{N_{\text{syn}}\, b} \sum_{m=1}^{N_{\text{syn}}} K_G\left(\frac{y - \tilde{y}_i^{(m)}}{b}\right).$$

The quantile function is obtained by linear interpolation between adjacent order statistics (Hyndman and Fan, 1996):

$$\hat{Q}_{i,\theta}(p) = (1 - \ell)\, \tilde{y}_{i,[j]} + \ell\, \tilde{y}_{i,[j+1]}, \qquad j = \lfloor N_{\text{syn}} p \rfloor, \quad \ell = N_{\text{syn}} p - j. \tag{3.8}$$

The corresponding quantile density function is given by the reciprocal of the marginal density evaluated at the quantile (Jones, 1992):

$$\hat{q}_{i,\theta}(p) = \frac{1}{\hat{g}_{i,\theta}\big(\hat{Q}_{i,\theta}(p)\big)}. \tag{3.9}$$

3.3.3. Estimation

Estimation is done in the usual copula setup, where we perform maximum likelihood estimation based on ranked pseudo-observations (see Section 3.2.1). The only special aspect in our case is that the copula is represented by the neural network model AC-NFCM. We split the pseudo-observations into a training set and a validation set. The validation set

is not used directly for parameter estimation but serves to monitor the training process and to implement early stopping. Maximum likelihood estimation is carried out on the training set only. Since the model is specified by a (possibly high-dimensional) parameter vector θ (the weights of the neural networks), optimization is performed using mini-batch gradient descent as in standard deep learning (Kingma and Ba, 2015).

It remains to specify the log-likelihood function. Let $\{\hat{\mathbf{u}}^k\}_{k=1}^N$ be the pseudo-observations in $(0,1)^d$. For a given parameter vector θ, the copula density of the AC-NFCM at $\hat{\mathbf{u}}^k = (\hat{u}_1^k, \dots, \hat{u}_d^k)^\top$ can be numerically computed via the density formula (3.5), using the marginal transforms from Section 3.3.2. Writing $\hat{Q}_{i,\theta}$ for the estimated marginal quantile function and $\hat{g}_{i,\theta}$ for the corresponding marginal density estimator of the i-th component of $g_\theta(\mathbf{X})$, we obtain the log-density

$$\begin{aligned}\log c_{\mathbf{X},\theta}(\hat{\mathbf{u}}^k) \approx \log\Big|\det J_{g_\theta^{-1}}\big(\hat{Q}_{1,\theta}(\hat{u}_1^k), \dots, \hat{Q}_{d,\theta}(\hat{u}_d^k)\big)\Big| \\ + \log f_{\mathbf{X}}\big(g_\theta^{-1}(\hat{Q}_{1,\theta}(\hat{u}_1^k), \dots, \hat{Q}_{d,\theta}(\hat{u}_d^k))\big) \\ - \sum_{i=1}^{d} \log \hat{g}_{i,\theta}\big(\hat{Q}_{i,\theta}(\hat{u}_i^k)\big).\end{aligned}$$

Here, $f_{\mathbf{X}}$ denotes the density of the base distribution and $J_{g_\theta^{-1}}$ the Jacobian of the inverse transformation g_θ^{-1}. The first two terms correspond to the change of variables from $\mathbf{X}$ to $g_\theta(\mathbf{X})$, and the last term subtracts the contribution of the marginal transformations, see (3.5). For a mini-batch $\mathcal{B}$ of size N_b consisting of pseudo-observations $\{\hat{\mathbf{u}}^k : k \in \mathcal{B}\}$, the mini-batch log-likelihood is then given by

$$L(\theta; \mathcal{B}) = \sum_{k \in \mathcal{B}} \log c_{\mathbf{X},\theta}(\hat{\mathbf{u}}^k) \tag{3.10}$$

and the corresponding loss to be minimized is $-L(\theta; \mathcal{B})$.

The form of the likelihood function illustrates the term "normalizing" in the name "normalizing flow": the pseudo-observations are mapped back, via the estimated marginal quantile functions and the inverse flow, to the base density $f_{\mathbf{X}}$, which is typically chosen as a simple reference distribution such as the multivariate standard normal.

Given the log-likelihood, the parameter vector θ is iteratively optimized during training. At each step, gradient descent is performed on a mini-batch of the training data, and the parameters of the neural networks are updated according to the gradient of the loss. After updating the model parameters with the current batch, the CDF-generator, which here appears through the quantile functions $\hat{Q}_{i,\theta}$, is recomputed as described in Section 3.3.2. During the iterative procedure, the validation data are used to dynamically determine the optimal number of training epochs, to adjust the learning rate, and to terminate the esti-

mation early to prevent overfitting (Prechelt, 1998). The overall procedure is summarized in Algorithm 2.

Algorithm 2: Optimizing AC-NFCM with Mini-Batch Gradient Descent.

Input: Predefined AC-NFCM $(\mathbf{X}, \Psi_\Theta)$, copula training dataset (pseudo-observations), copula validation dataset, mini-batch size N_b, maximum epochs N_E, learning rate reduction factor γ, initial learning rate η_0

Output: Optimized parameters θ_{opt} of the AC-NFCM

```
Initialize parameters θ;
Set learning rate η ← η_0;
for epoch = 1, 2, ..., N_E do
    for each mini-batch B in the copula training dataset do
        Determine the marginal quantile and quantile density functions of g_θ(X)
         (see (3.8) and (3.9));
        Compute log-likelihood L(θ; B) (see (3.10));
        Update parameters θ ← θ − η ∇_θ(−L(θ; B));
    if validation dataset shows no improvement then
        Reduce learning rate η ← γη;
    if early stopping criterion is met on the copula validation dataset then
        break;
```

Finally, it should be noted that in addition to the actual (hyper)parameters of the neural networks, the number of coupling transformations used and the number of synthetic data points generated for estimating the marginal distributions during and after training are also important hyperparameters in the proposed AC-NFCM. In practice, hyperparameters for a given dataset can be selected via standard hyperparameter optimization methods such as grid search or random search (see the discussion in Feurer and Hutter, 2019).

The following section presents simulation results for AC-NFCM, highlighting its performance in comparison to other nonparametric copula models.

3.4. Simulation Study

In this section, we compare the AC-NFCM to different nonparametric models such as kernel estimators for bivariate copulas (KDE; Nagler, 2018) and vine copula based kernel density estimator for multidimensional data (Vine KDE; Nagler, 2024), the empirical checkerboard copulas (CB_5, CB_{10}; Cuberos et al., 2020), the empirical Bernstein copulas

(BS_{10}, BS_{25}; Sancetta and Satchell, 2004) and the empirical beta copula (Beta; Segers et al., 2017). For two-dimensional cases the AC-NFCM has 4 coupling transformations and for multi-dimensional cases it has 6 coupling transformations. For further implementation details on the AC-NFCMs and baseline estimators considered, see Appendices A.2 and A.3. In our simulation study, various configurations of hyperparameter settings (including the number of coupling layers) were tested in preliminary investigations. The settings described in the implementation details proved to be robust with respect to different copula classes. For practical applications, we therefore recommend using these settings initially. For higher-dimensional problems, it may be useful to increase the number of coupling layers accordingly in order to improve model capacity and flexibility.

We perform experiments for bivariate copulas as well as for multivariate copulas with dimensions $d = 3, 4, 5$. First, we simulate 100 samples of size $n = 500$ with a target density c. Then, for each sample, we estimate the different copula estimators. We also test the performance of the estimators when the sample size is small with $n = 100$, but we are aware that this is a very small sample size for flexible non-parametric estimators, in particular for the multivariate cases.

For each replication, the full sample of size n is used as the estimation sample for all methods. When training the AC-NFCM, this sample is internally split into 80% data used to update the model parameters and 20% used as a validation set for early stopping and learning-rate reduction (see Algorithm 2). The classical nonparametric competitors (KDE, Vine KDE, CB, BS, Beta) do not require a validation set and are fitted on all n observations. Thus, all estimators are based on the same number of sample points.

To evaluate the performance of an estimator $\hat{c}$ of the copula density c, we use the integrated absolute error (IAE), that is, the L^1-distance between $\hat{c}$ and c on $[0,1]^d$:

$$\mathrm{IAE}[\hat{c}] = \int_{[0,1]^d} |\hat{c}(\mathbf{u}) - c(\mathbf{u})| \, d\mathbf{u}.$$

This criterion summarizes the global discrepancy between $\hat{c}$ and c and is widely used in nonparametric density estimation.

Since this integral is not available in closed form, we approximate it by Monte Carlo methods. A naive estimator based on uniformly sampled points on $[0,1]^d$ can suffer from high variance, in particular for higher dimensions. To obtain more stable and efficient estimates, we therefore adopt the importance sampling approach of Nagler and Czado (2016), using the true density c as the sampling distribution. We generate $N = 1,000$

Monte Carlo samples and approximate the IAE by

$$\widehat{IAE} \approx \frac{1}{N} \sum_{i=1}^{N} \frac{1}{c(\mathbf{u}_i)} \left| c(\mathbf{u}_i) - \hat{c}(\mathbf{u}_i) \right| .$$

3.4.1. Finite Sample Performance in the Bivariate Case

To examine the performance of the estimators with respect to different types of dependence, such as tail behavior, asymmetry, and extreme value behavior, we choose the following bivariate copulas, where lower and upper tail dependence are given in the parentheses (for details on the definitions and plots of the synthetic samples, see Appendix A.4). Note that we included two extreme value copulas (Galambos and t-EV). These are a special class of copulas that arise as the limiting distributions of componentwise maxima and play a fundamental role in multivariate extreme value theory (Gudendorf and Segers, 2010).

- Clayton copula with $\theta = 2$ ($\lambda_L \approx 0.707, \lambda_U = 0$);
- Gumbel copula with $\theta = 2.7$ ($\lambda_L = 0, \lambda_U \approx 0.707$);
- Gaussian copula with $\rho = 0.5$ ($\lambda_L = \lambda_U = 0$);
- t-copula with $\nu = 2$, and $\rho = 0.5$ ($\lambda_L = \lambda_U \approx 0.4$);
- Galambos copula with $\theta = 0.5$ ($\lambda_L = 0, \lambda_U \approx 0.25$);
- t-EV copula with $\rho = 0.5$ and $\nu = 4$ ($\lambda_L = 0, \lambda_U \approx 0.253$);
- kho_1: A Khoudraji device combines a Clayton copula and a Gumbel copula using the shape vector $s = (0.4, 0.95)$. The parameters of the Clayton and Gumbel copulas are both $\theta = 6$.
- kho_2: A Khoudraji device combines an independence copula with a Clayton copula using the shape vector $s = (0.95, 0.6)$. The parameter of the Clayton copula is $\theta = 2$.

Table 3.1 and Figure 3.2 show the results of experiments with moderate sample size (n=500). Overall, the kernel density estimator and the AC-NFCM clearly outperform the empirical checkerboard copulas, the empirical Bernstein copulas and the empirical beta copula for all considered copula families. While the AC-NFCM performs slightly better than the kernel density estimator for the two extreme-value copula families, Galambos and t-EV copula, the kernel density estimator provides significant better results for the remaining copula families. Furthermore, we see in Figure. 3.2 that although the AC-NFCM is based on neural networks and could therefore exhibit the variance associated with the

	Estimators						
Copulas	CB_5	CB_{10}	BS_{10}	BS_{25}	Beta	KDE	AC-NFCM
Clayton	0.300	0.353	0.262	0.184	0.485	**0.099**	0.108
Gumbel	0.384	0.363	0.398	0.239	0.430	**0.093**	0.102
Gaussian	0.207	0.329	0.101	0.110	0.557	**0.062**	0.069
t	0.298	0.374	0.186	0.160	0.541	**0.125**	0.132
Galambos	0.182	0.333	0.081	0.112	0.579	0.073	**0.066**
t-EV	0.184	0.330	0.083	0.115	0.582	0.069	**0.067**
kho1	0.438	0.441	0.418	0.339	0.512	**0.236**	0.262
kho2	0.205	0.332	0.126	0.120	0.568	**0.082**	0.103

Table 3.1.: Median IAE of the estimation results for bivariate copulas by using different non-parametric estimators for 100 samples with a size of 500 each. Bold entries denote the best result.

method (e.g., due to the random initialization of the neural networks), it shows a similarly small variance as the KDE. For the experiments with a small sample size ($n = 100$), the KDE still achieves the smallest median IAE across all copula families. Compared to the case $n = 500$, however, the empirical Bernstein estimators become considerably more competitive. AC-NFCM and, in particular, the Bernstein estimator BS_{10} show a comparable level of accuracy: AC-NFCM attains slightly smaller median IAEs for the Clayton, Gumbel, t and kho1 copulas, whereas BS_{10} performs slightly better for the Gaussian, Galambos, t-EV and kho2 copulas. However, a larger variance in the IAE results of AC-NFCM can be observed, which can be attributed to the nature of neural networks when dealing with small sample sizes (see Figure 3.3).

Since the IAE is an aggregated metric that summarizes performance information across the entire data distribution and, thus, only represents an average performance tendency, various aspects are considered below in order to comprehensively evaluate the performance of the estimates. In the following, we present the results for experiments with sample size $n = 500$. For clarity, we restrict the following analysis to the BS_{25}, Beta, KDE, and AC-NFCM estimators, which most frequently achieve the smallest IAE values across the considered copulas (see Table 3.1). The checkerboard estimators (CB_5, CB_{10}) typically yield the largest IAE values and are therefore omitted from the detailed discussion. Moreover, we only report BS_{25}, since BS_{10} exhibited a very similar behaviour in preliminary experiments. The results for experiments with sample size $n = 100$ can be found in Appendix A.5.

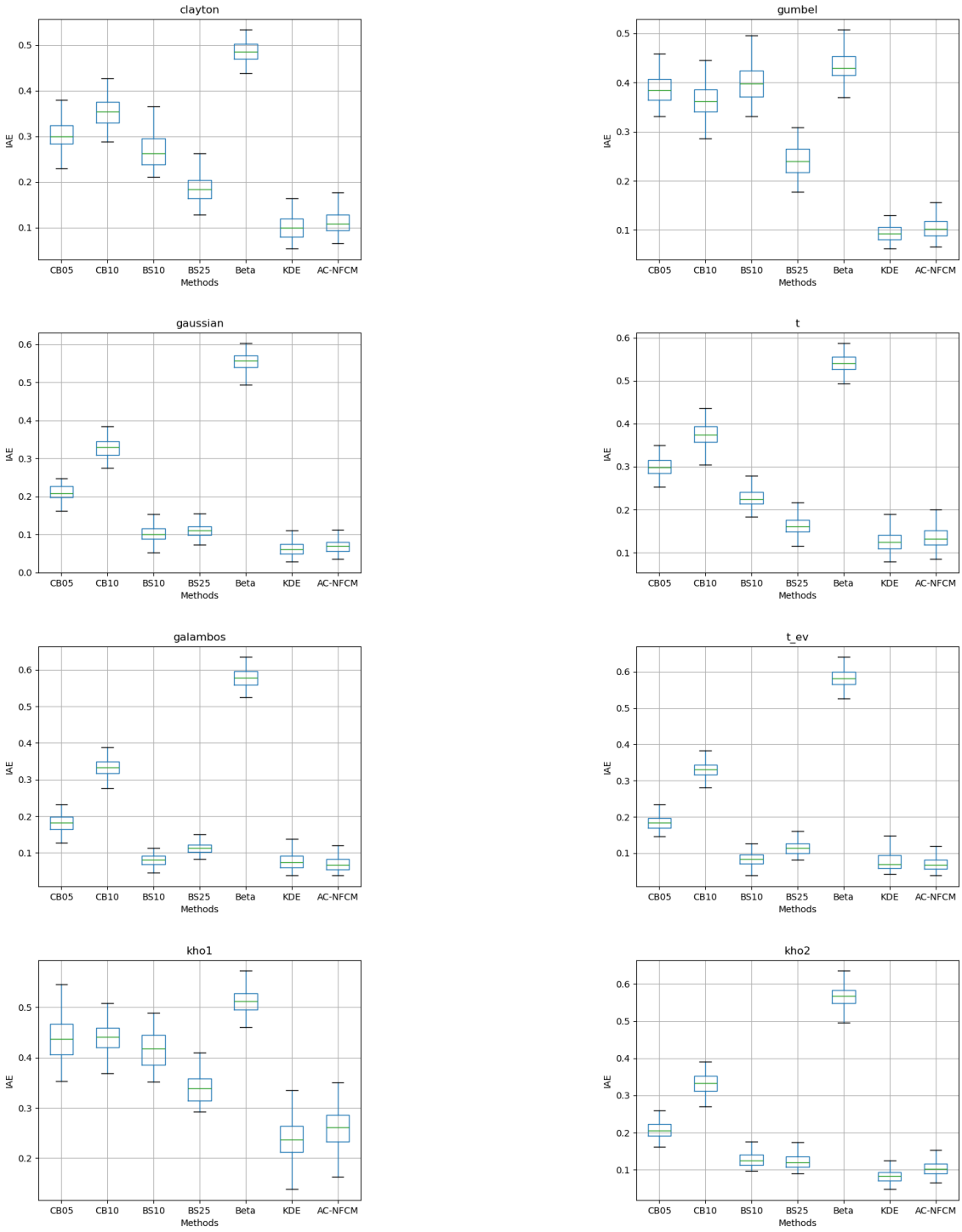


Figure 3.2.: Boxplots of IAE results from different estimators for 100 samples with a size of 500 each.

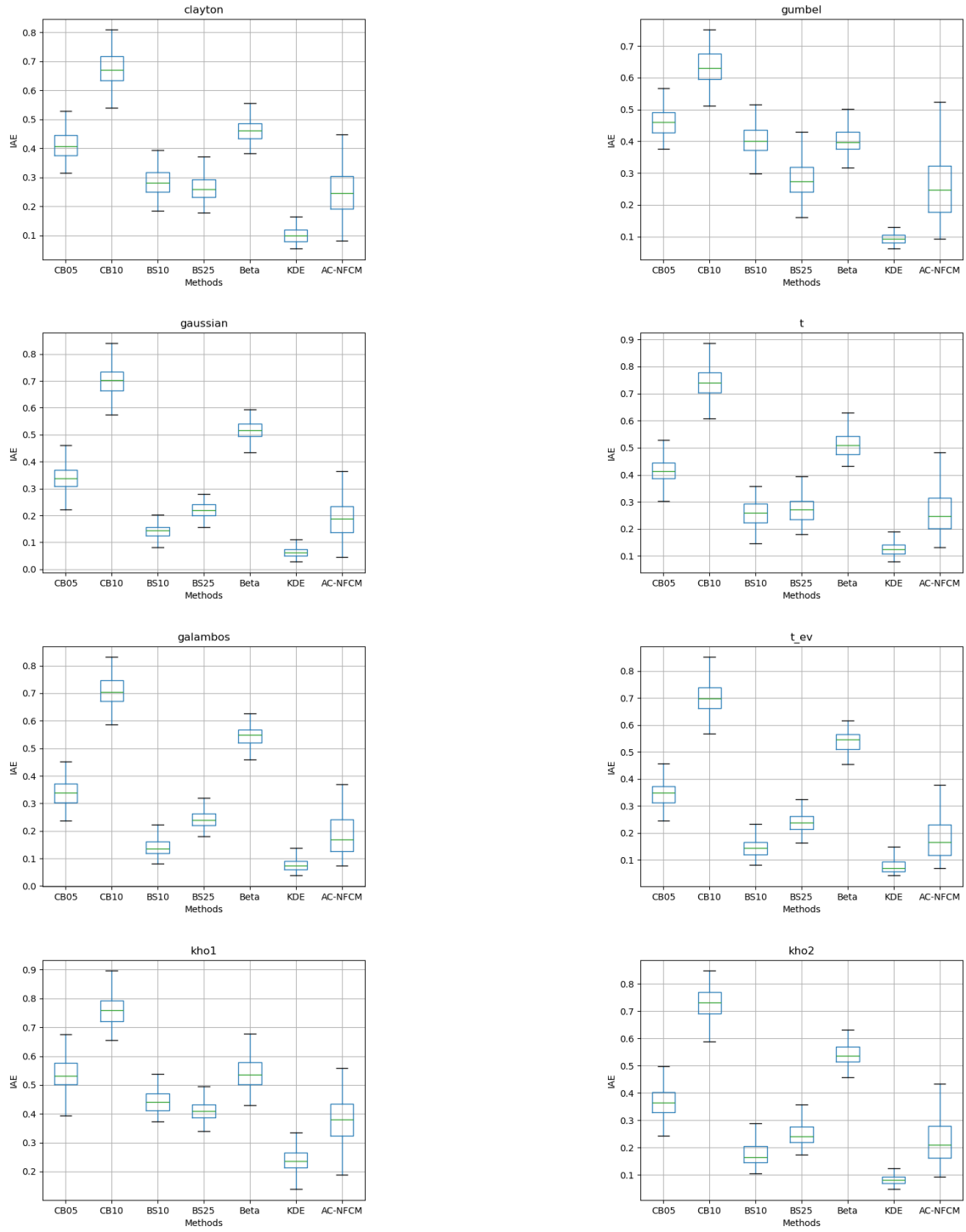


Figure 3.3.: Results for small sample size: Boxplots of IAE results from different estimators for 100 samples with a size of 100 each.

To assess how well the models reproduce tail dependence, we examine the empirical conditional tail probabilities, a diagnostic tool commonly used to visualize tail behavior in copula and extreme-value modeling (Frahm et al., 2005).

$$\mathrm{ETD}_n(t) = \frac{\sum_{i=1}^{n} \mathbb{1}(u_i \leq t, v_i \leq t)}{\sum_{i=1}^{n} \mathbb{1}(u_i \leq t)} \mathbb{1}_{(0,0.5]}(t) + \frac{\sum_{i=1}^{n} \mathbb{1}(u_i \geq t, v_i \geq t)}{\sum_{i=1}^{n} \mathbb{1}(u_i \geq t)} \mathbb{1}_{(0.5,1)}(t). \tag{3.11}$$

We generate synthetic samples (of 10,000 data points each) with the estimated models and use (3.11) to evaluate the empirical approximation functions of the tail dependence coefficients based on the synthetic data.

Figure 3.4 clearly shows that among the estimators considered, only the AC-NFCM is able to describe the tail behavior when tail dependence is present, while all estimators are able to describe the tail behavior very well when no tail dependence is present.

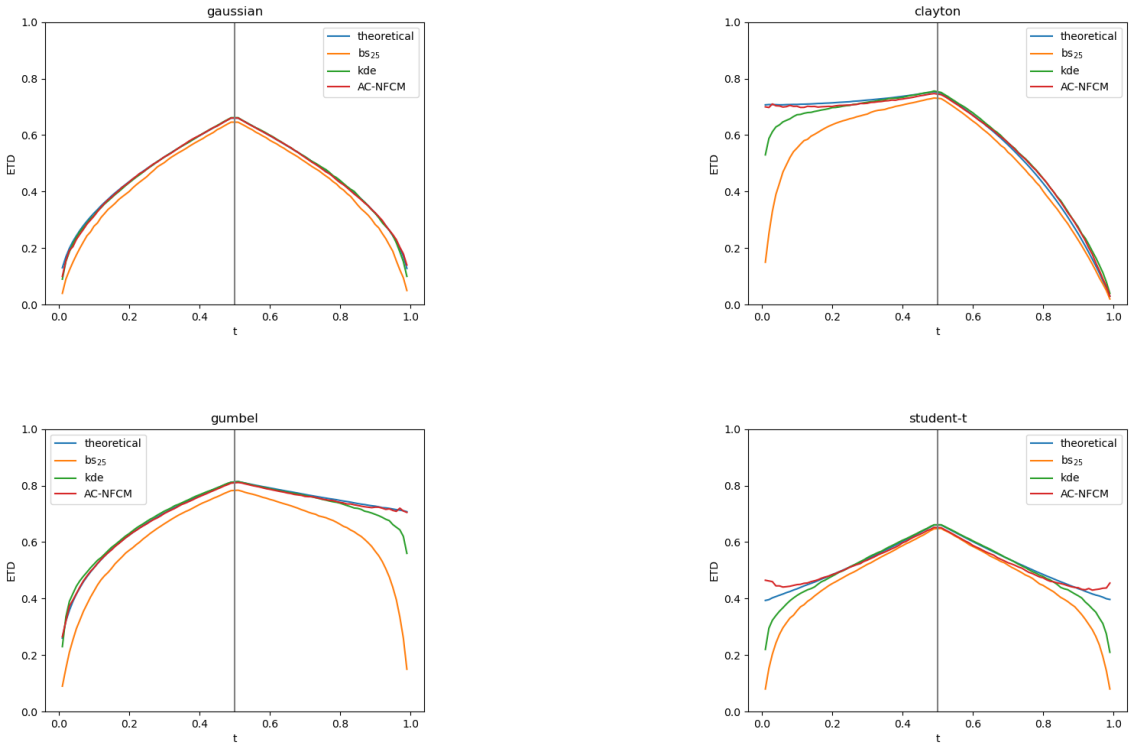


Figure 3.4.: Tail behavior of different non-parametric estimators, evaluated on synthetic data generated from the estimators: Gaussian copula data (top left), Clayton copula data (top right), Gumbel copula data (bottom left) and Student-t copula data (bottom right). We generate a total of 100 samples of size 10,000 and plot the median values here.

To evaluate how well the two extreme value copulas are learned, we consider the (empirical) Pickands dependence functions, which can be compared to the theoretical curves, since the Pickands dependence functions characterize the associated extreme value copulas (see also Appendix A.4.2). To do this, we first generate synthetic samples from the copula estimators and determine the Pickands dependence functions of the samples using the methods implemented in Hofert et al. (2024). The AC-NFCM provides the best fits for the Pickands dependence functions of the Galambos and t-EV copula as can be seen from Figure 3.5.

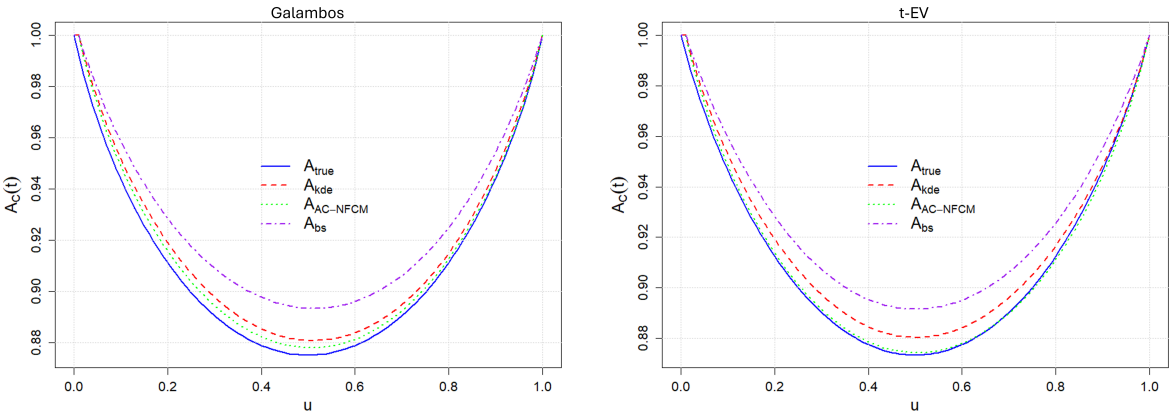
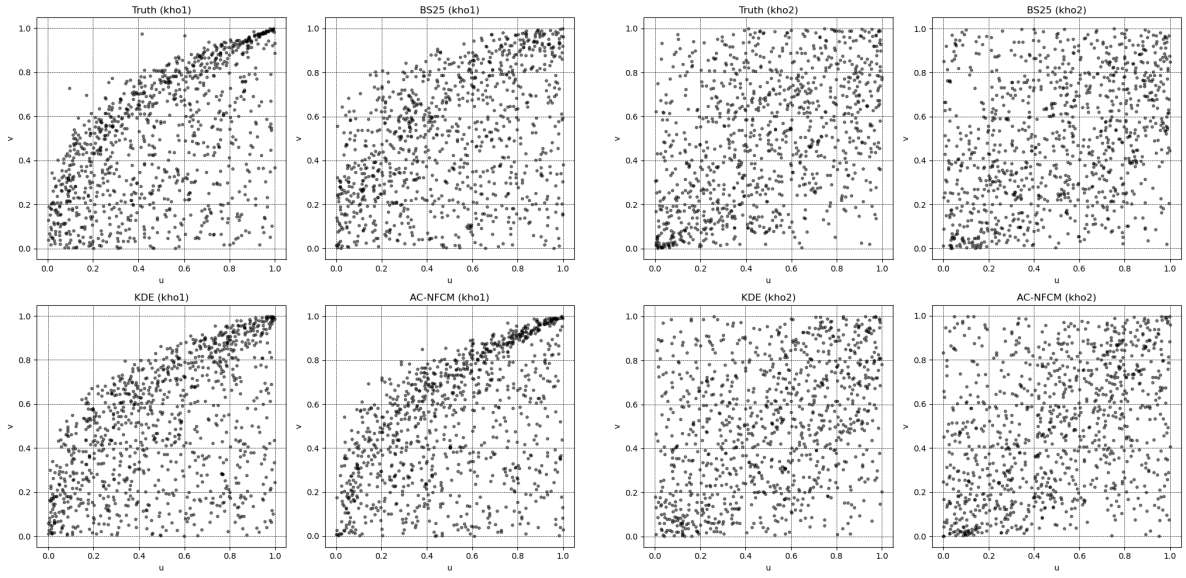


Figure 3.5.: Estimation of Pickands dependence functions by using synthetic data generated from different estimators. A total of 100 samples are generated for each copula family. The mean values of the determined curves are shown. Left panel: Galambos copula, right panel: t-EV copula.

Finally, we consider two copulas with distinctive structures, particularly shaped by the interplay of tail dependence and asymmetry (see Figure 3.6). Here as well, we observe that the AC-FCM successfully captures the underlying structure.

3.4.2. Finite Sample Performance in the Multivariate Case

To evaluate the performance for multivariate data, we test the AC-NFCM on 3-, 4- and 5-dimensional Clayton, Gumbel and Gaussian copula data. To ensure comparability of the different copula families, we choose the copula parameters so that all pair-wise Kendall's τ are equal to $\frac{1}{3}$. This means we choose Gaussian copula parameter $\rho = 0.5$, Clayton copula parameter $\theta = 1$ and Gumbel copula parameter $\theta = 1.5$. In addition, we construct a complex dependence structure using an R-vine comprising Gaussian, Clayton and Gumbel copulas with various parameters. For the specific definition of structure matrix and pa-



(a) kho1 **(b)** kho2

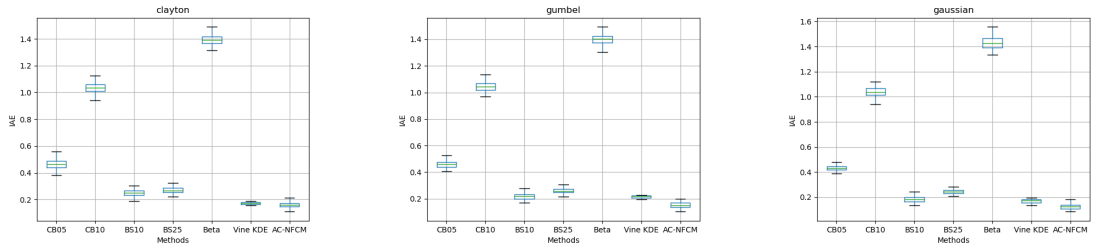
Figure 3.6.: Further copula examples with complex dependency structure. Panel a shows synthetic datasets generated by estimators from the `kho1` copula data, while Panel b corresponds to estimators from the `kho2` copula data.

rameter choices see Appendix A.4. The results for experiments with sample size $n = 100$ can be found in Appendix A.5; however, these small-sample results are as expected less conclusive, since all estimators show relative large errors, and we therefore focus on the $n = 500$ setting here.

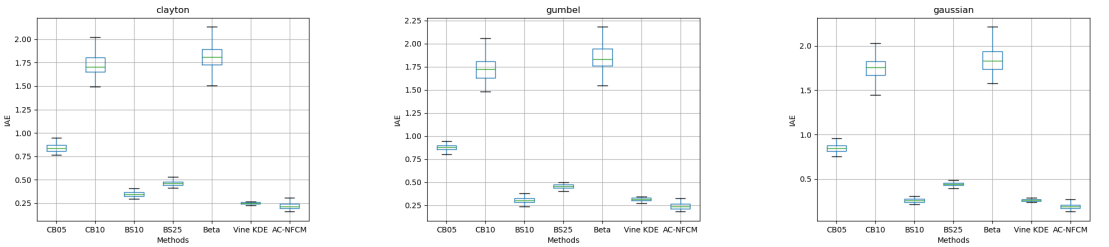
Similarly to the bivariate copulas, the AC-NFCM and the Vine-KDE outperform other non-parametric density estimators, with AC-NFCM showing slightly better results than Vine-KDE. Additionally, there is a tendency to recognize that the increase in error with increasing dimension is slower for AC-NFCM than for the Vine-KDE estimator (see Table 3.2). Figure 3.7 shows the boxplots of the IAEs. A similar variance of the KDE estimator and AC-NFCM can be observed here. Notably, the performance of the CB10 and Beta copulas decreases significantly with increasing dimensionality. Due to the combination of a small sample size and fine grid subdivision, the problem arises that, in many areas, the estimated density is zero while, in the few occupied cells, the estimated density is highly concentrated. This results in significant estimation inaccuracy and high dispersion of the IAE across experiments. Interestingly, the AC-NFCM clearly outperforms the Vine-KDE model for the complex 5-dimensional vine copula with respect to the median IAE. An example of the synthetic data generated is shown in Figure 3.8.

		Estimators						
Dimension	Copula	CB_5	CB_{10}	BS_{10}	BS_{25}	Beta	Vine KDE	AC-FCM
3	Clayton	0.466	1.034	0.249	0.269	1.391	0.174	**0.160**
	Gumbel	0.460	1.042	0.219	0.258	1.405	0.217	**0.148**
	Gaussian	0.433	1.039	0.182	0.245	1.429	0.173	**0.124**
4	Clayton	0.837	1.704	0.344	0.462	1.810	0.249	**0.216**
	Gumbel	0.875	1.722	0.299	0.457	1.830	0.313	**0.237**
	Gaussian	0.843	1.758	0.255	0.445	1.831	0.263	**0.187**
5	Clayton	1.370	1.860	0.459	0.714	1.877	0.351	**0.333**
	Gumbel	1.419	1.897	0.396	0.704	1.701	0.416	**0.325**
	Gaussian	1.408	1.931	0.347	0.705	1.750	0.345	**0.296**
	Vine	1.073	1.561	1.242	0.919	1.561	0.592	**0.467**

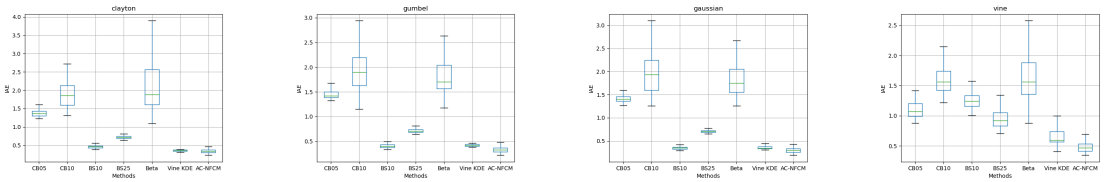
Table 3.2.: Median IAE of the estimation results for multivariate copulas using different non-parametric estimators for 100 samples with a size of 500 each. Median IAE is evaluated with importance sampling.



(a) 3D Experiments.



(b) 4D Experiments.



(c) 5D Experiments.

Figure 3.7.: Boxplots of IAE results from different estimators for 100 samples with a size of 500 each. First row: Results of the three dimensional experiments; second row: results of the four dimensional experiments; third row: results of the five dimensional experiments.

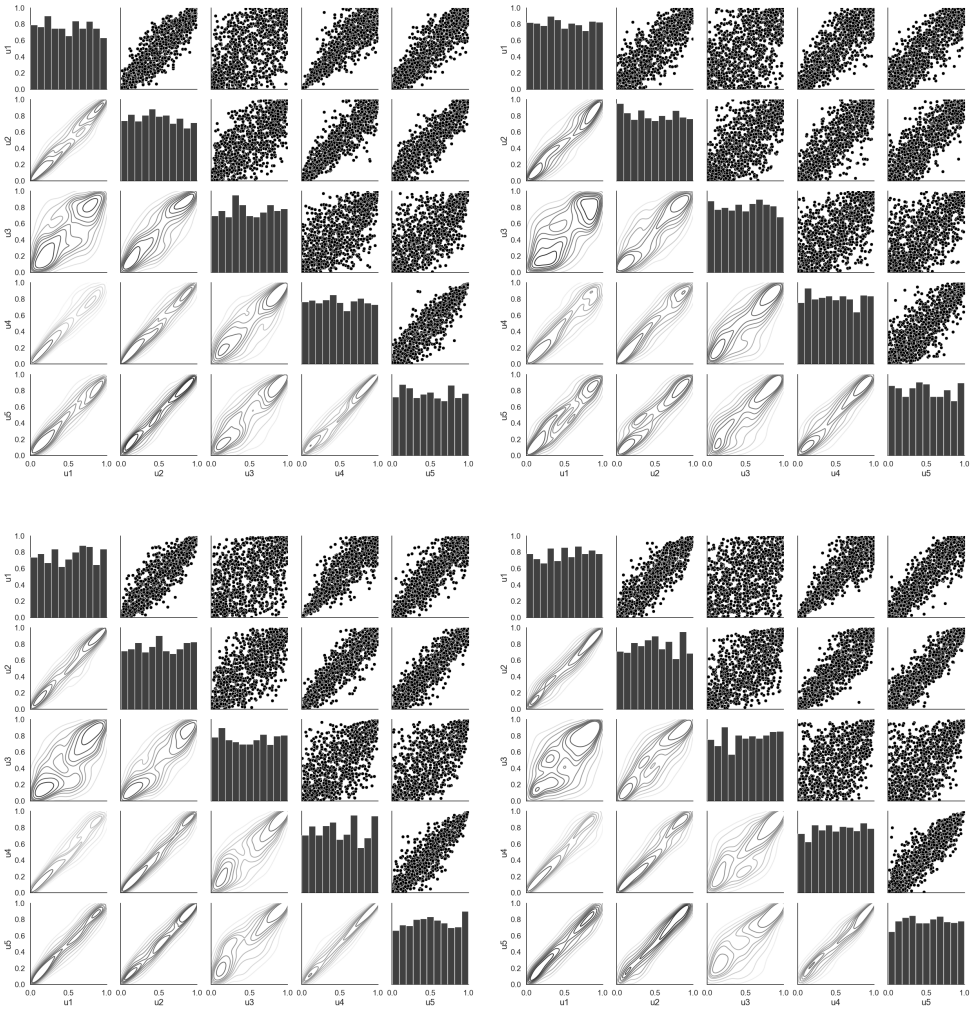


Figure 3.8.: Synthetic data generated by different estimators estimated from the 5D vine copula data. Top left: true sample; top right: BS_{25}; bottom left: Vine KDE, bottom right: AC-NFCM.

The following section presents an illustrative application of AC-NFCM to real-world data.

3.5. Real Data Examples

To evaluate the performance of the proposed AC-NFCM with respect to real-world data, we apply our methodology to an insurance dataset and a dataset from engineering. Implementation details can be found in Appendix A.2.

3.5.1. Loss-ALAE Dataset

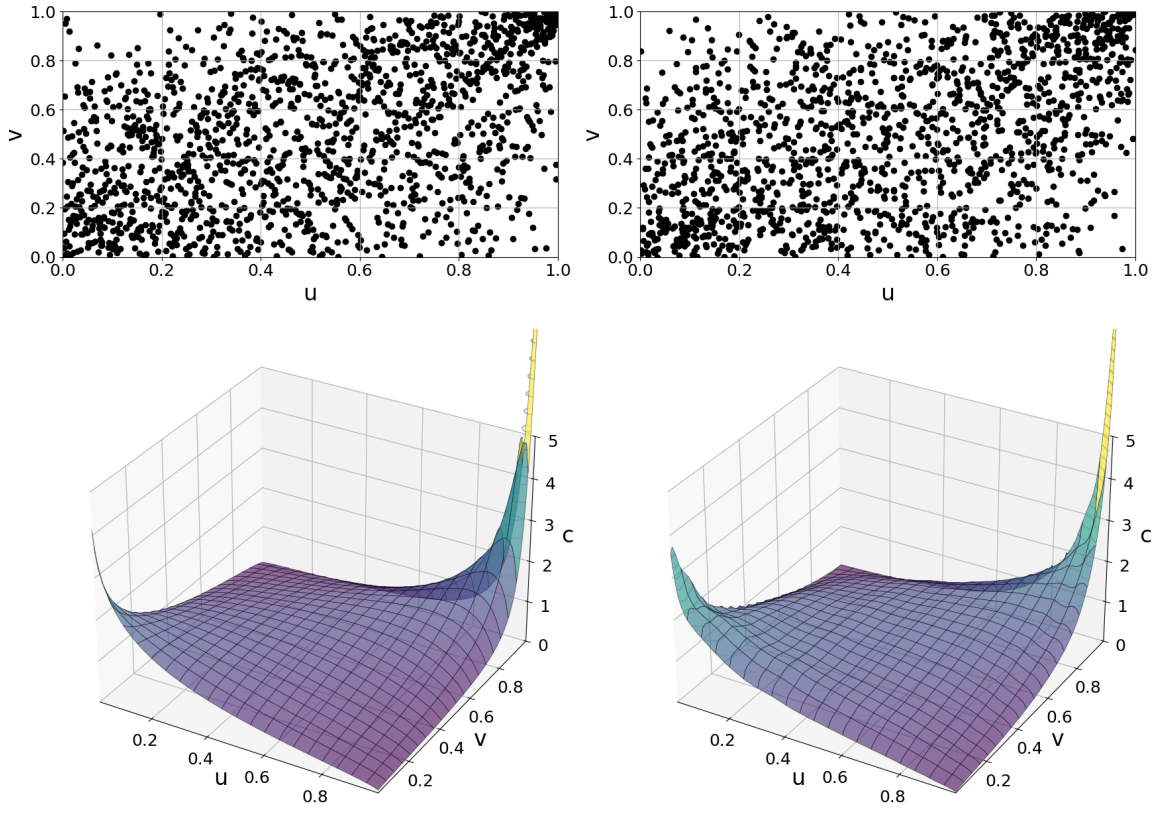


Figure 3.9.: Performance comparison on the Loss-ALAE dataset: Pseudo-observations (top left), 1500 synthetic data points generated by an AC-NFCM model (top right), proposed Gumbel copula density with parameter $\theta = 1.455$ (bottom left), and density estimation generated by the flow copula model (bottom right).

The first dataset we consider is the Loss-ALAE dataset (Frees and Valdez, 1998). The dataset consists of 1,500 random general liability claims, each consisting of the indemnity payment (Loss) and the allocated loss adjustment expenses (ALAE)[1]. We estimate an AC-NFCM with 4 coupling transformations. In Figure 3.9, the synthetic data generated by the AC-NFCM is presented. The generated synthetic data clearly reflects the characteristics

[1]To generate the pseudo observation of the real data we use the method implemented in R-Package Copula (Hofert et al., 2024) which assigns the ranks of tied observations pseudo-randomly, since the original dataset contained ties. This makes it easier to compare the learned copula with the original dataset.

of the training data. Furthermore, the estimated density function clearly shows that the AC-NFM identifies upper tail dependence between Loss and ALAE. The literature has shown that this dataset can be modeled particularly well with a Gumbel copula (Frees and Valdez, 1998; Chen et al., 2010; Geenens et al., 2017). Our experiment also shows that the AC-NFCM learns a copula density that is similar to the Gumbel copula density, see Figure 3.9.

3.5.2. Fuel Drops Dataset

We consider a dataset that describes the fuel droplets generated by an injection nozzle in a jet engine. The dataset contains five variables: Drop size, x-position, y-position, x-velocity and y-velocity. In Coblenz et al. (2020), this dataset is successfully modeled using a vine copula. We estimate two AC-NFCMs with 6 and 12 coupling transformations (AC-NFCM (6) and AC-NFCM (12)), respectively (for details see Appendix A.2). In Figure 3.10, synthetic data from the AC-NFCMs are compared with the pseudo observations and synthetic data generated from the vine copula model proposed in Coblenz et al. (2020).

Overall, the vine copula, AC-NFCM (6) and AC-NFCM (12) are able to successfully model the essential copula structure. AC-NFCM (12) shows a significant improvement compared to AC-NFCM (6) (e.g., the dependence structure between x-velocity and y-position). This result is consistent with the general property of a neural network model that the deeper the layers, the more complex and powerful the model tends to be.

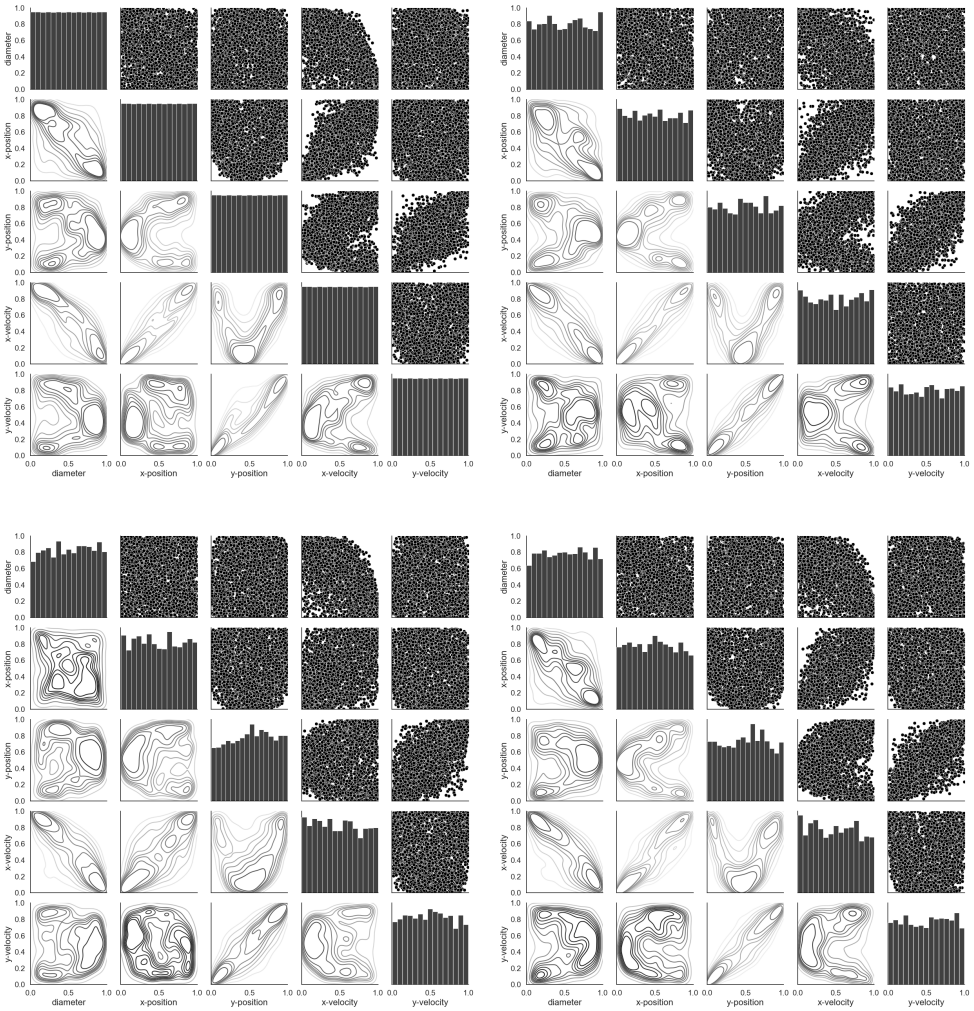


Figure 3.10.: Comparison of the data of the fuel droplet dataset: Pseudo observations (top left), data generated by vine copula (top right), data generated by AC-NFCM(6) (bottom left), data generated by AC-NFCM(12) (bottom right).

To quantitatively assess the estimation performance, we compare the joint distribution of the pseudo-observations and the generated synthetic data using two complementary goodness-of-fit measures. First, we apply a multivariate extension of the Kolmogorov-Smirnov (KS) test (Naaman, 2021) with the implementation in Laurent (2020). The KS statistic measures the maximal deviation between the empirical distribution functions of two samples; smaller values indicate a closer agreement of the two distributions. For a significance level of $\alpha = 0.05$, the corresponding critical value is 0.0926. If the KS statistic is below this critical value, the null hypothesis that both samples stem from the same distribution cannot be rejected at the 5% level. Second, we compute the Wasserstein distance (Flamary et al., 2021) between the pseudo-observations and the synthetic data. This

distance can be interpreted as the minimal "transport cost" required to transform one empirical distribution into the other and thus directly quantifies how close the two joint distributions are. Again, smaller values indicate a better match. The results in Table 3.3 show that AC-NFCM is able to generate synthetic samples that closely match the empirical dependence structure. In particular, AC-NFCM(12) attains the smallest KS statistic (well below the critical value for $\alpha = 0.05$) and the smallest Wasserstein distance, indicating that it captures the joint distribution most accurately among the considered models.

Measures	Vine Copula	AC-NFCM (6)	AC-NFCM (12)	Critical Value
KS-Statistics	0.032	0.043	**0.028**	0.0926
WS-Distance	0.140	0.160	**0.120**	-

Table 3.3.: Kolmogorov-Smirnov test results and Wasserstein distance between pseudo-observations and synthetic data.

3.6. Conclusion

This chapter introduces the normalizing flow copula model (NFCM), which parameterizes the copula through a learnable transformation of the base distribution and an associated CDF-generator acting on the transformed margins. We prove that increasing triangular transformations are sufficiently expressive to represent any absolutely continuous copula for a fixed base distribution, allowing the NFCM to be implemented without loss of generality using triangular maps. Based on this theoretical foundation, we propose a specific neural realization of the framework: the AC-NFCM, in which the transformation is represented by a normalizing flow consisting of affine coupling layers. Simulation studies demonstrate that the AC-NFCM successfully captures diverse copula families exhibiting dependence structures such as tail dependence, asymmetry, and extreme-value behavior. The model consistently outperforms nonparametric competitors such as empirical Bernstein copulas and achieves performance comparable to kernel-based copula density estimators in terms of median IAE. Notably, it is the only nonparametric approach among those considered that captures tail dependence. For multivariate copula data, the AC-NFCM yields particularly strong results, outperforming all competing models in terms of IAE. As with all flexible nonparametric methods, performance degrades for very small sample sizes. Applications to real datasets corroborate these findings, confirming that the normalizing flow copula model offers a flexible and robust tool for multidimensional copula modeling.

4. Neural Network-Based Copula Modeling via Perturbations of Independence

This chapter is based on joint work with Maximilian Coblenz and Oliver Grothe (Publ. II). We propose a neural-network-based estimator for copula densities, motivated by the idea of representing a copula as a perturbation of the independence copula. We prove that the model has sufficient expressive power to capture any square-integrable copula density.

4.1. Introduction

Modeling dependence structures between random variables often requires a flexible approach that goes beyond classical linear correlation. A rigorous framework is provided by copula theory, based on Sklar's Theorem (Sklar, 1959): any multivariate distribution F with continuous marginals $F_1, \dots, F_d$ can be uniquely written as $F(x_1, \dots, x_d) = C(F_1(x_1), \dots, F_d(x_d))$, where the copula $C : [0,1]^d \to [0,1]$ characterizes dependence separately from marginals. While numerous parametric copula families exist (for an introduction, see Nelsen, 2006; Durante and Sempi, 2015; Hofert et al., 2018), their ability to capture complex dependence patterns is inherently limited, motivating the development of nonparametric methods. To this end, we build on perturbation-based copula models reviewed below and develop a neural network-based estimator that learns the copula density directly from data without a priori assumptions.

Nonparametric copula density estimation has been approached in various ways, including kernel-based methods (Gijbels and Mielniczuk, 1990; Charpentier et al., 2007; Geenens et al., 2017; Wen and Wu, 2020; Nagler, 2018) and smoothing techniques such as the empirical beta copula (Segers et al., 2017) and the empirical Bernstein copula (Sancetta and Satchell, 2004; Weiß and Scheffer, 2012); for further methods and surveys, see Genest et al. (2009), Papapantoleon (2015), Gribkova and Lopez (2015), Ngounou Bakam and Pommeret (2025), Choroś et al. (2010), and Provost and Zang (2024).

The method proposed in this chapter rests on separable representations of copulas, which we now briefly review. Starting from the independence copula $\Pi(u, v) = uv$, one can introduce dependence through separable additive perturbations. Rodrıguez-Lallena and Úbeda-Flores (2004) introduce copulas of the form $C(u, v) = uv + F(u)G(v)$, generalizing the Farlie-Gumbel-Morgenstern family. Mesiar and Najjari (2014) extend this to finite sums $C(u, v) = uv + \sum_{k=1}^{K} \sigma_k F_k(u) G_k(v)$, deriving closed-form expressions for dependence measures such as Kendall's tau, Spearman's rho, and Gini's gamma, as well as tail behavior. Separable perturbation representations have also been studied directly on the density level. Mukhopadhyay and Parzen (2020) consider copula densities of the form $c(u, v) = 1 + \sum_{j,k\geq 1} \rho_{jk} f_j(u) g_k(v)$, where f_j and g_k are Gram-Schmidt orthonormalized polynomials derived from mid-ranks. Ngounou Bakam and Pommeret (2024) use the same representation with Legendre polynomials for hypothesis testing. For symmetric copulas, Longla (2024) use spectral theory to show that all absolutely continuous symmetric copulas with square-integrable densities admit an infinite separable representation with orthonormal eigenfunctions. Grothe and Rieger (2024) derive separable representations via singular value decomposition for checkerboard copulas, whose densities are piecewise constant. Additionally, they discuss continuous decompositions and provide the representation of the Gaussian copula in terms of transformed Hermite polynomials. Further extensions consider general base copulas (Saminger-Platz et al., 2021) or non-separable perturbations (Komorník et al., 2017).

Overall, this line of work has mainly focused on analytical properties and the construction of new parametric families, raising the question of whether such separable representations can be learned directly from data. We address this question with a neural network-based estimator for copula densities of the form $c(u, v) = 1 + \sum_{k=1}^{K} f_k(u) g_k(v)$. The architecture guarantees marginal uniformity by construction, while nonnegativity of the density is enforced via a penalty term. On the theoretical side, we show that the L^2-approximation error to any square-integrable copula density vanishes asymptotically for bounded nonnegative separable expansions, confirming the expressive power of this representation class. In simulations, the estimator performs competitively with kernel density estimation and outperforms existing methods on copulas with local structure. On real data, a vine copula based on the proposed estimator achieves the best fit among the considered methods.

The remainder of the chapter is structured as follows. Section 4.2 introduces the estimation procedure and the theoretical analysis of expressive power. Sections 4.3 and 4.4 present simulation studies and a real-data application. Section 4.5 concludes. Supplementary code for this chapter is available at Liu et al. (2026a).

4.2. Copula Density Estimation by Perturbing Independence

In this section, we develop a neural network-based approach for estimating copula densities from data, where the learned densities take the form

$$c(u,v) = 1 + \sum_{k=1}^{K} f_k(u)\, g_k(v). \tag{4.1}$$

The component functions f_k and g_k are learned directly from data using neural networks. This representation corresponds to a separable perturbation of the independence copula density. As shown in Section 4.2.3, separable expansions of this type arise naturally from the Schmidt decomposition, which represents any square-integrable copula density as an infinite series. Since any computational method must truncate the expansion to finitely many terms, we study the conditions under which such truncated expansions define valid copula densities. In Section 4.2.1, we show that so-called zero-mean constraints on the component functions ensure uniform marginals, and that boundedness is necessary for nonnegativity when K is finite. However, truncating the Schmidt decomposition of a copula density does not in general yield a valid copula density, since nonnegativity may be violated. Section 4.2.2 develops a neural network architecture with finite K that enforces these constraints by construction. Finally, Section 4.2.3 establishes that the L^2-approximation error of bounded nonnegative separable expansions vanishes asymptotically, so neither the boundedness constraint nor the nonnegativity constraint limits approximation capacity.

4.2.1. Conditions for Valid Copula Densities

We begin by identifying sufficient conditions: the following proposition shows that zero-mean constraints on the component functions, together with nonnegativity of the resulting density, are sufficient.

Proposition 4.2.1. *Let f_k, g_k be bounded functions on $[0,1]$ satisfying*

$$\int_0^1 f_k(t)\, dt = 0 \quad \textit{and} \quad \int_0^1 g_k(s)\, ds = 0 \quad \forall k = 1, \dots, K.$$

If $c(u,v) = 1 + \sum_{k=1}^{K} f_k(u) g_k(v) \geq 0$ on $[0,1]^2$, then c is a copula density.

Proof. For each fixed $v \in [0,1]$,

$$\int_0^1 c(u,v)\,du = \int_0^1 \Big(1 + \sum_{k=1}^{K} f_k(u)g_k(v)\Big)\,du = 1 + \sum_{k=1}^{K} g_k(v)\int_0^1 f_k(t)\,dt = 1.$$

Similarly, for each fixed $u \in [0,1]$,

$$\int_0^1 c(u,v)\,dv = 1 + \sum_{k=1}^{K} f_k(u)\int_0^1 g_k(s)\,ds = 1.$$

By Fubini's theorem,

$$\iint_{[0,1]^2} c(u,v)\,d(u,v) = \int_0^1 \left(\int_0^1 c(u,v)\,du\right) dv = \int_0^1 1\,dv = 1.$$

Since $c \geq 0$ and has uniform marginals, c is a copula density. □

As seen in the proof, the zero-mean constraints in Proposition 4.2.1 ensure that the resulting copula density has uniform marginal distributions. Rodrıguez-Lallena and Úbeda-Flores (2004) consider copulas of the form $C(u,v) = uv + F(u)G(v)$. The corresponding copula density is of the form $c(u,v) = 1 + f(u)g(v)$, i.e., the representation above with $K = 1$. The zero-mean constraints on the component functions correspond to the boundary conditions $F(0) = F(1) = G(0) = G(1) = 0$ (see Theorem 2.3 in Rodrıguez-Lallena and Úbeda-Flores 2004; for the extension to $K > 1$, see Theorem 2.2 in Mesiar and Najjari 2014).

Proposition 4.2.1 assumes that the component functions are bounded. This assumption is not merely a technical convenience: the following result shows that, for finite K, boundedness is in fact necessary for (4.1) to remain nonnegative on $[0,1]^2$. Equivalently, if $\sum_{k=1}^{K} f_k g_k \not\equiv 0$ and any component function is unbounded, then (4.1) cannot define a valid copula density.

Theorem 4.2.1 (Boundedness of component functions)**.** *Let $f_1, \ldots, f_K, g_1, \ldots, g_K : [0,1] \to \mathbb{R}$ be measurable functions satisfying*

$$\int_0^1 f_k(u)\,du = 0 \quad \textit{and} \quad \int_0^1 g_k(v)\,dv = 0 \quad \forall k = 1, \ldots, K,$$

and assume $\sum_{k=1}^{K} f_k(u)g_k(v) \not\equiv 0$. *If*

$$1 + \sum_{k=1}^{K} f_k(u)\, g_k(v) \geq 0 \quad \textit{for all } (u,v) \in [0,1]^2,$$

then each f_k *and each* g_k *is bounded on* $[0,1]$.

The proof is deferred to Appendix B.1. Building on these conditions, we next develop a neural network architecture that enforces the zero-mean and nonnegativity constraints by construction.

4.2.2. Neural Network-based Estimation

We now develop a neural network architecture that enforces the conditions of Proposition 4.2.1 by construction. We introduce a method based on Stein operators to construct component functions that automatically satisfy the zero-mean constraints, and describe the loss function and training procedure that enforces nonnegativity.

Construction of Zero-Mean Component Functions via Stein Operators

Our goal is to construct component functions $f_k, g_k : [0,1] \to \mathbb{R}$ satisfying the zero-mean constraints

$$\int_0^1 f_k(u)\, du = 0 \quad \text{and} \quad \int_0^1 g_k(v)\, dv = 0,$$

which, by Proposition 4.2.1, ensure that the copula density (4.1) has uniform marginals. To this end, we employ Stein operators, a well-established tool for generating functions with zero expectation under a reference distribution.

Stein's method was originally introduced by Stein (1972) in the context of normal approximation; see Chen et al. (2011) and Ross (2011) for comprehensive treatments. In modern statistics and machine learning, Stein operators underpin a variety of methods, including goodness-of-fit testing (Gorham and Mackey, 2015; Chwialkowski et al., 2016; Liu et al., 2016), score matching (Hyvärinen, 2005), variational inference (Liu and Wang, 2016; Ranganath et al., 2016), and variance reduction via control functionals (Oates et al., 2017; Si et al., 2020); see Anastasiou et al. (2023) for a recent review.

We use the Langevin-Stein operator associated with a continuously differentiable density $\omega > 0$ on $\mathbb{R}$. Following Gorham and Mackey (2015) and Liu et al. (2016), we define

for a continuously differentiable function $\phi : \mathbb{R} \to \mathbb{R}$,

$$\mathcal{A}_\omega[\phi](x) := \phi'(x) + \frac{\omega'(x)}{\omega(x)}\,\phi(x). \tag{4.2}$$

This operator satisfies $\mathbb{E}_\omega[\mathcal{A}_\omega[\phi](X)] = 0$ whenever $\phi(x)\omega(x) \to 0$ as $|x| \to \infty$, since

$$\int_{\mathbb{R}} \mathcal{A}_\omega[\phi](x)\,\omega(x)\,dx = \int_{\mathbb{R}} \frac{d}{dx}\big[\phi(x)\,\omega(x)\big]\,dx = 0.$$

To transfer this property from $\mathbb{R}$ to the unit interval, let Λ denote the distribution function of ω. By the change of variables $u = \Lambda(x)$,

$$\int_0^1 f(u)\,du = \int_{\mathbb{R}} f(\Lambda(x))\,\omega(x)\,dx$$

for any integrable $f : [0,1] \to \mathbb{R}$. Combining this with the Stein operator, we define

$$f := \mathcal{A}_\omega[\phi] \circ \Lambda^{-1}. \tag{4.3}$$

Proposition 4.2.2 (Zero-mean by construction)**.** *Let ω be a continuously differentiable density on $\mathbb{R}$ with $\omega > 0$, and let $\phi : \mathbb{R} \to \mathbb{R}$ be continuously differentiable with $\phi(x)\omega(x) \to 0$ as $|x| \to \infty$. Then $f := \mathcal{A}_\omega[\phi] \circ \Lambda^{-1}$ satisfies $\int_0^1 f(u)\,du = 0$.*

Proof. By the change of variables $u = \Lambda(x)$,

$$\int_0^1 f(u)\,du = \int_{\mathbb{R}} \mathcal{A}_\omega[\phi](x)\,\omega(x)\,dx = \int_{\mathbb{R}} \frac{d}{dx}\big[\phi(x)\,\omega(x)\big]\,dx = \big[\phi(x)\,\omega(x)\big]_{-\infty}^{\infty} = 0,$$

where the last equality follows from the boundary condition. □

We implement this construction using neural networks. Let $\phi_\theta : \mathbb{R} \to \mathbb{R}^K$ be a neural network with parameters θ that outputs all K components simultaneously. Writing $\phi_\theta = (\phi_{\theta,1}, \dots, \phi_{\theta,K})$, we define

$$f_k := \mathcal{A}_\omega[\phi_{\theta,k}] \circ \Lambda^{-1}, \quad k = 1, \dots, K.$$

By Proposition 4.2.2, each f_k has zero mean on $[0,1]$. Similarly, we parametrize the second family by an independent network $\psi_\eta : \mathbb{R} \to \mathbb{R}^K$ and define $g_k := \mathcal{A}_\omega[\psi_{\eta,k}] \circ \Lambda^{-1}$.

Specifically, ϕ_θ is a fully connected feedforward network with input dimension $d_0 = 1$, output dimension $d_L = K$, and $L - 1$ hidden layers with dimensions $d_1, \dots, d_{L-1}$. The

parameters $\theta = \{W^{(l)}, b^{(l)}\}_{l=1}^{L}$ consist of weight matrices $W^{(l)} \in \mathbb{R}^{d_l \times d_{l-1}}$ and bias vectors $b^{(l)} \in \mathbb{R}^{d_l}$. The network is defined recursively by

$$z^{(0)} = x, \quad z^{(l)} = \tau\left(W^{(l)} z^{(l-1)} + b^{(l)}\right) \text{ for } l = 1, \dots, L-1, \quad \phi_\theta(x) = W^{(L)} z^{(L-1)} + b^{(L)},$$

where τ denotes the activation function. We use SiLU (Sigmoid Linear Unit), defined by $\mathrm{SiLU}(z) = z/(1 + e^{-z})$ (Elfwing et al., 2018; Ramachandran et al., 2017), which is smooth and thus ensures that ϕ_θ is continuously differentiable as required by Proposition 4.2.2.

The construction (4.3) requires the derivative ϕ'_θ with respect to the input. Since neural networks are compositions of elementary differentiable functions, their derivatives can be computed exactly (up to numerical precision) by systematically applying the chain rule. This procedure, known as automatic differentiation, is efficiently supported by modern deep learning frameworks; see Griewank and Walther (2008) and Baydin et al. (2018) for comprehensive introductions.

For the reference distribution ω, we use the standard logistic distribution with distribution function $\Lambda(x) = 1/(1 + e^{-x})$ and density $\omega(x) = e^{-x}/(1 + e^{-x})^2$. The ratio ω'/ω simplifies to $1 - 2\Lambda(x)$. Transforming the unit interval to an unbounded domain via the logit is a common strategy in nonparametric copula density estimation to avoid boundary effects (Geenens et al., 2017; Wen and Wu, 2020); we adopt the logistic distribution for its simple analytical form and good performance in preliminary experiments. To ensure numerical stability, we restrict the domain to $[\delta, 1-\delta]^2$ for some small $\delta > 0$. On this compact set, the inverse Λ^{-1} is well-defined and continuous, and since ϕ_θ is continuously differentiable, the composition $f = \mathcal{A}_\omega[\phi_\theta] \circ \Lambda^{-1}$ is continuous on $[\delta, 1-\delta]$. Continuous functions on compact sets are bounded, so the component functions remain bounded on the restricted domain (cf. Theorem 4.2.1).

Loss Function and Training

The component functions f_k and g_k depend on the neural network parameters θ and η, respectively, which we make explicit by writing $f_k(u;\theta)$ and $g_k(v;\eta)$. The copula density is then

$$c_{\theta,\eta}(u,v) = 1 + \sum_{k=1}^{K} f_k(u;\theta) g_k(v;\eta).$$

Given observations $(x_i, y_i)_{i=1}^n$, we construct pseudo-observations as $u_i = R_i/(n+1)$ and $v_i = S_i/(n+1)$, where R_i and S_i denote the ranks of x_i and y_i, respectively; see Hofert et al. (2018), p. 139. The empirical negative log-likelihood is

$$\mathcal{L}(\theta,\eta) = -\frac{1}{n}\sum_{i=1}^{n} \log c_{\theta,\eta}(u_i, v_i).$$

While our construction guarantees marginal uniformity, the density $c_{\theta,\eta}$ must also be nonnegative to define a valid copula density. This constraint must hold over the entire domain $[0,1]^2$, not just at the training points. To enforce this, we introduce the penalty term

$$\mathcal{P}(\theta,\eta) = \mathbb{E}_{(u,v)\sim\mathrm{Unif}([0,1]^2)}\left[\max\left(0, -c_{\theta,\eta}(u,v)\right)\right].$$

Since $c_{\theta,\eta}$ is continuous, this penalty vanishes if and only if the density is nonnegative everywhere. During training, we approximate the expectation by Monte Carlo sampling.

The training objective combines likelihood and penalty:

$$\mathcal{J}_\mu(\theta,\eta) = \mathcal{L}(\theta,\eta) + \mu\,\mathcal{P}(\theta,\eta),$$

where $\mu > 0$ controls the strength of the penalty. This corresponds to a penalized formulation of the constrained problem

$$\min_{\theta,\eta}\ \mathcal{L}(\theta,\eta) \quad \text{subject to} \quad \mathcal{P}(\theta,\eta) = 0,$$

where the penalty parameter μ plays the role of a Lagrange multiplier. In preliminary experiments, we find that sufficiently large μ ensures that the nonnegativity constraint is satisfied at convergence.

In summary, our neural network construction ensures that $c_{\theta,\eta}$ integrates to one with uniform marginals, while the penalty term enforces $c_{\theta,\eta} \geq 0$ numerically throughout $[0,1]^2$. The entire framework is differentiable, enabling efficient gradient-based optimization with methods such as Adam (Kingma and Ba, 2015).

After training, evaluating the copula density at a point (u,v) amounts to a forward pass through both networks, computing $c_{\theta,\eta}(u,v) = 1 + \sum_{k=1}^{K} f_k(u;\theta) g_k(v;\eta)$. To sample from the fitted copula, we apply the conditional distribution method (Hofert et al., 2018). The separable structure yields the conditional distribution function

$$C_{V|U}(v \mid u;\theta,\eta) = v + \sum_{k=1}^{K} f_k(u;\theta)\, G_k(v;\eta), \qquad G_k(v;\eta) := \int_0^v g_k(t;\eta)\,dt. \tag{4.4}$$

To sample, draw $u \sim \mathrm{Unif}([0,1])$ and $w \sim \mathrm{Unif}([0,1])$, then solve $C_{V|U}(v \mid u) = w$ for v using standard root-finding methods (Brent, 1973).

Having developed a neural network-based estimator for the representation (4.1), we now turn to its expressive power and show that it can approximate arbitrary square-integrable copula densities.

4.2.3. Expressive Power

We study how general L^2 copula densities can be represented by separable expansions of the form (4.1). Our main tool is the Schmidt decomposition, the functional analogue of the singular value decomposition for matrices (see Schmidt, 1908; Ekert and Knight, 1995; for Hilbert-Schmidt and spectral theory more generally, see Mercer, 1909; Conway, 1990). In Section 4.2.3, we establish this representation and show that the component functions inherit the zero-mean property from the uniform copula margins. However, the Schmidt component functions need not be bounded, and the truncated approximations need not be nonnegative. In Section 4.2.3, we provide sufficient conditions under which the component functions are essentially bounded. Finally, in Section 4.2.3, we show that the L^2-approximation error of bounded nonnegative separable expansions vanishes asymptotically, so neither the boundedness constraint nor the nonnegativity constraint reduces expressive power.

Schmidt Representation and Zero-mean Property

For a general copula density $c \in L^2([0,1]^2)$, we write $c(u,v) = 1 + \mathcal{K}(u,v)$. The centered part $\mathcal{K}$ admits a Schmidt decomposition, which yields an infinite-series representation with orthonormal component functions.

Theorem 4.2.2 (Schmidt representation and zero-mean property)**.** *Let $c \in L^2([0,1]^2)$ be a copula density and set $\mathcal{K} := c - 1$. Then there exist singular values $(\sigma_k)_{k\geq 1} \subset [0,\infty)$ with*

$$\sigma_1 \geq \sigma_2 \geq \cdots \searrow 0,$$

and functions $(\alpha_k)_{k\geq 1}, (\beta_k)_{k\geq 1} \subset L^2([0,1])$ such that:

(i) Orthonormality and L^2-approximation. *The families $\{\alpha_k\}$ and $\{\beta_k\}$ are orthonormal in $L^2([0,1])$, and for*

$$S_K(u,v) := 1 + \sum_{k=1}^{K} \sigma_k \, \alpha_k(u) \, \beta_k(v)$$

one has

$$\| \, c - S_K \, \|_{L^2([0,1]^2)} \xrightarrow[K\to\infty]{} 0.$$

(ii) Zero-mean property. *For every k with $\sigma_k > 0$,*

$$\int_0^1 \alpha_k(u)\, du = 0, \qquad \int_0^1 \beta_k(v)\, dv = 0.$$

Proof. We work over $L^2([0,1])$ and use the inner product

$$\langle f, g\rangle_{L^2} := \int_0^1 f(t)\, g(t)\, dt.$$

(i) With $\mathcal{K} := c - 1$, the L^2-approximation by $S_K = 1 + \sum_{j\le K} \sigma_j \alpha_j \beta_j$ follows directly from the Schmidt decomposition in L^2; see Lemma 5 and Lemma 6 in Appendix B.2.
(ii) Since the copula margins are uniform,

$$\int_0^1 \mathcal{K}(u,v)\, dv = 0 \quad \text{for every } u, \qquad \int_0^1 \mathcal{K}(u,v)\, du = 0 \quad \text{for every } v.$$

Define the integral operator

$$(T_{\mathcal{K}} f)(u) := \int_0^1 \mathcal{K}(u,v)\, f(v)\, dv,$$

and its adjoint

$$(T_{\mathcal{K}}^* g)(v) := \int_0^1 \mathcal{K}(u,v)\, g(u)\, du.$$

Then $T_{\mathcal{K}}\mathbf{1} = T_{\mathcal{K}}^*\mathbf{1} = 0$. For $\sigma_k > 0$ one has $T_{\mathcal{K}}\beta_k = \sigma_k \alpha_k$ and $T_{\mathcal{K}}^*\alpha_k = \sigma_k \beta_k$. By the adjoint property, $\langle T_{\mathcal{K}}\beta_k, \mathbf{1}\rangle_{L^2} = \langle \beta_k, T_{\mathcal{K}}^*\mathbf{1}\rangle_{L^2}$, hence

$$\int_0^1 \alpha_k(t)\, dt = \langle \alpha_k, \mathbf{1}\rangle_{L^2} = \frac{1}{\sigma_k}\langle T_{\mathcal{K}}\beta_k, \mathbf{1}\rangle_{L^2} = \frac{1}{\sigma_k}\langle \beta_k, T_{\mathcal{K}}^*\mathbf{1}\rangle_{L^2} = 0,$$

and similarly $\int_0^1 \beta_k(s)\, ds = 0$. □

The Schmidt representation converges to the copula density in the L^2 sense, not pointwise. The limit is nonnegative almost everywhere since it equals the copula density almost everywhere. However, the truncations S_K are finite sums and need not be nonnegative; they can take negative values on parts of $[0,1]^2$. This means that S_K is generally not a valid copula density, even though it approximates one. In Appendix B.3, we provide a sufficient condition under which the truncations are also nonnegative almost everywhere.

Example 4.2.1 (Gaussian copula)**.** *The Gaussian copula density c_ρ with parameter $\rho \in (-1,1)$ is square-integrable. By a straightforward calculation, one sees that*

$$\int_{[0,1]^2} c_\rho(u,v)^2 \, du \, dv = \frac{1}{1-\rho^2}.$$

Furthermore, the density admits a known series expansion (see Grothe and Rieger, 2024):

$$c_\rho(u,v) = 1 + \sum_{k=1}^{\infty} \frac{\rho^k}{k!} \operatorname{He}_k\big(\Phi^{-1}(u)\big) \operatorname{He}_k\big(\Phi^{-1}(v)\big), \qquad u,v \in (0,1),$$

where He_k denotes the k-th probabilists' Hermite polynomial and Φ is the standard normal distribution function. This expansion fits the Schmidt structure of Theorem 4.2.2. Setting

$$\alpha_k(u) := \frac{\operatorname{He}_k(\Phi^{-1}(u))}{\sqrt{k!}}, \qquad \beta_k(v) := \operatorname{sgn}(\rho)^k \, \frac{\operatorname{He}_k(\Phi^{-1}(v))}{\sqrt{k!}}, \qquad \sigma_k := |\rho|^k \quad (k \geq 1),$$

we obtain $c_\rho(u,v) = 1 + \sum_{k=1}^{\infty} \sigma_k \, \alpha_k(u) \, \beta_k(v)$.

The zero-mean property follows by the change of variables $u = \Phi(\xi)$:

$$\int_0^1 \alpha_k(u) \, du = \frac{1}{\sqrt{k!}} \int_{\mathbb{R}} \operatorname{He}_k(\xi) \, \varphi(\xi) \, d\xi = 0,$$

and similarly for β_k. Orthonormality follows from the identity $\int_{\mathbb{R}} \operatorname{He}_k(u) \operatorname{He}_j(u) \, \varphi(u) \, du = k! \, \delta_{kj}$, which gives

$$\int_0^1 \alpha_k(u) \, \alpha_j(u) \, du = \delta_{kj}, \qquad \int_0^1 \beta_k(v) \, \beta_j(v) \, dv = \delta_{kj}.$$

However, the component functions are unbounded. Since $\operatorname{He}_1(x) = x$, we have $\alpha_1(u) = \Phi^{-1}(u)$, which diverges as $u \to 0$ or $u \to 1$. Higher Hermite polynomials grow polynomially in their argument, so all α_k and β_k are unbounded on $(0,1)$.

Essential Boundedness under Regularity Conditions

The Gaussian copula example shows that the Schmidt components need not be bounded, even for smooth copula densities. We now provide a sufficient condition under which the component functions (α_k) and (β_k) are essentially bounded. The condition requires that the section L^2-norms of the centered kernel $\mathcal{K}$ are uniformly bounded.

Since L^2-functions are defined only up to null sets, the appropriate notion is essential boundedness, meaning boundedness almost everywhere. This is not a weakening: every essentially bounded function admits a bounded representative, obtained by redefining it on a null set.

Theorem 4.2.3 (Essential boundedness)**.** *Let $c \in L^2([0,1]^2)$ be a copula density and write $\mathcal{K} := c - 1$. Assume that the section L^2-norms of $\mathcal{K}$ are essentially bounded:*

$$M_u := \operatorname*{ess\,sup}_{u\in[0,1]} \Big(\int_0^1 |\mathcal{K}(u,v)|^2 \, dv \Big)^{1/2} < \infty, \qquad M_v := \operatorname*{ess\,sup}_{v\in[0,1]} \Big(\int_0^1 |\mathcal{K}(u,v)|^2 \, du \Big)^{1/2} < \infty.$$

Then, for the singular values and component functions $(\sigma_k, \alpha_k, \beta_k)_{k\geq 1}$ provided by Theorem 4.2.2, each α_k and β_k is essentially bounded and satisfies

$$\|\beta_k\|_{L^\infty} \leq \frac{M_u}{\sigma_k}, \qquad \|\alpha_k\|_{L^\infty} \leq \frac{M_v}{\sigma_k} \qquad (k \geq 1,\ \sigma_k > 0).$$

Proof. Define the integral operator

$$(T_{\mathcal{K}} f)(u) := \int_0^1 \mathcal{K}(u,v)\, f(v)\, dv, \qquad f \in L^2([0,1]).$$

By Cauchy-Schwarz, for every $f \in L^2([0,1])$ and a.e. $u \in [0,1]$,

$$|(T_{\mathcal{K}} f)(u)| \leq \Big(\int_0^1 |\mathcal{K}(u,v)|^2 \, dv \Big)^{1/2} \|f\|_{L^2} \leq M_u \, \|f\|_{L^2}.$$

Taking the essential supremum over u gives $\|T_{\mathcal{K}} f\|_{L^\infty} \leq M_u \|f\|_{L^2}$. This means that $T_{\mathcal{K}}$ is a bounded operator from L^2 to L^∞, with operator norm

$$\|T_{\mathcal{K}}\|_{L^2\to L^\infty} := \sup_{\|f\|_{L^2}=1} \|T_{\mathcal{K}} f\|_{L^\infty} \leq M_u.$$

The adjoint operator $T_{\mathcal{K}}^*$ is given by

$$(T_{\mathcal{K}}^* g)(v) := \int_0^1 \mathcal{K}(u,v)\, g(u)\, du, \qquad g \in L^2([0,1]),$$

and the same argument with the roles of u and v exchanged yields

$$\|T_{\mathcal{K}}^*\|_{L^2\to L^\infty} \leq M_v.$$

Now let $(\sigma_k, \alpha_k, \beta_k)_{k\geq 1}$ be as in Theorem 4.2.2, so that $T_{\mathcal{K}}\beta_k = \sigma_k\alpha_k$ and $T_{\mathcal{K}}^*\alpha_k = \sigma_k\beta_k$ with $\|\alpha_k\|_{L^2} = \|\beta_k\|_{L^2} = 1$ and $\sigma_k > 0$. Applying the above bounds to $f = \beta_k$ and $g = \alpha_k$ gives

$$\|\alpha_k\|_{L^\infty} = \frac{\|T_{\mathcal{K}}\beta_k\|_{L^\infty}}{\sigma_k} \leq \frac{\|T_{\mathcal{K}}\|_{L^2\to L^\infty}\,\|\beta_k\|_{L^2}}{\sigma_k} \leq \frac{M_u}{\sigma_k},$$

and similarly

$$\|\beta_k\|_{L^\infty} = \frac{\|T_{\mathcal{K}}^*\alpha_k\|_{L^\infty}}{\sigma_k} \leq \frac{\|T_{\mathcal{K}}^*\|_{L^2\to L^\infty}\,\|\alpha_k\|_{L^2}}{\sigma_k} \leq \frac{M_v}{\sigma_k}.$$

Thus, each α_k and β_k is essentially bounded with the stated estimates. □

A particularly important case arises when the copula density itself is bounded, which implies the section L^2 bounds automatically.

Corollary 4.2.1 (Bounded components for bounded densities)**.** *Let $c \in L^\infty([0,1]^2)$ be a copula density. Then the section L^2 bounds in Theorem 4.2.3 hold with*

$$M_u, M_v \leq \|c-1\|_{L^\infty}.$$

Consequently, for the singular values and component functions $(\sigma_k, \alpha_k, \beta_k)_{k\geq 1}$ from Theorem 4.2.2 with $\sigma_k > 0$,

$$\|\beta_k\|_{L^\infty} \leq \frac{\|c-1\|_{L^\infty}}{\sigma_k}, \qquad \|\alpha_k\|_{L^\infty} \leq \frac{\|c-1\|_{L^\infty}}{\sigma_k}.$$

Proof. Since $c \in L^\infty([0,1]^2)$, we have $\mathcal{K} = c - 1 \in L^\infty([0,1]^2)$. For a.e. $u, v \in [0,1]$,

$$\Big(\int_0^1 |\mathcal{K}(u,v)|^2\,dv\Big)^{1/2} \leq \|\mathcal{K}\|_{L^\infty}, \qquad \Big(\int_0^1 |\mathcal{K}(u,v)|^2\,du\Big)^{1/2} \leq \|\mathcal{K}\|_{L^\infty}.$$

Thus the section L^2 bounds in Theorem 4.2.3 hold with $M_u, M_v \leq \|c-1\|_{L^\infty}$, and the stated estimates follow. □

Approximation by Bounded Nonnegative Separable Expansions

Theorem 4.2.3 and Corollary 4.2.1 show that bounded copula densities admit an exact Schmidt representation with bounded component functions. However, the truncated Schmidt approximations S_K need not be nonnegative, even though the target copula density is. The next result addresses both concerns: bounded nonnegative separable expansions are dense in L^2 copula densities, so neither the boundedness constraint nor the nonnegativity constraint limits the approximation capacity.

Theorem 4.2.4 (L^2-approximation by bounded nonnegative separable expansions)**.** *Let $c \in L^2([0,1]^2)$ be a copula density. Then there exists a sequence of approximations*

$$\tilde{S}_K := 1 + \sum_{k=1}^{K} f_k(u) g_k(v)$$

with zero-mean functions $f_k, g_k \in L^\infty([0,1])$ such that $\tilde{S}_K \geq 0$ a.e. and $\|c - \tilde{S}_K\|_{L^2([0,1]^2)} \to 0$ as $K \to \infty$.

Proof. Since c is a copula density, the centered function $\mathcal{K} := c - 1$ has vanishing marginals:

$$\int_0^1 \mathcal{K}(u,v)\,dv = 0 \quad \text{for a.e. } u, \qquad \int_0^1 \mathcal{K}(u,v)\,du = 0 \quad \text{for a.e. } v.$$

The problem reduces to approximating $\mathcal{K}$ by bounded separable sums with zero-mean factors, subject to a nonnegativity constraint.

Let $\mathcal{H} \subset L^2([0,1]^2)$ denote the subspace of functions with vanishing marginals. Define the continuous linear operators $T_1, T_2 : L^2([0,1]^2) \to L^2([0,1])$ by $(T_1 h)(u) = \int_0^1 h(u,v)\,dv$ and $(T_2 h)(v) = \int_0^1 h(u,v)\,du$. Then $\mathcal{H} = \ker(T_1) \cap \ker(T_2)$, which is closed as the intersection of kernels of continuous operators. By construction, $\mathcal{K} \in \mathcal{H}$. On a finite measure space, L^∞ is dense in L^2, so finite sums $\sum_k \hat{f}_k(u)\hat{g}_k(v)$ with $\hat{f}_k, \hat{g}_k \in L^\infty([0,1])$ are dense in $L^2([0,1]^2)$.

Let $P_{\mathcal{H}}$ denote the orthogonal projection onto $\mathcal{H}$. For $\hat{f}, \hat{g} \in L^\infty([0,1])$, write $\bar{f} = \int_0^1 \hat{f}(t)\,dt$ and $\bar{g} = \int_0^1 \hat{g}(s)\,ds$ for the means, and $\tilde{f} = \hat{f} - \bar{f}$ and $\tilde{g} = \hat{g} - \bar{g}$ for the centered parts. Then

$$\hat{f}(u)\hat{g}(v) = \tilde{f}(u)\tilde{g}(v) + \bar{f}\tilde{g}(v) + \bar{g}\tilde{f}(u) + \bar{f}\bar{g}.$$

The first term $\tilde{f}(u)\tilde{g}(v)$ lies in $\mathcal{H}$, since both factors are zero-mean. The remaining three terms lie in the orthogonal complement of $\mathcal{H}$. Hence $P_{\mathcal{H}}(\hat{f}(u)\hat{g}(v)) = \tilde{f}(u)\tilde{g}(v)$, which is bounded with zero-mean factors.

For each K, choose $s_K = \sum_{k=1}^{K} \hat{f}_k(u)\hat{g}_k(v)$ with $\hat{f}_k, \hat{g}_k \in L^\infty([0,1])$ such that $\|\mathcal{K} - s_K\|_{L^2} \to 0$ as $K \to \infty$. Since $\mathcal{K} \in \mathcal{H}$ and $P_{\mathcal{H}}$ is an orthogonal projection,

$$\|\mathcal{K} - P_{\mathcal{H}}(s_K)\|_{L^2} = \|P_{\mathcal{H}}(\mathcal{K} - s_K)\|_{L^2} \leq \|\mathcal{K} - s_K\|_{L^2} \to 0.$$

By linearity, $P_{\mathcal{H}}(s_K) = \sum_{k=1}^{K} \tilde{f}_k(u)\tilde{g}_k(v)$. Define the unconstrained approximation $S_K := 1 + \sum_{k=1}^{K} \tilde{f}_k(u)\tilde{g}_k(v)$, which satisfies $\|c - S_K\|_{L^2} \to 0$.

It remains to incorporate the nonnegativity constraint. For a function h, denote its negative part by $h^- := \max(-h, 0)$, so that $h^-(x) > 0$ if and only if $h(x) < 0$. Since $S_K \to c$

in L^2 and $c \geq 0$ a.e., we have

$$\|(S_K)^-\|_{L^2} \leq \|S_K - c\|_{L^2} \to 0.$$

This means that the set where S_K is negative shrinks in an L^2 sense as $K \to \infty$.

For each K, consider the constrained optimization problem: among all functions of the form $1 + \sum_{k=1}^{K} f_k(u)g_k(v)$ with $f_k, g_k \in L^\infty([0,1])$ zero-mean and satisfying $1 + \sum_{k=1}^{K} f_k(u)g_k(v) \geq 0$ a.e., find the one that minimizes the L^2-distance to c. Denote this constrained approximation by $\tilde{S}_K$.

If $S_K \geq 0$ a.e., then S_K satisfies the constraint and hence $\tilde{S}_K = S_K$. Since $S_K \to c$ in L^2 and $c \geq 0$ a.e., we have $\|(S_K)^-\|_{L^2} \to 0$, so the constraint $\tilde{S}_K \geq 0$ becomes asymptotically slack. Since $\tilde{S}_K$ minimizes $\|c - \cdot\|_{L^2}$ over the constrained set and the unconstrained minimizer S_K satisfies $\|c - S_K\|_{L^2} \to 0$, we conclude $\|c - \tilde{S}_K\|_{L^2} \to 0$ as $K \to \infty$. □

Remark 4.2.1 (Practical implementation)**.** *In our neural network architecture, the constrained approximation is obtained by minimizing the L^2-loss with an additional penalty term enforcing nonnegativity; see Section 4.2.2.*

The separable representation (4.1) provides a flexible framework for copula density estimation: the Schmidt decomposition shows that any L^2 copula density can be represented or approximated by separable expansions, and neither the boundedness constraint nor the nonnegativity constraint required by our neural network architecture limits the approximation capacity. We now turn to empirical evaluation of the proposed estimator.

4.3. Simulation Studies

We evaluate the performance of our proposed neural network-based copula density estimator through simulation studies. The goal is to assess its ability to capture dependence structures in bivariate copulas and to benchmark it against established nonparametric alternatives.

4.3.1. Study Setup

As baseline methods, we consider several nonparametric estimators for bivariate copulas. We use kernel density estimation with the TLL2nn estimator, which has demonstrated strong performance among copula KDE approaches (Nagler, 2018). For the Bernstein copula, we apply the empirical Bernstein estimator with automatic selection of the number of basis functions based on the method proposed by Weiß and Scheffer (2012) and Sancetta

and Satchell (2004). In addition, we consider the empirical beta copula, a smoothed variant of the empirical copula obtained via beta-kernel smoothing (Segers et al., 2017). The KDE and Bernstein-based estimators are provided by the `kdecopula` R package (Nagler, 2018).

For the neural network-based estimator proposed in Section 4.2.2, we use $K = 15$ terms in (4.1). The component functions f_k and g_k are based on two fully connected neural networks with seven hidden layers, using SiLU activations in all hidden layers and a linear output layer. The reference distribution is the logistic distribution (see Section 4.2.2). Based on preliminary investigations across different copula classes, we selected the number of terms K in (4.1) and the number of hidden layers. This configuration proved to be robust throughout our simulation study. The implementation uses Python with TensorFlow (Abadi et al., 2015). In the following, we refer to the neural network-based density estimator as the neural estimator via separable perturbation (NESP).

In our simulation study, we generate data from two types of bivariate copulas: parametric families with a global dependence structure controlled by a single parameter, and copulas exhibiting heterogeneous dependence patterns (described below). For each copula model, we simulate 100 independent datasets of size $n = 500$ and apply each estimator to every dataset to obtain copula density estimates, allowing us to compare estimation accuracy across methods.

For the parametric families, we consider well-known copulas with different tail behavior and symmetry. Let λ_L and λ_U denote lower and upper tail dependence, respectively:

- Clayton ($\theta = 2$): $\lambda_L \approx 0.71$, $\lambda_U = 0$; radially asymmetric.
- Joe ($\theta = 2.7$): $\lambda_L = 0$, $\lambda_U \approx 0.71$; radially asymmetric.
- Gaussian ($\rho = 0.5$): $\lambda_L = \lambda_U = 0$; radially symmetric.
- t ($\nu = 2$, $\rho = 0.5$): $\lambda_L = \lambda_U \approx 0.40$; radially symmetric.

Beyond these parametric families, we consider copulas with heterogeneous dependence patterns. The first is a trigonometric copula with density $c(u, v) = 1 + \sin(4\pi u)\cos(3\pi v)$, which has an explicit separable form. The remaining four copulas are constructed via the checkerboard principle: we partition $[0, 1]^2$ into a 40×40 grid, specify a nonnegative weight matrix encoding the desired local structure (e.g., Gaussian spots, sparse regions, or ring patterns), and apply Sinkhorn scaling (see Sinkhorn and Knopp, 1967) to obtain a bistochastic matrix that defines a valid copula with uniform margins. Details are given in Appendix B.4. Figure 4.1 shows example samples; Clayton and Gumbel copulas with varying tail dependence are omitted.

To study how estimation accuracy depends on the level of tail dependence, we vary the tail dependence parameter $\lambda \in \{0.2, 0.4, 0.6, 0.8\}$ using Clayton and Gumbel copulas. These two families provide a geometric contrast: in Clayton, lower-tail mass spreads over an area toward $(0, 0)$, whereas in Gumbel, upper-tail mass concentrates along a sharp ridge toward $(1, 1)$.

To assess estimation accuracy, we use the integrated absolute error (IAE), which measures the L^1-distance between $\hat{c}$ and c. The IAE is a widely adopted metric in copula density estimation (Nagler, 2018; Geenens et al., 2017):

$$\mathrm{IAE}[\hat{c}] = \int_{[0,1]^2} |\hat{c}(u, v) - c(u, v)| \, du \, dv.$$

We approximate this integral on the interior square $[0.01, 0.99]^2$ to avoid numerical boundary effects. Let

$$\mathcal{G}_N = \{(u_i, v_j) : u_i = 0.01 + i\Delta, \; v_j = 0.01 + j\Delta, \; i, j = 0, \dots, N-1\},$$

where $\Delta = 0.98/(N-1)$ and $N = 500$. The IAE is approximated by

$$\widehat{\mathrm{IAE}}_N = \frac{1}{N^2} \sum_{(u,v)\in\mathcal{G}_N} |\hat{c}(u, v) - c(u, v)|.$$

Recall that the copulas with heterogeneous dependence patterns are constructed on a 40×40 grid, where the true density is piecewise constant within each cell. The evaluation grid provides 12 to 13 points per cell along each axis, yielding between 144 and 169 evaluation points per cell, which ensures a stable approximation of the IAE.

4.3.2. Results

Table 4.1 reports the mean IAE and standard deviation across 100 replications for each copula and estimator. For parametric copulas, KDE achieves the lowest mean IAE overall. However, NESP performs comparably, and even outperforms KDE for the Gaussian copula. NESP also achieves lower errors than both the empirical Bernstein and beta estimators in three of the four parametric cases. The exception is the t copula, where NESP (0.186) is outperformed by both KDE (0.110) and the empirical Bernstein estimator (0.173).

NESP shows its strength on copulas with heterogeneous dependence patterns, achieving the lowest mean IAE for the trigonometric copula and three of the four checkerboard scenarios. This is consistent with the separable architecture of NESP, which can allocate

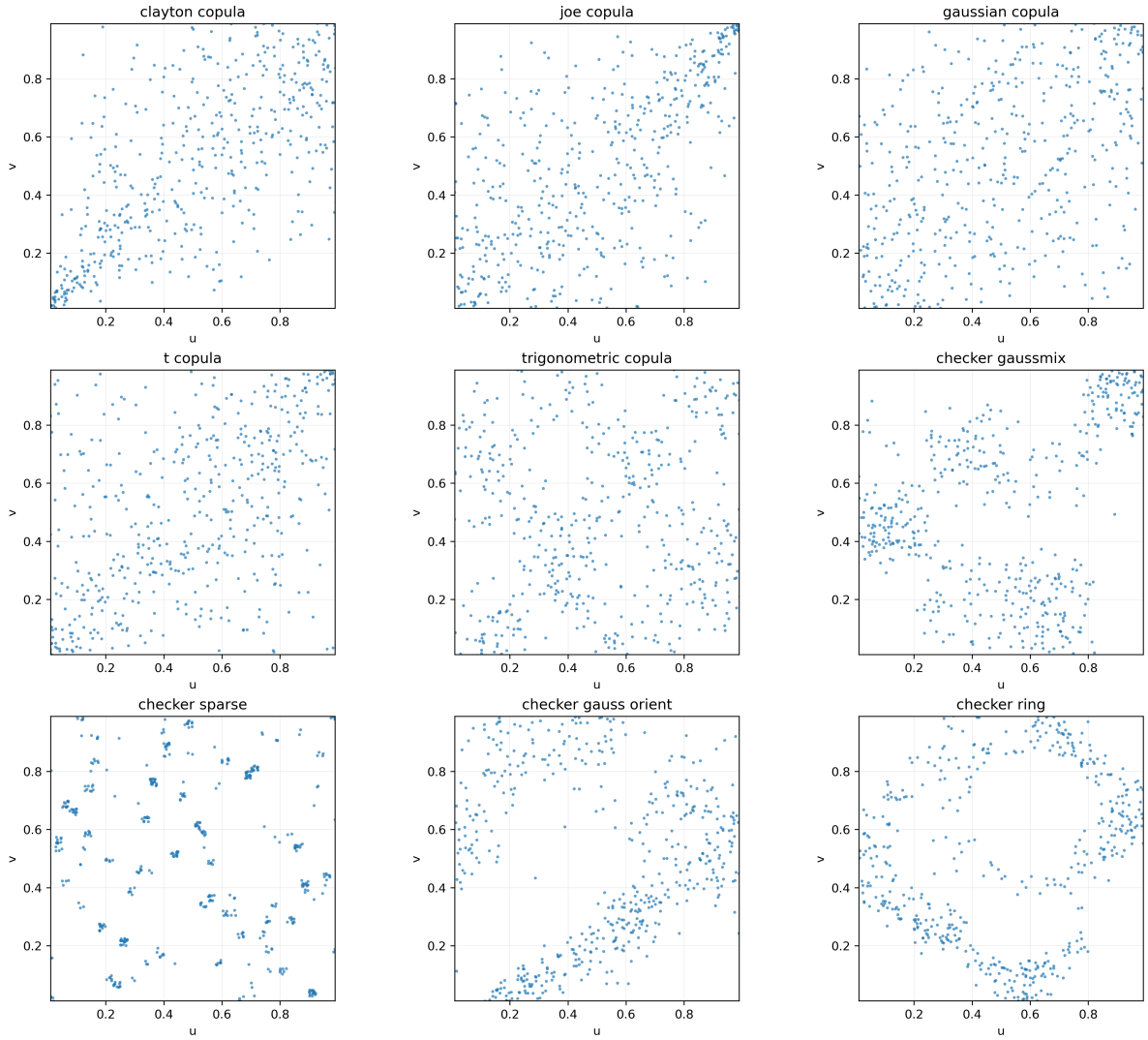


Figure 4.1.: Example copula training samples with 500 points each. The first four copulas are parametric families: Clayton ($\theta = 2$), Joe ($\theta = 2.7$), Gaussian ($\rho = 0.5$), and t ($\nu = 2$, $\rho = 0.5$). The remaining copulas exhibit heterogeneous dependence patterns: a trigonometric copula with separable density and four checkerboard copulas with varying local structure.

Copula	Empirical Beta	Empirical Bernstein	KDE	NESP
Gaussian	0.539 (0.014)	0.081 (0.018)	0.069 (0.031)	**0.065** (0.017)
t	0.535 (0.016)	0.173 (0.025)	**0.110** (0.025)	0.186 (0.056)
Clayton	0.474 (0.017)	0.191 (0.017)	**0.090** (0.023)	0.109 (0.020)
Joe	0.480 (0.017)	0.196 (0.021)	**0.103** (0.024)	0.127 (0.016)
Trigonometric	0.554 (0.020)	0.400 (0.007)	0.302 (0.053)	**0.221** (0.042)
Checkerboard Gaussian Mix	0.468 (0.029)	0.601 (0.020)	0.330 (0.027)	**0.260** (0.022)
Checkerboard Gaussian Orient	0.487 (0.030)	0.818 (0.025)	0.381 (0.099)	**0.319** (0.090)
Checkerboard Ring	0.477 (0.050)	0.888 (0.017)	0.425 (0.026)	**0.371** (0.040)
Checkerboard Sparse	**1.587** (0.062)	1.805 (0.000)	1.779 (0.007)	1.682 (0.020)

Table 4.1.: IAE: Mean and standard deviation of estimation errors for bivariate copulas (100 samples, each of size 500). Bold entries denote the best mean per copula.

capacity along directional features such as ridges and rings. In contrast, isotropic kernel smoothers spread smoothing uniformly, which blurs such patterns. Figure 4.2 illustrates this: the ring pattern remains sharply visible in the NESP estimate but is blurred by KDE. For the Checkerboard Sparse scenario, the empirical beta copula achieves the lowest error (1.587), followed by NESP (1.682). Despite being based on a neural network, NESP exhibits comparable variance to the deterministic methods across most scenarios.

Table 4.2 examines how estimation accuracy depends on the level of tail dependence; we omit the empirical beta copula due to its poor performance in the previous experiment. For all methods, estimation errors increase with λ, but more steeply for Gumbel than for Clayton. This suggests that estimation accuracy is not driven by tail dependence per se, but by the geometry of the tail mass: in Clayton, lower-tail mass spreads over an area toward $(0, 0)$, whereas in Gumbel, upper-tail mass concentrates along a sharp ridge toward $(1, 1)$. Such ridge-like diagonal concentration is harder to reconstruct for all estimators and exacerbates boundary effects.

4.4. Real Data Illustration

To illustrate how the proposed bivariate estimator can be used as a building block for vine copula construction, we apply it to the MAGIC Gamma Telescope dataset (Bock, 2004), which contains 19,020 observations of image parameters from atmospheric Cherenkov telescopes used for gamma-ray detection. Following Nagler and Czado (2016) and Nagler et al. (2017), we model the dependence among three variables: `fConc1` (the ratio of the highest pixel intensity over the total image intensity), `fM3Long`, and `fM3Trans` (the cube

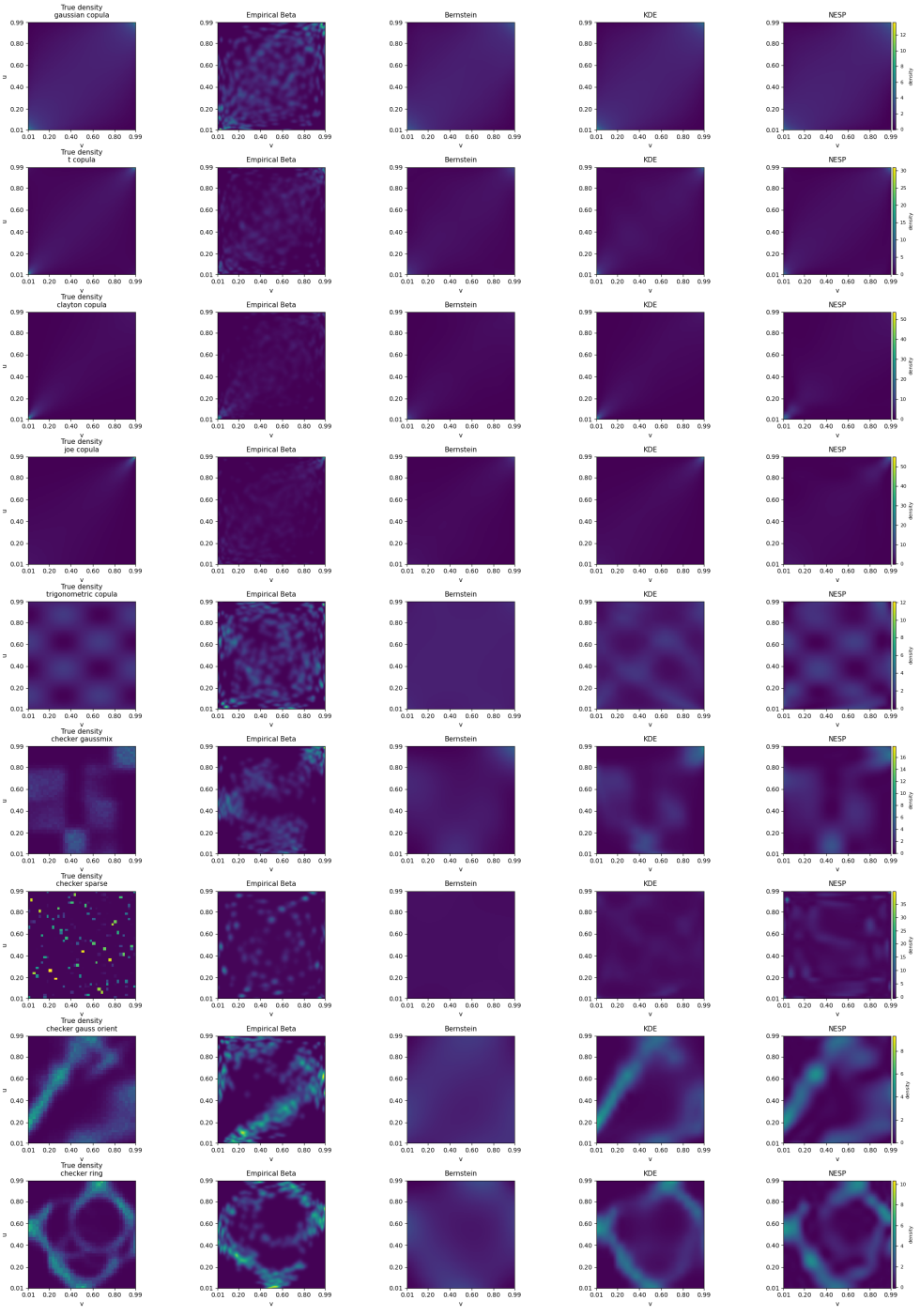


Figure 4.2.: Comparison of the true copula densities with the estimated densities from the empirical beta estimator, the empirical Bernstein estimator, KDE, and NESP.

Copula (λ)	Empirical Bernstein	KDE	NESP
Clayton (lower-tail λ_L)			
$\lambda_L = 0.2$	**0.070** (0.015)	0.072 (0.026)	0.078 (0.033)
$\lambda_L = 0.4$	0.094 (0.019)	**0.076** (0.028)	0.085 (0.028)
$\lambda_L = 0.6$	0.142 (0.020)	**0.085** (0.026)	0.093 (0.024)
$\lambda_L = 0.8$	0.277 (0.018)	**0.104** (0.020)	0.160 (0.029)
Gumbel (upper-tail λ_U)			
$\lambda_U = 0.2$	**0.063** (0.014)	0.075 (0.031)	0.080 (0.048)
$\lambda_U = 0.4$	0.102 (0.022)	**0.074** (0.026)	0.105 (0.034)
$\lambda_U = 0.6$	0.167 (0.024)	**0.085** (0.026)	0.148 (0.042)
$\lambda_U = 0.8$	0.390 (0.018)	**0.108** (0.022)	0.201 (0.022)

Table 4.2.: IAE: Mean and standard deviation of estimation errors on tail dependent copulas (100 samples, each of size 500). Bold entries denote the best mean per row. Clayton varies lower tail dependence λ_L; Gumbel varies upper tail dependence λ_U.

roots of the third moments along the major and minor ellipse axes, respectively). We denote by (u_1, u_2, u_3) the corresponding pseudo-observations.

For training, we draw a random subsample of $n = 2{,}000$ observations and use the remaining 17,020 observations for evaluation. Each bivariate copula density is estimated using NESP with $K = 15$ terms, providing sufficient flexibility for the vine construction.

We fit a three-dimensional C-vine under the simplifying assumption (Czado and Nagler, 2022), writing the copula density as

$$c(u_1, u_2, u_3) = c_{12}(u_1, u_2)\, c_{13}(u_1, u_3)\, c_{23|1}\big(C_{2|1}(u_2 \mid u_1),\, C_{3|1}(u_3 \mid u_1)\big). \tag{4.5}$$

The bivariate building blocks c_{12} and c_{13} are estimated first, as they directly enter the vine density and provide the conditional transforms needed for the higher-order pair. The pair (u_2, u_3) is not fitted at this stage, as its dependence is modeled conditionally via $c_{23|1}$ in the second tree of the vine.

The conditional sample points used for $c_{23|1}$ are obtained via the conditional distribution function (4.4). For the NESP parametrization with the logistic reference distribution (see Section 4.2.2), this takes the form

$$C_{2|1}(u_2 \mid u_1) = u_2 + \sum_{k=1}^{K} f_{12,k}(u_1)\, G_{12,k}(u_2),$$

$$C_{3|1}(u_3 \mid u_1) = u_3 + \sum_{k=1}^{K} f_{13,k}(u_1)\, G_{13,k}(u_3),$$

where $f_{ij,k}$ and $g_{ij,k}$ denote the component functions for the pair (u_i, u_j), and

$$G_{ij,k}(t) = \int_0^t g_{ij,k}(s)\, ds$$

is computed once on a fine grid via numerical integration and interpolation. We then estimate the conditional copula

$$c_{23|1}(a, b) = 1 + \sum_{k=1}^{K} f_{23|1,k}(a)\, g_{23|1,k}(b)$$

on the transformed sample $\{(a_i, b_i)\}_{i=1}^n$, where $a_i = C_{2|1}(u_{2,i} \mid u_{1,i})$ and $b_i = C_{3|1}(u_{3,i} \mid u_{1,i})$. Since the NESP parametrization is separable, the conditional distribution functions reduce to linear combinations of the pre-tabulated integrals $G_{ij,k}$, enabling efficient vectorized evaluation.

For comparison, we also fit a KDE-based vine copula and an empirical Bernstein vine copula using the `kdecopula` and `kdevine` implementations (Nagler, 2018; Nagler, 2024). Figure 4.3 shows, for each pair (u_1, u_2), (u_1, u_3), and (u_2, u_3), the training pseudo-observations and the corresponding bivariate copula density estimates obtained by Bernstein, KDE, and NESP.

Since the true data-generating density is unknown, we evaluate model fit by comparing synthetic samples against the held-out pseudo-observations. Specifically, we report (i) the multivariate two-sample Kolmogorov-Smirnov (KS) statistic, which measures the maximum discrepancy between the empirical distribution functions of two samples (Naaman, 2021; Laurent, 2020), and (ii) the Wasserstein distance, which measures the minimal cost of transporting mass from one distribution to the other (Flamary et al., 2021). Lower values indicate better agreement between the synthetic and held-out samples. For the KS test we list the critical value at $\alpha = 0.05$ and whether H_0 (equal distributions) is rejected.

Overall, NESP attains the smallest W and KS values among all methods, indicating a better match to the held-out pseudo-observations. While none of the methods leads to rejection of H_0 at the 5% level, NESP achieves notably lower values on both metrics, suggesting that the separable representation captures the dependence structure of this dataset well.

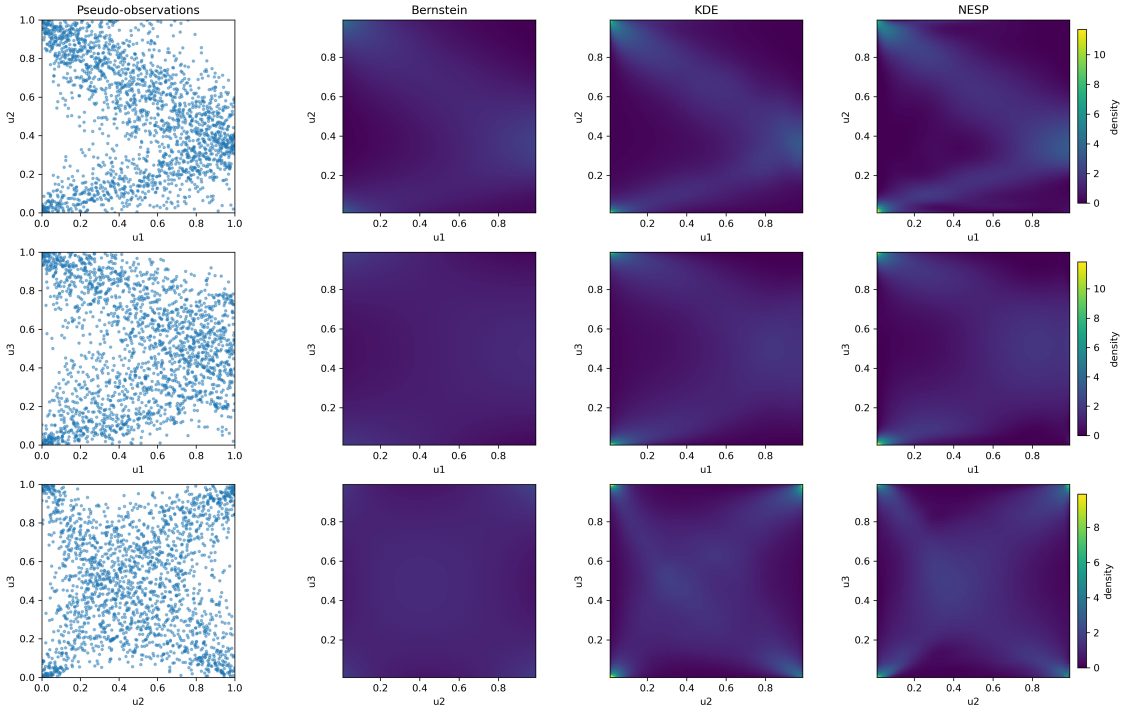


Figure 4.3.: Bivariate copula densities for the MAGIC Gamma Telescope data. Rows correspond to the variable pairs (u_1, u_2), (u_1, u_3), and (u_2, u_3); columns show the training pseudo-observations and the fitted densities obtained by the Bernstein estimator, KDE, and NESP. Color scales are shared within each row to facilitate comparison across estimators for a fixed pair.

Dataset	W	KS	Critical	Reject H_0?
Synthetic data from KDE	0.065341	0.042840	0.097133	No
Synthetic data from Bernstein	0.095981	0.059310	0.097133	No
Synthetic data from NESP	**0.063394**	**0.031713**	0.097133	No

Table 4.3.: Goodness-of-fit on pseudo-observations: Wasserstein W and multivariate KS (lower is better). The critical value corresponds to $\alpha = 0.05$; the last column indicates whether H_0 is rejected.

4.5. Conclusion

In this chapter, we develop a neural network-based copula density estimator for copula densities of the form $c(u, v) = 1 + \sum_{k=1}^{K} f_k(u)\, g_k(v)$ with bounded component functions f_k and g_k. We prove that bounded separable expansions with zero-mean component functions are dense in the space of square-integrable copula densities, justifying the expressive power of our approach. The proposed neural network architecture enforces marginal uniformity through a Stein-type construction and nonnegativity numerically during training. In simulation studies, the estimator performs competitively with kernel density estimators on standard parametric families and outperforms existing nonparametric methods for copulas with heterogeneous dependence patterns. On real data, we construct a vine copula model using the proposed bivariate estimator as a building block, achieving the best fit among the considered methods. Promising directions for future work include extensions to more general vine constructions and other multivariate copula models, as well as explicit control over the number of terms K in the separable representation.

5. Rank-Separable Smoothing for Checkerboard Copulas

This chapter is based on joint work with Maximilian Coblenz and Oliver Grothe (Publ. III). Based on the same idea as in Chapter 4, namely that a copula can be viewed as a perturbation of the independence copula, we propose a smoothing framework that maps checkerboard densities, for instance those arising from contingency tables, to a smooth copula density. The resulting construction removes the discontinuities inherent to checkerboard copulas while accurately preserving the empirical mass distribution.

5.1. Introduction

Dependence modeling is fundamental in multivariate statistics, with applications ranging from finance to engineering. When data are discrete or of mixed type, copula-based approaches face particular challenges: only the subcopula is uniquely identified, and extensions to the full unit cube are generally non-unique (Genest and Nešlehová, 2007; Geenens, 2020). Checkerboard copulas provide a convenient solution by multilinearly interpolating the subcopula, yielding a proper copula with uniform margins that connects naturally in contingency-table inference (Li et al., 1997; Genest et al., 2014). However, their piecewise constant density introduces block artifacts that hinder methods requiring smoothness. In this paper, we propose a smoothing framework that transforms a given checkerboard copula into a continuous copula density while preserving uniform margins and controlling the error in cellwise probability masses.

Copulas provide a fundamental framework for modeling multivariate dependence by separating marginal behavior from joint structure. For a d-variate cdf H with continuous margins $H_1, \dots, H_d$, Sklar's theorem (Sklar, 1959) implies a unique copula C such that

$$H(x_1, \dots, x_d) = C\big(H_1(x_1), \dots, H_d(x_d)\big).$$

Assuming C is absolutely continuous with density c, likelihood-based estimation is standard. However, when data are discrete or of mixed type, the marginal distributions exhibit jumps. Consequently, the copula is no longer uniquely determined; only the associated

subcopula is uniquely identified, that is, the restriction of the copula to $\mathrm{Ran}(H_1) \times \cdots \times \mathrm{Ran}(H_d)$ (see Nelsen, 2006, Definitions 2.10.5 and Theorem 2.10.9). Extending this subcopula to a copula on $[0,1]^d$ is generally not unique (Genest and Nešlehová, 2007; Geenens, 2020), raising the question of how to construct a principled extension.

A well-established answer is the multilinear interpolation of the subcopula (Genest et al., 2014). This construction extends the subcopula from the rank grid to the entire unit cube by linearly mixing the corner values within each cell, yielding a genuine copula with uniform margins and a piecewise constant density. In the literature, this is known as the checkerboard copula (Li et al., 1997; Carley and Taylor, 2002; Durante and Sempi, 2015), and in empirical settings corresponds to a multilinear smoothing of the empirical copula (Genest et al., 2017). The construction relies solely on ranks, is robust to ties, requires no bandwidth selection, and connects naturally to contingency-table inference (Genest et al., 2014; Genest et al., 2019). For applications, see Kuzmenko et al. (2020), Lin et al. (2025), Borwein and Howlett (2019), Cuberos et al. (2020), and Borwein et al. (2014). Despite these merits, the piecewise constant density remains a limitation when downstream methods require smoothness.

Our approach builds on the perturbation representation of copulas, which has been systematically investigated in Rodrıguez-Lallena and Úbeda-Flores (2004) and Mesiar and Najjari (2014). Starting from the independence copula $\Pi(u,v) = uv$, one can introduce dependence through separable additive perturbations of the form $C(u,v) = uv + \sum_k F_k(u)\,G_k(v)$. Based on this idea, a copula density can be represented as $c(u,v) = 1 + \sum_k f_k(u)\,g_k(v)$. Following the terminology used in Chapter 4, we refer to the univariate functions f_k and g_k in this separable expansion as component functions. For the resulting density to have uniform margins, these component functions must satisfy a so-called mean-zero condition. Chapter 4 makes this condition explicit, investigates the connection between such representations and the Schmidt decomposition, and develops a neural-network-based copula density estimator that learns the component functions f_k, g_k directly from data. For checkerboard copulas, Grothe and Rieger (2024) derive separable representations via singular value decomposition and establish connections between discrete and continuous decompositions, with a focus on low-rank approximation and graphical dependence analysis. We build on this representation to approximate checkerboard densities in finite-dimensional function spaces spanned by orthonormal bases. Related approaches have used function bases for copula construction more broadly. Shen et al. (2008) introduce linear B-spline copulas approximating a given copula from finitely many evaluations. Kauermann et al. (2013) propose a penalized spline approach on sparse-grid tensor bases, enforcing uniform margins via linear constraints. Ngounou Bakam and Pommeret (2025) develop a Legendre-polynomial-based estimator for copula densities. Lowin (2010) present a Fourier-based parametric copula family estimated via the

fast Fourier transform. Further smoothing methods for the empirical copula include the empirical beta copula and the empirical Bernstein copula (Segers et al., 2017; Sancetta and Satchell, 2004). Unlike these methods that estimate copulas directly from data, our approach specifically addresses the smoothing of given checkerboard copulas.

Motivated by the perturbation representation of copulas and its separable expansions, we develop a smoothing framework that approximates checkerboard copula densities in finite-dimensional subspaces spanned by orthonormal, mean-zero basis functions. We formulate the problem as a constrained quadratic optimization in L^2, where the objective naturally balances approximation quality with numerical stability through an implicit ridge-type regularization. We enforce nonnegativity of the estimated density as a constraint in the optimization problem, while the mean-zero property of the basis functions automatically guarantees uniform margins. We establish existence and uniqueness of the optimal density, prove consistency of the collocation-based discretization, and show that the error in cellwise probability masses is bounded by the density approximation error, even though mass preservation is not imposed explicitly. We investigate different basis choices, including Legendre polynomials, trigonometric functions, and B-splines, and demonstrate in simulations that the Legendre basis excels for smooth parametric targets, while the trigonometric basis is more efficient for sparse structures. Through applications to credit-rating transitions and U.S. age-income data, we show that the method removes artificial discontinuities while accurately maintaining empirical mass distributions. A reconstruction experiment further demonstrates that smooth copulas can recover fine-scale structure from aggregated data more accurately than checkerboard copulas.

The structure of the chapter is the following. Section 5.2 presents the theoretical framework for embedding checkerboard copulas into smooth densities and develops the computational approach. Section 5.3 provides an empirical study comparing different basis families. Section 5.4 illustrates the method on two real contingency tables: a one-year credit rating transition matrix and the 2023 U.S. age–income distribution. We conclude with remarks and an outlook. Supplementary code for this chapter is available at Liu et al. (2026b).

5.2. Continuous Embedding of a Checkerboard Copula Density

We begin by formally defining checkerboard copulas on general rectangular grids, which is essential for applications to contingency tables with differing row and column dimensions. Building on this, we formulate a continuous embedding problem in which the checkerboard copula density is approximated in L^2 by a separable representation within fixed

finite-dimensional subspaces, subject to pointwise nonnegativity of the resulting density on $[0,1]^2$. The associated computational scheme leads to a convex quadratic program after a collocation discretization, which can be solved efficiently. We establish the existence of solutions to the continuous embedding problem and show that the optimizers of the discrete collocation problems are consistent with the solution of the continuous problem.

5.2.1. Checkerboard Copulas

As briefly outlined in the introduction, the literature offers slightly different perspectives on the checkerboard copula: some are motivated by copula approximation (e.g., Li et al., 1997; Durante and Sempi, 2015), while others focus on modeling the dependence structure of discrete data (e.g., Genest et al., 2014; Genest and Nešlehová, 2007). In this work, we restrict attention to the bivariate case and specify the checkerboard copula directly via its density on a rectangular grid. We define a checkerboard copula as an absolutely continuous copula whose density is constant on each grid cell.

Definition 5.2.1 (Bivariate checkerboard copula on a general grid)**.** *Let* $0 = u_0 < u_1 < \cdots < u_m = 1$ *and* $0 = v_0 < v_1 < \cdots < v_n = 1$ *be (not necessarily equidistant) partitions of* $[0,1]$*. Using the notation* $[m] := \{1, \dots, m\}$*, for* $i \in [m]$ *and* $j \in [n]$*, we define the rectangular grid cells as*

$$I_i := \begin{cases} [0, u_1] & i = 1 \\ (u_{i-1}, u_i] & i \geq 2 \end{cases} \quad \text{and} \quad J_j := \begin{cases} [0, v_1] & j = 1 \\ (v_{j-1}, v_j] & j \geq 2 \end{cases}.$$

The corresponding cell widths are denoted by $\Delta u_i := u_i - u_{i-1}$ *and* $\Delta v_j := v_j - v_{j-1}$*.*

A copula $C^\sharp : [0,1]^2 \to [0,1]$ *is called a* checkerboard copula *(on the* $m \times n$ *grid) if it is absolutely continuous with a density* $c^\sharp : [0,1]^2 \to [0,\infty)$ *that is cellwise constant:*

$$c^\sharp(u,v) = c^\sharp_{ij} \quad \text{for } (u,v) \in I_i \times J_j,$$

where $c^\sharp_{ij} \geq 0$ *are the density values. Since* $C^\sharp$ *is a copula, it must have uniform margins, which implies:*

$$\sum_{j=1}^{n} c^\sharp_{ij} \, \Delta v_j = 1 \quad \forall\, i \in [m], \qquad \sum_{i=1}^{m} c^\sharp_{ij} \, \Delta u_i = 1 \quad \forall\, j \in [n].$$

Remark 5.2.1 (Mass-distribution matrix)**.** *Define*

$$\Pi = (\pi_{ij})_{i \in [m], j \in [n]} \in \mathbb{R}^{m \times n}_{\geq 0}, \quad \text{where} \quad \pi_{ij} := c^\sharp_{ij} \, \Delta u_i \, \Delta v_j$$

represents the probability mass in the cell $I_i \times J_j$. The uniform margin constraints then take the matrix form

$$\Pi \mathbf{1}_n = \Delta u, \qquad \mathbf{1}_m^\top \Pi = \Delta v^\top,$$

where $\mathbf{1}_m$ and $\mathbf{1}_n$ are vectors of ones, and $\Delta u = (\Delta u_1, \dots, \Delta u_m)^\top$, $\Delta v = (\Delta v_1, \dots, \Delta v_n)^\top$ collect the cell widths. Conversely, any nonnegative matrix Π satisfying these constraints induces a checkerboard copula via $c^\sharp_{ij} = \pi_{ij}/(\Delta u_i \, \Delta v_j)$.

5.2.2. Separable L^2 Approximation

The checkerboard copula density $c^\sharp$, as defined in Section 5.2.1, offers a flexible framework for modeling discrete dependencies. However, its cellwise constant nature typically yields a discontinuous density. This can be problematic in applications that require continuous (or even more regular) densities or when the discrete structure is an artifact of data discretization rather than an inherent feature. Therefore, a natural objective is to embed a given checkerboard copula into a richer class of absolutely continuous copulas that preserves the essential discrete structure while exhibiting desirable smoothness properties.

To address this question, we model the copula density using a finite *separable expansion* (Rodrıguez-Lallena and Úbeda-Flores, 2004; Mesiar and Najjari, 2014):

$$c(u, v) = 1 + \sum_{k=1}^{r} f_k(u) \, g_k(v), \qquad (u, v) \in [0, 1]^2, \tag{5.1}$$

where the constant term 1 corresponds to the independence copula density and the summation captures deviations from independence.

For the representation (5.1) to define a valid copula density, we impose the following two conditions (cf. Chapter 4, Proposition 4.2.1). First, the component functions f_k and g_k are assumed to belong to

$$L_0^2([0,1]) := \left\{ f \in L^2([0,1]) \, : \, \int_0^1 f(t) \, dt = 0 \right\}.$$

This mean-zero condition ensures that each term in the expansion integrates to zero along each coordinate. Consequently, the resulting bivariate density preserves the uniform margins of the constant baseline. Second, we require pointwise nonnegativity of the density itself, i.e.,

$$c(u, v) \geq 0 \quad \text{for all } (u, v) \in [0, 1]^2,$$

which guarantees that the function defined by (5.1) is a valid probability density on $[0, 1]^2$.

In the following, we restrict f_k and g_k to fixed finite-dimensional subspaces

$$\mathcal{V}_p \subset L_0^2([0,1]), \qquad \mathcal{W}_q \subset L_0^2([0,1]),$$

of dimensions p and q, respectively. We assume that these spaces are contained in $C([0,1]) \cap L_0^2([0,1])$.

To ensure that our formulation covers the entire tensor-product space $\mathcal{V}_p \otimes \mathcal{W}_q$ without imposing artificial restrictions on the dependence structure, we set the expansion rank to

$$r := \min(p, q).$$

Now we formulate the embedding problem as finding a representation of the form (5.1) that best approximates the checkerboard structure in the L^2 sense. This choice is theoretically grounded, as Chapter 4 establishes that every square-integrable copula density admits a (infinite) separable expansion; consequently, the L^2 norm provides the natural metric for quantifying the approximation error. Furthermore, this objective corresponds to the Mean Integrated Squared Error (MISE), a standard optimality criterion in nonparametric density estimation. Finally, this metric allows the objective function to be expressed as a convex quadratic form, rendering the optimization problem computationally tractable, as demonstrated in the discrete formulation in Section 5.2.3.

Problem 5.2.1 (L^2 density-fit in finite subspaces)**.** *Given a checkerboard copula density $c^\sharp$, finite-dimensional subspaces $\mathcal{V}_p, \mathcal{W}_q \subset C([0,1]) \cap L_0^2([0,1])$, and letting $r = \min(p,q)$, find functions*

$$f_k \in \mathcal{V}_p, \qquad g_k \in \mathcal{W}_q, \qquad k = 1, \dots, r,$$

that minimize the squared L^2 distance

$$\mathcal{J}\big((f_k, g_k)_{k=1}^r\big) := \int_0^1 \int_0^1 \left[c^\sharp(u,v) - \left(1 + \sum_{k=1}^r f_k(u) g_k(v)\right)\right]^2 du\, dv \tag{5.2}$$

subject to the pointwise nonnegativity of the expansion (5.1) *on* $[0,1]^2$*. The corresponding copula is then given by $C(u,v) = \int_0^u \int_0^v c(s,t)\, dt\, ds$.*

Remark 5.2.2 (Relaxation of the rank constraint)**.** *In Problem 5.2.1, despite setting $r = \min(p,q)$, we deliberately do not enforce linear independence within the families $\{f_k\}_{k=1}^r$ and $\{g_k\}_{k=1}^r$. Instead, we optimize over the set of functions with separable rank* at most *r. The rationale for this relaxation is topological: the set of functions with rank* exactly *r is not closed. A sequence of functions with rank r can converge to a limit with rank strictly less than r (e.g., by letting the amplitude of one component tend to zero). If we were to enforce the rank*

to be exactly r, the optimization problem would be ill-posed whenever the true dependence structure is simpler than the maximal rank r, leading to numerical instabilities. By working with the closure (rank ≤ r), we guarantee the existence of a minimizer.

Theorem 5.2.1 (Existence and uniqueness)**.** *Problem 5.2.1 admits a unique solution in terms of the optimal density. That is, there exists a unique density $c^* \in 1 + \mathcal{V}_p \otimes \mathcal{W}_q$ that minimizes the objective* (5.2) *subject to pointwise nonnegativity.*

Proof. See Appendix C.1. □

Remark 5.2.3 (Uniqueness of the density vs. the factorization)**.** *The uniqueness asserted in Theorem 5.2.1 refers to the optimal density c^*. The representation as a sum of separable terms (f_k, g_k) is generally not unique: in addition to sign changes and permutations of the summands, one may rescale factors, i.e., replace (f_k, g_k) by $(\alpha f_k, \alpha^{-1} g_k)$ for any $\alpha \neq 0$.*

The finite dimensionality of $\mathcal{V}_p$ and $\mathcal{W}_q$ provides the foundation for a numerical solution. In the following section, we will expand f_k and g_k in fixed bases to translate Problem 5.2.1 into an explicit matrix optimization problem. A further computational challenge is posed by the continuous nonnegativity constraint $c(u, v) \geq 0$. We will address this by relaxing the condition and enforcing nonnegativity only on a finite reference grid.

5.2.3. Discrete Matrix Formulation

To obtain a computable solution, we transform the functional optimization from Problem 5.2.1 into a finite-dimensional matrix problem. This transformation relies on representing the component functions in fixed orthonormal bases, which allows us to convert the integral objective into an algebraic form involving coefficient matrices.

Let $\mathcal{V}_p = \operatorname{span}\{\varphi_1, \dots, \varphi_p\}$ and $\mathcal{W}_q = \operatorname{span}\{\psi_1, \dots, \psi_q\}$ be subspaces of $L_0^2([0,1])$ spanned by orthonormal bases. In this setting, the resulting copula density is uniquely determined by a coefficient matrix $K \in \mathbb{R}^{p \times q}$ via

$$c_K(u, v) := 1 + \Phi(u)^\top K \Psi(v), \tag{5.3}$$

where $\Phi(u) := (\varphi_1(u), \dots, \varphi_p(u))^\top$ and $\Psi(v) := (\psi_1(v), \dots, \psi_q(v))^\top$.

We define the matrix $S \in \mathbb{R}^{p \times q}$ by the coefficients

$$S_{\mu\nu} := \iint_{[0,1]^2} (c^\sharp(u, v) - 1)\, \varphi_\mu(u)\, \psi_\nu(v)\, du\, dv, \qquad \mu = 1, \dots, p,\ \nu = 1, \dots, q.$$

The matrix S will serve as the finite-dimensional representation of the checkerboard target in the chosen bases and will allow us to rewrite the L^2 density-fit objective in a purely

algebraic form; see Proposition 5.2.1 and (5.5). Moreover, S is the coefficient matrix of the L^2-projection of $c^\sharp - 1$ onto the tensor-product space $\mathcal{V}_p \otimes \mathcal{W}_q$ with respect to the chosen orthonormal bases.

Using the cell-integral matrices $A_{i\mu} := \int_{I_i} \varphi_\mu(u)\,du$ and $B_{j\nu} := \int_{J_j} \psi_\nu(v)\,dv$, this target matrix can be computed efficiently via

$$S = A^\top(\mathbf{D}^\sharp - \mathbf{1}_{m\times n})B, \tag{5.4}$$

where $\mathbf{D}^\sharp \in \mathbb{R}^{m\times n}$ is defined by $(\mathbf{D}^\sharp)_{ij} := c^\sharp_{ij}$.

The following proposition establishes that minimizing the continuous L^2 error is mathematically equivalent to minimizing the Frobenius-norm error of the coefficient matrix.

Proposition 5.2.1 (Matrix representation of the L^2 error)**.** *For any coefficient matrix $K \in \mathbb{R}^{p\times q}$, the squared L^2 distance admits the decomposition*

$$\iint_{[0,1]^2} \Big[(c^\sharp - 1) - \Phi^\top K \Psi\Big]^2 \,du\,dv \;=\; const + \|K - S\|_F^2, \tag{5.5}$$

where the constant term depends only on $c^\sharp$ and S. Consequently,

$$\arg\min_K \| c^\sharp - c_K \|_{L^2}^2 \;=\; \arg\min_K \|K - S\|_F^2.$$

Proof. We expand the squared L^2 error:

$$\mathcal{J}(K) = \|c^\sharp - 1\|_{L^2}^2 - 2\iint_{[0,1]^2} (c^\sharp - 1)(\Phi^\top K \Psi)\,du\,dv + \|\Phi^\top K \Psi\|_{L^2}^2.$$

The linear term represents the inner product of the functions. Using the expansion $\Phi^\top K \Psi = \sum_{\mu,\nu} K_{\mu\nu}\varphi_\mu\psi_\nu$ and the definition of the projection matrix S, this simplifies to the Frobenius inner product:

$$\sum_{\mu,\nu} K_{\mu\nu} \underbrace{\iint (c^\sharp - 1)\varphi_\mu(u)\psi_\nu(v)\,du\,dv}_{=S_{\mu\nu}} = \sum_{\mu,\nu} K_{\mu\nu}S_{\mu\nu} = \langle S, K\rangle_F.$$

For the quadratic term, we expand the squared sum and apply linearity:

$$\begin{aligned}\|\Phi^\top K\Psi\|_{L^2}^2 &= \iint_{[0,1]^2} \left(\sum_{\mu,\nu} K_{\mu\nu}\varphi_\mu(u)\psi_\nu(v)\right)\left(\sum_{\mu',\nu'} K_{\mu'\nu'}\varphi_{\mu'}(u)\psi_{\nu'}(v)\right) du\, dv \\ &= \sum_{\mu,\nu}\sum_{\mu',\nu'} K_{\mu\nu}K_{\mu'\nu'} \underbrace{\int_0^1 \varphi_\mu\varphi_{\mu'}\, du}_{=\delta_{\mu\mu'}} \underbrace{\int_0^1 \psi_\nu\psi_{\nu'}\, dv}_{=\delta_{\nu\nu'}}.\end{aligned}$$

Due to the orthonormality of the bases, only terms with indices $(\mu,\nu) = (\mu',\nu')$ are nonzero. The quadruple sum thus collapses to the sum of squared entries:

$$\|\Phi^\top K\Psi\|_{L^2}^2 = \sum_{\mu,\nu} K_{\mu\nu}^2 = \|K\|_F^2.$$

Substituting these results back into the expansion yields

$$\mathcal{J}(K) = \|c^\sharp - 1\|_{L^2}^2 - 2\langle S, K\rangle_F + \|K\|_F^2.$$

Completing the square in the Frobenius norm, we obtain $\|K\|_F^2 - 2\langle S,K\rangle_F = \|K-S\|_F^2 - \|S\|_F^2$. Thus,

$$\mathcal{J}(K) = \underbrace{\left(\|c^\sharp - 1\|_{L^2}^2 - \|S\|_F^2\right)}_{\text{constant}} + \|K-S\|_F^2,$$

which proves that minimizing $\mathcal{J}(K)$ is equivalent to minimizing $\|K-S\|_F^2$. □

While the objective function simplifies algebraically, the global nonnegativity constraint $c_K(u,v) \geq 0$ requires a discretization. We enforce the condition on a finite reference grid $\Xi = \{(x_i, y_j)\} \subset [0,1]^2$ of size $N_g = N_x \times N_y$. Letting $\Phi_g \in \mathbb{R}^{N_x\times p}$ and $\Psi_g \in \mathbb{R}^{N_y\times q}$ denote the evaluation matrices of the basis functions on this grid, we arrive at the final optimization problem.

Problem 5.2.2 (Discrete Matrix Optimization)**.** *Given the matrix S defined in* (5.4) *and grid matrices Φ_g, Ψ_g, find*

$$K^* = \arg\min_{K\in\mathbb{R}^{p\times q}} \|K-S\|_F^2$$

subject to the elementwise linear inequality constraints

$$\Phi_g K \Psi_g^\top \geq -\mathbf{1}_{N_x\times N_y}.$$

Theorem 5.2.2 (Existence and Uniqueness)**.** *Problem 5.2.2 admits a unique solution K^*.*

Proof. The objective function $f(K) = \|K-S\|_F^2$ is strictly convex and coercive. The feasible set is closed (as the intersection of closed half-spaces defined by the linear inequalities) and non-empty (since $K = 0$ satisfies the constraints). A strictly convex function on a non-empty closed convex set always attains a unique minimum. □

Remark 5.2.4 (Implicit ridge regularization)**.** *The objective function in Problem 5.2.2 can be expanded as*

$$\|K - S\|_F^2 = \|K\|_F^2 - 2\langle K, S\rangle_F + \|S\|_F^2,$$

where $\langle A, B\rangle_F := \operatorname{tr}(A^\top B)$ *is the Frobenius inner product. Since* $\|S\|_F^2$ *is constant with respect to K, minimizing the distance is equivalent to*

$$\min_{K\in\mathbb{R}^{p\times q}} \Big\{ \underbrace{\|K\|_F^2}_{\text{ridge penalty}} - \underbrace{2\langle K, S\rangle_F}_{\text{alignment with } S} \Big\} \quad \text{subject to} \quad \Phi_g K \Psi_g^\top \geq -\mathbf{1}_{N_x\times N_y}.$$

This reveals that our formulation inherently acts as a ridge-regularized optimization with parameter $\lambda = 1$*. The term* $\|K\|_F^2$ *penalizes large coefficients, thereby controlling the deviation from the independence copula (which corresponds to* $K = 0$*). This implicit regularization ensures numerical stability by keeping the objective strictly convex and well-conditioned. Furthermore, it prevents extreme oscillations in the fitted density, effectively balancing fidelity to the checkerboard projection S with the smoothness requirements imposed by the basis functions.*

Once the optimal matrix K^* is determined, the copula density is fully defined via (5.3), which inherently implies a separable form as in (5.1). However, in a direct expansion, the resulting families of component functions $\{f_k\}$ and $\{g_k\}$ are not guaranteed to be orthogonal within their respective spaces. In this context, the singular value decomposition offers a convenient way to derive an equivalent formulation of the sum: it recovers a minimal representation where the singular values act as explicit weights and the associated component functions form orthonormal systems.

Proposition 5.2.2 (Separable representation)**.** *Let* $K^* \in \mathbb{R}^{p\times q}$ *be the solution to Problem 5.2.2 with effective rank* $r_{\text{eff}} := \operatorname{rank}(K^*)$*, and let* $K^* = U\Sigma V^\top$ *be its compact singular value decomposition, where* $U \in \mathbb{R}^{p\times r_{\text{eff}}}$ *and* $V \in \mathbb{R}^{q\times r_{\text{eff}}}$ *have orthonormal columns, and* $\Sigma = \operatorname{diag}(\sigma_1, \dots, \sigma_{r_{\text{eff}}})$ *with* $\sigma_1 \geq \dots \geq \sigma_{r_{\text{eff}}} > 0$*. We denote the k-th column of U by* $U_{:,k}$ *and similarly for V. Then the optimal density is given by*

$$c_{K^*}(u, v) = 1 + \sum_{k=1}^{r_{\text{eff}}} \sigma_k f_k(u) g_k(v),$$

where the component functions

$$f_k(u) := \Phi(u)^\top U_{:,k} \quad \textit{and} \quad g_k(v) := \Psi(v)^\top V_{:,k}$$

are defined using the k-th columns $U_{:,k}$ and $V_{:,k}$ of the singular matrices. These functions form orthonormal systems in $L_0^2([0,1])$.

Proof. First, we substitute the SVD factorization $K^* = \sum_{k=1}^{r_{\text{eff}}} \sigma_k U_{:,k} V_{:,k}^\top$ into the matrix density formula (5.3):

$$c_{K^*}(u,v) = 1 + \Phi(u)^\top \left(\sum_{k=1}^{r_{\text{eff}}} \sigma_k U_{:,k} V_{:,k}^\top \right) \Psi(v) = 1 + \sum_{k=1}^{r_{\text{eff}}} \sigma_k \underbrace{\left(\Phi(u)^\top U_{:,k}\right)}_{=f_k(u)} \underbrace{\left(V_{:,k}^\top \Psi(v)\right)}_{=g_k(v)}.$$

This confirms the expansion formula. To verify the orthonormality of the functions $\{f_k\}$, we compute their L^2 inner product explicitly. Recalling that the vector-matrix product $f_k(u) = \Phi(u)^\top U_{:,k}$ corresponds to the sum $\sum_{\mu=1}^{p} U_{\mu k} \varphi_\mu(u)$, we have:

$$\langle f_k, f_l \rangle_{L^2} = \int_0^1 \left(\sum_{\mu=1}^{p} U_{\mu k} \varphi_\mu(u) \right) \left(\sum_{\nu=1}^{p} U_{\nu l} \varphi_\nu(u) \right) du.$$

By pulling the sums out of the integral and using the orthonormality of the basis functions ($\int \varphi_\mu \varphi_\nu = \delta_{\mu\nu}$), this simplifies to:

$$\langle f_k, f_l \rangle_{L^2} = \sum_{\mu=1}^{p} \sum_{\nu=1}^{p} U_{\mu k} U_{\nu l} \underbrace{\int_0^1 \varphi_\mu(u) \varphi_\nu(u)\, du}_{=\delta_{\mu\nu}} = \sum_{\mu=1}^{p} U_{\mu k} U_{\mu l}.$$

The remaining sum is exactly the dot product of the k-th and l-th columns of the matrix U. Since U has orthonormal columns (i.e., $U^\top U = I$), it holds that $\sum_\mu U_{\mu k} U_{\mu l} = \delta_{kl}$. Therefore, $\langle f_k, f_l \rangle_{L^2} = \delta_{kl}$. The proof for the functions $\{g_k\}$ follows analogously using the orthonormality of the columns of V. □

Remark 5.2.5 (Weighted vs. unweighted separable representation)**.** *The general form* (5.1) *does not include explicit weights, whereas the SVD-based representation in Proposition 5.2.2 involves singular values σ_k. These formulations are equivalent: the weights can be absorbed into the component functions, e.g., by defining $\tilde{f}_k := \sigma_k f_k$. However, this absorption destroys the orthonormality of $\{f_k\}$.*

5.2.4. Consistency and Mass Preservation

Having established the computational framework, we now analyze the theoretical properties of this discrete approximation. We first show that, for fixed finite-dimensional bases $\mathcal{V}_p$ and $\mathcal{W}_q$, our discretization strategy based on *collocation* (i.e., enforcing the nonnegativity constraint strictly on a finite grid) consistently approximates the continuous problem. To that end, it is convenient to formulate the optimization problem directly in terms of the coefficient matrix K.

For fixed bases $\mathcal{V}_p$ and $\mathcal{W}_q$ with associated evaluation maps Φ and Ψ, consider the "continuous" optimization problem

$$\min_{K\in\mathbb{R}^{p\times q}} \|K-S\|_F^2 \quad \text{subject to} \quad c_K(u,v) = 1 + \Phi(u)^\top K \Psi(v) \geq 0 \; \forall (u,v) \in [0,1]^2. \tag{5.6}$$

By the equivalence established in the proof of Theorem 5.2.1, problem (5.6) admits a unique solution.

Theorem 5.2.3 (Consistency of the collocation approximation)**.** *Fix finite-dimensional bases $\mathcal{V}_p$ and $\mathcal{W}_q$ and the corresponding matrix S. Let $(\mathcal{G}_\ell)_{\ell\in\mathbb{N}}$ be a sequence of collocation grids*

$$\mathcal{G}_\ell = \{(x_i^{(\ell)}, y_j^{(\ell)}) : i = 1,\dots,N_x^{(\ell)},\ j = 1,\dots,N_y^{(\ell)}\} \subset [0,1]^2$$

such that for every open set $U \subset [0,1]^2$, there exists $L \in \mathbb{N}$ with $\mathcal{G}_\ell \cap U \neq \emptyset$ for all $\ell \geq L$. For each ℓ, let K_ℓ^* denote the solution of the discrete problem

$$\min_{K\in\mathbb{R}^{p\times q}} \|K-S\|_F^2 \quad \textit{subject to} \quad c_K(x_i^{(\ell)}, y_j^{(\ell)}) \geq 0 \quad \forall (x_i^{(\ell)}, y_j^{(\ell)}) \in \mathcal{G}_\ell,$$

and denote the corresponding densities by $c_{K_\ell^*}(u,v) = 1 + \Phi(u)^\top K_\ell^* \Psi(v)$.

Then

$$K_\ell^* \to K^\star \quad \textit{and} \quad c_{K_\ell^*} \to c_{K^\star} \textit{ in } L^2([0,1]^2),$$

where $K^\star$ is the unique solution of the continuous problem (5.6)*. In particular, $c_{K^\star}(u,v) \geq 0$ for all $(u,v) \in [0,1]^2$ and $c_{K^\star}$ solves Problem 5.2.1 with the fixed finite-dimensional spaces $\mathcal{V}_p$ and $\mathcal{W}_q$.*

Proof. See Appendix C.2. □

This theorem confirms that our discrete computational approach provides a consistent approximation of the continuous nonnegativity constraint: as the collocation grid becomes dense, the discrete quadratic programs recover the solution of the "ideal" continuous problem (5.6), and thus of Problem 5.2.1 with fixed finite-dimensional bases.

Remark 5.2.6 (Regularization by finite dimensions)**.** *The consistency result in Theorem 5.2.3 relies on fixed basis dimensions p and q. While increasing these dimensions to infinity would theoretically minimize the approximation error, it would also reconstruct the discontinuities of the underlying checkerboard density. Since our objective is to obtain a smooth copula density, we treat the basis dimensions as regularization parameters. By fixing finite values for p and q, we restrict the solution to a subspace of smooth functions and effectively filter out the artifacts of the discrete grid.*

While our primary objective is to approximate the checkerboard density $c^\sharp$ in the L^2 sense, a natural question arises about how well the resulting copula c_{K^*} preserves the original mass distribution Π. Notably, although mass preservation is not imposed explicitly, we can show that accurate density approximation entails accurate mass approximation.

Proposition 5.2.3 (Mass preservation through density approximation)**.** *Let c_{K^*} be the solution of Problem 5.2.2 and $c^\sharp$ the checkerboard target density. For each cell $I_i \times J_j$ define the probability mass error*

$$E_{ij} := \iint_{I_i \times J_j} (c_{K^*} - c^\sharp)\, du\, dv.$$

Equivalently, $E_{ij} = \pi^*_{ij} - \pi^\sharp_{ij}$, where

$$\pi^*_{ij} := \iint_{I_i \times J_j} c_{K^*}(u,v)\, du\, dv, \qquad \pi^\sharp_{ij} := \iint_{I_i \times J_j} c^\sharp(u,v)\, du\, dv.$$

Then

$$|E_{ij}| \le \sqrt{\Delta u_i \Delta v_j}\, \|c_{K^*} - c^\sharp\|_{L^2(I_i \times J_j)}$$

and consequently

$$\|E\|_F \le \max_{i,j} \sqrt{\Delta u_i \Delta v_j}\, \|c_{K^*} - c^\sharp\|_{L^2([0,1]^2)}.$$

For equidistant grids with $\Delta u_i = 1/m$ and $\Delta v_j = 1/n$ this simplifies to

$$\|E\|_F \le \frac{1}{\sqrt{mn}}\, \|c_{K^*} - c^\sharp\|_{L^2([0,1]^2)}.$$

Proof. Fix a cell $I_i \times J_j$ and set $f = c_{K^*} - c^\sharp$ and $g \equiv 1$ on $I_i \times J_j$. By the Cauchy-Schwarz inequality,

$$|E_{ij}| = \left| \iint_{I_i \times J_j} f g\, du\, dv \right| \le \|f\|_{L^2(I_i \times J_j)} \|g\|_{L^2(I_i \times J_j)} = \|c_{K^*} - c^\sharp\|_{L^2(I_i \times J_j)} \sqrt{\Delta u_i \Delta v_j}.$$

Squaring this inequality yields

$$E_{ij}^2 \leq \Delta u_i \Delta v_j \, \|c_{K^*} - c^\sharp\|^2_{L^2(I_i \times J_j)}.$$

Since the cells $\{I_i \times J_j\}$ partition $[0,1]^2$, summing over all (i,j) yields

$$\|E\|_F^2 = \sum_{i,j} E_{ij}^2 \leq \max_{i,j}(\Delta u_i \Delta v_j) \sum_{i,j} \|c_{K^*} - c^\sharp\|^2_{L^2(I_i \times J_j)} = \max_{i,j}(\Delta u_i \Delta v_j) \, \|c_{K^*} - c^\sharp\|^2_{L^2([0,1]^2)}.$$

Taking square roots completes the proof. □

This proposition highlights an important feature of our approach: the mass distribution error is naturally controlled by the cell size. For practical applications where both density smoothness and mass preservation are important, this bound ensures that our density-focused optimization still yields reliable approximations of the underlying dependence structure.

5.3. Empirical Study

Having established the theoretical framework for embedding checkerboard copulas into smooth densities, we now investigate the practical performance of our method. This section compares smoothing performance across different basis families to understand how the choice of subspace affects the fidelity of the continuous approximation. We consider three subspace constructions: (i) a Legendre basis as a polynomial model, (ii) a Cosine basis as a trigonometric model (referencing the Sine/Cosine basis functions used in spectral methods), and (iii) a cubic B-spline basis. In all cases, the bases are constructed to be orthonormal on $[0,1]$ and mean-free. Formal definitions and construction details are provided in Appendix C.3. Throughout the study, we use $N_g = 40 \times 40$ collocation points for enforcing the nonnegativity constraints.

As a first generic test case, we generate a random checkerboard copula on a 10×10 grid. To ensure that the target mass distribution matrix Π satisfies the marginal constraints, we construct it as a convex combination of random permutation matrices. This yields a valid doubly-stochastic matrix with an irregular structure, allowing us to test the method in a scenario without strong parametric assumptions. Next, we evaluate the method's ability to approximate checkerboard copulas derived from three parametric continuous copulas: Clayton ($\theta = 1.5$), Gumbel ($\theta = 1.5$), and Gaussian ($\rho = 0.5$). We discretize each copula by partitioning $[0,1]^2$ into a 10×10 grid. On each grid cell, we define a piecewise-constant approximation by setting the density equal to its average value over that cell. Finally, we

investigate the challenging scenario of sparse or singular targets, specifically the comonotonic and countermonotonic copulas, as well as two distinct permutation patterns.

To assess the quality of the embedding, we employ two complementary metrics:

1. The relative Frobenius error (RFE) of the reconstructed cell-mass matrix to measure discrete mass preservation:

$$\mathrm{RFE} = \frac{\|\Pi^{\text{target}} - \Pi^{\text{reconstructed}}\|_F}{\|\Pi^{\text{target}}\|_F}.$$

2. The absolute L^2 density error ($\mathcal{E}_{L^2}$) to measure continuous shape fidelity:

$$\mathcal{E}_{L^2} = \|c_K - c_{\text{ref}}\|_{L^2([0,1]^2)}.$$

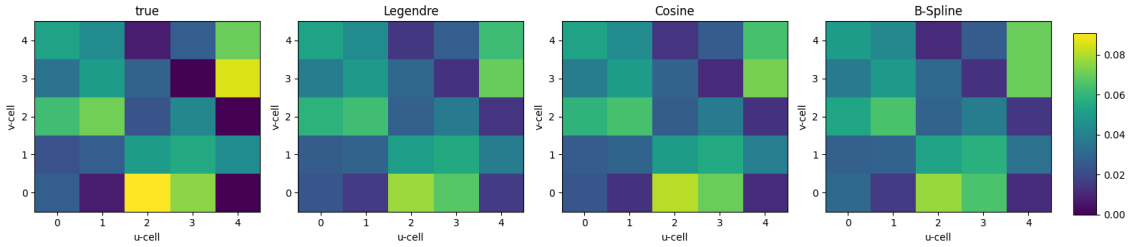
For the random checkerboard copula, the reconstructed mass distributions are visually indistinguishable from the target across all three bases (Fig. 5.1). However, the quantitative analysis of the densities reveals subtle differences. While the Legendre and B-spline bases produce very smooth approximations, the Cosine basis proves slightly more efficient at capturing the irregular structure of the random mixture. Consequently, it achieves the lowest density error (0.280) compared to Legendre (0.304) and B-spline (0.303) in this scenario. To ensure that this result is not merely an artifact of a single realization, we explicitly validate this finding across 100 independent trials in the Monte Carlo study at the end of this section.

In the case of the smooth parametric targets (Clayton, Gaussian, Gumbel), the Legendre basis consistently yields the most accurate approximations. As shown in Figures 5.2 and 5.3, it captures the gradual transitions and peak structures, such as the sharp tail dependence of the Clayton copula, with high fidelity. The Cosine basis preserves the discrete mass well but exhibits larger deviations from the reference density in these cases.

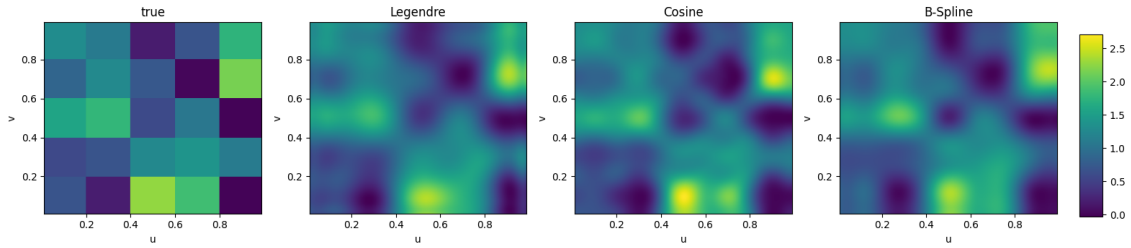
A different picture emerges for the sparse permutation structures. For the comonotonic, countermonotonic, and permutation-based copulas shown in Figures 5.4 and 5.5, the Cosine model consistently achieves the lowest reconstruction errors in terms of both mass preservation and density shape. Its spectral nature allows it to represent the singular behavior of the underlying permutation matrices more efficiently than the polynomial bases. The polynomial models tend to produce wider smoothing artifacts around the concentrated masses.

Table 5.1 summarizes the quantitative benchmarks for all test cases and highlights the lowest errors in bold.

Finally, we analyze the influence of the subspace dimension $p = q$. To ensure statistical robustness, we performed a Monte Carlo study with 100 independent realizations gener-



(a) Reconstructed mass

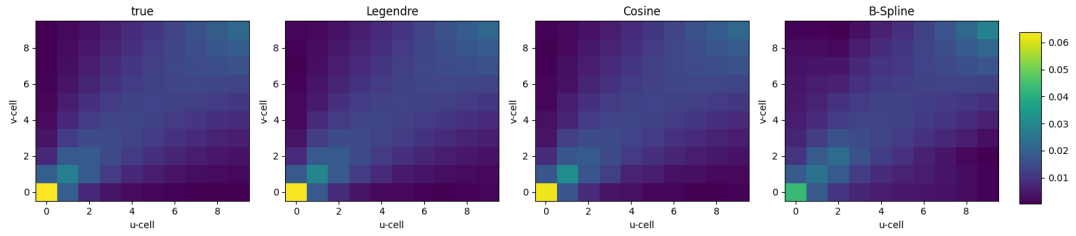


(b) Reconstructed densities c_K

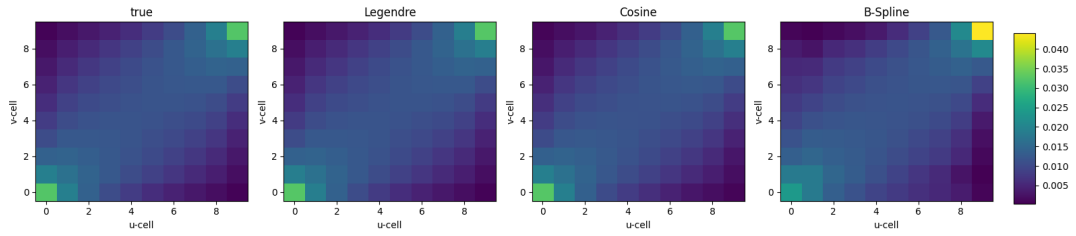
Figure 5.1.: Reconstructed mass distributions (top) and corresponding smooth densities (bottom) for the random checkerboard target. Columns from left to right display: target, Legendre, Cosine, and B-spline reconstructions.

	Legendre		Cosine		B-spline	
Target	RFE	$\mathcal{E}_{L^2}$	RFE	$\mathcal{E}_{L^2}$	RFE	$\mathcal{E}_{L^2}$
Checkerboard from Continuous Copulas						
Clayton ($\theta = 1.5$)	**0.025**	**1.183**	0.047	1.286	0.213	1.249
Gaussian ($\rho = 0.5$)	**0.002**	**0.066**	0.006	0.154	0.158	0.271
Gumbel ($\theta = 1.5$)	**0.007**	**0.587**	0.061	0.690	0.180	0.648
Checkerboard Structures						
Random checkerboard	0.163	0.304	**0.134**	**0.280**	0.162	0.303
Comonotonic	0.416	1.101	**0.325**	**0.962**	0.394	1.071
Countermonotonic	0.416	1.101	**0.325**	**0.962**	0.377	1.032
Permutation 1	0.429	1.138	**0.324**	**0.974**	0.380	1.071
Permutation 2	0.427	1.130	**0.326**	**0.975**	0.396	1.123

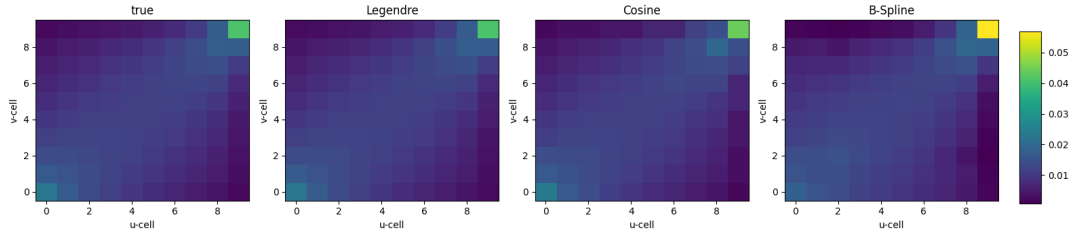
Table 5.1.: Benchmark summary reporting mass error (RFE) and continuous density error ($\mathcal{E}_{L^2}$). Minimum errors for each metric are highlighted in bold.



(a) Clayton ($\theta = 1.5$)



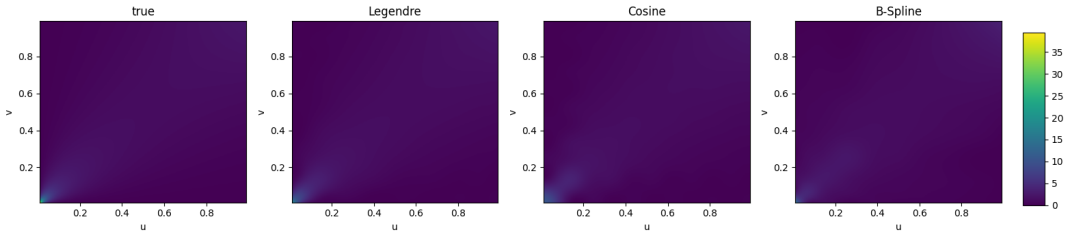
(b) Gaussian ($\rho = 0.5$)



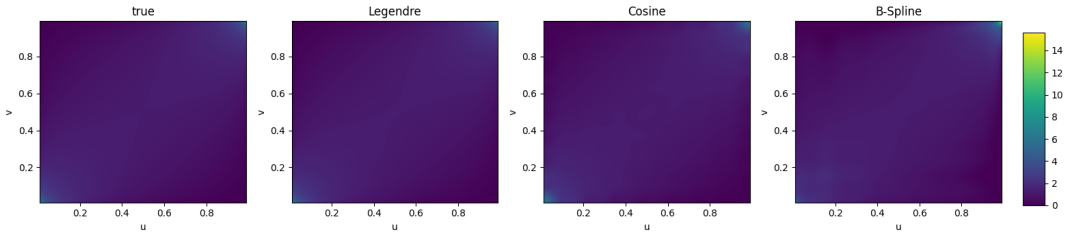
(c) Gumbel ($\theta = 1.5$)

Figure 5.2.: Reconstructed mass distributions for continuous parametric targets. In each row, columns from left to right display: target, Legendre, Cosine, and B-spline.

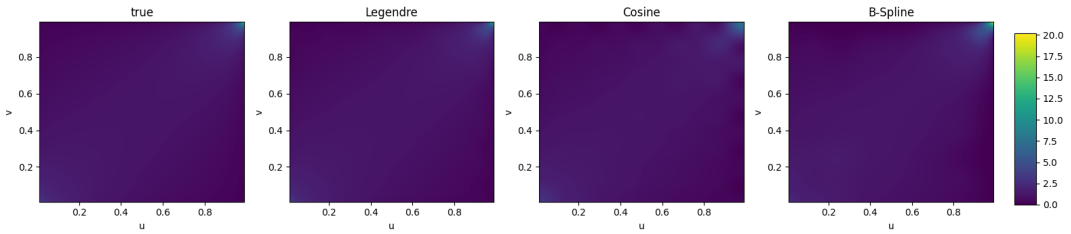
ated as convex combinations of random permutation matrices on a 5×5 grid. The results in Table 5.2 show that the approximation error decreases for all bases as the dimension increases. Confirming the trend observed in the initial single example, the Cosine basis yields the lowest errors across all tested dimensions. Notably, at the lowest dimension $p = 4$, the Cosine basis provides a significantly better approximation than the Legendre and B-spline bases, indicating a higher efficiency in representing these random mixtures with few parameters. This convergence behavior is visually illustrated in Figure 5.6.



(a) Clayton Density

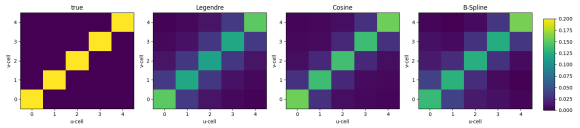


(b) Gaussian Density

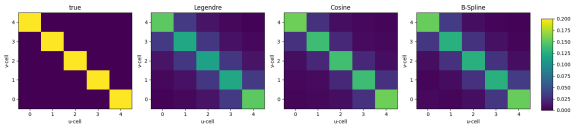


(c) Gumbel Density

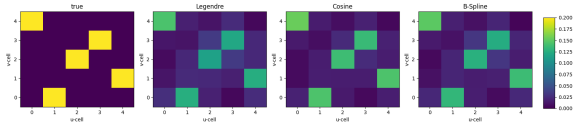
Figure 5.3.: Reconstructed smooth densities c_K for continuous parametric targets. In each row, columns from left to right display: target, Legendre, Cosine, and B-spline.



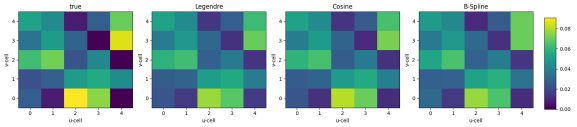
(a) Comonotonic



(b) Countermonotonic

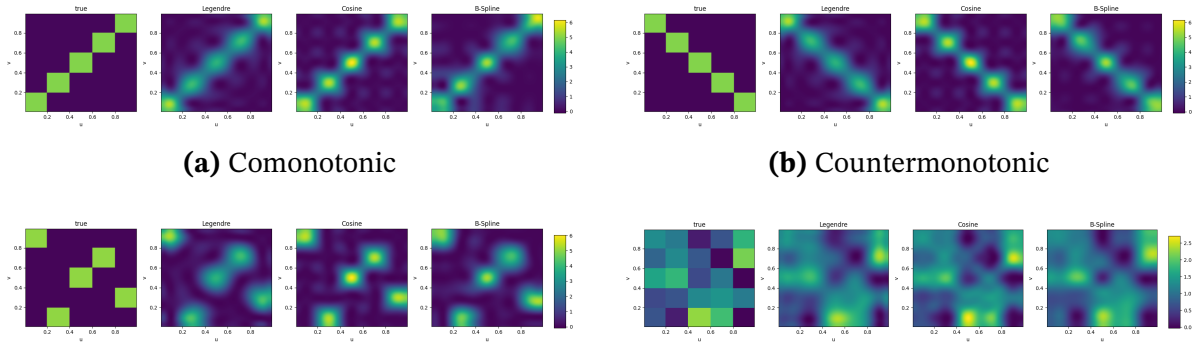


(c) Permutation 1



(d) Permutation 2

Figure 5.4.: Reconstructed mass distributions for sparse permutation targets. Columns within each plot from left to right: target, Legendre, Cosine, B-spline.



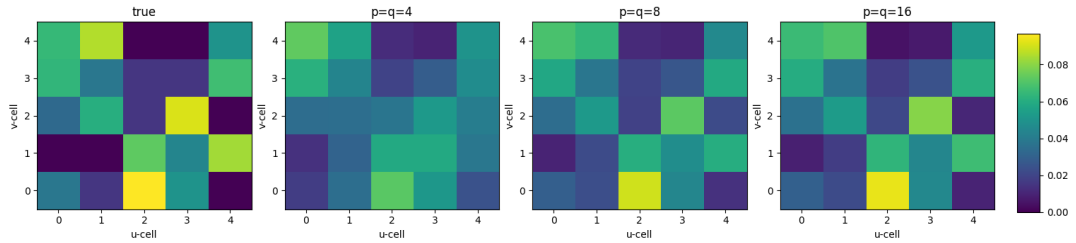
(a) Comonotonic **(b)** Countermonotonic

(c) Permutation 1 **(d)** Permutation 2

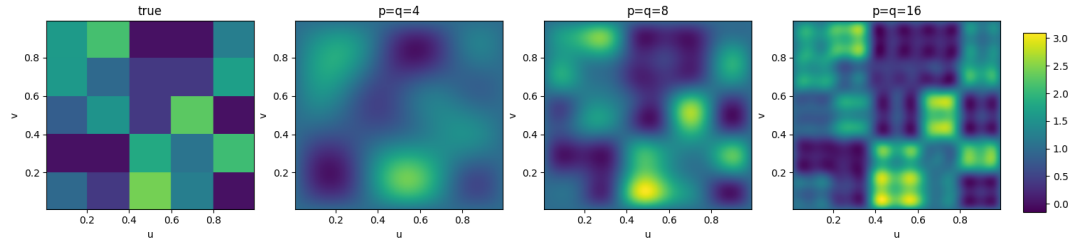
Figure 5.5.: Reconstructed smooth densities c_K for sparse permutation targets. Columns within each plot from left to right: target, Legendre, Cosine, B-spline.

	RFE (Mass Error)			$\mathcal{E}_{L^2}$ (Density Error)		
Method	$p = 4$	$p = 8$	$p = 16$	$p = 4$	$p = 8$	$p = 16$
Legendre	0.470 $\pm$ 0.063	0.260 $\pm$ 0.042	0.192 $\pm$ 0.030	0.681 $\pm$ 0.110	0.484 $\pm$ 0.080	0.429 $\pm$ 0.065
Cosine	**0.343** $\pm$ 0.055	**0.219** $\pm$ 0.035	**0.159** $\pm$ 0.026	**0.563** $\pm$ 0.094	**0.438** $\pm$ 0.067	**0.422** $\pm$ 0.063
B-spline	0.489 $\pm$ 0.061	0.254 $\pm$ 0.043	0.167 $\pm$ 0.027	0.703 $\pm$ 0.110	0.484 $\pm$ 0.077	0.431 $\pm$ 0.066

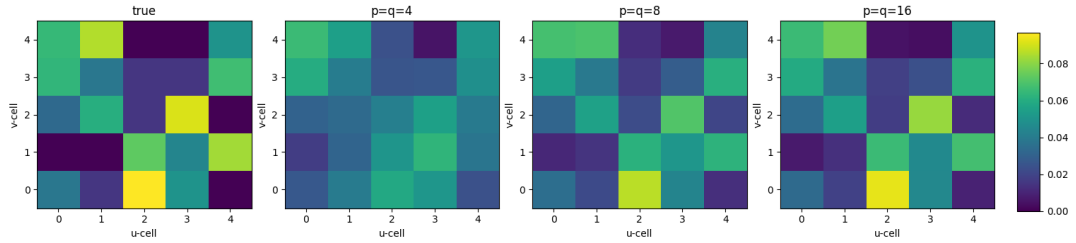
Table 5.2.: Impact of basis dimension $p = q$ on approximation error for 100 randomly generated checkerboard copulas (mean $\pm$ std). Column-wise minima (highlighting the best performing basis for each dimension) are in bold.



(a) Legendre



(b) Cosine



(c) B-spline

Figure 5.6.: Visual convergence of the reconstructed densities for increasing subspace dimension $p = q \in \{4, 8, 16\}$. Columns 2–4 correspond to $p = 4, 8, 16$.

5.4. Real Data Illustration

We illustrate the smoothing procedure with two real-world contingency tables that differ in structure and application domain: a 9×10 credit rating transition table from Scope Ratings (Kulo and Poulain, 2025), and a 16×4 age-by-income table from the 2023 American Community Survey (U.S. Census Bureau, 2024).

5.4.1. Rating Transitions

Credit ratings assess the creditworthiness of bond issuers on an ordinal scale from AAA (highest quality) through intermediate grades (AA, A, BBB, BB, B, CCC, CC, C) to D (de-

fault). Transition tables record how ratings evolve over a fixed horizon and are widely used in credit risk management (Cantor, 2004).

We analyze data from Scope Ratings' *Credit Rating Transition and Default Study 2024* (Kulo and Poulain, 2025), which tracks 995 credit ratings over the one-year period from December 31, 2023 to December 31, 2024. The original data is a contingency table where each row corresponds to a rating class at the beginning of the period, each column to a rating class at the end, and the entries indicate the percentage of issuers transitioning between classes (see Appendix C.4 for the complete table). In addition to the nine rating classes (AAA through C) and default, the table includes columns for withdrawn ratings (WR) and paid-off obligations.

Since withdrawn ratings and paid-off obligations represent administrative exits rather than credit events, we exclude these columns from our analysis. After removing them, we renormalize each row to sum to one, yielding a proper transition probability matrix. The resulting joint distribution is shown in Fig. 5.8 (left panel).

To separate the marginal distributions from the dependence structure, we transform this transition matrix into a checkerboard copula. Each rating class is mapped to a quantile interval on $[0, 1]$ based on its marginal frequency. Specifically, AAA-rated issuers constitute 13.0% of the population and are mapped to $u \in [0, 0.130)$; AA-rated issuers (8.4%) are mapped to $u \in [0.130, 0.214)$; and so forth for the remaining classes. Applying the same transformation to the columns, each cell (i, j) of the transition matrix corresponds to a rectangle $[u_{i-1}, u_i) \times [v_{j-1}, v_j)$ on the unit square, and the copula density is constant within each rectangle. The resulting checkerboard copula density (Fig. 5.7, left panel) exhibits the expected structure: strong concentration along the diagonal (i.e., issuers tend to retain their current rating), off-diagonal mass capturing upgrades and downgrades, and a narrow region for default transitions.

The checkerboard representation introduces artificial discontinuities that arise from the discrete rating grid rather than the underlying credit dynamics. Fig. 5.9 illustrates this issue: the conditional probability $\Pr\{\text{rating} \geq A \mid U = u\}$ drops abruptly from 98.8% to 4.8% at $u = 0.4885$, the boundary between A and BBB ratings.

In line with structural models of credit risk and their continuous latent variable framework (Merton, 1974; Lando, 2004), we interpret this discontinuity as an artifact of the classification scheme rather than a feature of the underlying phenomenon. Credit ratings discretize a latent continuous variable, the issuer's creditworthiness, into a finite number of categories. Two issuers rated A and BBB may have nearly identical default probabilities if they lie close to the boundary; the rating difference reflects the granularity of the classification, not a fundamental distinction in credit quality. Rating agencies implicitly acknowledge this by subdividing broad categories into finer notches (e.g., BBB+, BBB, BBB-) and by assigning outlooks that signal likely future changes (Bannier and Hirsch,

2010). Market prices provide further evidence: bond spreads vary smoothly across the credit spectrum without jumps at rating boundaries (Zhu, 2006).

Beyond the conceptual argument, a smooth representation offers practical advantages. Many downstream applications, such as interpolating transition probabilities for unobserved rating notches, computing derivatives for sensitivity analysis, or integrating over the copula for risk aggregation, require or benefit from continuous functions. A step function, while consistent with the observed data, complicates these tasks unnecessarily.

We resolve this issue by fitting a smooth copula density using Problem 5.2.2. We employ the Legendre basis with dimension $p = q = 20$ and enforce nonnegativity on a grid of $N_g = 40 \times 40$ collocation points. The resulting smooth density (Fig. 5.7, right panel) preserves the empirical cell masses (compare Fig. 5.8, left and right panels) while interpolating continuously between them. The conditional threshold probability now decreases gradually from 82.3% at $u = 0.45$ to 22.0% at $u = 0.55$ (Fig. 5.9, blue curve), reflecting the economic reality that issuers near rating boundaries exhibit similar, not dramatically different, transition behavior.

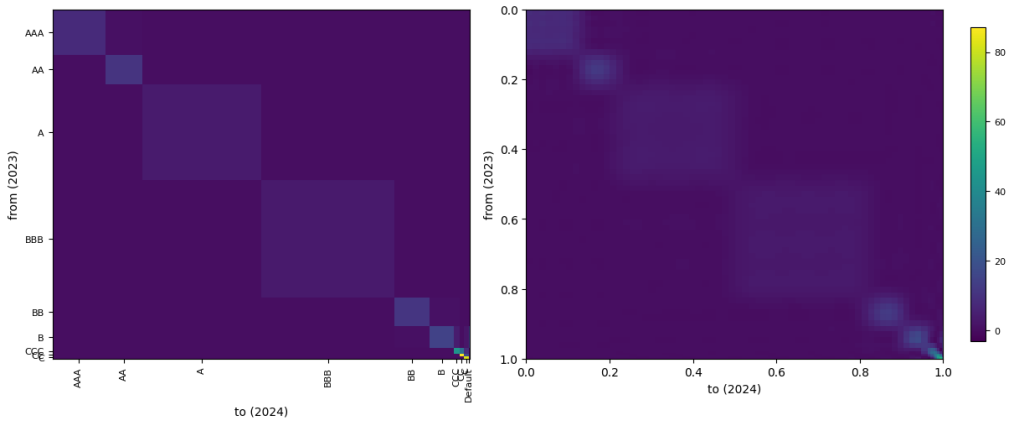


Figure 5.7.: Copula density comparison: checkerboard copula density (left, cellwise constant) vs. fitted smooth copula density on the Legendre subspace (right).

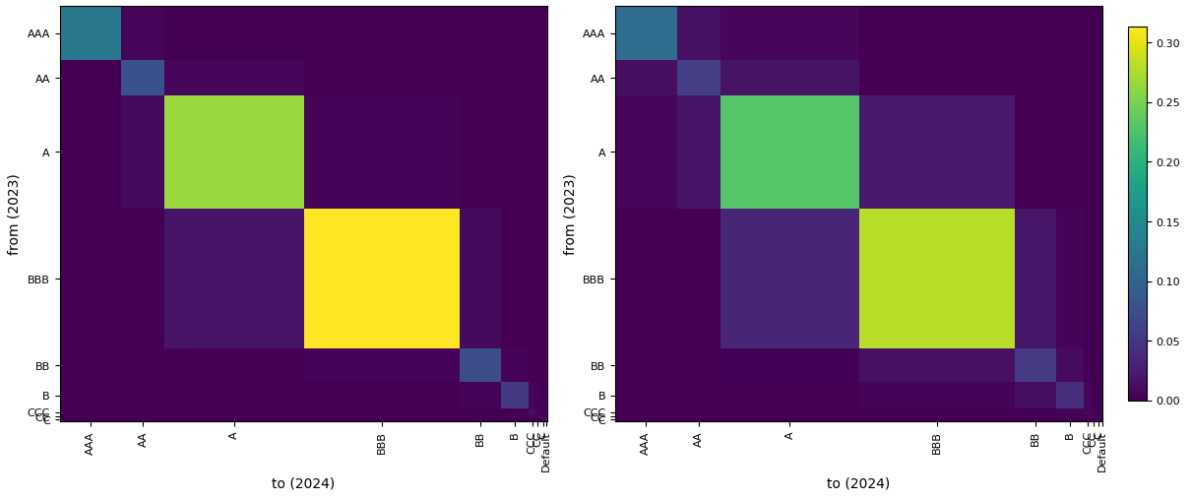


Figure 5.8.: Audit of mass preservation under the smooth copula: observed joint distribution (left) vs. reconstructed cell masses (right).

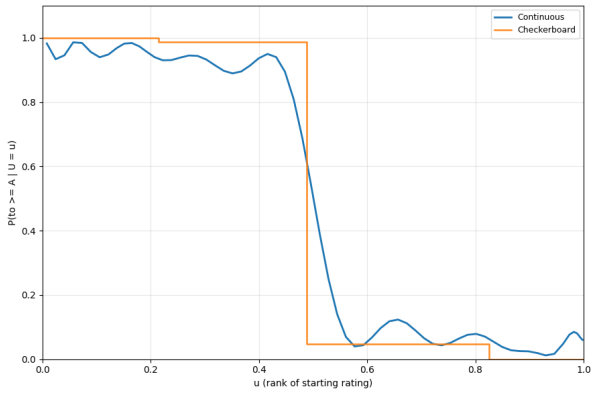


Figure 5.9.: Conditional threshold $\Pr\{\text{to} \geq \text{A} \mid U = u\}$: step-shaped under the checkerboard versus smooth under the fitted density.

5.4.2. Age-Income Dependence in the USA

The relationship between age and household income reflects lifecycle patterns of earnings, savings, and retirement (White, 2005). We analyze data from the 2023 American Community Survey (U.S. Census Bureau, 2024), which covers approximately 134 million U.S. households.

The data take the form of a contingency table with 16 rows corresponding to income brackets (ranging from “Less than $10,000” to “$200,000 or more”) and four columns corresponding to age groups based on the householder’s age (under 25, 25–44, 45–64, and

65 and over); see Appendix C.5 for the complete table. The entries represent household counts, which we normalize to obtain a joint probability distribution. This 16×4 table reveals characteristic lifecycle patterns: younger households (under 25) concentrate in lower income brackets with 76.0% earning below \$50,000, middle-aged households (45–64) show peak earnings with the highest proportion in upper income brackets, while the 65+ group shows increased concentration in lower incomes reflecting retirement, though with a substantial minority maintaining higher incomes.

We transform this joint distribution into a checkerboard copula by mapping each income bracket and each age group to quantile intervals on $[0, 1]$ based on their respective marginal frequencies. The resulting checkerboard copula density (Fig. 5.10, left panel) has non-uniform cell widths reflecting the unequal marginal probabilities. We then solve Problem 5.2.2 with Legendre basis ($p = q = 10$) and $N_g = 40 \times 40$ collocation points to obtain a smooth copula density (Fig. 5.10, right panel). Fig. 5.11 compares the original joint distribution with the reconstructed cell masses obtained by integrating the smooth copula over the original rectangles. The close agreement confirms that our smoothing procedure preserves the empirical mass distribution while removing artificial discontinuities.

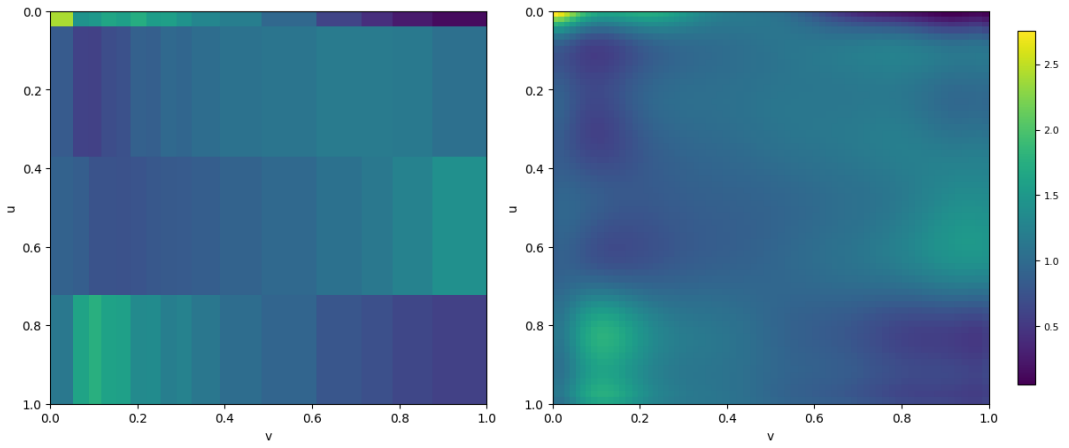


Figure 5.10.: Copula representations of age-income dependence. Left: checkerboard copula with non-uniform cell widths from empirical margins. Right: smooth copula density obtained via Legendre basis optimization. The horizontal axis corresponds to income (16 brackets), the vertical axis to age (4 groups).

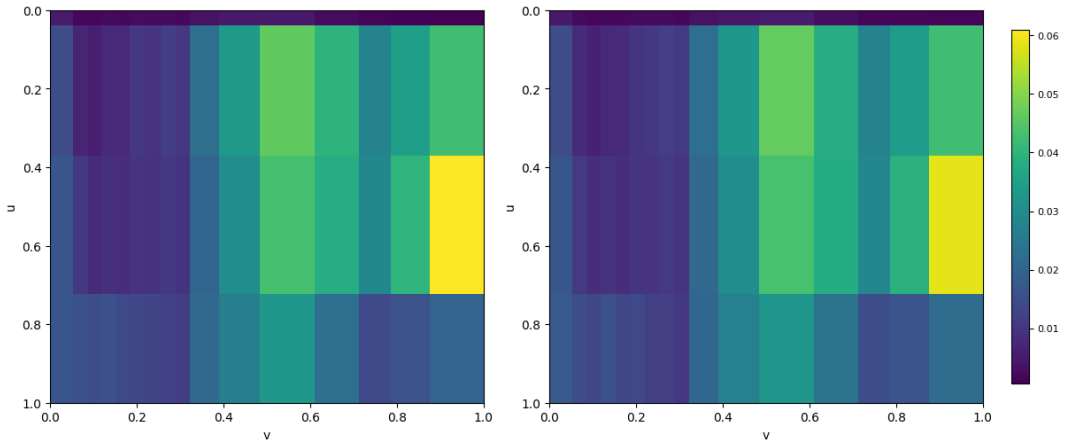


Figure 5.11.: Mass preservation audit. Left: original joint frequency distribution on the copula scale. Right: reconstructed cell masses from integrating the smooth copula density. The horizontal axis corresponds to income, the vertical axis to age.

Beyond removing discontinuities, smooth copulas can potentially recover fine-scale structure from aggregated data. To investigate this, we design a reconstruction experiment using the age-income data.

The original contingency table has 16 income brackets and 4 age groups, yielding a 16×4 table of cell counts. We refer to this as the fine table. To simulate a situation where only aggregated data is available, we construct a coarse table by merging every four adjacent income brackets into one, reducing the income dimension from 16 to 4. The result is a 4×4 table. Importantly, the coarse table contains strictly less information than the fine table: within each coarse income block, we no longer know how the counts are distributed across the original four fine brackets.

Given only the coarse table, can we reconstruct the fine table? That is, can we estimate the cell counts of the original 16×4 table using only the aggregated 4×4 table?

We fit two copula models to the coarse table. The first is a checkerboard copula, which assigns constant density within each of the 16 coarse cells. When we integrate this density over the fine grid defined by the original 16 income brackets, the mass within each coarse cell is distributed uniformly across the four fine cells it contains. The second is a smooth copula fitted using the Legendre basis. Because the density varies continuously, the mass within each coarse cell is distributed non-uniformly across the fine cells, guided by the global shape of the fitted density.

Table 5.3 compares the reconstructed fine tables to the true fine table using three error metrics. The smooth model reduces RMSE by 4%, MAE by 5%, and KL divergence by 3% compared to the checkerboard baseline. These improvements are achieved without

any additional information, demonstrating that the continuity assumption embedded in the smooth copula extracts information that the piecewise-constant checkerboard model cannot exploit. This finding is consistent with the fundamental rationale of smoothing techniques, which exploit local continuity to recover structure from aggregated or sparse data (Simonoff, 1996).

Model	RMSE	MAE	KL
Checkerboard	0.001894	0.001277	0.005720
Smooth	**0.001818**	**0.001217**	**0.005529**

Table 5.3.: Reconstruction accuracy for fine-scale prediction from coarse data (lower is better).

5.5. Conclusion

In this chapter, we present a method for transforming checkerboard copulas with cell-wise constant densities into smooth copula densities while preserving the empirical mass distribution. We formulate the embedding as an L^2 approximation problem with pointwise nonnegativity constraints, which yields a convex quadratic program that can be solved efficiently with standard optimization methods. By restricting the search to finite-dimensional subspaces spanned by orthonormal, mean-zero basis functions, we obtain a matrix optimization problem, which leads to a separable representation of the fitted density.

On the theoretical side, we establish existence and uniqueness of the optimal density and prove consistency of the collocation-based discretization. In our empirical study, we compare several orthonormal, mean-zero bases and highlight their respective strengths. Legendre bases capture smooth transitions particularly well in continuous copulas such as the Clayton, Gaussian, and Gumbel copulas. The Cosine basis, in contrast, is more efficient for sparse or singular structures such as permutation-based copulas. Increasing the basis dimension consistently improves mass-preservation accuracy across all bases.

In applications to credit rating transitions, our method removes economically implausible jumps at rating boundaries, replacing them with gradual changes that better align with market observations and rating agency practices. For age-income dependence in the American Community Survey, the smooth copula enables a more faithful reconstruction of fine-scale structure from aggregated data.

While the approach effectively smooths checkerboard copulas, computational complexity increases with grid size, and basis selection currently requires manual consideration of

the data structure. Future work will extend the framework to higher-dimensional checkerboard copulas. The method is modular and can incorporate additional criteria, such as entropy constraints for maximum entropy copulas or moment-matching conditions, either as objectives or as constraints.

6. Short-Term Grid Frequency Forecasting Using Gaussian Processes

While the previous chapters focused on copula models and copula density estimation, we now turn our attention to a concrete application in the energy sector: the modeling of the electrical grid frequency. This frequency serves as the central stability indicator, reflecting the real-time balance between power generation and consumption. However, accurately forecasting it is becoming increasingly challenging due to the volatility introduced by renewable energy sources.

This chapter is based on joint work with Maximilian Coblenz and Oliver Grothe (Publ. IV). In this chapter, we present a fully data-driven approach that combines Gaussian processes with sequence models for short-term forecasting of electrical grid frequency deviations using external techno-economic features.

6.1. Introduction

In our modern society, a stable electrical power supply is crucial for our daily lives and ensures economic activities. Power supply is linked to grid stability for which a key indicator is the grid frequency. If there is an imbalance between power generation and power consumption, the grid frequency deviates from the reference frequency. Therefore, an accurate model of the grid frequency is extremely important for simulations and predictions related to grid stability. However, the increasing share of renewables in the energy supply makes grid frequency modeling even more challenging due to the volatility and unpredictability of renewable energy (see Ourahou et al., 2020, for a review).

In the literature, an oscillator model or equation of motion motivated by the physical nature of the grid is generally assumed when considering the grid frequency (e.g., Filatrella et al., 2008; Wood et al., 2014; Schäfer et al., 2017). In Kraljic (2023), the stochastic Ornstein-Uhlenbeck process is modified and a fractal noise statistic is proposed to realistically model the grid frequency in Great Britain, taking into account statistical properties

such as fat tails and bimodality. In addition to random components, e.g., fluctuations, external influences such as technical and economic conditions must be modeled realistically (cf. Kruse et al., 2021b). In Gorjão et al. (2020), a dynamic model is formulated whose parameters take into account the influences of the fundamental control systems, the market and noise. In order to develop an accurate model for grid frequency dynamics, even more features need to be taken into account. However, it is a major challenge to incorporate the technical and economic features into the modeling and prediction of the short-term development of the frequency deviation at the level of seconds, as the features are usually recorded hourly. A recent study in Kruse et al. (2023) represents a special step towards solving this problem, in which a physically based machine learning model is presented whose physical model equations can take into account the influence of operating conditions in the form of techno-economic parameters on the short-term dynamics of the frequency control system.

Although physical models undoubtedly provide a solid basis for modeling the dynamics of grid frequency, the interesting question is whether a fully data-driven approach without detailed modeling of the physical principles can lead to comparable results. The aim of this chapter is to provide exactly this kind of data-driven model. In the following, the frequency behavior in Continental Europe is considered as an example to demonstrate the methodology. Note that the methodology presented is not only applicable to large power operation systems, but also to micro or distributed energy sources. For example, characteristics can be extracted from the data of a smart meter and used to model or predict the local frequency deviation. In this work, we want to explore the potential of data-driven modeling of an energy system without precise physical models, which are not readily available for every energy system in reality.

A popular approach for modeling stochastic processes is the Gaussian process, which has proven to be a valuable tool in many areas due to its flexibility and strong modeling capabilities. Also, Gaussian processes have become increasingly important in energy forecasting. For example, in Yang et al. (2018) the hourly probability density of the electricity load is predicted using quantile regression of the Gaussian process. In addition to Gaussian processes, sequence models such as long short-term memory (LSTM), gated recurrent units (GRUs), or transformers have proven to be particularly useful for processing sequential data. These techniques were originally developed to understand and process natural language and have achieved great success in many areas (see Islam et al., 2023; Yu et al., 2019). Furthermore, they have become increasingly important for the prediction of time series. In Kim and Cho (2019), a CNN-LSTM model is successfully used to extract complex energy consumption features and predict the energy consumption of residential buildings. In Zerveas et al. (2021), it is discussed how a transformer-encoder architecture can be used to learn multivariate time series representations.

This chapter presents a fully data-driven approach based on a combination of Gaussian processes and sequence models in order to model and predict the short-term evolution of frequency deviations, complementing models that take into account physical properties. For training, we use external feature values (e.g. day-ahead forecasts of load, generation, price and more, see Appendix D.3 for details) recorded at the beginning of an hour, in conjunction with the corresponding time series data of grid frequency that were recorded within the same hour.

The remainder of the chapter is structured as follows. Section 6.2 introduces the Gaussian process as a model for the frequency deviation and discusses solution approaches to account for correlations between time points. In Section 6.3, we show how the information in the techno-economic features can be extracted using the GRU and transformer architectures to directly predict the Gaussian process without physical modeling. In particular, we show the possibilities to consider the fat-tail behavior by changing the marginal distributions of the stochastic processes. Section 6.4 contains an overview of the data used and the models developed. In addition to the training details, we also present the baseline models and evaluation measures here. In Section 6.5, we present the evaluation results of our approaches to probabilistic forecasting and synthetic data generation in comparison to various baseline models. The chapter ends with a conclusion. The source code of this work is available at Liu et al. (2024b).

6.2. Grid Frequency Model based on Gaussian Process

In this section, we derive a Gaussian process as a framework for modeling the short-term evolution of grid frequency. We address the difficulty of modeling the correlation between time points and present solutions using covariance matrices with particular structures.

6.2.1. Model Setup

In the following, we denote the reference grid frequency by f_{ref}. For example, a reference frequency of 50 Hz is used in continental Europe. As the reference grid frequency remains constant over time, it is sufficient to model the deviation of the actual frequency from the reference, denoted as $\Delta f := f - f_{\text{ref}}$.

Let $(\Omega, \mathcal{A}, P)$ be a suitable probability space. We assume that the short-term dynamics of the grid frequency over a period of one hour, starting at time t_{start}, can be described by a Gaussian process $\Delta f : T \to L^2(\Omega)$ with time interval $T = [t_{\text{start}}, t_{\text{end}}]$. The Gaussian process Δf is uniquely defined by a mean function $\mu(t) := \mathbb{E}[\Delta f(t)]$ and a covariance function $\mathbf{C}(t, \tau) := \text{Cov}[\Delta f(t), \Delta f(\tau)]$, $t, \tau \in T$. L^2 is the space of square integrable functions

and guarantees that variance $\sigma^2(t)$ and covariance $\mathrm{Cov}(t,\tau), t \neq \tau$, are finite. In particular, the Gaussian process has the useful property that the random vector $(\Delta f(t_1), \dots, \Delta f(t_n))^T$ follows an n-dimensional Gaussian distribution for any choice of $t_1, \dots, t_n \in T$. A detailed discussion of Gaussian processes and their typical applications in machine learning can be found in Rasmussen and Williams (2006).

The flexibility and power of the Gaussian process, as evidenced by its successful application in various fields (Jain et al., 2018; Hensman et al., 2013; Diggle et al., 1998), make it a suitable candidate for modeling the complex dynamics of frequency deviation. Furthermore, since many physical models (such as the diffusion equation) ultimately lead to (multivariate) normal distributions under regularity conditions, it is justified in this sense to ask whether one can skip the intermediate step via (physical) model equations and, in our case, model the frequency deviation directly with a univariate Gaussian process. In particular, we do not use a specific physical model and associated stochastic differential equations, but aim to learn the Gaussian process directly from the available day-ahead data. This purely data-driven model offers great flexibility in modeling.

For the model, we now consider an index set with discrete time points $I = \{t_0, \dots, t_{N-1}\}$. For $N = 3600$, the discretized Gaussian process captures the grid frequency at a temporal resolution of one second. In this case, the Gaussian process can be represented by a multivariate Gaussian distribution with the N-dimensional mean vector $\mu \in \mathbb{R}^N$ and the covariance matrix $\mathbf{C} \in \mathbb{R}^{N \times N}$. In the following, we introduce approaches that identify the mean vector and the covariance matrix directly using advanced sequence models. Note that the discretization is not a strong constraint for a Gaussian process with independent time points, as the mean and variance functions can be reconstructed using interpolation techniques such as splines.

6.2.2. Treatment of Serial Dependency

The serial dependence structure of a Gaussian process is generally determined by its covariance function. For the discrete stochastic process Δf, we represent this dependence either directly by covariance matrices or by kernel models. Handling a large and full covariance matrix and estimating all its parameters can be very challenging and computationally intensive. To reduce complexity and improve identifiability, special matrix structures can be adopted. In particular, we assume a block structure of the covariance matrix (with a diagonal matrix as the simplest candidate) and design suitable kernel specifications. Both are discussed in more detail below.

As a structural simplification for the covariance matrix, a band structure or a diagonal block structure can be assumed. A band-structured covariance matrix for the Gaussian process $\Delta f : T \to L^2(\Omega)$ means that the frequency values are correlated only within a

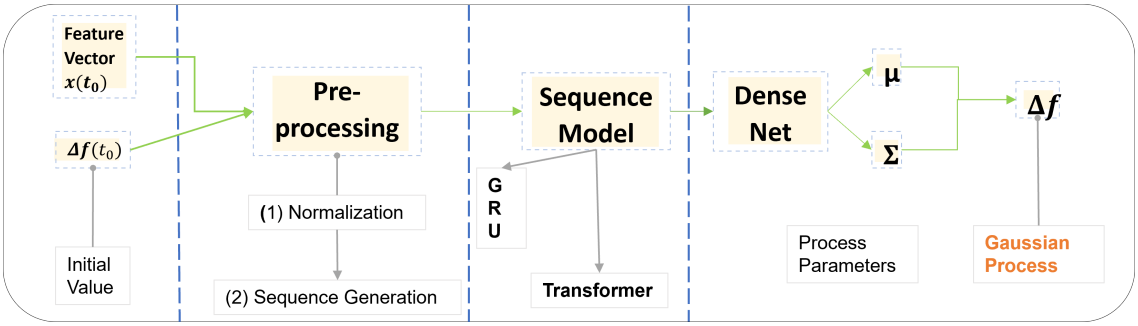


Figure 6.1.: Overview of the model structure. The techno-economic features and the initial frequency deviation are first pre-processed and then fed into sequence models (GRU or transformer) to extract temporal dependencies. The resulting representations are passed through a dense network to predict the mean vector and covariance matrix of the frequency deviations, modeled as a Gaussian process within the hour.

certain rolling time window, while the block structure of a covariance matrix implies that the time points can be divided into groups (represented by the blocks) within which significant relationships (covariances) exist, while no or only weak relationships exist between the groups. A simple but important special case of the band structure (bandwidth equal to 1) and the diagonal block matrix is the diagonal covariance matrix. In this case, the time points are independent of each other and only the variance has to be modeled for each time point. It could also be useful to divide the time points into four groups with time points of 15 minutes each. For example, market trading takes place at discrete intervals (such as 15 minutes). At the start of a new 15-minute interval, power generation is rapidly adjusted to meet the new trading conditions and demands. This leads to regular jumps in frequency dynamics (Kruse et al., 2020).

Another flexible yet powerful method for defining covariance functions for Gaussian processes is the use of special kernel functions that have a predefined functional form with a limited number of hyperparameters. When using predefined kernel matrices, only the hyperparameters need to be identified, which significantly reduces the complexity of determining the complete covariance matrix. A kernel frequently used in practice is the exponentiated quadratic kernel (also known as the radial basis function kernel):

$$k_{\mathrm{EQ}}(t, t') = \sigma^2 \exp\left(-\frac{(t - t')^2}{2\ell^2}\right), \tag{6.1}$$

where σ^2 denotes the variance (signal amplitude) and ℓ is the characteristic length scale. The rational quadratic kernel can model variations over multiple length scales with the

additional shape parameter $\alpha > 0$

$$k_{\mathrm{RQ}}(t, t') = \sigma^2 \left(1 + \frac{(t - t')^2}{2\alpha\ell^2}\right)^{-\alpha}. \tag{6.2}$$

A more detailed discussion of the properties of kernel functions can be found in Rasmussen and Williams (2006). Appendix D.1 shows different kernels and synthetic data for various hyperparameters. In particular, covariance matrices defined by kernel functions can have an approximate band structure (see Fig. D.1 in Appendix D.1). Note that one advantage of using standard kernels is that certain kernel combinations also produce valid covariance matrices. For example, the addition or matrix multiplication of two covariance matrices again results in a covariance matrix, so that specific synthetic data can be generated if kernels with different patterns are suitably combined.

In this chapter, we first assume that there is no correlation between the frequency deviations Δf at different points in time. In addition to the independence assumption, we also investigate to what extent the consideration of correlations by kernels can improve the results (see Section 6.5). Having discussed the basic model setup, we now turn to the next key step: identifying Gaussian process parameters using sequence models.

6.3. Process Learning Using Sequence Models

In this section, we first formally introduce the underlying learning task. From the requirements, we derive specific customized sequence models to efficiently and effectively process and extract the information from the techno-economic features, which are then used to predict distribution parameters of the Gaussian processes (cf. Fig. 6.1).

6.3.1. Learning Task and Loss functions

To complete our frequency model based on Gaussian processes, the process parameters (i.e. the mean vector μ and the covariance matrix $\mathbf{C}$) must be determined. As discussed in detail in Kruse et al. (2023) and Kruse et al. (2021b), the stochastic process of the grid frequency is influenced by external techno-economic properties. Therefore, we could design a method to predict the process parameters based on the external techno-economic features, see Fig. 6.1. Specifically for forecasting purposes, we focus below on constructing a model that processes the available day-ahead features. In particular, we use the following day-ahead features: day-ahead forecasts of the load, renewable generation, day-ahead electricity prices, the planned generation, their respective increases and information on the hour of the day (for details see Appendix D.3).

In Kruse et al. (2023), the frequency value at the beginning of a period is used to initialize the learned stochastic processes and is not listed directly as a feature. Here we also use the frequency deviation at the start time as a feature value directly. For a time interval, we can synthesize a feature vector by combining the hourly resolved values of the external features described above and the initial value of the grid frequency at the beginning of this time interval. The learning task presented is a typical supervised learning task. For each input feature vector representing the techno-economic state at the beginning of an hour, we use the data set of N grid frequency values within the corresponding hour as the true values for training and testing.

Since we are modeling the grid frequency with Gaussian processes, we now need a model that predicts the parameters of the Gaussian process from the features. A major challenge is that the values of the available features are typically published at a much lower temporal resolution, e.g. only at the beginning of an hour, resulting in an unbalanced size of the input and output dimensions. Therefore, models are needed that meaningfully transform the input data into higher dimensional data spaces, taking into account the communication possibilities between the values in the output sequences. This consideration leads to the use of a recurrent neural network structure, in particular a gated recurrent unit (GRU), which is an efficient and effective method for modeling time series data (see Section 6.3.2 for implementation details). Another idea to solve the problem described above is to use a transformer-like structure based on the attention mechanism. By using multi-head attention, different aspects of the feature vector can be learned. After information processing by the GRU or transformer, the learned information about the relationships is further processed by fully connected layers to compute the process parameters. A simple model structure such as a dense neural network with fully connected layers will not be suitable since fully connected layers do not directly take into account the latent structure of the data at either the input or the output, so learning the data representation in this way is inefficient.

Before presenting the adapted sequence models for predicting process parameters, we first introduce the following loss functions. Denoting the function learned by the neural network as φ, we define the negative log-likelihood loss function for X, a batch of input feature vectors $\mathbf{x}_k$, as follows

$$\mathcal{L}(X) = \frac{1}{K} \sum_{k=1}^{K} -\log(p_{\varphi(\mathbf{x}_k)}(\mathbf{y}_k)),$$

where $p_{\varphi(\mathbf{x}_k)}(\cdot)$ is the density function of the multivariate normal distribution with parameters encoded in the form of $\varphi(\mathbf{x}_k)$ and $\mathbf{y}_k \in \mathbb{R}^N$ is the time series of frequency associated with the feature vector $\mathbf{x_k}$.

Assuming that the frequency is uncorrelated between different points in time, the loss function simplifies to

$$\mathcal{L}(X) = \frac{1}{K}\frac{1}{N}\sum_{k=1}^{K}\sum_{n=0}^{N-1}\Big(\frac{1}{2}\log(2\pi) + \log(\sigma_n(\mathbf{x}_k)) + \frac{1}{2\sigma_n(\mathbf{x}_k)^2}(y_{k,n} - \mu_n(\mathbf{x}_k))^2\Big),$$

where $\mu_n(\mathbf{x}_k)$ and $\sigma_n(\mathbf{x}_k)$ denote the predicted mean and standard deviation at time point t_n given the feature vector $\mathbf{x}_k$, and $y_{k,n}$ is the observed frequency deviation at time t_n.

If the correlation of the frequency is modeled by a kernel matrix $\mathbf{K}$, then the loss function is calculated as

$$\mathcal{L}(X) = \frac{1}{K}\sum_{k=1}^{K}\big(\frac{N}{2}\log(2\pi) + \frac{1}{2}\log(|\mathbf{K}(\mathbf{x}_k)|) + \frac{1}{2}(\mathbf{y}_k - \mu(\mathbf{x}_k))^T(\mathbf{K}(\mathbf{x}_k))^{-1}(\mathbf{y}_k - \mu(\mathbf{x}_k))\big).$$

As shown in Kruse et al. (2023) and Kraljic (2023), large frequency deviations are more likely than a normal distribution would predict. Since in a Gaussian process the expected value function and covariance function do not need to be constant, the aggregate distribution of all time points of a Gaussian process can produce a different tail behavior than a normal distribution. However, one could ask whether a relaxation of the Gaussian limits to fat-tail distributions could lead to an even more realistic representation of the tail behavior. To this end, we also consider the Student's t-distribution and the Cauchy distribution, which is a special case of a Student's t distribution, for each time point. The dynamics of the frequency deviation is modeled by a stochastic process where we have a fat-tail distribution at each time point. In this case, we also assume that the time points are independent. In particular, we will learn a location-scale Student's t distribution to account for different mean positions and scattering behavior at each time point. For a stochastic process with marginal Cauchy distributions, we focus on learning the median and interquartile range. Details on the loss functions for fat-tail marginal distributions are provided in Appendix D.2.

Here the flexibility of our approach of using sequence models (see Section 6.3.2 for details) for feature processing becomes apparent. In order to take different stochastic processes into account, we only need to exchange the loss function and the rest of the model structure remains the same.

6.3.2. Information Extraction using Sequence-to-Sequence Models

In this section, we present the GRU and transformer-like neural network structure specially adapted for modeling the short-term dynamics of grid frequency. To solve the problem that the values of the available features are recorded hourly, but the frequency values are recorded every second, we use custom GRU and transformer structures. Whereas in GRU the techno-economical feature vector is artificially repeated for a number of points in time and then processed by an "autoregressive" type of network structure, the transformer-like structure attempts to learn different aspects from a static feature vector by multi-head attention.

GRU Structure

We assume that the macro-techno-economic state remains constant for the entire hour. The discrete time points within this hour are given by the index set $I = \{t_0, \dots, t_{N-1}\}$. In addition, we assume a hidden state h_n for each time point t_n, $n = 0, \dots, N-1$, which determines the (distribution of the) frequency deviation, e.g., the mean and the variance at that time point. Following an autoregressive modeling approach, one would calculate the hidden state h_{n+1} for time t_{n+1} as a function of h_n and the global techno-economic state $\mathbf{x}$. This modeling principle is implemented below using a GRU structure Cho et al. (2014). For this purpose, we consider the following process in Fig. 6.2. For each time step, we assume the same techno-economic feature vector as input. This can be achieved by repeating the input feature vector $\mathbf{x}$ for all N prediction time points.

To make a prediction for the time point t_n, a preliminary hidden state h'_n is first estimated. To do this, the feature vector $\mathbf{x}$ is processed with fully connected layers to calculate a value $r_n \in [0, 1]$. r_n represents the proportion of information from h_{n-1} that propagates from time t_{n-1} to t_n. If r_n is close to 0, it means that the sub-neural network has reset the information from the previous state. It is therefore also referred to as a reset gate. A value of $r_n = 1$ means that the influence of the past is strongly taken into account, which can be interpreted as a complete propagation of the previous state component.

The preliminary hidden state h'_n can then be calculated from $r_n \cdot h_{n-1}$ and $\mathbf{x}$. To get a final estimate for the hidden state, the preliminary hidden state h'_n can be weighted by the previous hidden state h_{n-1} with

$$h_n = z_n \cdot h'_n + (1 - z_n) \cdot h_{n-1}.$$

z_n is calculated similarly to r_n. The reset gate and the update gate, which are controlled by different fully connected layers, are trained to dynamically trade off remembering previous information and recognizing new information Chung et al. (2014) and Cho et al. (2014).

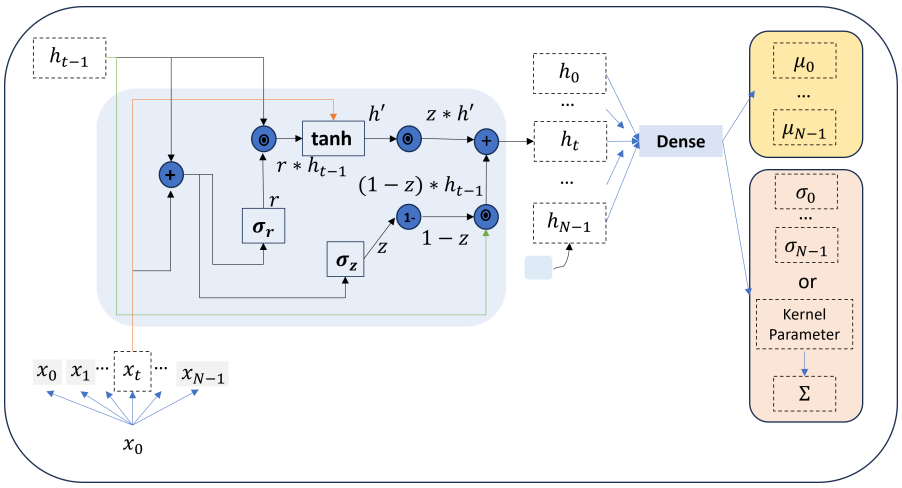


Figure 6.2.: Information processing using the GRU to predict frequency deviations. At each time point, the feature vector is combined with the previous hidden state to compute the current hidden state. The hidden states are then passed through a dense network to predict the mean vector and covariance matrix. For notational simplicity, the figure uses t instead of t_n to denote time steps.

After obtaining the hidden states h_n, $n = 0, \dots, N-1$, for the frequency dynamics for all time points in the period considered, we can use this learned information to compute, for example, the means and variances of $\Delta f(t_n)$, $n = 0, \dots, N-1$, by inserting a dense network between the outputs of the GRU and the outputs of the entire model. Depending on whether we take the correlation into account or not, we can then use different loss functions (cf. Section 6.3.1) to fit the model using a stochastic optimisation procedure such as ADAM.

Transformer Structure

In contrast to the GRU, we do not artificially duplicate the static feature vector to create pseudo-time series as inputs, but instead aim to generate different aspects of the data from the static feature vector that encode the dependencies of the individual feature values, which can then be used directly to predict the distribution parameters of the Gaussian processes Δf.

In particular, from a static feature vector of length L, we generate a sequence of L embedding vectors, which are then processed by multi-head attention as presented in Vaswani et al. (2017).

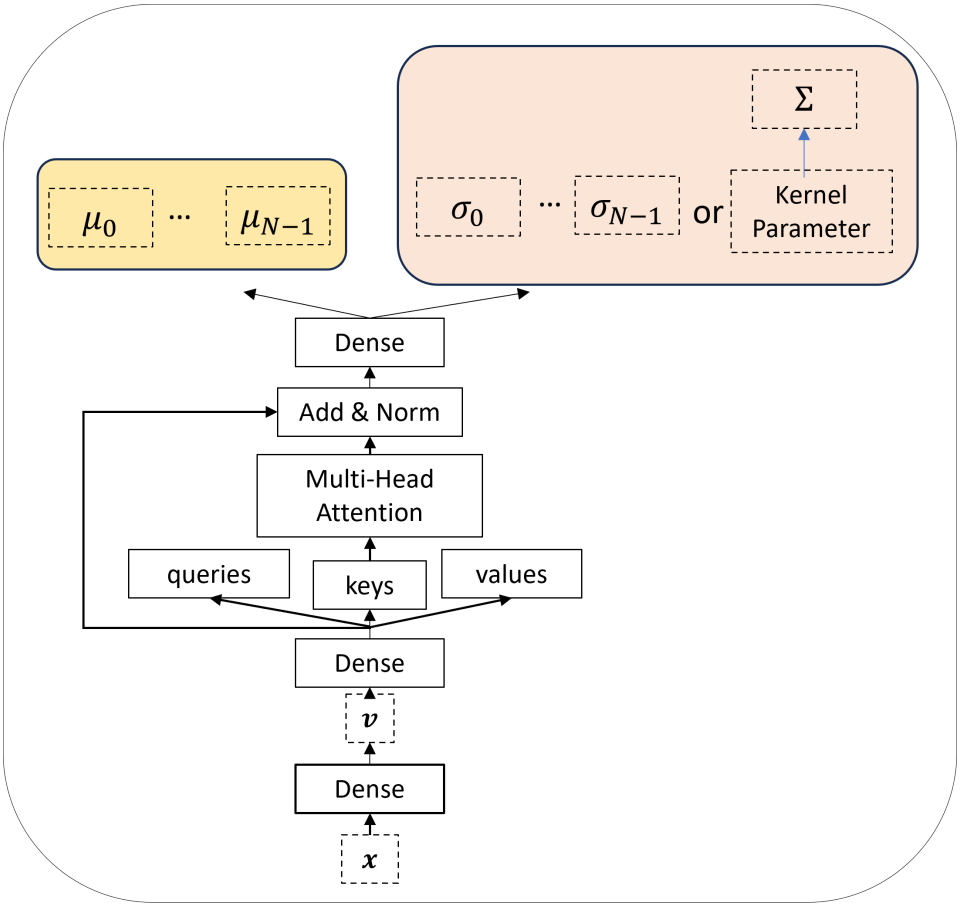


Figure 6.3.: Extraction of feature relationships using a transformer encoder structure. The static feature vector is used to generate embedding vectors. Multi-head attention is then applied to learn different aspects of the feature relationships. The resulting representations are passed through a dense network to predict the parameters of the Gaussian process. Parts of this illustration are based on Vaswani et al. (2017).

Since all features in the feature vector are captured at the same time, we do not use positional encoding, unlike in typical transformer setups. The embedding vector is transformed by three different dense nets to obtain three different representations: queries, keys, and values. From queries and keys, weights are computed via scalar products, which weight values and give an aggregated vector of value elements. The scaled scalar products are called attention scores, and the attention computation unit is called a single head. To look at different aspects of our techno-economic features simultaneously and thus learn complex patterns and relationships more effectively, multiple heads can then be used simultaneously. The results of all the heads are combined by concatenation and reprojection to produce the final output of the multi-head attention layer. As in the original trans-

former (Vaswani et al., 2017), we also use residual connections and layer normalization to stabilize the training process. Fig. 6.3 shows the entire model structure.

This aggregated information about the relationships between feature values can then be used to build another dense network to predict, for example, the mean vector or covariance matrix parameters. As with the GRU structure, different loss functions from Section 6.3.1 can be used here, depending on the purpose.

6.4. Study Setup

In the following section, we present the setup of our study, in which different models based on sequence models are built and evaluated. First, we present the data used to train the models. Then we give an insight into the models created and the details of the training. Finally, we briefly present the different baseline models and the evaluation methodology.

6.4.1. Data Set Description, Model Input and Output

We use the same database as in Kruse et al. (2023) for reasons of comparability. For details on the data, see Appendix D.3. The complete dataset consists of 26,859 data points. Each data point belongs to a time interval and consists of a feature vector and a time series of the grid frequency. The feature vector consists of values of external techno-economic variables at the beginning of the hour, temporal information and the initial value of the grid frequency at the beginning of the hour. Specifically, we considered eleven day-ahead features containing price, generation and load information and their ramp values (for details see Table D.1). With two time features and one initial value of the grid frequency, we obtain then a 14-dimensional static feature vector.

While all trained models based on the Gaussian process use the above-mentioned 14-dimensional static feature vector as input, the output of the models is different. With independent Gaussian process models, the outputs of the models are the mean values and the standard deviations of the grid frequency deviation for each point in time (e.g. every second). To limit the computational complexity of training with covariance matrices, models for Gaussian processes with correlations are trained to compute predictions for every 15th second instead of every second of an hour. The outputs of the models are the mean of the grid frequency deviation at every 15th second and the kernel parameters of the kernel matrix of all predicted time points. To generate the training data for this, every 15th element in each frequency sequence is selected. The feature data remains the same.

6.4.2. Training Details

The network structures are implemented as described in Section 6.3.2. Table 6.1 provides an overview of the models developed and their assumptions. More details on the model structures can be found in the Appendix D.4.

We use the negative log-likelihood functions as the loss functions. To reduce the computational cost for GRU-based models, we assume that the time points can be divided into consecutive groups of time points and the dependency information of each group can be encoded by a separate latent state. Our preliminary experiments show that 180 latent states are sufficient to achieve good results. Therefore, for performance evaluation in GRU-based models, we implement 180 latent states h_n for 3600 time points, each of which encodes the dependency information of 20 time points. By subsequently applying a dense network of suitable dimension (e.g. 3600 if we can learn a mean and variance at each time point), we again obtain the outputs for each time point. For transformer-based models we always use one attention block with 4 heads.

We trained models considering Gaussian marginals with both the transformer and the GRU structure. The models with fat-tail marginal distributions were trained with the transformer structure. An implementation with the GRU is also possible. However, our preliminary experiments show that the transformer-based structures can be trained faster compared to the GRU structure.

All models were trained for a maximum of 100 epochs with a batch size of 128. The validation loss was monitored during each training. We used early stopping and learning rate reduction techniques to avoid overfitting and improve model performance. Training was stopped if it did not improve over 5 epochs. The learning rate was multiplied by a factor of 0.1 if no improvement was observed after 3 epochs. This helps e.g. to fine-tune the model by taking smaller steps when a learning plateau is reached. This reduction in the learning rate continues until a lower limit is reached and training is terminated by early stopping. Data from 2015 to 2018 was used for training and validation, while data from 2019 was used for evaluation.

6.4.3. Baseline Models and Evaluation Measures

To evaluate the performance of the models, we compare our models with other base models. We consider models that make probabilistic predictions as the Gaussian processes above do and also models that make point predictions. In addition to the day-ahead and ex-post models from Kruse et al. (2023), we also consider other data-driven models such as daily profiles and constant profiles. For the constant profile, a Gaussian distribution with the global mean and standard deviation of the frequency data from 2015 to 2018 is

Marginal Distribution	Serial Dependency	Network Architecture
Gaussian	independent	GRU
Gaussian	independent	transformer
Gaussian	exponentiated quadratic kernel	GRU
Gaussian	rational quadratic kernel	GRU
Gaussian	exponentiated quadratic kernel	transformer
Gaussian	rational quadratic kernel	transformer
Student's t	independent	transformer
Cauchy	independent	transformer

Table 6.1.: Overview of the trained models and their configurations.

assumed, following Kruse et al. (2023). For the baseline models for the point forecast, we use all mean estimators of the probabilistic models as baseline models. We also included simple point estimators, such as the stepwise constant profile, which assumes that the frequency deviations are equal to the frequency deviation at the beginning of the hour. In addition, a simple nearest neighbor model was developed that calculates a weighted sum of the frequency sequences of the nearest feature vectors in the historical data. Table D.4 in Appendix D.5 provides an overview and detailed information of all the comparative models considered.

Measures such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) are used to compare the point forecasts. We evaluate the probabilistic forecasts based on normal distributions with independent time points using negative log-likelihood and Continuous Ranked Probability Score (Hersbach, 2000). Histograms of the realized quantiles are also created. In addition, we use energy scores (Gneiting and Raftery, 2007) to compare the predictions of our model with kernels. Details of the measures used can be found in the Appendix D.6.

6.5. Results

In this section, we evaluate the results of the various models presented. First, we compare these models, which are based on the assumption of independent time points, with different baseline models. We then illustrate the performance of the Gaussian sequence models using kernels. We consider both the performance of the point prediction and that of the probability prediction. Finally, we examine the properties of synthetically generated data and compare them with the properties of the real frequency deviation data. In particular, we investigate whether replacing the marginal distributions with fat tails leads to better

Model	MAE	MSE	RMSE
Constant zero	0.1255	0.0262	0.1543
Global mean	0.1233	0.0254	0.1524
Initial value	0.1772	0.0499	0.2199
Daily profile (mean)	0.0939	0.0145	0.1190
KNN profile	0.0847	0.0116	0.1072
PIML day-ahead (mean)	0.0882	0.0126	0.1117
PIML ex-post (mean)	0.0881	0.0125	0.1110
Independent Gaussian, GRU (mean)	**0.0814**	**0.0106**	**0.1027**
Independent Gaussian, transformer (mean)	0.0821	0.0109	0.1038
Cauchy, transformer (median)	0.0828	0.0111	0.1048
Student's t, transformer (mean)	0.0819	0.0108	0.1036

Table 6.2.: Evaluation results for point predictions.

results. To present the results in a standardized way, we always compare the angular frequency deviation $\omega = 2\pi \cdot \Delta f$.

6.5.1. Prediction Performance

We start with the evaluation of the point forecasts using the measures MAE, MSE and RMSE. Since the models in Kruse et al. (2023) were evaluated with a 15-minute time interval to achieve their best performance, we compare the measures for a 15-minute time interval here. As Table 6.2 shows, the mean predictions of all models based on sequence modeling outperform other data-based models and the two PIML models. In particular, the GRU-based Gaussian process model achieves the best result for all measures.

We calculate the negative log-likelihood loss and the continuous ranked probability score to evaluate the performance of the probabilistic forecasts. Again, both our GRU-based and transformer-based models are slightly better than day-ahead and ex-post models in Kruse et al. (2023) and clearly outperform the daily profile and the constant profile (see Table 6.3). In the Appendix D.8, we also provide the evaluation results of the measures on one-hour intervals and histograms of the realized quantiles.

To compare specific prediction examples, we choose the same time windows as in Kruse et al. (2023) (see Fig. 6.4). Since the time points are independent of each other, we can use the standard deviation function to draw the enveloping lines around the mean value function. Our results when using day-ahead features are comparable to those of Kruse et al. (2023), which use day-ahead and ex-post features, for both the good cases (see Fig. 6.4 (a), (c)) and the bad ones (see Fig. 6.4 (b), (d)). This means, first, that our data-driven methods

Model	Neg. Log-Likelihood	CRPS
Constant profile	−531.89	0.0882
Daily profile	−765.53	0.0663
PIML day-ahead	−824.78	0.0625
PIML ex-post	−825.23	0.0623
Independent Gaussian, GRU	−871.49	**0.0575**
Independent Gaussian, transformer	**−872.95**	0.0581

Table 6.3.: Evaluation results for probabilistic predictions using the median of negative log-likelihood and CRPS.

Model	Neg. Log-Likelihood	Energy Score
Independent Gaussian	−236.87	2.05
Gaussian, GRU, rational quadratic kernel	**−352.21**	**1.91**
Gaussian, transformer, rational quadratic kernel	−347.79	1.96
Gaussian, GRU, exponentiated quadratic kernel	−318.33	2.07
Gaussian, transformer, exponentiated quadratic kernel	−312.83	2.15

Table 6.4.: Comparison of models with independent time points versus kernel-based covariance structures, evaluated using negative log-likelihood and energy score.

can learn the complex nature of stochastic differential equations under the assumption of independent Gaussian processes, and second, that they confirm indirectly the correctness of the choice of model equations in Kruse et al. (2023).

Taking into account the correlations between time points, the exponentiated quadratic and rational quadratic kernel models outperform independent Gaussian processes in terms of negative log-likelihood (see Table 6.4). Again, the GRU-based models are slightly better than the transformer-based models. For example, the energy scores of the exponentiated quadratic kernel models are worse than the energy scores of the independent Gaussian process models. However, in terms of energy scores, the results with rational quadratic kernels are still the best. In Appendix D.7, we provide conditional predictions using the correlations learned for the bad cases above that could not be predicted well under the assumption of independent time points.

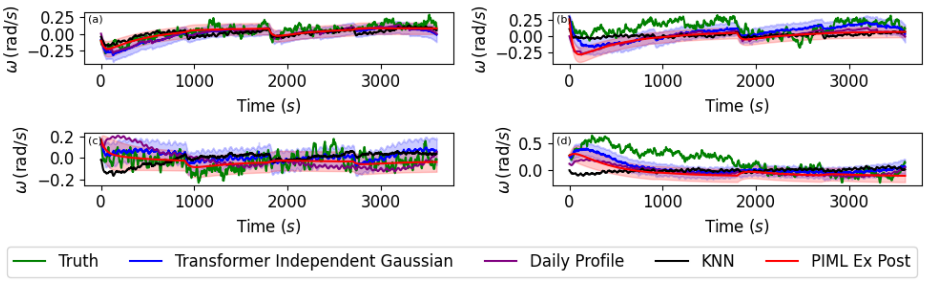


Figure 6.4.: Examples of good (a, c) and poor (b, d) predictions. The independent Gaussian model with transformer architecture shows similar performance to the PIML ex-post model. Both models outperform the daily profile and the KNN model in favorable scenarios and perform similarly to the daily profile in challenging scenarios.

6.5.2. Synthetic Data Generation

Our models can generate realistic synthetic data from techno-economic features, e.g., for optimization or simulation purposes. For a given techno-economic vector, our models first learn the distribution parameters (depending on whether correlations are taken into account or not). Synthetic time series can then be generated from the multivariate Gaussian distribution. To illustrate this, we generated frequency data for the period January 2019 with external features and checked whether the synthetic data correspond to the typical features of frequency data. In particular, we compare the estimated probability density functions (PDFs) of the frequency deviation of the real and synthetic data. The tail behavior of frequency fluctuations is well reproduced by the synthetic time series of an independent Gaussian process with transformer structure and outperforms a simple estimate of a normal distribution. The Cauchy marginal distribution overestimates the presence of fat tails, while the Student's t marginal distribution also reproduces the tail behavior very accurately due to its flexibility (see Fig. 6.5). Fig. 6.6 shows the autocorrelation functions (ACF) of the frequency deviation and the frequency increments $\Delta\omega(t) = \omega(t + 1s) - \omega(t)$, which were calculated from the synthetic data of a Gaussian process model with transformer structure and rational quadratic kernel. An exact mapping of the autocorrelation of increments is generally a difficult task. For example, Kruse et al. (2023) are not able to capture this aspect well, since an analytical solution of the Fokker-Planck equation requires the assumption of uncorrelated fluctuations. Here, the ACF of the frequency and the frequency increments are well represented by the synthetic data. For more details on the autocorrelations of other models with different kernels, see Appendix D.8.3.

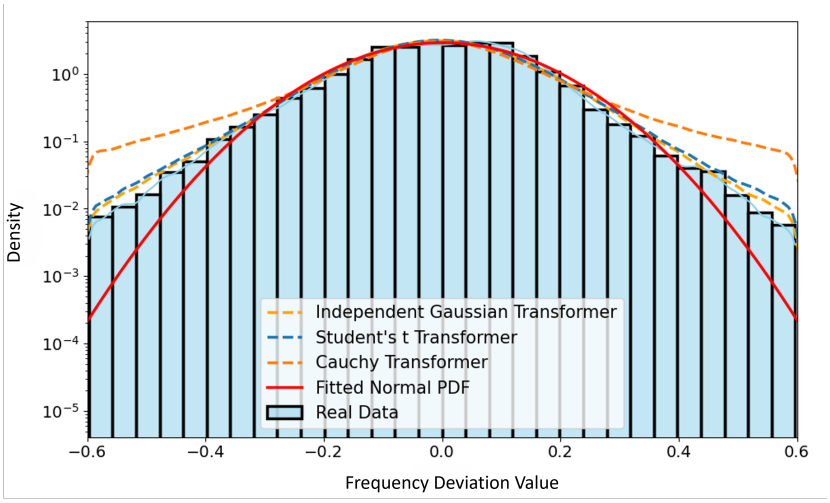


Figure 6.5.: Comparison of estimated probability density functions for real data and synthetic data generated using models with Gaussian, Student's t, and Cauchy marginal distributions.

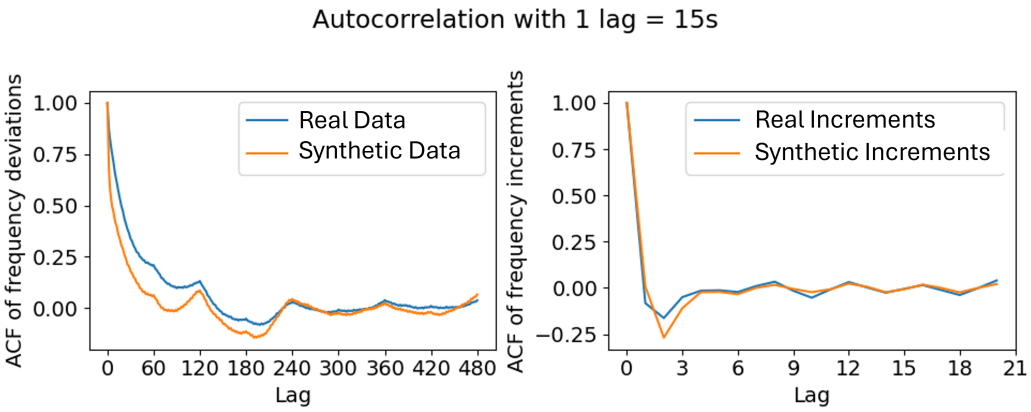


Figure 6.6.: Comparison of the autocorrelation function (ACF) for the frequency deviation and its increments between real data and synthetic data generated by a GRU-based Gaussian process model with a rational quadratic kernel. Good agreement is observed.

6.6. Conclusion

This chapter presents a fully data-driven approach to modeling and predicting the short-term evolution of frequency deviations. This fully data-driven approach is based on a combination of Gaussian processes and sequence models such as GRU and transformer.

Different models have been trained using various sequence models and kernels, and evaluated using multiple measures for both point predictions and probabilistic predictions, including negative log-likelihood, CRPS, and energy scores. Although we do not explicitly model the underlying physical properties, we achieve results comparable to physics-informed machine learning models and obtain slightly better performance across a range of evaluation measures. Our models outperform simpler data-driven baselines. The GRU structure performs slightly better than the transformer-based process models, but the latter are faster and easier to train. The synthetic data generated by our models with Gaussian and Student's t marginal distributions exhibit typical stochastic properties of frequency data, such as fat-tailed behavior. The behavior of the autocorrelation functions of the frequency deviation and its increments is also well captured by synthetic data generated from the correlated Gaussian process model. Compared to simple data-driven methods such as the k-nearest neighbors algorithm, the probabilistic models presented here offer powerful simulation capabilities, enabling the generation of realistic data sets for various scenarios that can be used for testing and validation purposes.

7. Probabilistic Prediction of Grid Frequency Dynamics

This chapter is based on joint work with Maximilian Coblenz and Oliver Grothe (Publ. V). Building on the grid frequency forecasting approach developed in the previous chapter, we extend the existing point predictor to a probabilistic estimator. We analyze its historical forecast errors in feature-defined subspaces and model the regime-specific error distributions with nonparametric copulas to capture serial dependencies, yielding more reliable uncertainty estimates.

7.1. Introduction

Power grid frequency reflects the balance between power generation and consumption in a transmission network and characterizes the stability and security of a power grid. Deviations of the grid frequency from the nominal frequency reflect imbalances between power generation and consumption, which can lead to overloads, shutdowns, or, in the worst case, impair critical infrastructures (see Kirby, 2003, for a detailed discussion). In the course of integrating renewable energy sources, monitoring and controlling the electricity grid frequency has become a challenge due to the volatile nature of renewable energies (Saha et al., 2023; Prakash et al., 2022).

A prediction model of grid frequency can detect grid instability situations early, and the forecast information can be used to efficiently deploy control power to balance grid fluctuations in a timely manner and prevent bottlenecks and failures. A particularly interesting prediction task in this context is to forecast the dynamic development of the grid frequency based on information available at the beginning of an hourly interval (Kruse et al., 2020; Kruse et al., 2023). Various approaches exist in the literature. For example, Kruse et al. (2023) developed a physics-informed machine learning model that builds upon physically meaningful model equations which take into account the influence of operating conditions. In another approach, Liu et al. (2024a) presented a purely data-driven methodology, where the model architecture is constructed without domain knowledge and the model learns directly from the data.

Although the above approaches can effectively model many aspects, they are fundamentally based on the assumption of Gaussian processes, which either assume independent points in time within the hourly interval to be predicted, or can only account for linear dependencies through correlation.

In this chapter, we address this problem by transforming an existing point predictor of grid frequency based on historical data into a probabilistic estimator capable of recognizing and flexibly modeling serial dependencies in the power grid frequency dynamics. In particular, we analyze the forecast errors generated by the point estimator on historical data by considering different subspaces of features defined based on the observed errors to identify different error states. The subspace-based decomposition allows us to capture localized error patterns that may be missed in a global analysis. We model the error distribution within each subspace using nonparametric copula estimators. Our methodology is motivated by the approaches of Schefzik et al. (2013) and Grothe et al. (2023), and the underlying idea has been successfully applied to day-ahead electricity price forecasting (see Grothe et al., 2023). Here, we extend this approach by incorporating feature-based error correction within the identified subspaces. We demonstrate our methodology using historical data from the European continental grid and show that by accounting for dependence structures between the time points, more potent multivariate estimators can be constructed. Furthermore, we demonstrate that failing to consider serial dependencies significantly underestimates the uncertainty in power grid frequency forecasts.

The remainder of the chapter is structured as follows. Section 7.2 introduces the probabilistic estimator construction for power grid frequency through feature-based error correction and the associated sampling process. In Section 7.3, we conduct a study to implement the methodology for modeling the grid frequency dynamics in continental Europe and evaluate the performance using the energy score and the marginal average continuous ranked probability score. In particular, we demonstrate that the performance of our copula based probabilistic forecasting models is significantly better when predicting the average grid frequency deviation. The chapter ends with a conclusion. The source code of this work is available at Liu et al. (2025a).

7.2. Methodology

Let $\hat{\mathbf{f}} : \mathcal{X} \to \mathbb{R}^d$ be an existing point predictor for the evolution of the grid frequency over d time points within a one-hour interval, i.e., it outputs a deterministic prediction of the grid frequency trajectory based on input features associated with the respective one-hour interval. Since time is considered in discrete steps, the frequency evolution can be represented as a d-dimensional grid frequency vector. Such a point predictor can, e.g., be

derived from a probabilistic model by using the expected value function of the Gaussian distribution-based approaches, as employed in the models of Liu et al. (2024a) and Kruse et al. (2023).

We denote by $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{f}_i) \mid \mathbf{x}_i \in \mathcal{X}, \mathbf{f}_i \in \mathbb{R}^d, i = 1, \dots, N\}$ a dataset consisting of N sample points, where each $\mathbf{x}_i$ is a historical feature vector and $\mathbf{f}_i$ is the corresponding observed true frequency vector. For instance, the training data used to train the point predictor $\hat{\mathbf{f}}$ can be utilized as $\mathcal{D}$. For $i = 1, \dots, N$, let $\hat{\mathbf{f}}_i := \hat{\mathbf{f}}(\mathbf{x}_i) \in \mathbb{R}^d$ denote the prediction of the point predictor at $\mathbf{x}_i$, where $(\mathbf{x}_i, \mathbf{f}_i) \in \mathcal{D}$.

The methodology presented below aims to develop a prediction model based on the point estimator $\hat{\mathbf{f}}$ and the dataset $\mathcal{D}$. The model is designed to provide a probabilistic prediction of the grid frequency vector for a given hour, represented as a d-dimensional conditional distribution F given a feature vector $\mathbf{x}$.

For this purpose, the feature space $\mathcal{X}$ is partitioned into meaningful feature subspaces derived from the clustering results of the observed feature vectors in the historical dataset. This allows for a more fine-grained analysis of the point predictor's error behavior. The prediction errors observed in $\mathcal{D}$ are analyzed independently across these clusters, and we estimate a separate error distribution for each subspace. The point predictor is then corrected by the error distributions to obtain a probabilistic predictor that can also map non-linear serial dependencies between different points in time in a predicted hour. Our methodology is motivated by the work of Grothe et al. (2023) and extends the approach by feature-space-based prediction error considerations. In contrast to the approach in Grothe et al. (2023), where the error distribution is considered over the entire historical data set, here we attempt to consider the error distribution over different subspaces of the entire feature space to account for variation in error behavior as a function of features. In addition, we propose the use of the empirical Bernstein copula for modeling the dependence of different time points. The complete construction procedure is given in the Algorithm 3.

7.2.1. Feature Space Decomposition

The distribution of the prediction error of the point estimator depends on the features that represent the conditions for the prediction. To account for this feature dependency in the subsequent error correction, the dataset $\mathcal{D}$ is first clustered according to features to obtain different prediction states. It should be noted that the feature space for the construction of the probabilistic predictor can generally be broader or different from the space of features originally used to train the point estimator. This has the advantage that we are able to extend the point predictor to a probabilistic predictor in aspects that are interesting for certain applications.

Algorithm 3: Copula-based probabilistic grid frequency predictor construction

Input: Point predictor for grid frequency $\hat{\mathbf{f}}$, Dataset
$\mathcal{D} = \{(\mathbf{x}_i, \mathbf{f}_i), \mathbf{x}_i \in \mathcal{X}, \mathbf{f}_i \in \mathbb{R}^d, i = 1, \dots, N\}$

Output: Probabilistic grid frequency predictor $\mathbf{f}_p$ with distribution P

Compute the error statistics from forecasting error time series;

$\epsilon_i = \mathbf{f}_i - \hat{\mathbf{f}}(\mathbf{x}_i), i = 1, \cdots, N$;

Compute the mutual information between all error statistics and the feature space;

Compute the normalized vectors of the average feature importance by using the computed mutual information;

Use the average feature importance vector to perform a k-means method with the scaled distance function to cluster the feature vectors in $\mathcal{D}$ into the clusters $\{\mathcal{D}_1, \cdots, \mathcal{D}_K\}$ and decompose the feature space into K subspaces $\{\mathcal{X}_1, \cdots, \mathcal{X}_K\}$;

for $k \leftarrow 1$ **to** K **do**

  Compute the copula structure $C^{(k)}$ between the d time points using the empirical Bernstein copula. For the copula estimation, the forecast error vectors of the feature vectors in $\mathcal{D}_k$ are used;

  for $j \leftarrow 1$ **to** d **do**

    Compute the empirical marginal distribution of the frequency values (in the cluster $\mathcal{D}_k$) with the time point index j: $\hat{F}_j^{(k)}$;

  Build the probabilistic predictor P_k on the feature subspace $\mathcal{X}_k$ with $\hat{F}^{(k)}(f_1, \dots, f_d \mid \mathbf{x}) = C^{(k)}\left(\hat{F}_1^{(k)}(f_1 - \hat{f}_1(\mathbf{x})), \dots, \hat{F}_d^{(k)}(f_d - \hat{f}_d(\mathbf{x}))\right)$;

$P = \{P_k, k = 1, ..., K\}$;

return P;

Additionally, different predictors often use varying amounts of information, meaning they operate on different datasets. A point estimator may initially be trained on a limited feature space, utilizing only a restricted amount of information. Later, it is possible to correct or adjust this estimator by including additional information. By expanding the data set, the estimator can be recalibrated in this sense, which can then provide better predictions

Multiple features are often used to predict grid frequency. For example, the data-driven model in Liu et al. (2024a) uses 14 techno-economic features and the ex-post physics-informed machine learning model in Kruse et al. (2023) uses 51 features.

To identify different feature subspaces for which we want to determine separate prediction error distributions of the point predictors, we use a feature-weighted k-means clustering approach (Xing et al., 2002; Bilenko et al., 2004). This approach enhances clustering

accuracy by giving higher weights to relevant features while reducing the influence of irrelevant or noisy features (De Amorim, 2011).

To this end, we systematically analyze the relationship between individual features and prediction errors. For each observed feature point in $\mathcal{D}$, the point predictor is evaluated to obtain the corresponding predicted grid frequency vector, which is then compared to the observed (true) grid frequency vector within the same hourly interval to calculate the prediction error values. This sequence of prediction errors over the associated interval can be represented as a d-dimensional error vector

$$\epsilon_i = \mathbf{f}_i - \hat{\mathbf{f}}_i, \quad i = 1, \dots, N. \tag{7.1}$$

From a prediction error vector ϵ_i, we can extract various characteristic statistics such as mean, standard deviation, maximum, minimum, and autocorrelation values. To quantify the strength of association between features and these error statistics, we employ mutual information, which measures the amount of shared information between two random variables and is particularly well-suited for detecting nonlinear dependencies (see Cover and Thomas, 2006, for background). Higher mutual information between an error statistic and a feature indicates that the feature strongly influences this error statistic.

Mutual information can be empirically estimated from the feature and error statistics data (see, e.g., Kraskov et al., 2004; Ross, 2014). After calculating the mutual information between individual features and a specific error statistic, these mutual information values can be normalized so that their sum equals 1. We perform this procedure for all selected error statistics. For each error statistic, we obtain a mutual information vector, where each component represents the mutual information between the error statistic and a specific feature. To derive a summary feature importance vector, we compute the mean of all mutual information vectors across the different error statistics. We refer to this aggregated vector as the feature importance vector λ.

Here, the standard k-means distance function is then modified with λ to incorporate the varying relevance of features for prediction errors. Let p denote the number of features, i.e., the dimension of the feature space $\mathcal{X}$. The weighted distance is defined as

$$d_\lambda(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{j=1}^{p} \lambda_j (x_j - y_j)^2}, \quad \mathbf{x}, \mathbf{y} \in \mathcal{X}, \tag{7.2}$$

where p denotes the number of features (i.e., the dimension of $\mathbf{x} \in \mathcal{X}$).

It should be noted that features are already standardized prior to weighting to avoid scale effects.

To determine the optimal number of clusters, the silhouette coefficient can be optimized (for details, see Rousseeuw, 1987).

Assume that the historical feature vectors in the dataset $\mathcal{D}$ are partitioned into disjoint clusters $\mathcal{D}_k$, for $k = 1, \dots, K$, through the clustering process described above. Each cluster $\mathcal{D}_k$ induces a corresponding feature subspace $\mathcal{X}_k \subset \mathcal{X}$. A feature vector $\mathbf{x} \in \mathcal{X}$ is assigned to the feature subspace $\mathcal{X}_k$ if the distance between $\mathbf{x}$ and the center $\mathbf{z}_k$ of cluster $\mathcal{D}_k$ is minimal among all K clusters, i.e.,

$$k = \arg \min_{1 \leq j \leq K} d_\lambda(\mathbf{x}, \mathbf{z}_j). \tag{7.3}$$

7.2.2. Multivariate Frequency Distribution via Copula Model

Following the procedure described in Section 7.2.1, the clusters $\mathcal{D}_k$ and the feature subspaces $\mathcal{X}_k$, $k = 1, \dots, K$, are obtained.

For each $\mathcal{D}_k$, we consider the associated set of observed prediction error vectors, defined as

$$\mathcal{E}_k := \left\{ \epsilon_i \in \mathbb{R}^d \mid (\mathbf{x}_i, \mathbf{f}_i) \in \mathcal{D}_k \right\}, \quad k = 1, \dots, K. \tag{7.4}$$

For each of these subsets of prediction error vectors $\mathcal{E}_k$, we estimate a multivariate probability distribution P_k that captures the stochastic characteristics of the errors within the respective cluster. A key aspect of this process is modeling the dependency structure of the prediction error vector.

Since the prediction error vector represents the temporal progression of errors within a one-hour interval, the estimated dependency structure reflects how error values at different time points within that interval are interrelated. To flexibly capture these temporal dependencies and represent the hourly stochastic dynamics within each cluster, we employ the copula framework.

While a Gaussian distribution can only model linear dependencies through correlation, a copula model allows capturing complex, nonlinear dependency structures. In general, any multivariate distribution function F of a random vector $(Z_1, \dots, Z_d)$ can be decomposed into a dependence structure, represented by a copula C, and its marginal distributions F_j, as stated by Sklar's theorem (Sklar, 1959):

$$F(z_1, \dots, z_d) = C\left(F_1(z_1), \dots, F_d(z_d)\right). \tag{7.5}$$

If all marginal distributions $F_i, i = 1, \dots, d$ are continuous, the copula is unique.

To estimate a copula for the prediction error vector in the feature subspace $\mathcal{X}_k$, the residual data $\mathcal{E}_k$ can be used. Following the approach proposed by Genest and Favre (2007), we first compute the so-called pseudo-observations and then estimate a copula from these data. This can be done either using a parametric model, such as the Gaussian copula, if

one wishes to model a linear dependence between the frequencies at different time points. However, without knowing the underlying dependency structure, we have to estimate the frequency dependence with a nonparametric copula.

Let the elements of $\mathcal{E}_k$ be indexed as $\epsilon_i^{(k)} \in \mathbb{R}^d$, for $i = 1, \dots, n_k$. The pseudo-observations are defined as

$$\hat{U}_{ij}^{(k)} = \frac{R_{ij}^{(k)}}{n_k + 1}, \tag{7.6}$$

where $R_{ij}^{(k)}$ denotes the rank of the j-th component of $\epsilon_i^{(k)}$ among the values $\epsilon_{1j}^{(k)}, \epsilon_{2j}^{(k)}, \dots, \epsilon_{n_k j}^{(k)}$. In the following, we omit the index k for clarity in the representation of the copula formula and write n for the sample size n_k.

A simple non-parametric estimation method is to use the empirical copula

$$C_n(u_1, \dots, u_d) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\{\hat{U}_{i1} \leq u_1, \dots, \hat{U}_{id} \leq u_d\}. \tag{7.7}$$

In the literature, several approaches exist for nonparametric copula estimation. In this work, we employ the empirical Bernstein copula with a smoothing degree m, typically an integer between 1 and n. It is defined as

$$C_n^B(u_1, \dots, u_d) = \sum_{v_1=0}^{m} \dots \sum_{v_d=0}^{m} \prod_{j=1}^{d} \binom{m}{v_j} u_j^{v_j} (1 - u_j)^{m - v_j} C_n\left(\frac{v_1}{m}, \dots, \frac{v_d}{m}\right), \tag{7.8}$$

which constitutes a smoothed version of the empirical copula (Sancetta and Satchell, 2004; Segers et al., 2017). It generalizes families of polynomial copulas (Sancetta and Satchell, 2004) and has already been successfully applied in the literature for various modeling purposes (see, e.g., Sarmiento et al., 2018; Yamut and Hudaverdi, 2023; Diers et al., 2012).

A total of K copulas $\{C^{(k)},\ k = 1, \dots, K\}$ are estimated based on $\mathcal{E}_k, k = 1, \dots, K$. In addition to the temporal dependency structure $C^{(k)}$, the marginal distributions also need to be determined for each $\mathcal{E}_k$. This refers to the distributions of the prediction error from the point estimator at specific time points within the predicted hour interval. The empirical distribution function (denoted as $\hat{F}_i^{(k)}, i = 1, \dots, d$) and the empirical quantile function (denoted as $\hat{Q}_i^{(k)}, i = 1, \dots, d$) can be used for this purpose. Note that this also generalizes the Gaussian model, as it allows us to model skewness, kurtosis, and other properties of the marginal distributions.

The learned copula functions and the empirical marginal distributions of individual time points within an hourly interval can be combined into a probabilistic predictor of the

grid frequency. For a given feature vector $\mathbf{x} \in \mathcal{X}_k$ and a point predictor $\hat{\mathbf{f}}$, this *probabilistic predictor* is represented by the conditional multivariate distribution function:

$$\hat{F}^{(k)}(f_1, \dots, f_d \mid \mathbf{x}) = C^{(k)}\left(\hat{F}_1^{(k)}(f_1 - \hat{f}_1(\mathbf{x})), \dots, \hat{F}_d^{(k)}(f_d - \hat{f}_d(\mathbf{x}))\right), \tag{7.9}$$

where $\hat{f}_i(\mathbf{x})$ denotes the i-th component of the point prediction $\hat{\mathbf{f}}(\mathbf{x})$.

To generate a concrete prediction from the probabilistic predictor for a feature vector $\mathbf{x} \in \mathcal{X}_k$, i.e., to sample a data point from the conditional distribution $\hat{F}^{(k)}$, one first samples a point $(\hat{u}_1, \dots, \hat{u}_d)$ from the learned copula $C^{(k)}$. Then, for each dimension $i = 1, \dots, d$, the estimated quantile function $\hat{Q}_i^{(k)}$ is applied individually to obtain the residual term

$$\hat{\epsilon} := \left(\hat{Q}_1^{(k)}(\hat{u}_1), \dots, \hat{Q}_d^{(k)}(\hat{u}_d)\right). \tag{7.10}$$

By correcting the point prediction $\hat{\mathbf{f}}(\mathbf{x})$ with $\hat{\epsilon}$, the final prediction from the probabilistic predictor is given by

$$\hat{\mathbf{f}}_p(\mathbf{x}) = \hat{\mathbf{f}}(\mathbf{x}) + \hat{\epsilon}. \tag{7.11}$$

7.3. Results

In the following, we demonstrate how a trained point estimator can be transformed into a probabilistic estimator using the methodology described above using real grid frequency data of continental Europe. We show that the prediction performance with respect to the energy score is thereby improved. Furthermore, through a simulation study on the average frequency deviation in an hourly interval, we show that a Gaussian-based frequency prediction model that does not account for dependencies underestimates the uncertainty of the forecast. The developed copula-based model is better suited in this regard.

7.3.1. Study Setup

We use the same data and setup as in Liu et al. (2024a) and Kruse et al. (2023). In particular, we use the historical dataset $\mathcal{D}$ to construct the probabilistic predictors, which contains historical grid frequency data and external feature data for continental Europe from 2015 to 2018. This is the same data used to train the point estimators described below, which serve as baselines in our evaluation. The performance of the probabilistic predictors is then evaluated using data from 2019, which was completely excluded from the training process for both the probabilistic models and the baseline point estimators, and thus serves as a test set.

Type	Feature	Unit
External	Load Day-Ahead	MW
External	Solar Day-Ahead	MW
External	Offshore Wind Day-Ahead	MW
External	Onshore Wind Day-Ahead	MW
External	Load Ramp Day-Ahead	MW/h
External	Generation Ramp Day-Ahead	MW/h
External	Solar Ramp Day-Ahead	MW/h
External	Offshore Wind Ramp Day-Ahead	MW/h
External	Onshore Wind Ramp Day-Ahead	MW/h
External	Price Day-Ahead	EUR/MWh
External	Price Ramp Day-Ahead	EUR/MWh/h
Time	$\cos(\pi/12$ Hour$)$	-
Time	$\sin(\pi/12$ Hour$)$	-
Initial Value	Initial Grid Frequency Value	Hz

Table 7.1.: Included Features in Dataset $\mathcal{D}$ (see also Liu et al., 2024a; Kruse et al., 2023).

The external techno-economic features are recorded hourly, while the associated grid frequency data are available as time series at different timestamps within the corresponding hourly interval. In this work, we analyze minute-by-minute predictions of frequency values within a given hourly interval. The output of a point predictor is a 60-dimensional grid frequency vector, whereas the output of a probabilistic predictor is a 60-dimensional multivariate distribution. An overview of the techno-economic features can be found in Table 7.1. Details on the collection and preprocessing of the data can be found in Kruse et al. (2021b) and Kruse et al. (2021a).

We use selected predictors from Kruse et al. (2023) and Liu et al. (2024a) as the basis for the point estimation. Table 7.2 provides an overview of the different point predictors and their properties. While the KNN model from Liu et al. (2024a) directly provides a point prediction (i.e., a predicted frequency vector for a given hourly interval), the other models are probabilistic in nature. They are based on Gaussian distributions and model the frequencies at several discrete time points within an hourly interval either as independent or correlated. Given a set of input features, they predict a mean frequency vector along with either a vector of standard deviations (assuming independence between time points) or a full covariance matrix (in the case of correlated time points). To derive point predictions from these probabilistic models, we use the predicted mean vector as a deterministic estimate.

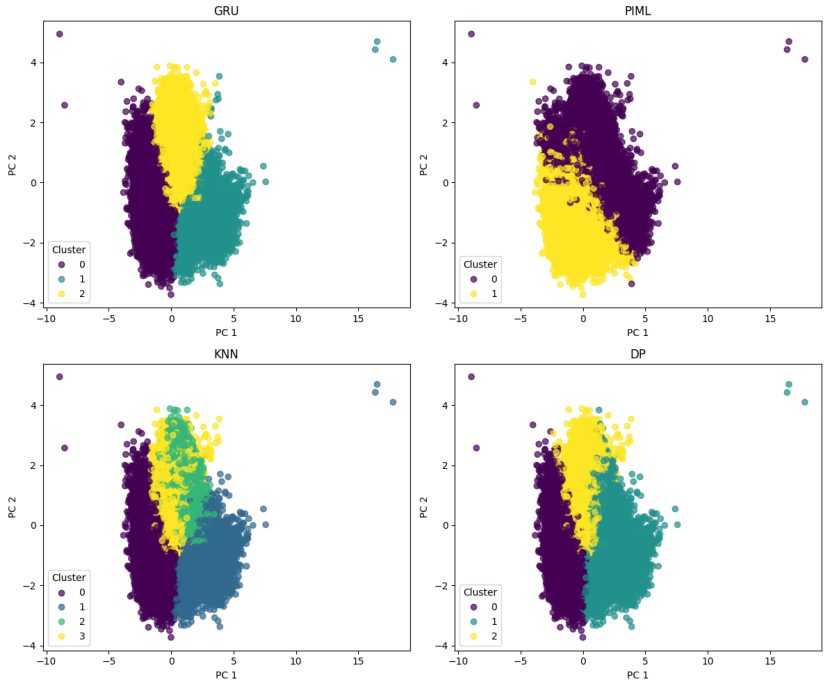
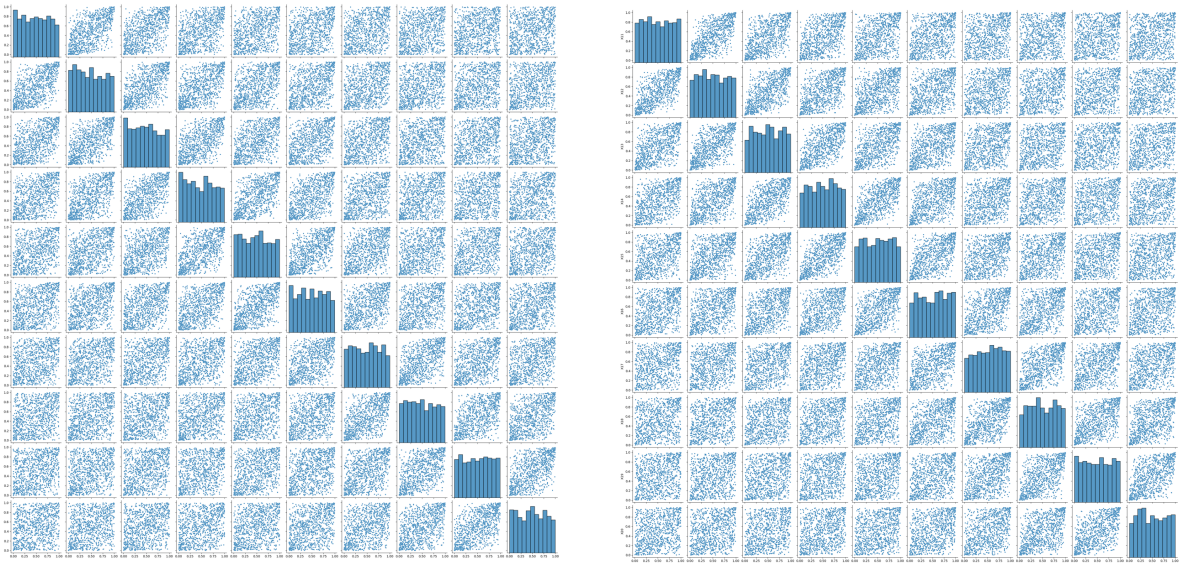


Figure 7.1.: Error based feature space clustering results of different point predictors. The 2D coordinates obtained from the PCA transformation of the clustered points are shown.

Point Predictor	Description	Features Used for Training
GRU Independent Gaussian	The mean frequency value prediction of the independent Gaussian process model based on GRU	Same as in $\mathcal{D}$
Day-Ahead PIML	The mean frequency value prediction of the day-ahead physics-informed machine learning model	Same as in $\mathcal{D}$
KNN	Prediction based on the feature distance of historical data	Same as in $\mathcal{D}$
Daily Profile	The mean of all training data at this time of day	Only temporal features (time of day)

Table 7.2.: Overview of point predictors and their feature usage.



(a) GRU-based Point Estimator **(b)** Daily Profile Point Estimator

Figure 7.2.: Pairwise plots of simulation data (1,000 points) generated from the respective learned empirical Bernstein copula of the error distribution within one of the respective clusters. The plots illustrate the pairwise dependencies between time points within a 10-minute interval, from the 10th to the 19th minute (10-dimensional).

7.3.2. Clustering and Copula Results

For each considered predictor, we perform feature-weighted k-means clustering. The corresponding feature importance weights are first determined and then used in the respective clustering process. To compute the error statistics within a one-hour interval, we analyze the time series of prediction errors. We consider several metrics: First, we calculate the mean and standard deviation of the errors to quantify the average error level and its variability. Second, we assess the autocorrelation of the error time series to capture the temporal dependency structure within the interval. Specifically, we focus on the average autocorrelation over the first 15 lags (equivalent to 15 minutes).

We then compute the mutual information between each dimension of the feature space and the error statistics on $\mathcal{D}$. As described in Section 7.2.1, the feature importance vector is subsequently used as the weights for the distance function. A clear dependency can be observed between temporal features and error behavior across all the models considered. Interestingly, while the GRU-based baseline and the physics-informed machine learning model exhibit similar performance (see Liu et al., 2024a), their prediction errors are based on different features, apart from the temporal ones (see Table 7.3).

Feature	GRU	PIML	KNN	DP
Solar Day-Ahead	0.0510	0.0597	**0.1283**	**0.0816**
Wind On Day-Ahead	0.0760	0.0375	0.0184	0.0654
Wind Off Day-Ahead	0.0686	**0.1332**	0.0601	0.0106
Prices Day-Ahead	0.0940	**0.1808**	0.0338	0.0436
Load Day-Ahead	0.0447	**0.1538**	0.0732	0.0546
Load Ramp Day-Ahead	**0.1080**	0.0000	0.0714	0.0562
Total Gen Ramp Day-Ahead	0.0770	0.0877	**0.1209**	0.0602
Wind Off Ramp Day-Ahead	0.0159	0.0352	0.0084	0.0141
Wind On Ramp Day-Ahead	0.0000	0.0765	0.0158	0.0000
Solar Ramp Day-Ahead	0.0167	0.0447	0.0710	0.0751
Price Ramp Day-Ahead	**0.1040**	0.0126	0.0887	0.0703
Hour (sin)	**0.1148**	0.0005	**0.1445**	**0.1838**
Hour (cos)	**0.1585**	**0.1069**	**0.1291**	**0.1217**
Initial Value	0.0707	0.0709	0.0362	**0.1627**

Table 7.3.: Average Mutual Information Between Error Statistics and Feature Space

Now, the feature weighted k-means algorithm can be performed using the calculated feature importance weights. The optimal number of clusters is determined by maximizing the silhouette coefficient. Figure 7.1 illustrates the clustering results concerning different error behaviors of the point predictors. To visualize the clustering results, we first performed a principal component analysis (PCA) on the feature space and then plotted the clustering results using the 2D coordinates obtained from the PCA transformation of the clustered points.

With these cluster results, we calculate the respective error distributions for each point predictor for all their clusters, as described in Section 7.2.2, using the empirical Bernstein copula. For practical estimation, we use the Python package OpenTURNS (Baudin et al., 2016). For each point predictor, a separate copula function is estimated for each cluster to model the temporal dependence between the deviations from the corresponding point predictions at the 60 time steps within a one-hour interval.

We illustrate some examples of the learned dependency structures. First, we sample 1000 60-dimensional vectors from a trained empirical Bernstein copula. The resulting pair plots visualize the dependency structure between two dimensions, i.e., between two time points within a one-hour interval. Due to space limitations, we only present the learned dependency structure for 10 of the 60 dimensions (see Figures 7.2a and 7.2b). Here, one can observe that the greater the distance between two time points in an hourly interval, the less dependence is present. Although many pairwise dependencies have a Gaussian

copula-like shape, there are many pairwise dependencies that are not similar to a Gaussian copula.

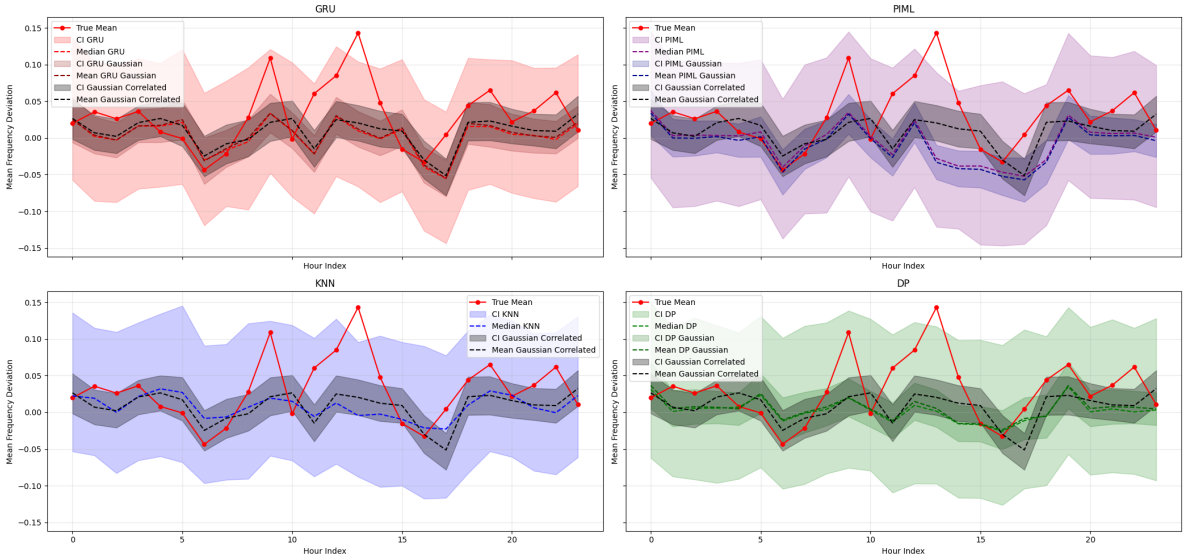


Figure 7.3.: Confidence intervals for the hourly average grid angular frequency deviation. For each hour, 1000 time series are sampled from the respective probabilistic model. From these time series, the mean frequency deviations are calculated and the confidence intervals are derived. The red line indicates the true progression of the hourly average frequency deviation in an example day.

7.3.3. Prediction Performance

We evaluate the obtained probabilistic predictors in comparison to the original probabilistic predictors, from which the point predictors are derived, using data from 2019 as the test set. The evaluation is based on energy scores and marginal CRPS scores (i.e., the average of the CRPS scores of the marginal predictors). Additionally, we consider a probabilistic GRU-based Gaussian process model with a rational quadratic kernel from Liu et al. (2024a), which accounts for the correlation between different time points in the considered hour interval.

We evaluate the scores on the test data using *scoringrules*, a Python library for assessing probabilistic forecasts (Zanetta and Allen, 2024), and present the median values in Table 7.4. It is evident that the copula-based probabilistic methods improve the energy score of their corresponding baseline models. Furthermore, the result of the copula-GRU model is also better than the considered correlated Gaussian process model. Additionally, Wilcoxon signed-rank tests (Woolson, 2005) were conducted to assess the statistical

Method	Energy Score	Marginal Avg. CRPS	*p*-value vs Gaussian Baseline (ES)	*p*-value vs Corr. Gaussian (ES)	*p*-value vs Gaussian Baseline (CRPS)	*p*-value vs Corr. Gaussian (CRPS)
Copula-based						
GRU	**0.4774**	**0.0497**	< 0.0001	< 0.0001	0.0252	< 0.0001
PIML	0.4936	0.0509	< 0.0001	< 0.0001	< 0.0001	< 0.0001
KNN	0.4868	0.0507	--	< 0.0001	--	< 0.0001
DP	0.5108	0.0532	< 0.0001	< 0.0001	< 0.0001	< 0.0001
Independent Gaussian						
GRU	0.4823	0.0499	--	< 0.0001	--	< 0.0001
PIML	0.5001	0.0515	--	< 0.0001	--	< 0.0001
DP	0.5940	0.0612	--	< 0.0001	--	< 0.0001
Correlated Gaussian						
Rational Quadratic	0.4827	0.0503	--	--	--	--

Table 7.4.: Median values of energy score and marginal average CRPS for copula-based, independent Gaussian, and correlated Gaussian models, evaluated on the full test dataset. *p*-values from the Wilcoxon test for comparisons to Gaussian baselines and the correlated Gaussian model are presented. -- indicates that no test was performed, either because it would be a comparison with itself or because no baseline is available (as in the case of KNN).

significance of differences between the score values. As shown in Table 7.4, the differences in model performance are statistically significant at the 0.01% level for nearly all comparisons (with the exception of the comparison of marginal CRPS values between the copula-based GRU model and the Gaussian GRU model).

7.3.4. Prediction of Hourly Average Grid Frequency Deviation

To highlight the importance of accurately accounting for dependencies across different time points, we consider the hourly average frequency deviation, which is defined as the average of the frequency deviations across all 60 discrete time points within a one-hour interval. A prediction for this quantity can be directly derived from probabilistic predictors of the grid frequency vector. To do so, a prediction of the grid frequency vector is first generated. Then, the average of the predicted grid frequency values over all time points within the hour is computed, and the deviation of this average from the nominal grid frequency is taken.

To illustrate the performance of different probabilistic models, we use the learned probabilistic predictors to generate synthetic grid frequency vectors for each of the 24 hourly intervals of a given example day in the test dataset, based on the corresponding input features. For each interval, we perform a Monte Carlo simulation by generating 1,000 synthetic grid frequency vectors, resulting in 1,000 corresponding predictions of the aver-

age frequency deviation. Based on these simulated predictions, we estimate a confidence interval for the mean frequency deviation for each hourly interval.

The confidence level is set between the quantile level 0.02275 and the quantile level 0.97725, which corresponds to an uncertainty range of $\mu \pm 2\sigma$ in a normal distribution. We compare the results of copula-based probabilistic models with independent Gaussian models and the GRU-based correlated Gaussian model. In Figure 7.3, one can clearly see that the mean frequency deviation is very often outside the 2 σ range of the independent Gaussian models or the confidence interval of the correlated Gaussian model, while the uncertainty regions of the copula-based predictors cover the fluctuations of the mean frequency deviation much better. Interestingly, none of the models are able to accurately predict the mean frequency deviation around midday on the example day.

7.4. Conclusion

In this chapter, we presented a grid frequency prediction methodology where an existing point predictor can be transformed into a copula-based probabilistic predictor using feature space-based probabilistic error correction. The created probabilistic models are able to capture the dependencies between different time points in a prediction hour interval and show better performance in terms of energy scores compared to their independent baseline counterparts. Additionally, by simulating the mean grid frequency deviation from different models, we have shown that both temporally independent models such as independent Gaussian process-based models as well as correlated Gaussian process models cannot account for enough uncertainty, while the copula-based models perform much better with respect to this. The feature-based error correction technique can also be used for other energy data prediction tasks.

8. Conclusion

This thesis develops new methods at the intersection of copula theory and deep learning, advancing both nonparametric copula density estimation and dependence-aware probabilistic prediction. The first two contributions introduce neural network-based copula estimators using normalizing flows and separable perturbations of independence. The third contribution provides a smoothing framework that transforms discrete checkerboard copulas into continuous densities. The final two contributions apply probabilistic modeling techniques, including copula-based approaches, to power grid frequency prediction. Throughout, the emphasis lies on flexible, data-driven methods that learn dependence structures without imposing restrictive parametric assumptions.

Chapter 3 introduces the normalizing flow copula model, which parameterizes the copula through a learnable transformation of a base distribution. We prove that increasing triangular transformations are sufficiently expressive to represent any absolutely continuous copula, allowing the model to be implemented without loss of generality using triangular maps. The proposed affine coupling-based architecture consistently outperforms nonparametric competitors such as empirical Bernstein copulas and achieves performance comparable to kernel-based estimators. Notably, it is the only nonparametric approach among those considered that accurately captures tail dependence. For multivariate data, the model yields particularly strong results, outperforming all competing methods. Applications to insurance and engineering data confirm that the normalizing flow copula model offers a flexible and robust tool for multidimensional copula modeling.

Chapter 4 develops an alternative neural network-based copula density estimator using separable perturbations of independence. The approach represents the copula density as the independence density plus a separable sum, where each term is the product of two univariate functions. We prove that bounded separable expansions with mean-zero component functions are dense in the space of square-integrable copula densities, justifying the expressive power of the approach. The proposed architecture enforces marginal uniformity through a Stein-type construction. In simulations, the estimator performs competitively with kernel density estimators on standard parametric families and outperforms existing methods for copulas with heterogeneous local dependence patterns. On real data, a vine copula model using the proposed bivariate estimator as a building block achieves the best fit among the considered methods.

Chapter 5 presents a method for transforming checkerboard copulas into smooth copula densities while preserving the empirical mass distribution. The approach formulates the embedding as an L^2 approximation problem with pointwise nonnegativity constraints, leading to a convex quadratic program that can be efficiently solved using standard optimization methods. Different basis families exhibit distinct properties: Legendre polynomials are ideal for capturing smooth transitions in continuous copulas, trigonometric bases are better suited for visualizing sparse structures, and B-splines provide balanced performance. Applied to credit rating transitions, the method resolves economically implausible discontinuities where firms at adjacent rating boundaries show abrupt transition probabilities. For age-income dependencies from the American Community Survey, the smooth copula enables more accurate reconstruction of fine-scale structure from aggregated data.

Chapter 6 presents a fully data-driven approach to modeling and predicting short-term grid frequency dynamics based on a combination of Gaussian processes and sequence models. Using gated recurrent units and transformer architectures, we extract information from techno-economic features to predict the parameters of a Gaussian process governing within-hour frequency evolution. Although the approach does not explicitly model physical properties, it achieves results comparable to physics-informed machine learning models and outperforms simple data-driven baselines. The synthetic data generated by our models fulfill the typical stochastic properties of frequency data, including fat-tail behavior and realistic autocorrelation structure, offering simulation capabilities for testing purposes in different scenarios.

Chapter 7 extends the grid frequency modeling framework by transforming an existing point predictor into a copula-based probabilistic predictor using feature space-based error correction. The resulting models capture dependencies between different time points within a prediction interval and show better performance in terms of energy scores compared to independent baseline counterparts. Importantly, we demonstrate that both temporally independent Gaussian process models and correlated Gaussian process models underestimate uncertainty, while the copula-based models provide a more accurate representation of predictive uncertainty.

Several directions for future work emerge from this thesis. The normalizing flow copula model of Chapter 3 could be extended to conditional copula estimation, where the dependence structure varies with covariates. For the separable perturbation approach of Chapter 4, promising directions include extensions to more general vine constructions and explicit control over the number of terms in the separable representation. The checkerboard smoothing framework of Chapter 5 can be extended to higher-dimensional copulas and augmented with additional criteria such as entropy constraints or moment-matching conditions. For the grid frequency models, the feature-based error correction technique

can be applied to other energy data prediction tasks, and the copula-based probabilistic correction could be combined with more advanced point predictors or foundation models for time series.

Taken together, this thesis advances the intersection of copula theory and deep learning by developing flexible neural network-based copula estimators, providing a principled framework for smoothing discrete copula representations, and demonstrating the practical utility of copula-based dependence modeling for probabilistic prediction of power grid frequency dynamics.

Appendices

A. Appendix to Chapter 3

A.1. Dependency Structure

One phenomenon that often needs to be modeled in practice, especially in financial risk management, is tail dependence (for detailed discussions, see, e.g., Embrechts et al., 2001), which is intended to represent the dependence of extreme events. It can be described by the tail dependence coefficients (Nelsen, 2006).

Definition A.1.1 (Tail Dependence Coefficient)**.** *Let X and Y be two continuous random variables with distribution functions F_X and F_Y respectively. Provided that the following limits exist, the coefficient of lower tail dependence λ_U of (X, Y) is defined by*

$$\lambda_L = \lim_{q \to 0^+} \mathbb{P}\left(Y \leq F_Y^{-1}(q) \mid X \leq F_X^{-1}(q)\right) \tag{A.1}$$

and the coefficient of upper tail dependence λ_L of (X, Y) is defined by

$$\lambda_U = \lim_{q \to 1^-} \mathbb{P}\left(Y > F_Y^{-1}(q) \mid X > F_X^{-1}(q)\right) \tag{A.2}$$

Under the assumption in Definition A.1.1, tail dependence coefficients are uniquely determined by the underlying copula $C_{X,Y}$ of the random vector (X, Y). In particular, the following applies

$$\lambda_{L,(X,Y)} = \lim_{q \to 0^+} \frac{C(q,q)}{q} \quad \text{and}$$

$$\lambda_{U,(X,Y)} = \lim_{q \to 1^-} \frac{1 - 2q + C(q,q)}{1 - q}.$$

It is interesting to note, for example, that the Gaussian copula has no tail dependence, while the Clayton copula has lower tail dependence and the Gumbel copula has upper tail dependence, where the tail dependence is controlled by the corresponding copula parameters. A table of tail dependence coefficients of different parametric copula families can be found in Joe (1997).

Another important concept is asymmetric dependence. In general, there are the following two types of asymmetry for bivariate copulas (Hofert et al., 2018, Definition 2.5.4).

Definition A.1.2 (Bivariate Radial Symmetry)**.** *A bivariate copula C is called radial symmetric if* $C(u, v) = u + v - 1 + C(1 - u, 1 - v)$.

Definition A.1.3 (Exchangeability)**.** *A bivariate copula C is called exchangeable if* $C(u, v) = C(v, u)$.

A bivariate Gaussian copula is both radially symmetric and exchangeable and, e.g., a Clayton copula is not radially symmetric but exchangeable.

A.2. Implementation Details on the AC-NFCM

For the 2D experiments in the simulation studies, we use an AC-NFCM with a total of 4 coupling transformations and for multi-dimensional experiments 6 coupling layers. For the 2D real insurance data, we use the same model and settings as for the 2D simulation studies. For the 5D engineering dataset, we consider a flow copula model with 6 and 12 coupling transformations.

The t- and s-functions of each coupling function (see Section 3.3.1) are implemented using a 6 hidden layer neural network with 128 neurons and Gaussian Error Linear Units as the activation function for the intermediate layers (Hendrycks and Gimpel, 2016). The output layer of t has the linear activation, while the output layer of s has the tanh activation, because s controls the scaling in a coupling transformation and the boundary property of tanh leads to a stable training process. For univariate kernel density estimation, the Gaussian kernel method implemented in the scipy package in python (Virtanen et al., 2020) is used. The training process is carried out with a batch size of 128, a number of epochs of 100 and a learning rate of 0.0001, where we also use early stopping and a learning rate reduction technique described next.

The validation error is monitored during the training process. For the 2D experiments, training is stopped if the validation loss does not improve for 10 epochs and the learning rate is reduced by a factor of 0.1 if the validation loss does not improve for 5 epochs. To accelerate the multidimensional experiments, training is stopped if the validation loss does not improve for 5 epochs and the learning rate is reduced by a factor of 0.1 if the validation loss does not improve for 2 epochs.

The models are implemented and trained using keras (Watson et al., 2024) and tensorflow (Abadi et al., 2015). The underlying normalizing flow architecture is based on the framework of an implementation of RealNVP (Dinh et al., 2016) provided by Keras (Mandolini et al., 2020), and is modified for our purposes according to the design ideas in 3.3.1 to obtain an AC-NFCM.

For AC-NFCM training, we perform a preprocessing step when only a few data points are available for training. The proposed AC-NFCM is based on deep neural networks,

which are primarily designed to process and learn from large amounts of data. Such a deep learning model can run into problems if only a small sample is available, as in this case overfitting can be achieved quickly, resulting in the models working well on the training data but generalizing poorly. In addition, the training process can be unstable, so that the results contain large variances.

To address the small sample problem in estimating AC-NFCMs, we propose applying a resampling technique to expand the dataset available for estimation. Let the original dataset be denoted by $\mathcal{D} = \{x_i\}_{i=1}^{N}$. An augmented dataset $\mathcal{D}^* = \{x_j^*\}_{j=1}^{N_a}$, with $N_a > N$, is constructed by sampling with replacement from $\mathcal{D}$. This preprocessing step artificially augments the dataset, where we use $N_a = 5000$, and our experiments indicate an improved robustness of the estimation procedure. Pseudo observations are then calculated from the enlarged data set using rank transformation.

Note that in settings with limited sample size, resampling prior to rank transformation increases the incidence of rank ties. Applying random tie-breaking assigns tied observations to distinct order positions within their admissible rank interval, thereby inducing stochastic dispersion of the resulting pseudo-observations. This mechanism constitutes an implicit smoothing of the empirical copula input.

A.3. Baseline Estimators

In the following, we will briefly discuss the considered baseline estimators and how they are implemented in the simulation study. We first recall the empirical copula C_N, noting that the empirical checkerboard, empirical Bernstein, and empirical beta copulas can all be interpreted as smoothed versions of C_N.

A.3.1. Empirical Copula

The empirical copula C_N provides a nonparametric estimator of the underlying copula and is defined by (see, e.g., Hofert et al. (2018, Section 4.2.1))

$$C_N(\mathbf{u}) = \frac{1}{N} \sum_{i=1}^{N} \prod_{j=1}^{d} \mathbf{1}(\hat{u}_j^{\,i} \leq u_j), \qquad \mathbf{u} \in [0,1]^d,$$

where $\hat{\mathbf{u}}^{\,i} = (\hat{u}_1^{\,i}, \dots, \hat{u}_d^{\,i})^\top$, $i = 1, \dots, N$, are the pseudo-observations.

A.3.2. Empirical Checkerboard Copula

We define the empirical checkerboard copula according to Cuberos et al. (2020). Let $[0,1]^d$ be the d-dimensional unit cube and let λ denote the Lebesgue measure. The checkerboard copula approximates a copula on the m^d sub-cubes with side length $\frac{1}{m}$

$$I_{\mathbf{i},m} = \prod_{j=1}^{d} \left(\frac{i_j - 1}{m}, \frac{i_j}{m} \right], \qquad \mathbf{i} \in \{1, \dots, m\}^d.$$

For a copula with measure μ, the checkerboard copula C_m^* is defined by

$$C_m^*(\mathbf{u}) = \sum_{\mathbf{i} \in \{1,\dots,m\}^d} m^d \mu(I_{\mathbf{i},m}) \lambda([\mathbf{0}, \mathbf{u}] \cap I_{\mathbf{i},m}).$$

For a given sample, the empirical checkerboard copula is defined analogously, using the measure $\hat{\mu}$ of the empirical copula instead of the measure μ of the true copula.

The (empirical) checkerboard copula has constant density over each sub-cube. For the simulation study, we use the implementation from Laverny (2020) and consider the cases $m = 5, 10$.

A.3.3. Empirical Bernstein Copula

We follow the definition and notation in Segers et al. (2017). For a function $f : [0,1]^d \to \mathbb{R}$, the Bernstein polynomial of order $\mathbf{m} = (m_1, \dots, m_d) \in \mathbb{N}^d$ of f is defined as

$$B_{\mathbf{m}}(f)(\mathbf{u}) = \sum_{s_1=0}^{m_1} \cdots \sum_{s_d=0}^{m_d} f\left(\frac{s_1}{m_1}, \dots, \frac{s_d}{m_d} \right) \prod_{j=1}^{d} p_{m_j,s_j}(u_j),$$

where $\mathbf{u} = (u_1, \dots, u_d) \in [0,1]^d$ and

$$p_{m_j,s_j}(u_j) = \binom{m_j}{s_j} u_j^{s_j} (1 - u_j)^{m_j - s_j}, \qquad u_j \in [0,1],\ m_j \in \mathbb{N},\ s_j \in \{0, \dots, m_j\}.$$

The function $B_{\mathbf{m}}(C)$ is called the Bernstein copula of C. Let C_N be the empirical copula of a sample of size N. The empirical Bernstein copula is then defined as $B_{\mathbf{m}}(C_N)$. In Segers et al. (2017), it is shown that the empirical Bernstein copula $B_{\mathbf{m}}(C_N)$ is a proper copula if and only if all polynomial degrees $m_1, \dots, m_d$ are divisors of N. For the simulation study, we use the implementation in Baudin et al. (2016) and consider the cases $m = 10, 25$.

A.3.4. Empirical Beta Copula

We follow the definition in Segers et al. (2017). For a d-dimensional sample of size N, the empirical beta copula is defined as

$$C_N^{\beta}(\mathbf{u}) = \frac{1}{N} \sum_{i=1}^{N} \prod_{j=1}^{d} F_{N,R_{ij}}(u_j),$$

where $F_{N,r}$ denotes the CDF of the beta distribution with parameters r and $N+1-r$, and R_{ij} is the rank of x_j^i among $x_j^1, \dots, x_j^N$. In Segers et al. (2017), it is shown that the empirical beta copula is a special case of the empirical Bernstein copula $B_{\mathbf{m}}(C_N)$ with $\mathbf{m} = (N, \dots, N)$. For the simulation study, we use the implementation in Baudin et al. (2016).

A.3.5. Kernel Density Estimator

For a bivariate sample $\mathbf{u}^1, \dots, \mathbf{u}^N \in [0,1]^2$, the basic kernel density estimator is given by

$$\hat{c}_N(\mathbf{u}) = \frac{1}{N} \sum_{i=1}^{N} K_{\mathbf{H}}(\mathbf{u} - \mathbf{u}^i), \qquad \mathbf{u} \in [0,1]^2,$$

where K is a kernel function (e.g., a bivariate standard normal density) and $\mathbf{H}$ is the bandwidth matrix. Many refined approaches exist in the literature (Nagler, 2018). For the simulation study, we consider the default method in Nagler (2018), the transformation method TLL2nn, which approximates the log-density by quadratic polynomials (see Geenens et al., 2017).

The bivariate approach can be extended to higher dimensions via the vine copula structure (Nagler and Czado, 2016). Under the simplifying assumption (Czado, 2019, p. 78), a d-dimensional copula density can be decomposed into $d(d-1)/2$ (conditional) bivariate copula densities, each of which can be estimated using the kernel density estimator described above. For the simulation study, we use the implementation in Nagler (2024).

A.4. Copula Examples for Simulation Studies

In the following, we present details on the copulas used in the simulation study. For details on particular copula families, we refer to Joe (2014).

A.4.1. Bivariate Copula Families

- **Clayton Copula**:

$$C_\theta(u,v) = \left(\max\left\{u^{-\theta} + v^{-\theta} - 1, 0\right\}\right)^{-1/\theta}, \quad \theta > 0$$

- **Gumbel Copula**:

$$C_\theta(u,v) = \exp\left(-\left((-\ln(u))^\theta + (-\ln(v))^\theta\right)^{1/\theta}\right), \quad \theta \geq 1$$

- **Gaussian Copula**:

$$C_\rho(u,v) = \Phi_\rho\left(\Phi^{-1}(u), \Phi^{-1}(v)\right)$$

where Φ_ρ is the CDF of the multivariate standard normal distribution and Φ is the CDF of the standard normal distribution.

- **t-Copula**:

$$C_{\nu,\rho}(u,v) = t_{\nu,\rho}(t_\nu^{-1}(u), t_\nu^{-1}(v))$$

where $t_{\nu,\rho}$ is the CDF of the multivariate t-distribution and t_ν is the CDF of the t-distribution.

- **Galambos Copula**:

$$C_\theta(u,v) = \exp\left(-\left((-\ln(u))^{-\theta} + (-\ln(v))^{-\theta}\right)^{-1/\theta}\right), \quad \theta > 0$$

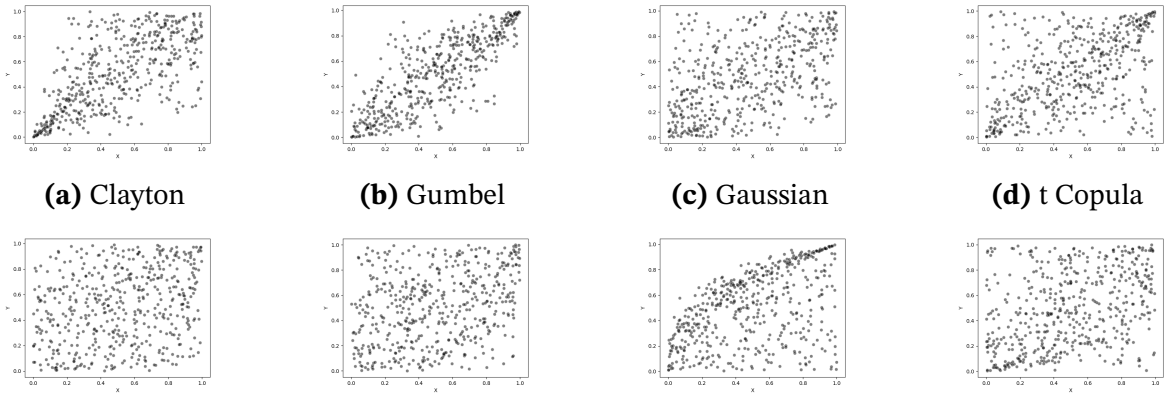
- **t-EV Copula**:

$$C_{\rho,\nu}(u,v) = \exp\left(-(x+y)B\left(\frac{x}{x+y}; \rho, \nu\right)\right)$$

where $x = -\log(u)$, $y = -\log(v)$ and

$$\begin{aligned} B(w;\rho,\nu) = {} & wT_{\nu+1}\left(\frac{\sqrt{\nu+1}}{\sqrt{1-\rho^2}}\left(\left(\frac{w}{1-w}\right)^{1/\nu} - \rho\right)\right) \\ & + (1-w)T_{\nu+1}\left(\frac{\sqrt{\nu+1}}{\sqrt{1-\rho^2}}\left(\left(\frac{1-w}{w}\right)^{1/\nu} - \rho\right)\right), \end{aligned}$$

where $T_{\nu+1}$ is the CDF of the univariate t-distribution with $\nu+1$ degrees of freedom.



(a) Clayton **(b)** Gumbel **(c)** Gaussian **(d)** t Copula

(e) Galambos **(f)** t-EV Copula **(g)** kho 1 **(h)** kho 2

Figure A.1.: Synthetic data of bivariate copula families, 500 points each.

Definition A.4.1 (Khoudraji's device in 2D (Khoudraji, 1995; Hofert et al., 2018))**.** *Given two bivariate copulas C_1 and C_2, and a shape vector $s = (s_1, s_2) \in [0,1]^2$, Khoudraji's device is defined as:*

$$kho_{(s_1,s_2)}(C_1, C_2)(u_1, u_2) = C_1\left(u_1^{1-s_1}, u_2^{1-s_2}\right) C_2\left(u_1^{s_1}, u_2^{s_2}\right), \quad u_1, u_2 \in [0,1].$$

A.4.2. Extreme Value Copulas

Lemma 4 (Characterization of bivariate Extreme-Value Copulas (Hofert et al., 2018))**.** *A copula C is an extreme-value copula if and only if there exists a function $A : [0,1] \to [1/2, 1]$ such that, for any $u_1, u_2 \in (0,1)$, the copula can be written as:*

$$C(u_1, u_2) = \exp\left(\log u_1 + \log u_2 \cdot A\left(\frac{\log u_2}{\log u_1 + \log u_2}\right)\right).$$

The function A is called the Pickands dependence function associated with the copula C.

A.4.3. Vine Copula Example

For the simulation, we use the Package VineCopula of Nagler et al. (2024) in R (R Core Team, 2024) to generate synthetic data of the following 5-dimensional R-Vine Copula with the structure matrix M_R, the copula family matrix F_R, and the parameter matrix P_R for a 5-dimensional R-Vine Copula, defined as follows:

$$M_R = \begin{pmatrix} 5 & 2 & 3 & 1 & 4 \\ 0 & 2 & 3 & 4 & 1 \\ 0 & 0 & 3 & 4 & 1 \\ 0 & 0 & 0 & 4 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$F_R = \begin{pmatrix} 0 & 1 & 3 & 4 & 4 \\ 0 & 0 & 3 & 4 & 1 \\ 0 & 0 & 0 & 4 & 1 \\ 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$P_R = \begin{pmatrix} 0 & 0.2 & 0.9 & 1.5 & 3.9 \\ 0 & 0 & 1.1 & 1.6 & 0.9 \\ 0 & 0 & 0 & 1.9 & 0.5 \\ 0 & 0 & 0 & 0 & 4.8 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

The entries in the family matrix F denote the type of copula used for each pair of variables. The numbers correspond to specific copula families:

- 1: Gaussian copula
- 3: Clayton copula
- 4: Gumbel copula

For details about constructions of R-Vine Copula, we refer to Czado (2019). For Matlab users, we recommend the MATVines package for vine copulas (Coblenz, 2021).

A.5. Further Results of Small Sample Experiments

In the following, in Figure A.2, A.4, A.3 the results of the simulation studies of samples of size $n = 100$ for bivariate copula families and in Figure A.5 and Table A.1 the results of the multidimensional copulas can be found.

The results for $n = 100$ should be interpreted with caution. At this sample size, particularly in dimensions $d > 3$, all nonparametric estimators operate near their limits and all show large errors. The estimators exhibit different responses to the small sample regime: some show increased variance, while others display inherent bias induced by

their structural assumptions, for instance, Bernstein estimators through polynomial basis constraints or AC-NFCM through preprocessing steps. A detailed bias-variance decomposition of these estimators in small-sample settings would be interesting, but lies beyond the scope of this work.

		Estimators						
	Copulas	CB_5	CB_{10}	BS_{10}	BS_{25}	Beta	Vine KDE	AC-NFCM
Dim 3	Clayton	0.831	1.661	**0.331**	0.500	1.112	0.334	0.372
	Gumbel	0.852	1.666	**0.312**	0.506	1.131	0.357	0.366
	Gaussian	0.825	1.669	**0.275**	0.502	1.153	0.336	0.371
Dim 4	Clayton	1.427	1.790	**0.471**	0.820	1.559	0.520	0.594
	Gumbel	1.484	1.821	**0.464**	0.825	1.573	0.535	0.595
	Gaussian	1.465	1.830	**0.423**	0.832	1.592	0.513	0.651
Dim 5	Clayton	1.717	1.742	**0.647**	1.119	1.768	0.696	0.817
	Gumbel	1.822	1.745	**0.650**	1.142	1.718	0.713	0.799
	Gaussian	1.771	1.817	**0.605**	1.156	1.760	0.695	0.828
	Vine	1.277	1.662	1.007	0.907	1.317	0.814	**0.767**

Table A.1.: Median IAE of the estimation results for multivariate copulas by using different non-parametric estimators for 100 samples with a size of 100 each.

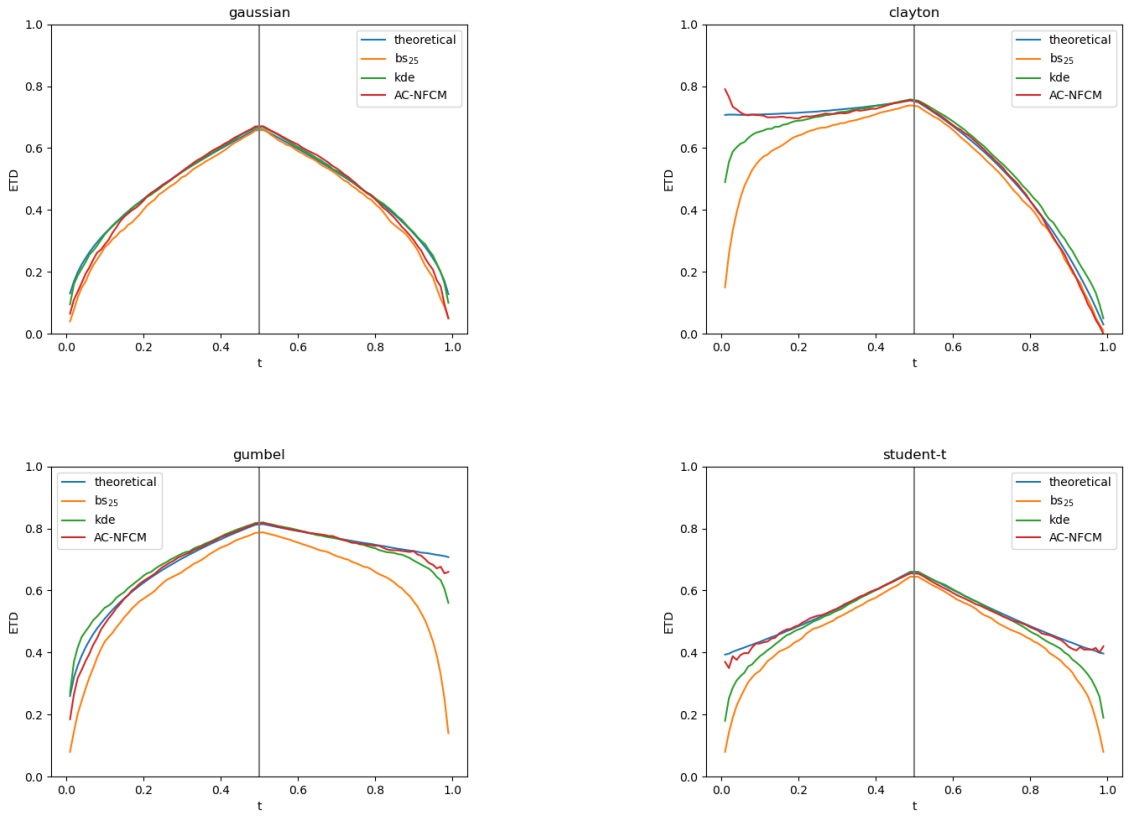


Figure A.2.: Results for experiments with small samples: Tail behavior of different non-parametric estimators. Evaluated on synthetic data generated from the estimators: Gaussian copula data (top left), Clayton copula data (top right), Gumbel copula data (bottom left) and Student-t copula data (bottom right). We generated a total of 100 samples of size 10,000 and plot the median values here.

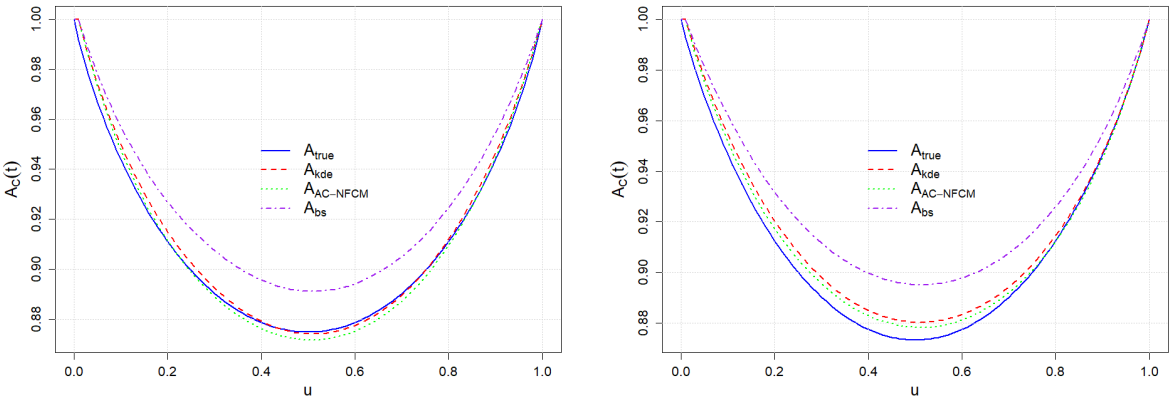
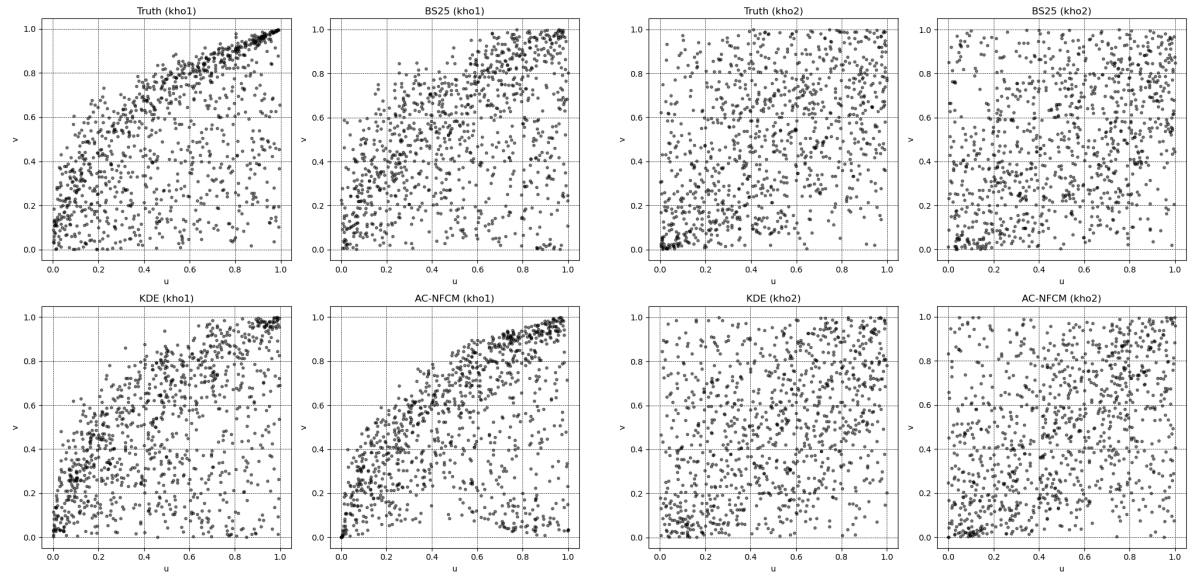
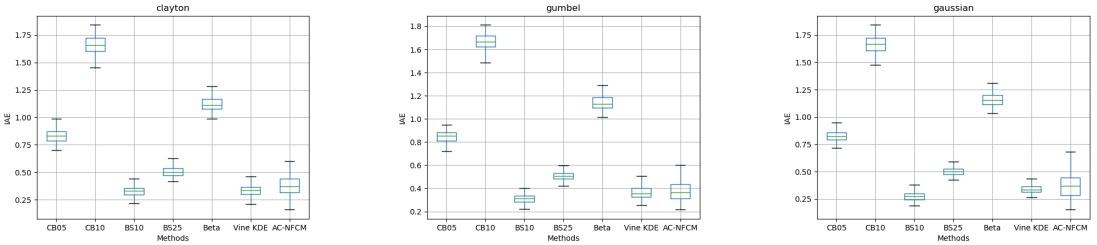


Figure A.3.: Results for experiments with small samples: Estimation of Pickands dependence functions by using synthetic data generated from different estimators. A total of 100 samples are generated for each copula family. The mean values of the approximated curves are shown. Left: Galambos copula; right: t-EV copula.

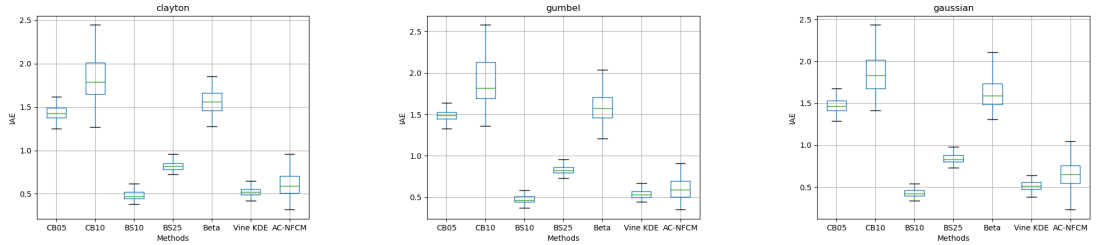


(a) kho1 **(b)** kho2

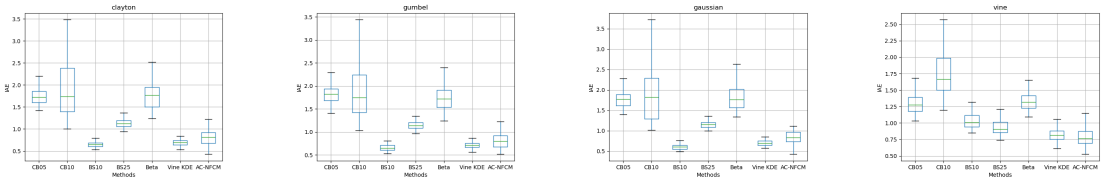
Figure A.4.: Results for experiments with small samples. Left: synthetic datasets generated by estimators from kho1 copula data. Right: synthetic datasets generated by estimators from kho2 copula data.



(a) 3D Experiments.



(b) 4D Experiments.



(c) 5D Experiments.

Figure A.5.: Boxplots of IAE results from different estimators for 100 samples with a size of 100 each.

B. Appendix to Chapter 4

B.1. Proof of Theorem 4.2.1

Proof. Without loss of generality, we may assume that the families $\{f_k\}_{k=1}^K$ and $\{g_k\}_{k=1}^K$ are each linearly independent in $L^1([0,1])$: if not, the representation can be reduced to one with fewer, linearly independent terms while preserving the zero-mean constraints and nonnegativity. The assumption $\sum_{k=1}^K f_k g_k \not\equiv 0$ ensures that this reduction does not yield the trivial case $c \equiv 1$.

Suppose, to the contrary, that at least one of the functions $f_1, \dots, f_K$ is unbounded on $[0,1]$. Then there exists a sequence $(u_m)_{m\in\mathbb{N}} \subset [0,1]$ such that

$$M_m := \Big(\sum_{k=1}^K f_k(u_m)^2 \Big)^{1/2} \xrightarrow[m\to\infty]{} +\infty.$$

Define $a_{k,m} := f_k(u_m)/M_m$, so that $a_m := (a_{1,m}, \dots, a_{K,m})$ lies on the unit sphere in $\mathbb{R}^K$. Passing to a subsequence, we may assume $a_{k,m} \to a_k$ for each k, with $\sum_{k=1}^K a_k^2 = 1$. For each fixed $v \in [0,1]$ and every m, the assumption gives

$$0 \le 1 + \sum_{k=1}^K f_k(u_m)\, g_k(v) = 1 + M_m \sum_{k=1}^K a_{k,m}\, g_k(v),$$

hence

$$0 \le \frac{1}{M_m} + \sum_{k=1}^K a_{k,m}\, g_k(v).$$

Since $a_{k,m} \to a_k$ for all k, taking $m \to \infty$ yields

$$\sum_{k=1}^K a_k\, g_k(v) \ge 0 \quad \text{for all } v \in [0,1].$$

Integrating and using the zero-mean constraints gives

$$\int_0^1 \sum_{k=1}^{K} a_k\, g_k(v)\, dv = \sum_{k=1}^{K} a_k \int_0^1 g_k(v)\, dv = 0.$$

Since $\sum_{k=1}^{K} a_k\, g_k(v) \geq 0$ everywhere with integral zero, we have $\sum_{k=1}^{K} a_k\, g_k(v) = 0$ for all v. Linear independence of $\{g_k\}$ implies $a_k = 0$ for all k, contradicting $\sum_{k=1}^{K} a_k^2 = 1$. Thus all f_k are bounded. The argument for $g_1, \dots, g_K$ is symmetric, interchanging the roles of u and v and of the families $\{f_k\}$ and $\{g_k\}$. □

Remark B.1.1. *The zero-mean constraint is essential for the boundedness result. Appendix B.6 provides an example in which the zero-mean constraint is violated, all f_k and g_k change sign, some are unbounded, yet $1 + \sum_{k=1}^{K} f_k(u)g_k(v) \geq 0$ everywhere. This shows that sign changes alone do not enforce boundedness.*

B.2. Schmidt Decomposition

We consider the low-rank approximation of bivariate functions with sufficient regularity. The goal is to represent such functions efficiently as finite sums of separable terms. The theoretical foundation for this approach is the singular value decomposition of compact operators, which provides the best rank-r approximation in the operator norm.

Lemma 5 (Schmidt Decomposition for L^2 Functions)**.** *Let $\mathcal{K} \in L^2([0,1]^2)$. Then there exist orthonormal systems $\{\alpha_k\}_{k\geq 1} \subset L^2(0,1)$ and $\{\beta_k\}_{k\geq 1} \subset L^2(0,1)$, and a nonincreasing sequence of singular values*

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq 0, \qquad \sigma_k \to 0,$$

such that

$$\mathcal{K}(u,v) = \sum_{k=1}^{\infty} \sigma_k\, \alpha_k(u)\, \beta_k(v) \quad \text{in } L^2([0,1]^2).$$

Proof. See Griebel and Li, 2018, pp. 977-978. □

Lemma 6 (Best Rank-r Approximation Schwab and Todor, 2006, Lemma 2.7)**.** *Let $\mathcal{K} \in L^2([0,1]^2)$ admit the Schmidt decomposition from Lemma 5 with $\sigma_1 \geq \sigma_2 \geq \cdots \geq 0$. For $r \in \mathbb{N}$ define the truncation*

$$\mathcal{K}_r(u,v) := \sum_{k=1}^{r} \sigma_k\, \alpha_k(u)\, \beta_k(v).$$

Then $\mathcal{K}_r$ is a best rank-r approximation of $\mathcal{K}$ in the L^2 norm:

$$\|\mathcal{K} - \tilde{\mathcal{K}}\|_{L^2} \geq \|\mathcal{K} - \mathcal{K}_r\|_{L^2} \quad \textit{for all } \tilde{\mathcal{K}} = \sum_{k=1}^{r} a_k(u)b_k(v),\ a_k, b_k \in L^2(0,1),$$

and the error satisfies

$$\|\mathcal{K} - \mathcal{K}_r\|_{L^2}^2 = \sum_{k>r} \sigma_k^2.$$

Remark B.2.1. *The Schmidt decomposition of functions appears under different names across various disciplines. It is known as the Schmidt decomposition in quantum information theory (Nielsen and Chuang, 2010), the Karhunen-Loève expansion in stochastic processes (Ghanem and Spanos, 2003), and the Proper Orthogonal Decomposition in fluid dynamics (Berkooz et al., 1993).*

B.3. Vertex Nonnegativity and Low-Rank Approximation

The Schmidt decomposition of a bivariate L^2 function provides truncations that are L^2-optimal among all rank-r separable approximations. In our copula setting, the truncations

$$S_r(u,v) = 1 + \sum_{k=1}^{r} \sigma_k\, \alpha_k(u)\, \beta_k(v)$$

preserve the uniform margins, but they are not guaranteed to be nonnegative almost everywhere. Hence a lower-rank truncation need not be a valid copula density.

In the following, we provide a sufficient condition for the nonnegativity of partial sums. To motivate the idea, we first explain why it suffices to test only finitely many parameter choices.

For fixed $v \in (0,1)$, the map

$$(\varepsilon_1, \dots, \varepsilon_K) \longmapsto 1 + \sum_{k=1}^{K} \varepsilon_k\, g_k(v)$$

is affine in the parameters $(\varepsilon_1, \dots, \varepsilon_K)$, while the admissible set $[a_1, b_1] \times \cdots \times [a_K, b_K]$ is a convex box (a polytope). Affine functions attain their minima on convex polytopes at *vertices*. Hence, verifying nonnegativity for all $\varepsilon_k \in [a_k, b_k]$ reduces to checking the finitely many corner points $\{a_1, b_1\} \times \cdots \times \{a_K, b_K\}$. We encode this finite vertex test in the following notion.

Definition B.3.1 (Vertex Nonnegativity)**.** *Let $K \in \mathbb{N}$ and $a_k < b_k$ for $k = 1, \dots, K$ be given. For*

$$g_k : (0,1) \to \mathbb{R}, \quad k = 1, \dots, K,$$

we say that the family $(g_k)_{k=1}^K$ satisfies vertex nonnegativity (with respect to $\{a_k, b_k\}_{k=1}^K$) if

$$1 + \sum_{k=1}^{K} \varepsilon_k \, g_k(v) \geq 0 \tag{B.1}$$

for all $v \in (0,1)$ and all $(\varepsilon_1, \dots, \varepsilon_K) \in \{a_1, b_1\} \times \cdots \times \{a_K, b_K\}$.

Lemma 7 (Partial-sum nonnegativity under vertex nonnegativity)**.** *Let $K \in \mathbb{N}$ and fix intervals $a_k < b_k$ with $0 \in [a_k, b_k]$ for $k = 1, \dots, K$. Suppose*

$$f_k : (0,1) \to \mathbb{R}, \qquad g_k : (0,1) \to \mathbb{R} \quad \forall\, k = 1, \dots, K$$

satisfy

$$a_k \leq f_k(u) \leq b_k \quad \text{for all } u \in (0,1),\ k = 1, \dots, K, \tag{B.2}$$

and that $(g_k)_{k=1}^K$ satisfies vertex nonnegativity*, i.e.*

$$1 + \sum_{k=1}^{K} \varepsilon_k \, g_k(v) \geq 0 \quad \text{for all } v \in (0,1),\ (\varepsilon_1, \dots, \varepsilon_K) \in \prod_{k=1}^{K} \{a_k, b_k\}. \tag{B.3}$$

Then, for every index $K' \leq K$,

$$1 + \sum_{k=1}^{K'} f_k(u) \, g_k(v) \geq 0 \quad \text{for all } (u,v) \in (0,1)^2. \tag{B.4}$$

Proof. Fix $(u,v) \in (0,1)^2$ and define $\varepsilon_k^* \in \{a_k, b_k\}$ by

$$\varepsilon_k^* = \begin{cases} a_k, & \text{if } g_k(v) > 0, \\ b_k, & \text{if } g_k(v) < 0, \\ a_k \text{ (or } b_k), & \text{if } g_k(v) = 0. \end{cases}$$

By (B.2), for all $k \leq K'$ we have $\varepsilon_k^* g_k(v) \leq f_k(u) g_k(v)$. Since $0 \in [a_k, b_k]$, for $k > K'$ the choice above yields $\varepsilon_k^* g_k(v) \leq 0$. Hence

$$1 + \sum_{k=1}^{K} \varepsilon_k^* g_k(v) \leq 1 + \sum_{k=1}^{K'} f_k(u) g_k(v).$$

Vertex nonnegativity (B.3) gives that the left-hand side is nonnegative, which proves (B.4). □

Remark B.3.1 (Two immediate sufficient conditions)**.** *The vertex nonnegativity condition for $(g_k)_{k=1}^K$ with respect to $\{a_k, b_k\}$ follows from either of the following elementary hypotheses:*

1. Sign consistency. *If for all k either $a_k, b_k \geq 0$ and $g_k(v) \geq 0$ for all $v \in (0,1)$, or $a_k, b_k \leq 0$ and $g_k(v) \leq 0$ for all v, then*

$$1 + \sum_{k=1}^{K} \varepsilon_k g_k(v) \geq 1 \quad \text{for all } (\varepsilon_1, \dots, \varepsilon_K) \in \prod_{k=1}^{K} \{a_k, b_k\},$$

   hence vertex nonnegativity holds (indeed, with a margin).

2. Uniform smallness. *Set $M_k := \max\{|a_k|, |b_k|\}$ and assume there exist bounds $C_k \geq 0$ with $\sup_{v\in(0,1)} |g_k(v)| \leq C_k$ and $\sum_{k=1}^K M_k C_k < 1$. Then, for any choice of endpoints,*

$$\Big|\sum_{k=1}^{K} \varepsilon_k g_k(v)\Big| \leq \sum_{k=1}^{K} M_k C_k < 1,$$

   so $1 + \sum_{k=1}^K \varepsilon_k g_k(v) > 0$, and vertex nonnegativity holds.

Remark B.3.2 (Infinite case)**.** *Assume that $0 \in [a_k, b_k]$ for all $k \in \mathbb{N}$ and that, for every $v \in (0,1)$ and every sequence $(\varepsilon_k)_{k\geq 1} \in \prod_{k\geq 1}\{a_k, b_k\}$, the series $\sum_{k=1}^\infty \varepsilon_k g_k(v)$ converges and*

$$1 + \sum_{k=1}^{\infty} \varepsilon_k g_k(v) \geq 0.$$

If, in addition, $f_k : (0,1) \to \mathbb{R}$ satisfy $a_k \leq f_k(u) \leq b_k$ for all $u \in (0,1)$ and all k, then for every $K' \in \mathbb{N}$ and all $(u, v) \in (0,1)^2$,

$$1 + \sum_{k=1}^{K'} f_k(u)\, g_k(v) \geq 0.$$

This follows by passing to the limit from the finite case of Lemma 7.

The vertex nonnegativity constraint then guarantees that the lower-rank approximation is a valid copula.

Corollary B.3.1 (Best rank-r approximation under vertex nonnegativity)**.** *Let C be an absolutely continuous bivariate copula with density $c \in L^2([0,1]^2)$ and Schmidt decomposition*

$$c(u, v) = 1 + \sum_{k=1}^{\infty} \sigma_k\, \alpha_k(u)\, \beta_k(v),$$

where $(\alpha_k)_{k\geq 1}, (\beta_k)_{k\geq 1} \subset L^2(0,1)$ *are orthonormal families with* $\int_0^1 \alpha_k = 0 = \int_0^1 \beta_k$ *for all* k*, and* $(\sigma_k)_{k\geq 1}$ *are nonincreasing singular values. Assume there exist bounds* $M_k > 0$ *such that* $|\alpha_k(u)| \leq M_k$ *for a.e.* u *and set*

$$a_k := -\sigma_k M_k, \qquad b_k := \sigma_k M_k,$$

so that $0 \in [a_k, b_k]$ *and* $a_k \leq \sigma_k \alpha_k(u) \leq b_k$ *for all* u*. Assume moreover that* $(\beta_k)_{k\geq 1}$ *satisfies* (infinite) vertex nonnegativity with respect to $\{a_k, b_k\}$*, i.e., for every* $v \in (0,1)$ *and every sequence* $(\varepsilon_k)_{k\geq 1} \in \prod_{k\geq 1}\{a_k, b_k\}$ *the series* $\sum_{k=1}^{\infty} \varepsilon_k \beta_k(v)$ *converges and*

$$1 + \sum_{k=1}^{\infty} \varepsilon_k \beta_k(v) \geq 0.$$

Then, for every $r \in \mathbb{N}$*, the truncation*

$$c_r(u,v) = 1 + \sum_{k=1}^{r} \sigma_k \, \alpha_k(u) \, \beta_k(v)$$

is a copula density (nonnegative with uniform margins) and is the best rank-r approximation of c *in the* L^2*-norm.*

Proof. With $f_k := \sigma_k \alpha_k$ and $g_k := \beta_k$, we have $0 \in [a_k, b_k]$ and $a_k \leq f_k(u) \leq b_k$ for all u. By Lemma 7 and Remark B.3.2, each truncation c_r is nonnegative on $(0,1)^2$. The centering of α_k and β_k preserves the copula margins, hence c_r is a copula density. Optimality in L^2 follows from the Eckart-Young theorem (cf. Lemma 6 or Theorem 4.2.2). □

Example B.3.1. *Consider*

$$c(u,v) = 1 + \frac{\sin\big(2\pi(u+v)\big)}{5 - 4\cos\big(2\pi(u+v)\big)} + \frac{\sin\big(2\pi(u-v)\big)}{5 - 4\cos\big(2\pi(u-v)\big)}, \qquad (u,v) \in [0,1]^2.$$

This is a copula density with the series representation

$$c(u,v) = 1 + \sum_{k\geq 1} 2^{-k} \sin(2\pi k u)\cos(2\pi k v) \quad \textit{(cf. Appendix B.5).}$$

Let $\mathcal{K}(u,v) := c(u,v) - 1$ *and use the orthonormal systems*

$$\alpha_k(t) = \sqrt{2}\,\sin(2\pi k t), \qquad \beta_k(t) = \sqrt{2}\,\cos(2\pi k t), \qquad k \in \mathbb{N}.$$

Then

$$\mathcal{K}(u,v) = \sum_{k=1}^{\infty} \sigma_k \, \alpha_k(u) \, \beta_k(v), \qquad \sigma_k = \frac{1}{2^{k+1}}.$$

Hence

$$f_k(u) := \sigma_k \alpha_k(u) = \frac{1}{\sqrt{2}\,2^k} \sin(2\pi k u), \qquad g_k(v) := \beta_k(v) = \sqrt{2}\,\cos(2\pi k v).$$

Since $|f_k(u)| \le \frac{1}{\sqrt{2}\,2^k}$ *for all* u*, choose*

$$a_k = -\frac{1}{\sqrt{2}\,2^k}, \qquad b_k = \frac{1}{\sqrt{2}\,2^k}.$$

For any $v \in (0,1)$ *and any* $\varepsilon_k \in \{a_k, b_k\}$*,*

$$1 + \sum_{k=1}^{\infty} \varepsilon_k g_k(v) \ge 1 - \sum_{k=1}^{\infty} |\varepsilon_k|\,|g_k(v)| \ge 1 - \sum_{k=1}^{\infty} \frac{1}{\sqrt{2}\,2^k} \cdot \sqrt{2} = 1 - \sum_{k=1}^{\infty} \frac{1}{2^k} = 0.$$

Thus the (infinite) vertex nonnegativity condition (with respect to $\{a_k, b_k\}$*) holds.*

Hence each truncation

$$c_r(u,v) = 1 + \sum_{k=1}^{r} f_k(u)\, g_k(v)$$

is nonnegative and, by Corollary B.3.1, is a valid copula density and the best rank-r approximation in L^2 *(see Figure B.1 for the low-rank approximations).*

Remark B.3.3. *In the example above, both the copula density and its truncation belong to the class of so-called trigonometric copulas (Amblard and Girard, 2002; Chesneau, 2022; Longla, 2024).*

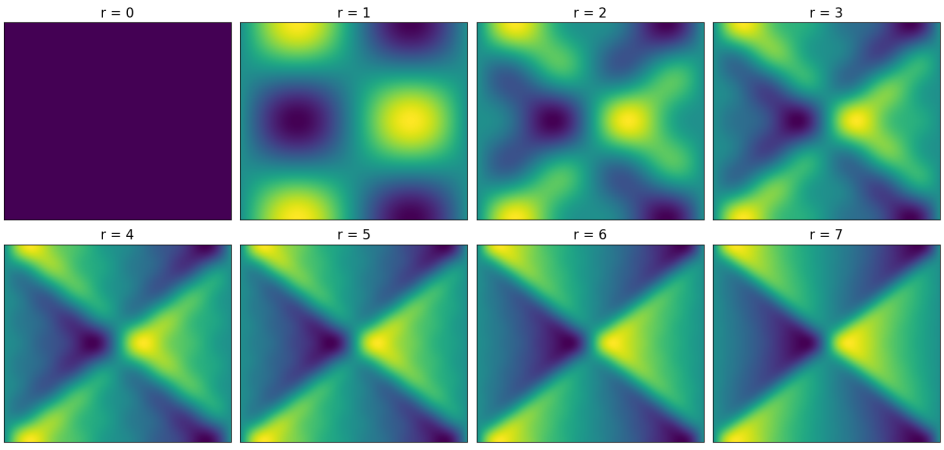


Figure B.1.: Rank-r approximation of a copula under vertex nonnegativity condition.

B.4. Copulas with Heterogeneous Dependence Patterns

Trigonometric Copula For $(u, v) \in [0,1]^2$ define

$$c(u,v) \;=\; 1 + \sin(4\pi u)\,\cos(3\pi v).$$

Since $\sin, \cos \in [-1,1]$, we have $c(u,v) \in [0,2]$ (hence nonnegative). Moreover,

$$\int_0^1 c(u,v)\,dv = 1 + \sin(4\pi u)\int_0^1 \cos(3\pi v)\,dv = 1, \qquad \int_0^1 c(u,v)\,du = 1,$$

and therefore

$$\iint_{[0,1]^2} c(u,v)\,du\,dv = 1.$$

Thus c is a valid copula density.

The copula C is obtained by double integration:

$$C(u,v) = \int_0^u \int_0^v c(s,t)\,dt\,ds = uv + \frac{1}{12\pi^2}\big(1-\cos(4\pi u)\big)\sin(3\pi v).$$

Checkerboard Copulas via Weight Matrices and Sinkhorn Scaling We partition $[0,1]^2$ into an $m \times m$ grid and specify a nonnegative weight matrix $W \in \mathbb{R}_+^{m\times m}$. Sinkhorn scaling produces a bistochastic matrix R (all row and column sums equal $1/m$), which ensures uniform margins and therefore defines a valid copula (see Sinkhorn and Knopp, 1967; Idel and Wolf, 2015).

In the fixed examples below we use $m = 40$.

Given $W \geq 0$, the scaling yields R satisfying

$$\sum_{j=1}^m R_{ij} = \sum_{i=1}^m R_{ij} = \frac{1}{m}, \qquad \sum_{i=1}^m \sum_{j=1}^m R_{ij} = 1.$$

The induced piecewise-constant copula density is

$$c(u,v) = m^2\,R_{ij} \quad \text{for} \quad (u,v) \in \Big[\tfrac{i-1}{m}, \tfrac{i}{m}\Big) \times \Big[\tfrac{j-1}{m}, \tfrac{j}{m}\Big), \;\; i,j \in \{1,\dots,m\}.$$

Sampling proceeds by drawing a cell (i,j) with probability R_{ij} and then sampling uniformly within that cell.

(1) Isotropic Gaussian Spots

Place K isotropic Gaussian bumps with common width $\sigma > 0$ and convex weights:

$$W_{ij} \mathrel{+}= \sum_{k=1}^{K} w_k \exp\left(-\frac{(i-c_k^x)^2 + (j-c_k^y)^2}{2\sigma^2}\right).$$

Parameter choice: $K = 10$, $\sigma = \frac{m}{15}$ (for $m = 40$: $\sigma \approx 2.67$).

(2) Anisotropic, Oriented Gaussian Spots

Each bump is elliptical with scales $s_{x,k}, s_{y,k} > 0$ and orientation φ_k; in rotated coordinates (x_r, y_r),

$$W_{ij} \mathrel{+}= \sum_{k=1}^{K} w_k \exp\Big(-\tfrac{1}{2}\big((x_r/s_{x,k})^2 + (y_r/s_{y,k})^2\big)\Big).$$

Parameter choice: $K = 10$; scales $s_{x,k}, s_{y,k} \in \left[\frac{m}{30}, \frac{m}{7}\right]$; orientations concentrated around $\varphi_0 = \frac{\pi}{4}$ with dispersion 0.25.

(3) Sparse Structure

Only a fraction dens $\in (0, 1)$ of cells carry substantial mass; remaining cells receive a negligible background level.
Parameter choice: dens $= 0.10$ (for $m = 40$: about $0.10 \cdot m^2 = 160$ active cells).

(4) Ring Mixture

Mass is concentrated on rings of target radius r_0 around multiple centers. With $r_{ijk} = \sqrt{(i-c_k^x)^2 + (j-c_k^y)^2}$ and ring width $\sigma > 0$,

$$W_{ij} \mathrel{+}= \sum_{k=1}^{K} w_k \exp\left(-\frac{(r_{ijk} - r_0)^2}{2\sigma^2}\right).$$

Parameter choice: $K = 6$, $r_0 = \frac{m}{4}$ (for $m = 40$: $r_0 = 10$), $\sigma = \frac{m}{35}$ (for $m = 40$: $\sigma \approx 1.14$).

B.5. Example Copula Density and its Series Representation

We consider

$$c(x,y) = 1 + \frac{\sin\big(2\pi(x+y)\big)}{5 - 4\cos\big(2\pi(x+y)\big)} + \frac{\sin\big(2\pi(x-y)\big)}{5 - 4\cos\big(2\pi(x-y)\big)}, \qquad (x,y) \in [0,1]^2.$$

For brevity set $h(\theta) := \dfrac{\sin\theta}{5 - 4\cos\theta}$. Then $c(x,y) = 1 + h\big(2\pi(x+y)\big) + h\big(2\pi(x-y)\big)$.

Since

$$h'(\theta) = \frac{5\cos\theta - 4}{\big(5 - 4\cos\theta\big)^2},$$

the extrema occur at $\cos\theta = 4/5$, where $\sin\theta = \pm 3/5$. Hence

$$h_{\max} = \frac{3/5}{5 - 16/5} = \frac{1}{3}, \qquad h_{\min} = -\frac{1}{3},$$

so $h \in [-1/3, 1/3]$ and consequently

$$c(x,y) \in \Big[1 - \tfrac{2}{3},\, 1 + \tfrac{2}{3}\Big] = \Big[\tfrac{1}{3},\, \tfrac{5}{3}\Big].$$

In particular, $c \geq 0$ on $[0,1]^2$.

Now let $g(z) := 5 - 4\cos(2\pi z)$. Then

$$\frac{d}{dz}\log g(z) = \frac{g'(z)}{g(z)} = \frac{8\pi\sin(2\pi z)}{5 - 4\cos(2\pi z)}.$$

Over one period,

$$\int_0^1 \frac{\sin(2\pi z)}{5 - 4\cos(2\pi z)}\, dz = \frac{1}{8\pi}\Big[\log g(z)\Big]_0^1 = 0.$$

Fix $x \in [0,1]$. By periodicity and the substitutions $z = x \pm y$,

$$\int_0^1 c(x,y)\, dy = 1 + \int_0^1 h\big(2\pi(x+y)\big)\, dy + \int_0^1 h\big(2\pi(x-y)\big)\, dy = 1,$$

and, by symmetry, $\int_0^1 c(x,y)\, dx = 1$. With the proven nonnegativity, c therefore defines a copula density.

Now using the complex geometric series, for $|r| < 1$,

$$\sum_{k=0}^{\infty} (re^{i\theta})^k = \frac{1}{1 - re^{i\theta}} \quad \Longrightarrow \quad \sum_{k=0}^{\infty} r^k \sin(k\theta) = \Im\Big(\frac{1}{1 - re^{i\theta}}\Big) = \frac{r\sin\theta}{1 - 2r\cos\theta + r^2}.$$

With $r = \frac{1}{2}$ we obtain

$$\frac{\sin\theta}{5-4\cos\theta} = \frac{1}{2}\sum_{k=1}^{\infty}\left(\tfrac{1}{2}\right)^k \sin(k\theta).$$

Apply this with $\theta = 2\pi(x+y)$ and $\theta = 2\pi(x-y)$, and use $\sin A+\sin B = 2\sin\frac{A+B}{2}\cos\frac{A-B}{2}$, to get

$$c(x,y) = 1 + \sum_{k=1}^{\infty}\left(\tfrac{1}{2}\right)^k \sin(2\pi kx)\cos(2\pi ky)\,.$$

B.6. Nonnegativity Despite Unbounded Component Functions

We consider, for $x \in (0,1]$ and $y \in [0,1]$, the functions

$$f_1(x) = \frac{2}{\sqrt{x}} - 4 + \frac{1}{4}\sin(2\pi x), \qquad f_2(x) = -\frac{1}{\sqrt{x}} + 2 - \frac{1}{8}\sin(2\pi x) + \frac{1}{8}\cos(2\pi x),$$

$$g_1(y) = y - \frac{1}{2}, \qquad g_2(y) = 2y - \frac{3}{2}.$$

By direct verification, f_1, f_2 as well as g_1, g_2 are linearly independent and each changes sign on its domain. Moreover, f_1 and f_2 are unbounded below on $(0,1]$ due to the terms $\pm x^{-1/2}$ as $x \downarrow 0$.

The expression

$$1 + f_1(x)\,g_1(y) + f_2(x)\,g_2(y)$$

is affine in y and can be rewritten as

$$1 + f_1(x)\,g_1(y) + f_2(x)\,g_2(y) = y\big(f_1(x) + 2f_2(x)\big) + \Big[1 - \tfrac{1}{2}f_1(x) - \tfrac{3}{2}f_2(x)\Big],$$

with

$$f_1(x) + 2f_2(x) = \frac{1}{4}\cos(2\pi x).$$

Hence, for fixed x, the minimum over $y \in [0,1]$ is always attained at a boundary point:

$$\min_{y\in[0,1]}\big(1 + f_1 g_1 + f_2 g_2\big) = \begin{cases} 1 - \frac{1}{2}f_1(x) - \frac{3}{2}f_2(x), & \text{if } \cos(2\pi x) \geq 0 \text{ (minimum at } y = 0), \\ 1 + \frac{1}{2}f_1(x) + \frac{1}{2}f_2(x), & \text{if } \cos(2\pi x) < 0 \text{ (minimum at } y = 1). \end{cases}$$

A direct estimate of these boundary values gives a positive lower bound in both cases. Using $a\sin t + b\cos t \geq -\sqrt{a^2+b^2}$ for all t,

$$\text{if } \cos(2\pi x) \geq 0: \quad 1 - \tfrac{1}{2}f_1(x) - \tfrac{3}{2}f_2(x) = \frac{1}{2\sqrt{x}} + \frac{1}{16}\sin(2\pi x) - \frac{3}{16}\cos(2\pi x)$$
$$\geq \frac{1}{2} - \frac{\sqrt{10}}{16} > 0,$$

$$\text{if } \cos(2\pi x) < 0: \quad 1 + \tfrac{1}{2}f_1(x) + \tfrac{1}{2}f_2(x) = \frac{1}{2\sqrt{x}} + \frac{1}{16}\big(\sin(2\pi x) + \cos(2\pi x)\big)$$
$$\geq \frac{1}{2} - \frac{\sqrt{2}}{16} > 0.$$

Consequently,

$$1 + f_1(x)\,g_1(y) + f_2(x)\,g_2(y) \geq 0 \qquad \text{for all } (x,y) \in (0,1] \times [0,1].$$

C. Appendix to Chapter 5

C.1. Proof of Theorem 5.2.1

Proof. By assumption, $\mathcal{V}_p$ and $\mathcal{W}_q$ are finite-dimensional subspaces of $L_0^2([0,1])$ spanned by bounded and continuous functions. Without loss of generality, let

$$\mathcal{V}_p = \operatorname{span}\{\varphi_1, \dots, \varphi_p\}, \qquad \mathcal{W}_q = \operatorname{span}\{\psi_1, \dots, \psi_q\}$$

where the bases are orthonormal in $L^2([0,1])$.

Since $r = \min(p,q)$ in Problem 5.2.1, any function in the tensor-product space $\mathcal{V}_p \otimes \mathcal{W}_q$ can be represented in coefficient form by a matrix $K \in \mathbb{R}^{p\times q}$ via

$$c_K(u,v) := 1 + \Phi(u)^\top K \Psi(v),$$

where

$$\Phi(u) := (\varphi_1(u), \dots, \varphi_p(u))^\top, \qquad \Psi(v) := (\psi_1(v), \dots, \psi_q(v))^\top.$$

Conversely, every $K \in \mathbb{R}^{p\times q}$ defines such a density in $1 + \mathcal{V}_p \otimes \mathcal{W}_q$. Hence, optimizing over the families $\{f_k\}_{k=1}^r \subset \mathcal{V}_p$ and $\{g_k\}_{k=1}^r \subset \mathcal{W}_q$ is equivalent to optimizing over $K \in \mathbb{R}^{p\times q}$.

By Proposition 5.2.1, the squared L^2 objective satisfies

$$\mathcal{J}(K) = \|c^\sharp - 1\|_{L^2}^2 - \|S\|_F^2 + \|K - S\|_F^2,$$

where $S \in \mathbb{R}^{p\times q}$ is the coefficient matrix defined in (5.4). Thus, minimizing $\mathcal{J}$ over feasible expansions is equivalent to minimizing the continuous function $K \mapsto \|K - S\|_F^2$.

Define the feasible set in coefficient form by

$$\mathcal{F}_K := \Big\{K \in \mathbb{R}^{p\times q} \;:\; 1 + \Phi(u)^\top K \Psi(v) \geq 0 \;\; \forall (u,v) \in [0,1]^2\Big\}.$$

For each fixed (u,v) the map $K \mapsto 1 + \Phi(u)^\top K \Psi(v)$ is linear and hence continuous. Therefore the set $\{K \;:\; 1 + \Phi(u)^\top K \Psi(v) \geq 0\}$ is a closed half-space. As $\mathcal{F}_K$ is an intersection of such closed half-spaces over $(u,v) \in [0,1]^2$, it is closed. Moreover, $\mathcal{F}_K$ is nonempty since $K = 0$ yields $c_K \equiv 1$.

Let $(K_n)_{n\in\mathbb{N}} \subset \mathcal{F}_K$ be a minimizing sequence, i.e., $\|K_n - S\|_F^2 \to \inf_{K\in\mathcal{F}_K} \|K - S\|_F^2$. Since $K \mapsto \|K - S\|_F^2$ is coercive, the sequence (K_n) is bounded. By the Bolzano–Weierstrass theorem in $\mathbb{R}^{p\times q}$, there exists a convergent subsequence $K_{n_j} \to K^\star$. Since $\mathcal{F}_K$ is closed, $K^\star \in \mathcal{F}_K$. By continuity of the objective, $\|K^\star - S\|_F^2 = \inf_{K\in\mathcal{F}_K} \|K - S\|_F^2$, so the minimum is attained at $K^\star$. Since $\mathcal{F}_K$ is convex (as an intersection of half-spaces) and the map $K \mapsto \|K - S\|_F^2$ is strictly convex, this minimizer is unique.

Finally, let $K^\star = U\Sigma V^\top$ be the compact singular value decomposition with $r_{\text{eff}} := \operatorname{rank}(K^\star) \le \min(p, q) = r$ and singular values $\sigma_1, \dots, \sigma_{r_{\text{eff}}} > 0$. Then

$$K^\star = \sum_{\ell=1}^{r_{\text{eff}}} \sigma_\ell \, U_{:,\ell} V_{:,\ell}^\top,$$

and therefore

$$c_{K^\star}(u, v) - 1 = \sum_{\ell=1}^{r_{\text{eff}}} \big(\Phi(u)^\top U_{:,\ell}\big)\, \sigma_\ell \, \big(V_{:,\ell}^\top \Psi(v)\big).$$

Thus $c_{K^\star} - 1$ admits a separable representation with at most r terms:

$$c_{K^\star}(u, v) - 1 = \sum_{\ell=1}^{r_{\text{eff}}} \sigma_\ell f_\ell(u) g_\ell(v),$$

where

$$f_\ell(u) := \Phi(u)^\top U_{:,\ell}, \qquad g_\ell(v) := \Psi(v)^\top V_{:,\ell}, \qquad \ell = 1, \dots, r_{\text{eff}}.$$

This yields an optimal feasible representation for Problem 5.2.1. □

C.2. Proof of Theorem 5.2.3

Proof. We first establish that the sequence $(K_\ell^*)_{\ell\in\mathbb{N}}$ is uniformly bounded. Since $c_0 \equiv 1 \ge 0$, the matrix $K = 0$ is feasible for all discrete problems. By optimality of K_ℓ^*,

$$\|K_\ell^* - S\|_F \le \|0 - S\|_F = \|S\|_F \quad \text{for all } \ell \in \mathbb{N}.$$

The triangle inequality then yields

$$\|K_\ell^*\|_F \le \|K_\ell^* - S\|_F + \|S\|_F \le 2\|S\|_F,$$

so the sequence is uniformly bounded. Since $\mathbb{R}^{p\times q}$ is finite-dimensional, there exists a subsequence (not relabeled) and a matrix $\widetilde{K} \in \mathbb{R}^{p\times q}$ such that

$$K_\ell^* \to \widetilde{K} \quad \text{as } \ell \to \infty.$$

We next establish convergence of the corresponding densities and then verify feasibility and optimality of the limit.

For any coefficient matrix $K \in \mathbb{R}^{p\times q}$ and $(u, v) \in [0,1]^2$ we have

$$c_K(u,v) = 1 + \Phi(u)^\top K \Psi(v),$$

where $\Phi(u) = (\varphi_1(u), \dots, \varphi_p(u))^\top$ and $\Psi(v) = (\psi_1(v), \dots, \psi_q(v))^\top$. Since $\varphi_\mu, \psi_\nu \in C([0,1])$ and $[0,1]$ is compact, all basis functions are bounded. Hence there exists $M < \infty$ such that $\|\Phi(u)\|_2 \le M$ and $\|\Psi(v)\|_2 \le M$ for all $u, v \in [0,1]$.

For the difference of two densities we obtain

$$|c_{K_\ell^*}(u,v) - c_{\widetilde{K}}(u,v)| = \left|\Phi(u)^\top (K_\ell^* - \widetilde{K})\Psi(v)\right| \le \|\Phi(u)\|_2 \, \|K_\ell^* - \widetilde{K}\|_F \, \|\Psi(v)\|_2 \le M^2 \|K_\ell^* - \widetilde{K}\|_F.$$

Taking the supremum over $(u, v) \in [0,1]^2$ yields

$$\|c_{K_\ell^*} - c_{\widetilde{K}}\|_{L^\infty([0,1]^2)} \le M^2 \|K_\ell^* - \widetilde{K}\|_F \xrightarrow[\ell\to\infty]{} 0.$$

In particular, the densities converge uniformly and hence also in L^2,

$$c_{K_\ell^*} \to c_{\widetilde{K}} \quad \text{in } L^2([0,1]^2).$$

We next show that $\widetilde{K}$ satisfies the nonnegativity constraint of the continuous problem (5.6). By construction of K_ℓ^* as the solution of the ℓ-th discrete problem, we have

$$c_{K_\ell^*}(x_i^{(\ell)}, y_j^{(\ell)}) \ge 0 \quad \text{for all } (x_i^{(\ell)}, y_j^{(\ell)}) \in \mathcal{G}_\ell.$$

Suppose for contradiction that $\widetilde{K}$ were not feasible for (5.6). Then there would exist $(u_0, v_0) \in [0,1]^2$ such that

$$c_{\widetilde{K}}(u_0, v_0) < 0.$$

By continuity of $(u, v) \mapsto c_{\widetilde{K}}(u, v)$, there exist $\varepsilon > 0$ and an open neighborhood $U \subset [0,1]^2$ of (u_0, v_0) such that

$$c_{\widetilde{K}}(u, v) \le -2\varepsilon \quad \text{for all } (u, v) \in U.$$

Uniform convergence implies the existence of $L \in \mathbb{N}$ with

$$\|c_{K_\ell^*} - c_{\widetilde{K}}\|_{L^\infty([0,1]^2)} < \varepsilon \quad \text{for all } \ell \geq L,$$

so for all $(u, v) \in U$ and $\ell \geq L$,

$$c_{K_\ell^*}(u, v) \leq c_{\widetilde{K}}(u, v) + \varepsilon \leq -\varepsilon.$$

Since $\mathcal{G}_\ell$ becomes dense in $[0, 1]^2$, for all sufficiently large ℓ there exists a collocation point in U, hence

$$c_{K_\ell^*}(x_i^{(\ell)}, y_j^{(\ell)}) < 0,$$

contradicting feasibility. Therefore,

$$c_{\widetilde{K}}(u, v) \geq 0 \quad \text{for all } (u, v) \in [0, 1]^2,$$

so $\widetilde{K}$ is feasible for (5.6).

It remains to establish optimality of $\widetilde{K}$. Let

$$\mathcal{F}_{\text{cont}} := \{K \in \mathbb{R}^{p \times q} : c_K(u, v) \geq 0 \; \forall (u, v) \in [0, 1]^2\}$$

denote the feasible set of the continuous problem. The preceding argument shows that $\widetilde{K} \in \mathcal{F}_{\text{cont}}$. Let $\bar{K} \in \mathcal{F}_{\text{cont}}$ be a minimizer of the continuous problem, i.e.

$$\|\bar{K} - S\|_F^2 = \min_{K \in \mathcal{F}_{\text{cont}}} \|K - S\|_F^2.$$

For each ℓ, the feasible set of the discrete problem is

$$\mathcal{F}_\ell := \{K \in \mathbb{R}^{p \times q} : c_K(x_i^{(\ell)}, y_j^{(\ell)}) \geq 0 \; \forall (x_i^{(\ell)}, y_j^{(\ell)}) \in \mathcal{G}_\ell\}.$$

Since $\mathcal{G}_\ell \subset [0, 1]^2$ and $c_{\bar{K}} \geq 0$ on $[0, 1]^2$, we have $\bar{K} \in \mathcal{F}_\ell$ for all ℓ. By optimality of K_ℓ^*,

$$\|K_\ell^* - S\|_F^2 \leq \|\bar{K} - S\|_F^2 \quad \text{for all } \ell \in \mathbb{N}.$$

Passing to the limit along the convergent subsequence yields

$$\|\widetilde{K} - S\|_F^2 = \lim_{\ell \to \infty} \|K_\ell^* - S\|_F^2 \leq \|\bar{K} - S\|_F^2,$$

so $\widetilde{K}$ is also a minimizer of the continuous problem.

It remains to show that the entire sequence (K_ℓ^*) converges, not just a subsequence. By Theorem 5.2.1, the continuous problem (5.6) admits a unique minimizer, say $K^\star$. Since

(K_ℓ^*) is bounded, every subsequence has a further convergent subsequence, and any such limit must equal $K^\star$. Thus $K^\star$ is the unique cluster point of (K_ℓ^*), which implies $K_\ell^* \to K^\star$.

Finally, since the map $K \mapsto c_K$ is continuous from $\mathbb{R}^{p\times q}$ to $L^2([0,1]^2)$ by the estimate above, we conclude

$$c_{K_\ell^*} \to c_{K^\star} \quad \text{in } L^2([0,1]^2),$$

as asserted. □

C.3. Basis Constructions in Empirical Study

This appendix describes the three one-dimensional basis families used in the experiments. The constructions are presented for a generic dimension p and a generic grid partition of size N. In the context of the bivariate problem in Section 5.2.1, these definitions apply to the u-axis (with $N = m$, basis dimension p, yielding matrix A) and analogously to the v-axis (with $N = n$, basis dimension q, yielding matrix B).

Throughout, functions are defined on $[0,1]$, inner products are with respect to the Lebesgue measure, and "mean-free" means zero integral over $[0,1]$. We denote the N grid cells by $\{I_i = [a_i, b_i]\}_{i=1}^N$ and the p basis functions by $\{\varphi_k\}_{k=1}^p$.

C.3.1. Shifted Legendre basis (polynomial model)

Let P_k denote the k-th Legendre polynomial on $[-1,1]$. Define the shifted, mean-free, L^2-orthonormal basis by

$$\varphi_k(x) := \sqrt{2k+1}\, P_k(2x-1), \qquad k = 1, 2, \dots, p.$$

Since P_k is orthogonal to $P_0 \equiv 1$ for $k \geq 1$, it holds that $\int_0^1 \varphi_k(x)\,dx = 0$. The orthonormality follows from the scaling. Cell integrals needed for the matrix $A \in \mathbb{R}^{N\times p}$ are available in closed form via the primitive of P_k:

$$A_{ik} = \int_{I_i} \varphi_k(x)\,dx = \frac{\sqrt{2k+1}}{2}\Big[\mathcal{P}_k(2x-1)\Big]_{x=a_i}^{x=b_i},$$

where $I_i = [a_i, b_i]$ and $\mathcal{P}_k$ is any antiderivative of P_k.

C.3.2. Cosine basis (trigonometric model)

Define the mean-free, L^2-orthonormal cosine basis by

$$\varphi_k(x) := \sqrt{2}\,\cos(\pi k x), \qquad k = 1, 2, \dots, p.$$

For $k \geq 1$, the integral over $[0, 1]$ vanishes: $\int_0^1 \varphi_k(x)\,dx = 0$. The orthonormality condition $\int_0^1 \varphi_k \varphi_{k'} = \delta_{kk'}$ holds for the proposed scaling. The cell integrals have the explicit form

$$A_{ik} = \int_{a_i}^{b_i} \sqrt{2}\,\cos(\pi k x)\,dx = \frac{\sqrt{2}}{\pi k}\Big[\sin(\pi k b_i) - \sin(\pi k a_i)\Big].$$

C.3.3. Cubic B-spline basis (spline model)

Let $T = (t_0, \dots, t_{p+4})$ be an open, uniform (clamped) knot vector on $[0, 1]$ and let $\{N_j\}_{j=1}^{p+1}$ be the associated cubic B-splines. Note that we initially construct $p + 1$ functions to compensate for the degree of freedom removed by the zero-mean constraint. These functions are nonnegative, locally supported, and form a partition of unity (Boor, 1978, Chapter 4). To obtain a mean-free, L^2-orthonormal family of size p, proceed as follows.

First, assemble the Gram matrix $G \in \mathbb{R}^{(p+1)\times(p+1)}$ with

$$G_{jj'} = \int_0^1 N_j(x)\,N_{j'}(x)\,dx,$$

compute a Cholesky factorization $G = R^\top R$ (Golub and Loan, 1996, Sec. 4.2), and define intermediate orthonormal functions

$$\phi_j(x) := \sum_{l=1}^{p+1} N_l(x)\,(R^{-1})_{lj}, \qquad j = 1, \dots, p+1.$$

Second, to enforce the zero-mean property, form the vector $g \in \mathbb{R}^{p+1}$ with entries $g_j = \int_0^1 \phi_j(x)\,dx$. Construct an orthogonal matrix $Q \in \mathbb{R}^{(p+1)\times(p+1)}$ whose first column is parallel to g (e.g., via a Householder reflector mapping g to $\|g\|_2 e_1$; see Golub and Loan, 1996, Section 5.1–5.2). The basis functions are then defined by rotating the constant part into the first component and discarding it:

$$\tilde{\phi}_k(x) := \sum_{j=1}^{p+1} \phi_j(x)\,Q_{j,k+1}, \qquad k = 1, \dots, p.$$

By construction, $\int_0^1 \tilde{\phi}_k(x)\,dx = 0$ and the system remains orthonormal. Finally, we set $\varphi_k := \tilde{\phi}_k$ for $k = 1, \dots, p$.

For the matrix $A \in \mathbb{R}^{N\times p}$, the entries

$$A_{ik} = \int_{I_i} \varphi_k(x)\,dx$$

are computed by Gauss-Legendre quadrature (Atkinson, 1989, Section 5.3) on each cell I_i.

C.4. Credit Rating Transition Table

Table C.1 shows the original one-year transition table from Kulo and Poulain (2025) used in Section 5.4.1. The table tracks 995 credit ratings over the period from December 31, 2023 to December 31, 2024. Each row corresponds to a rating class at the beginning of the period, each column to a rating class at the end, and the entries indicate the percentage of issuers transitioning between classes. The rightmost columns record withdrawn ratings (WR), paid-off obligations, and defaults; the final column shows the total number of issuers in each initial rating class. For our analysis, we exclude the WR and paid-off columns and renormalize each row to obtain a proper transition probability matrix.

	2024												
2023	AAA	AA	A	BBB	BB	B	CCC	CC	C	WR	Paid-off	Default	No. of ratings
AAA	87%	3%	0%	0%	0%	0%	0%	0%	0%	3%	7%	0%	130
AA	1%	76%	5%	0%	0%	0%	0%	0%	0%	14%	4%	0%	84
A	0%	2%	79%	1%	0%	0%	0%	0%	0%	15%	3%	0%	274
BBB	0%	0%	4%	77%	2%	0%	0%	0%	0%	13%	4%	0%	337
BB	0%	0%	0%	3%	86%	4%	1%	1%	0%	4%	1%	0%	80
B	0%	0%	0%	0%	5%	78%	5%	0%	2%	3%	5%	3%	64
CCC	0%	0%	0%	0%	0%	0%	58%	25%	0%	0%	8%	8%	12
CC	0%	0%	0%	0%	0%	0%	0%	67%	17%	0%	0%	17%	6
C	0%	0%	0%	0%	0%	0%	0%	0%	100%	0%	0%	0%	8

Table C.1.: One-year credit rating transition table (December 31, 2023 to December 31, 2024). Entries are percentages; rows sum to 100%.

C.5. American Community Survey 2023 Age-Income Table

Table C.2 shows the age-by-income contingency table from the 2023 American Community Survey (U.S. Census Bureau, 2024) used in Section 5.4.2. Each row corresponds to an income bracket (16 categories ranging from "Less than $10,000" to "$200,000 or more"), each column to an age group based on the householder's age (4 categories), and the entries represent household counts. The table covers approximately 134 million U.S. households. For our analysis, we normalize the entries to obtain a joint probability distribution.

Income	Under 25	25–44	45–64	65+	Total
Less than $10,000	669,553	1,886,075	2,202,148	2,205,797	6,963,573
$10,000 to $14,999	263,407	885,159	1,373,446	2,056,683	4,578,695
$15,000 to $19,999	233,790	740,615	1,027,414	1,910,111	3,911,930
$20,000 to $24,999	295,536	1,012,227	1,196,482	2,019,144	4,523,389
$25,000 to $29,999	260,227	994,228	1,075,926	1,822,729	4,153,110
$30,000 to $34,999	324,877	1,379,801	1,266,354	1,747,215	4,718,247
$35,000 to $39,999	277,487	1,277,323	1,245,328	1,682,879	4,483,017
$40,000 to $44,999	298,772	1,532,358	1,355,953	1,594,372	4,781,455
$45,000 to $49,999	247,781	1,331,890	1,248,318	1,480,710	4,308,699
$50,000 to $59,999	454,148	2,942,273	2,649,018	2,738,426	8,783,865
$60,000 to $74,999	589,137	4,366,305	3,922,315	3,469,643	12,347,400
$75,000 to $99,999	638,157	6,078,421	5,674,421	4,291,686	16,682,685
$100,000 to $124,999	320,712	5,190,879	5,011,852	2,900,502	13,423,945
$125,000 to $149,999	158,351	3,632,945	3,795,062	1,874,211	9,460,569
$150,000 to $199,999	117,445	4,514,415	5,228,226	2,080,925	11,941,011
$200,000 or more	84,499	5,577,264	8,003,677	2,605,330	16,270,770
Total	4,433,879	43,342,178	43,875,940	42,680,363	134,332,360

Table C.2.: U.S. household counts by income and age of householder (ACS 2023)

D. Appendix to Chapter 6

D.1. Kernels

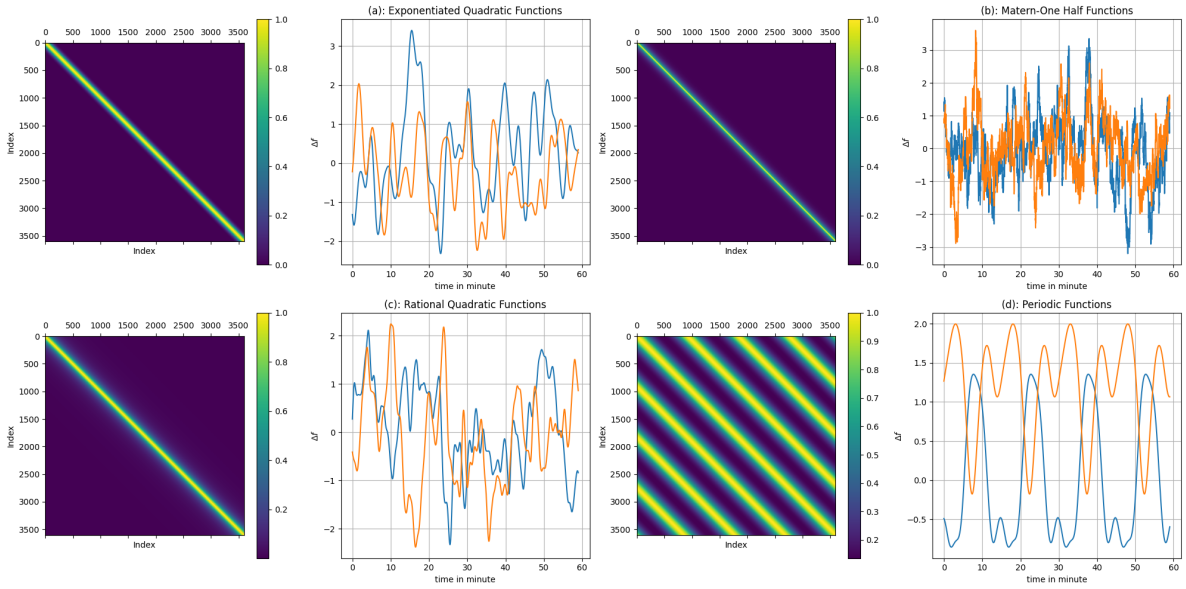


Figure D.1.: Kernels and two generated synthetic time series in each case.

We show here four different kernels and two generated synthetic time series in each case (see Fig. D.1). The amplitude and the length scale are set to one for all kernels shown. For the periodic kernel, the period parameter is set to 15.

D.2. Loss Functions for Fat-Tail Marginal Distributions

The Student's t-distribution with ν degrees of freedom has the density

$$f(t) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\,\Gamma\left(\frac{\nu}{2}\right)}\left(1+\frac{t^2}{\nu}\right)^{-\frac{\nu+1}{2}},$$

where Γ is the gamma function. For ν greater than 30, the Student's t-distribution can be approximated by the normal distribution. This means with the assumption of the Student's t-distribution, the model is still capable of learning a Gaussian process-like process because ν is learnable. In particular, we learn for each time point t_n a location-scale Student's t-distribution $a_n \cdot \mathcal{T} + b_n$ to account for different mean positions and scattering behavior.

The Cauchy distribution is a special case of the Student's t-distribution with $\nu = 1$. Note that the Cauchy distribution has neither an expected value nor a variance (DasGupta, 2010). Therefore, for a stochastic process with marginal Cauchy distributions, we focus on learning the median and interquartile range.

For the true value $y \in \mathbb{R}^N$, the loss function using the likelihood function can then be calculated for the process with Student's t-distributions

$$\begin{aligned}\mathcal{L}_{\mathrm{t}}(y, a, b, \nu) = -\frac{1}{N}\sum_{n=0}^{N-1}\Bigg(&\log\Gamma\left(\frac{\nu_n+1}{2}\right) - \log\Gamma\left(\frac{\nu_n}{2}\right) - \frac{1}{2}\log(\nu_n\pi) \\ &- \log(a_n) - \frac{\nu_n+1}{2}\log\left(1 + \frac{(y_n-b_n)^2}{\nu_n a_n^2}\right)\Bigg),\end{aligned}$$

and for a process with Cauchy distributions

$$\mathcal{L}_{\mathrm{Cauchy}}(y, m, \gamma) = \frac{1}{N}\sum_{n=0}^{N-1}\left[\log(\pi\gamma_n) + \log\left(1 + \left(\frac{y_n - m_n}{\gamma_n}\right)^2\right)\right],$$

where the median m_n is the position parameter and γ_n is the half-width at time point t_n.

D.3. Data

For the models, we consider hourly recorded external techno-economic features and the associated grid frequency data in a temporal resolution of seconds in Continental Europe between 2015-2019. An overview of the features is shown in Tab. D.1. In particular, we used the cleansed data from Kruse et al. (2021b) and Kruse et al. (2021a). The raw feature data of the cleansed data are from the ENTSO-E Transparency Platform and the raw frequency data are from TransnetBW GmbH (Kruse et al., 2021a). In addition, we used the code from Kruse (2023) to generate training and test data to then ensure comparability with results from the physics-informed machine learning models of Kruse et al. (2023).

Type	**Feature**	Unit
Extern	Load day-ahead	MW
Extern	Solar day-ahead	MW
Extern	Offshore wind day-ahead	MW
Extern	Onshore wind day-ahead	MW
Extern	Load ramp day-ahead	MW/h
Extern	Generation ramp day-ahead	MW/h
Extern	Solar ramp day-ahead	MW/h
Extern	Offshore wind ramp day-ahead	MW/h
Extern	Onshore wind ramp day-ahead	MW/h
Extern	Price day-ahead	EUR/MWh
Extern	Price ramp day-ahead	EUR/MWh/h
Time	$\cos(\pi/12 \times \text{hour})$	-
Time	$\sin(\pi/12 \times \text{hour})$	-
Initial value	Grid frequency deviation at the beginning of the hour	Hz

Table D.1.: Overview of Features.

D.4. Details on Model Structures

Layer (type)	Output Shape
Input Layer	[(None, 14)]
Repeat Vector	(None, 180, 14)
GRU	(None, 128)
Dropout	(None, 128)
6 fully connected layers	(None, 128)
Dropout	(None, 128)
Output Layer	(None, 7200)

Table D.2.: Model structure of a GRU-based model for independent Gaussian process.

The model structure for models with Gaussian process and GRU is shown in Tab. D.2. The structure of the models based on transformer can be found in Tab. D.3. We always use the same structures for different processes (whether Gaussian or fat-tail marginal distribution, or whether the time points are dependent) and only exchange the loss functions for training and learning the concrete parameters of the models (and thus the output shape). This also shows the flexibility of our approach of using sequence models. For hidden layers, we use the ReLU function as the activation function. For output layers, we usually use the linear activation function. To take into account the properties of variances or the scaling parameters of kernels, we also use the softplus function in the loss function to guarantee their positivity.

D.5. Baseline Models

An overview and descriptions of the baseline models used can be found in Tab. D.4. In particular, when implementing the KNN profile, we search the feature space of the training

Layer (type)	Output Shape
Input Layer	[(None, 14)]
Initial Dense with 3 layers	(None, 448)
Reshape	(None, 14,32)
Three dense nets for query, key and value, each with three hidden layers of 64 neurons	3 times (None, 128)
multi head attention	(None, 14,32)
Add	(None, 14, 32)
LayerNormalization	(None, 14, 32)
Flatten	(None, 448)
6 fully connected layers	(None, 128)
Dropout	(None, 128)
Output Layer	(None, 480)

Table D.3.: Model structure of a transformer-based model for a correlated Gaussian process (with time step = 15 s).

data for the features that have the smallest distance to the feature of the current time interval to be predicted and then calculate the prediction as a weighted sum of the frequency series for the given feature vectors. To optimize the hyperparameters, cross-validation is performed to determine the number of neighbors k. A KNN regressor is instantiated and trained with the optimal k value. For the implementation we use the "KNeighborsRegressor" from the "sklearn.neighbors" library. For the data generation of test data using the PIML day-ahead model, PIML ex-post model, constant profile and daily profile, we use the code provided by Kruse (2023). In addition, we include all means of all probabilistic models as point predictors.

D.6. Evaluation Measures

A point predictor $\bar{y}$ of the true value y for n predictions could be evaluated with MAE (mean absolute error), MSE (mean squared error) and RMSE (root mean squared error)

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \bar{y}_i|$$

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y}_i)^2$$

$$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y}_i)^2}.$$

Type	Name	Description
point	constant zero	constant frequency deviation with 0 Hz
point	global mean	global mean of the whole frequency data
point	KNN profile	prediction based on the feature distance of historical data
point	begin value profile	predictions for frequency values of a one-hour interval is the frequency value at beginning of the time interval
probabilistic	constant profile	Gaussian distribution with the global mean and variance of the whole frequency data
probabilistic	daily profile	the prediction for a time of day t is equal to a normal distribution with the mean and standard deviation of all training data at this time of day
probabilistic	PIML day-ahead	physics-informed machine learning model using day-ahead features from Kruse et al. (2023)
probabilistic	PIML ex-post	physics-informed machine learning model using ex-post features from Kruse et al. (2023)
point	mean daily profile	mean of daily profile
point	mean PIML day-ahead	mean of PIML day-ahead model
point	mean PIML ex-post	mean of PIML ex-post model

Table D.4.: Overview of baseline models.

For a one dimensional probabilistic predictor $\hat{Y}$ for Y with the distribution function F, density function f and a realized value y, one could use the negative log likelihood and CRPS (Continuous Ranked Probability Score) (Jordan et al., 2019; Gneiting and Raftery, 2007)

$$\log \mathcal{L}(y) = -\log f(y)$$

$$CRPS(F, y) = \mathbb{E}_F |Y_1 - y| - \frac{1}{2}\mathbb{E} |Y_1 - Y_2| ,$$

where Y_1 and Y_2 are i.i.d. random variables with distribution F. There is an explicit formula for a normal distribution (Gneiting and Raftery, 2007)

$$CRPS(N(\mu, \sigma^2), x) = \sigma \left[\frac{x-\mu}{\sigma} \left(2\Phi \left(\frac{x-\mu}{\sigma} \right) - 1 \right) + 2\phi \left(\frac{x-\mu}{\sigma} \right) - \frac{1}{\sqrt{\pi}} \right],$$

where ϕ and Φ denote the probability density function and the cumulative distribution function of a standard Gaussian variable. We used the explicit formula to implement CRPS for our study. In the case of a multidimensional probabilistic predictor $\hat{Y}$, the negative log likelihood could be used, too. In addition, the energy score can be used as a generalised form of CRPS to evaluate the performance of $\hat{Y}$

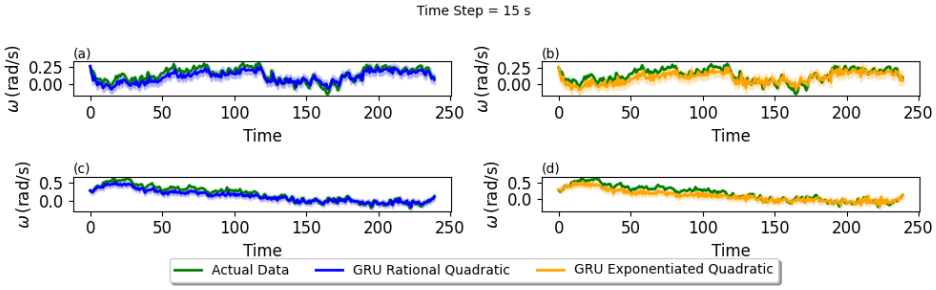


Figure D.2.: Conditional prediction with time step = 15s. We use here the GRU Structure. The distribution of the next point in time is predicted based on the realized values. The predictions can map the trend of the dynamic development of the frequency deviation very well.

$$ES(F, \hat{y}) = \mathbb{E}\|Y - \hat{y}\| - \frac{1}{2}\mathbb{E}\|Y - Y'\|,$$

where Y and Y' are independent identically distributed random variables from the distribution F, and $\|\cdot\|$ represents the Euclidean norm.

Like Jordan et al. (2019), we also use the Monte Carlo method to approximate the energy score by sampling for Y and Y' and then taking the empirical mean from the above expression.

For our study, we calculate the above measures for each data point in the test data and then use the median of the values as the final evaluation measure.

D.7. Conditional Forecasting

In the following, we provide conditional forecasting examples. Since the time points are now dependent, we cannot simply draw the mean function and the fill lines to calculate prediction examples. Instead, we draw the next unknown time point with realized time points using the learned correlations and draw the enveloping line using the conditional standard deviation. We repeat this process for the worst cases in Fig. 6.4 (b) and (d), where the independent assumptions are obviously not sufficient.

Since the entire distribution of all time points is subject to a Gaussian process, this conditional distribution $\Delta f_{n+1}|\Delta f_n, \dots, \Delta f_0$ is also a normal distribution $\mathcal{N}(\mu_{n+1|0:n}, \sigma^2_{n+1|0:n})$ with

$$\mu_{n+1|0:n} = \mu_{n+1} + \Sigma_{n+1,0:n}\Sigma^{-1}_{0:n,0:n}(\mathbf{x}_{0:n} - \mu_{0:n})$$

and

$$\sigma^2_{n+1|0:n} = \Sigma_{n+1,n+1} - \Sigma_{n+1,0:n}\Sigma^{-1}_{0:n,0:n}\Sigma_{0:n,n+1},$$

Model	MAE	MSE	RMSE
Constant zero	0.100163	0.016221	0.123233
Global mean	0.100212	0.016226	0.123266
Initial value	0.151541	0.036620	0.189428
Daily profile (mean)	0.081699	0.010686	0.102458
KNN profile	0.078300	0.009728	0.098325
PIML day-ahead (mean)	0.079684	0.010182	0.100159
PIML ex-post (mean)	0.080332	0.010224	0.100493
Independent Gaussian, GRU (mean)	0.075965	0.009120	0.095231
Independent Gaussian, transformer (mean)	0.076791	0.009349	0.096402
Cauchy, transformer (median)	0.077286	0.009513	0.097261
Student's t, transformer (mean)	0.076565	0.009304	0.096163

Table D.5.: Evaluation results for point predictions for one-hour intervals.

Model	Neg. Log-Likelihood	CRPS
Constant profile	−2659.64	0.0708
Daily profile	−3411.74	0.0575
PIML day-ahead	−3440.49	0.0563
PIML ex-post	−3417.16	0.0567
Independent Gaussian, GRU	−3575.84	0.0536
Independent Gaussian, transformer	−3549.58	0.0542

Table D.6.: Evaluation results for probabilistic predictions for one-hour intervals.

where $\Sigma_{n+1,0:n} \in \mathbb{R}^{1\times(n+1)}$ denotes the covariance vector between time point t_{n+1} and all previous time points, $\Sigma_{0:n,0:n} \in \mathbb{R}^{(n+1)\times(n+1)}$ is the covariance matrix of all observed time points, and $\Sigma_{0:n,n+1} = \Sigma_{n+1,0:n}^T$.

That is, with realized values $x_0, x_1, \dots, x_n$, the conditional distribution of x_{n+1} can then be predicted as a Gaussian distribution $\mathcal{N}(\mu_{n+1|0:n}, \sigma^2_{n+1|0:n})$.

The prediction samples in Fig. D.2 illustrate that our models, taking into account the correlations, can describe the evolution of the frequency deviation accurately.

D.8. Further Results On Prediction Performance and Synthetic Data

D.8.1. Prediction Performance on One-hour Intervals

The models for point and probabilistic forecasts are also for hourly intervals (3600 seconds) evaluated (see Tab. D.5 and D.6). Again, we can observe that the sequence model based models are better than the other baseline models.

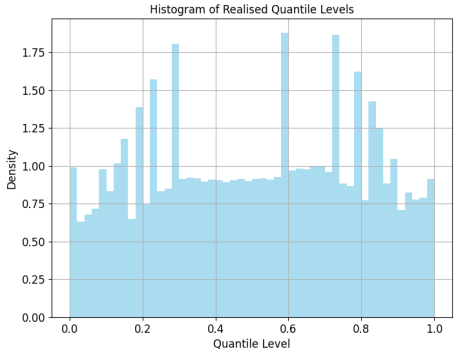
D.8.2. Histogram of Realized Quantiles

The Fig. D.3 shows a histogram of quantile levels for each probability model based on an independent Gaussian distribution. The quantile levels are calculated from the cumulative distribution function of the normal distribution for each true value using the estimated means and standard deviations for each time point. Note that the quantile level of a true value indicates the percentage of the distribution below which that value falls. Here, we have independent time points. In an ideally calibrated model, due to the nature of the probability integral transformation, the histogram of realized quantile levels should have an approximately uniform distribution. The Fig. D.3 show that all models clearly outperform the constant profile and that the calibration quality of our sequence-model-based Gaussian processes is comparable to the two PIML models.

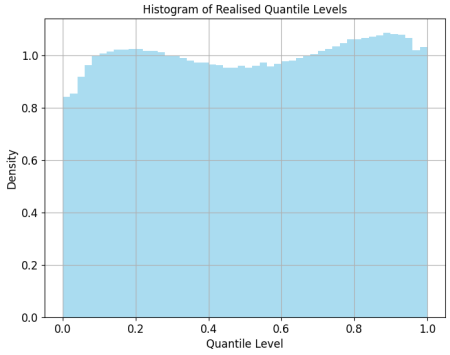
D.8.3. Autocorrelation of Synthetic Data

Here, we show the ACFs of the synthetically generated data using different models (see Fig. D.4, D.5 and D.6). While good agreement can generally be observed for both variables for all models, there is a clear difference in the ACF values of the increments at lag 2 (30s) between the models with exponentiated quadratic and rational quadratic kernels.

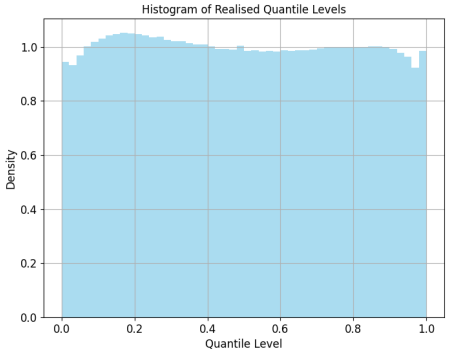
(a) constant profile



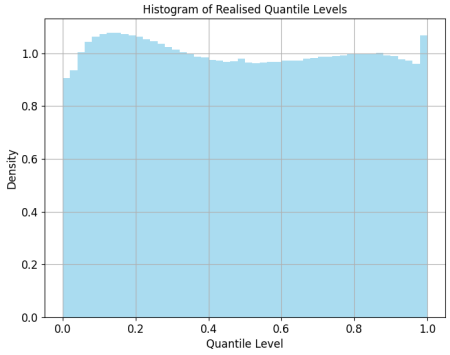
(b) daily profile



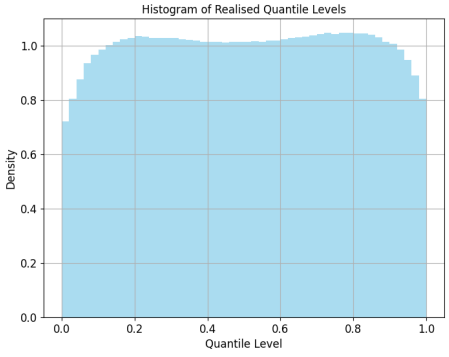
(c) PIML day-ahead



(d) PIML ex-post



(e) independent Gaussian GRU



(f) independent Gaussian transformer

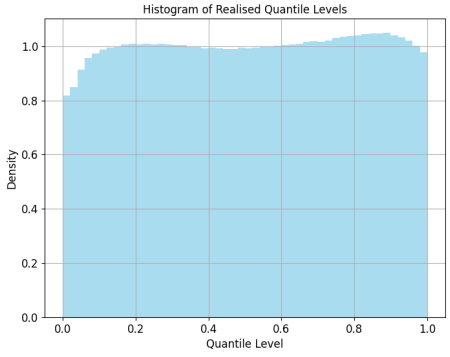


Figure D.3.: Histograms of realized quantiles for all models based on independent normal distributions.

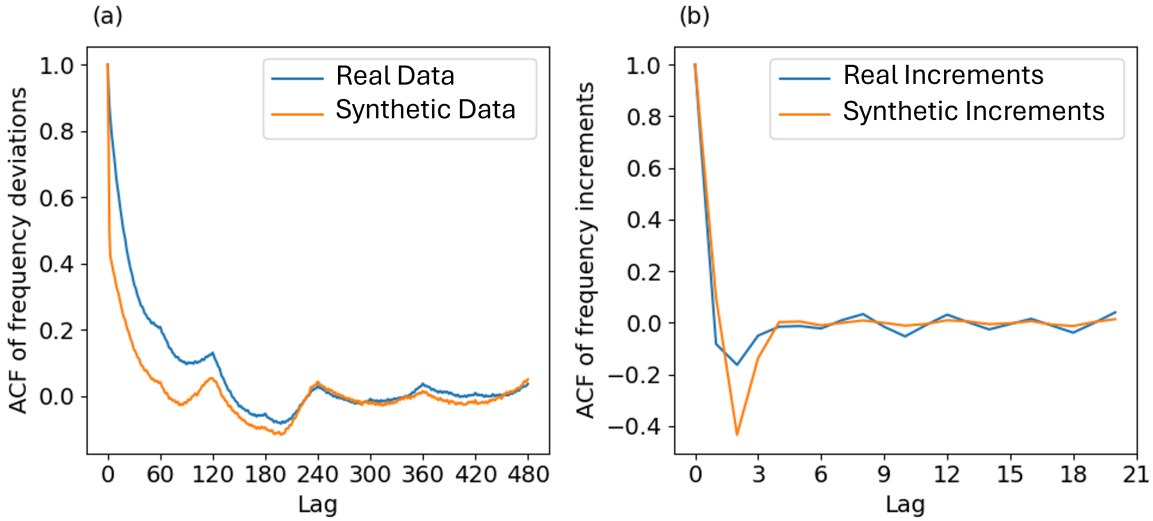


Figure D.4.: Autocorrelation functions for frequency deviation and its increments generated by a GRU-based Gaussian process model with exponentiated quadratic kernel.

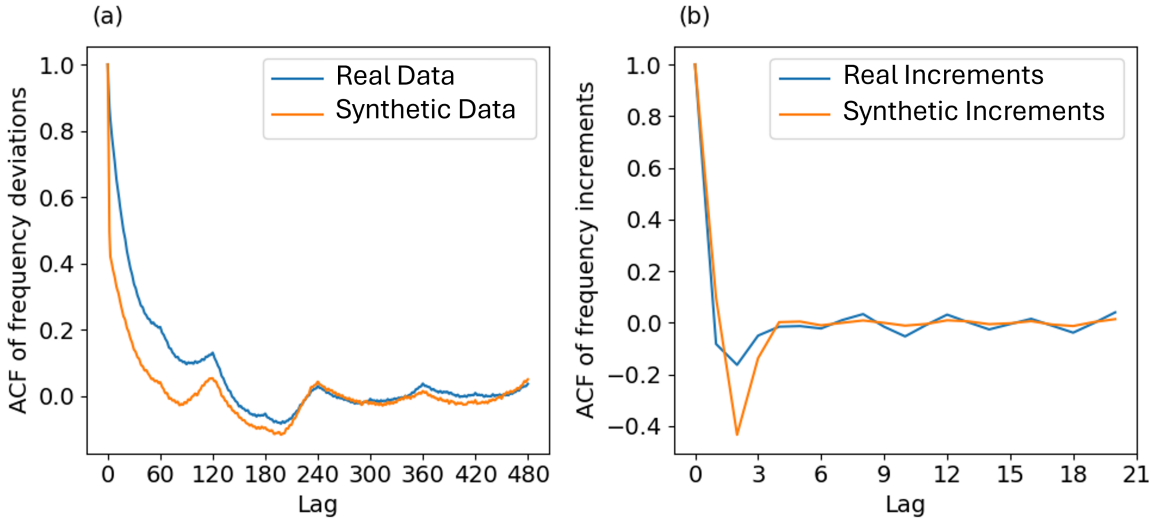


Figure D.5.: Autocorrelation functions for frequency deviation and its increments generated by a transformer-based Gaussian process model with exponentiated quadratic kernel.

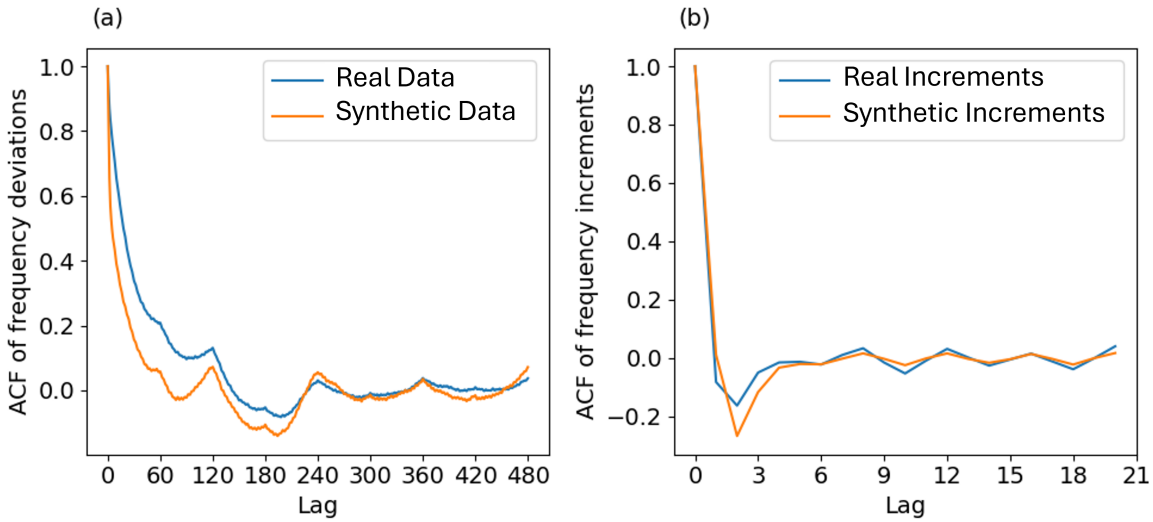


Figure D.6.: Autocorrelation functions for frequency deviation and its increments generated by a transformer-based Gaussian process model with rational quadratic kernel.

List of Author's Publications and Presentations

Publications included in this dissertation

Publ. I Liu, Bolin, Maximilian Coblenz, and Oliver Grothe. "The copula via transformation representation and its estimation with normalizing flows". Submitted to: *Computational Statistics*.

Publ. II Liu, Bolin, Maximilian Coblenz, and Oliver Grothe. "Learning Copula Densities as Separable Perturbations from Independence with Neural Networks". *Working Paper*.

Publ. III Liu, Bolin, Maximilian Coblenz, and Oliver Grothe. "Rank-Separable Smoothing for Checkerboard Copulas". *Working Paper*.

Publ. IV Liu, Bolin, Maximilian Coblenz, and Oliver Grothe (2024). "Predicting grid frequency short-term dynamics with Gaussian processes and sequence modeling". In: *Proceedings of the 15th ACM International Conference on Future and Sustainable Energy Systems*. E-Energy '24: The 15th ACM International Conference on Future and Sustainable Energy Systems. Singapore, Singapore: Association for Computing Machinery, pp. 535–550. doi:10.1145/3632775.3662160.

Publ. V Liu, Bolin, Maximilian Coblenz, and Oliver Grothe (2025). "Copula-based Probabilistic Prediction of Grid Frequency Dynamics". In: *Proceedings of the 16th ACM International Conference on Future and Sustainable Energy Systems*. E-Energy '25: The 16th ACM International Conference on Future and Sustainable Energy Systems. New York, NY, USA: Association for Computing Machinery, pp. 733–741. doi:10.1145/3679240.3734641.

Further Publications

Publ. VI Bott, Alexander, Bolin Liu, Alexander Puchta, and Juergen Fleischer. "Machine Learning-Driven RUL Prediction and Uncertainty Quantification for Ball Screw Drives in a Cloud-Ready Maintenance Framework". In: *Journal of Machine Engineering* 24.3 (2024), pp. 17–31. doi:10.36897/jme/192681.

Publ. VII Rieger, Jonas, Bolin Liu, Bernd Saugel, and Oliver Grothe. "On the assessment of the ability of measurements, nowcasts, and forecasts to track changes". In: *BMC Medical Research Methodology* 24.1 (2024), p. 275. doi:10.1186/s12874-024-02397-x.

Publ. VIII Coblenz, Maximilian, Oliver Grothe, Bolin Liu, and David Weniger. "Copulas and Deep Learning: A Review". Submitted to Dependence Modeling.

Conference presentations

Conf. I Liu*, Bolin, Maximilian Coblenz and Oliver Grothe (June 4, 2024). "Predicting grid frequency short-term dynamics with Gaussian processes and sequence modeling". e-Energy 2024: 15th ACM International Conference on Future and Sustainable Energy Systems (Singapore).

Conf. II Liu*, Bolin, Oliver Grothe and Maximilian Coblenz (August 30, 2024). "Copula estimation with flow copula models". COMPSTAT 2024: International Conference on Computational Statistics (Gießen, Germany).

Conf. III Liu, Bolin, Oliver Grothe and Maximilian Coblenz* (December 14, 2024). "The Flow Copula Class". CMStatistics 2024: 18th International Joint Conference CFE-CMStatistics (King's College London, UK).

Conf. IV Liu*, Bolin, Maximilian Coblenz and Oliver Grothe (March 25, 2025). "The Factor Flow Copula Model". DAGStat 2025: 7th Joint Statistical Meeting (Berlin, Germany).

Conf. V Liu*, Bolin, Maximilian Coblenz and Oliver Grothe (June 17, 2025). "Copula-based Probabilistic Prediction of Grid Frequency Dynamics". e-Energy 2025: 16th ACM International Conference on Future and Sustainable Energy Systems (Rotterdam, The Netherlands).

The starred name refers to the presenter at the conference.

Bibliography

Aas, Kjersti et al. (2009). "Pair-copula constructions of multiple dependence". In: *Insurance: Mathematics and Economics* 44.2, pp. 182–198.

Abadi, Martín et al. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. URL: `https://www.tensorflow.org` (visited on 01/06/2026).

Amblard, Cécile and Stéphane Girard (2002). "Symmetry and dependence properties within a semiparametric family of bivariate copulas". In: *Journal of Nonparametric Statistics* 14.6, pp. 715–727.

Anastasiou, Andreas et al. (2023). "Stein's Method Meets Computational Statistics: A Review of Some Recent Developments". In: *Statistical Science* 38.1, pp. 120–139.

Atkinson, Kendall E. (1989). *An Introduction to Numerical Analysis*. 2nd ed. New York: John Wiley & Sons.

Autin, Florent, Erwan Le Pennec, and Karine Tribouley (2010). "Thresholding methods to estimate copula density". In: *Journal of Multivariate Analysis* 101.1, pp. 200–222.

Bannier, Christina E. and Christian W. Hirsch (2010). "The economic function of credit rating agencies–What does the watchlist tell us?" In: *Journal of Banking & Finance* 34.12, pp. 3037–3049.

Baudin, Michaël et al. (2016). "OpenTURNS: An Industrial Software for Uncertainty Quantification in Simulation". In: *Handbook of Uncertainty Quantification*. Ed. by Roger Ghanem, David Higdon, and Houman Owhadi. Cham: Springer International Publishing, pp. 1–38.

Baydin, Atılım Güneş et al. (2018). "Automatic Differentiation in Machine Learning: a Survey". In: *Journal of Machine Learning Research* 18.153, pp. 1–43.

Berkooz, Gal, Philip Holmes, and John L. Lumley (1993). "The proper orthogonal decomposition in the analysis of turbulent flows". In: *Annual Review of Fluid Mechanics* 25.1, pp. 539–575.

Bilenko, Mikhail, Sugato Basu, and Raymond J. Mooney (2004). "Integrating constraints and metric learning in semi-supervised clustering". In: *Proceedings of the Twenty-First International Conference on Machine Learning*. ICML '04. Banff, Alberta, Canada: Association for Computing Machinery, pp. 81–88.

Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning*. Information Science and Statistics. New York: Springer.

Bock, R. (2004). *MAGIC Gamma Telescope*. UCI Machine Learning Repository. URL: `https://archive.ics.uci.edu/ml/datasets/magic+gamma+telescope` (visited on 01/06/2026).

Bogachev, Vladimir Igorevich, Aleksandr Viktorovich Kolesnikov, and Kirill Vladimirovich Medvedev (2005). "Triangular transformations of measures". In: *Sbornik: Mathematics* 196.3, pp. 309–335.

Boor, Carl de (1978). *A Practical Guide to Splines*. New York: Springer.

Borwein, Jonathan and Phil Howlett (2019). "Checkerboard copulas of maximum entropy with prescribed mixed moments". In: *Journal of the Australian Mathematical Society* 107.3, pp. 302–318.

Borwein, Jonathan, Phil Howlett, and Julia Piantadosi (2014). "Modelling and Simulation of Seasonal Rainfall Using the Principle of Maximum Entropy". In: *Entropy* 16.2, pp. 747–769.

Bouezmarni, Taoufik, Anouar El Ghouch, and Abderrahim Taamouti (2013). "Bernstein estimator for unbounded copula densities". In: *Statistics & Risk Modeling* 30.4, pp. 343–360.

Brent, Richard P. (1973). *Algorithms for Minimization without Derivatives*. Englewood Cliffs: Prentice-Hall.

Cantor, Richard (2004). "An introduction to recent research on credit ratings". In: *Journal of Banking and Finance* 28.11, pp. 2565–2573.

Carley, H. and M. D. Taylor (2002). "A New Proof of Sklar's Theorem". In: *Distributions With Given Marginals and Statistical Modelling*. Ed. by Carles M. Cuadras, Josep Fortiana, and José A. Rodriguez-Lallena. Dordrecht: Springer Netherlands, pp. 29–34.

Casella, George and Roger L. Berger (2002). *Statistical Inference*. 2nd ed. Pacific Grove: Duxbury.

Charpentier, Arthur, Jean-David Fermanian, and Olivier Scaillet (2007). "The estimation of copulas: theory and practice". In: *Copulas: From Theory to Application in Finance*. Ed. by Jörn Rank. London: Risk Books, pp. 35–64.

Chatrabgoun, Omid and G. Parham (2016). "Copula density estimation using multiwavelets based on the multiresolution analysis". In: *Communications in Statistics–Simulation and Computation* 45.9, pp. 3350–3372.

Chen, Louis H. Y., Larry Goldstein, and Qi-Man Shao (2011). *Normal Approximation by Stein's Method*. Berlin: Springer.

Chen, Xiaohong et al. (2010). "Estimation and model selection of semiparametric multivariate survival functions under general censorship". In: *Journal of Econometrics* 157.1, pp. 129–142.

Chen, Yen-Chi (2017). "A tutorial on kernel density estimation and recent advances". In: *Biostatistics & Epidemiology* 1.1, pp. 161–187.

Chesneau, Christophe (2022). "Theoretical study of some angle parameter trigonometric copulas". In: *Modelling* 3.1, pp. 140–163.

Cho, Kyunghyun et al. (2014). "Learning phrase representations using RNN encoder-decoder for statistical machine translation". In: *arXiv preprint 1406.1078*.

Choroś, Barbara, Rustam Ibragimov, and Elena Permiakova (2010). "Copula estimation". In: *Copula Theory and Its Applications: Proceedings of the Workshop Held in Warsaw, 25-26 September 2009*. Springer, pp. 77–91.

Chung, Junyoung et al. (2014). "Empirical evaluation of gated recurrent neural networks on sequence modeling". In: *arXiv preprint 1412.3555*.

Chwialkowski, Kacper, Heiko Strathmann, and Arthur Gretton (2016). "A kernel test of goodness of fit". In: *Proceedings of the 33rd International Conference on Machine Learning*. ICML'16. JMLR.org, pp. 2606–2615.

Coblenz, Maximilian (2021). "MATVines: A vine copula package for MATLAB". In: *SoftwareX* 14, p. 100700.

Coblenz, Maximilian et al. (2020). "Modelling fuel injector spray characteristics in jet engines by using vine copulas". In: *Journal of the Royal Statistical Society Series C: Applied Statistics* 69.4, pp. 863–886.

Conway, John B. (1990). *A Course in Functional Analysis*. 2nd ed. New York: Springer-Verlag.

Cover, Thomas M. and Joy A. Thomas (2006). *Elements of Information Theory*. 2nd ed. Hoboken: Wiley.

Cuberos, A., E. Masiello, and V. Maume-Deschamps (2020). "Copulas checker-type approximations: Application to quantiles estimation of sums of dependent random variables". In: *Communications in Statistics–Theory and Methods* 49.12, pp. 3044–3062.

Cybenko, George (1989). "Approximation by Superpositions of a Sigmoidal Function". In: *Mathematics of Control, Signals and Systems* 2.4, pp. 303–314.

Czado, Claudia (2019). *Analyzing Dependent Data with Vine Copulas: A Practical Guide With R*. Cham: Springer.

Czado, Claudia and Thomas Nagler (2022). "Vine copula based modeling". In: *Annual Review of Statistics and Its Application* 9.1, pp. 453–477.

DasGupta, Anirban (2010). "Continuous Random Variables". In: *Fundamentals of Probability: A First Course*. New York, NY: Springer New York, pp. 153, 165.

De Amorim, Renato Cordeiro (2011). "Learning Feature Weights for K-Means Clustering Using the Minkowski Metric". PhD thesis. London: Birkbeck, University of London.

Deheuvels, Paul (1979). "La fonction de dépendance empirique et ses propriétés. Un test non paramétrique d'indépendance". In: *Bulletins de l'Académie Royale de Belgique* 65.1, pp. 274–292.

Diers, Dorothea, Martin Eling, and Sebastian D Marek (2012). "Dependence modeling in non-life insurance using the Bernstein copula". In: *Insurance: Mathematics and Economics* 50.3, pp. 430–436.

Diggle, Peter J, Jonathan A Tawn, and Rana A Moyeed (1998). "Model-based geostatistics". In: *Journal of the Royal Statistical Society Series C: Applied Statistics* 47.3, pp. 299–350.

Dinh, Laurent, Jascha Sohl-Dickstein, and Samy Bengio (2016). "Density estimation using Real NVP". In: *arXiv preprint 1605.08803*.

Durante, Fabrizio, Rachele Foschi, and Peter Sarkoci (2010). "Distorted copulas: constructions and tail dependence". In: *Communications in Statistics–Theory and Methods* 39.12, pp. 2288–2301.

Durante, Fabrizio and Carlo Sempi (2015). *Principles of Copula Theory*. Boca Raton: Chapman and Hall/CRC.

Ekert, Artur and Peter L Knight (1995). "Entangled quantum systems and the Schmidt decomposition". In: *American Journal of Physics* 63.5, pp. 415–423.

Elfwing, Stefan, Eiji Uchibe, and Kenji Doya (2018). "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning". In: *Neural Networks* 107. Special issue on deep reinforcement learning, pp. 3–11.

Embrechts, Paul, Filip Lindskog, and Alexander McNeil (2001). "Modelling dependence with copulas". In: *Rapport technique, Département de mathématiques, Institut Fédéral de Technologie de Zurich, Zurich* 14, pp. 1–50.

Feurer, Matthias and Frank Hutter (2019). "Hyperparameter optimization". In: *Automated Machine Learning: Methods, Systems, Challenges*. Springer International Publishing Cham, pp. 3–33.

Filatrella, Giovanni, Arne Hejde Nielsen, and Niels Falsig Pedersen (2008). "Analysis of a power grid using a Kuramoto-like model". In: *The European Physical Journal B* 61, pp. 485–491.

Flamary, Rémi et al. (2021). "POT: Python optimal transport". In: *Journal of Machine Learning Research* 22.78, pp. 1–8.

Frahm, Gabriel, Markus Junker, and Rafael Schmidt (2005). "Estimating the tail-dependence coefficient: Properties and pitfalls". In: *Insurance: Mathematics and Economics* 37.1. Papers presented at the DeMoSTAFI Conference, Québec, 20-22 May 2004, pp. 80–100.

Frees, Edward W and Emiliano A Valdez (1998). "Understanding relationships using copulas". In: *North American Actuarial Journal* 2.1, pp. 1–25.

Geenens, G., A. Charpentier, and D. Paindaveine (2017). "Probit Transformation for Nonparametric Kernel Estimation of the Copula Density". In: *Bernoulli* 23.3, pp. 1848–1873.

Geenens, Gery (2020). "Copula modeling for discrete random vectors". In: *Dependence Modeling* 8.1, pp. 417–440.

Genest, C et al. (Jan. 2019). "Testing for independence in arbitrary distributions". In: *Biometrika* 106.1, pp. 47–68.

Genest, C., K. Ghoudi, and L.-P. Rivest (1995). "A Semiparametric Estimation Procedure of Dependence Parameters in Multivariate Families of Distributions". In: *Biometrika* 82.3, pp. 543–552.

Genest, Christian and Anne-Catherine Favre (2007). "Everything you always wanted to know about copula modeling but were afraid to ask". In: *Journal of hydrologic engineering* 12.4, pp. 347–368.

Genest, Christian, Esterina Masiello, and Karine Tribouley (2009). "Estimating copula densities through wavelets". In: *Insurance: Mathematics and Economics* 44.2, pp. 170–181.

Genest, Christian and Johanna Nešlehová (2007). "A Primer on Copulas for Count Data". In: *ASTIN Bulletin* 37.2, pp. 475–515.

Genest, Christian, Johanna G. Nešlehová, and Bruno Rémillard (2014). "On the empirical multilinear copula process for count data". In: *Bernoulli* 20.3, pp. 1344–1371.

– (2017). "Asymptotic Behavior of the Empirical Multilinear Copula Process under Broad Conditions". In: *Journal of Multivariate Analysis* 159, pp. 82–110.

Ghanem, Roger G. and Pol D. Spanos (2003). *Stochastic Finite Elements: A Spectral Approach*. Mineola: Dover Publications.

Gijbels, Iène and Jan Mielniczuk (1990). "Estimating the density of a copula function". In: *Communications in Statistics–Theory and Methods* 19.2, pp. 445–464.

Gneiting, Tilmann and Adrian E Raftery (2007). "Strictly proper scoring rules, prediction, and estimation". In: *Journal of the American statistical Association* 102.477, pp. 359–378.

Golub, Gene H. and Charles F. Van Loan (1996). *Matrix Computations*. 3rd ed. Baltimore: The Johns Hopkins University Press.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep Learning*. Cambridge, MA: MIT Press.

Gorham, Jackson and Lester Mackey (2015). "Measuring Sample Quality with Stein's Method". In: *Advances in Neural Information Processing Systems*. Vol. 28, pp. 226–234.

Gorjão, Leonardo Rydin et al. (2020). "Data-driven model of the power-grid frequency dynamics". In: *IEEE access* 8, pp. 43082–43097.

Gribkova, Svetlana and Olivier Lopez (2015). "Non-parametric copula estimation under bivariate censoring". In: *Scandinavian Journal of Statistics* 42.4, pp. 925–946.

Griebel, Michael and Guanglian Li (2018). "On the decay rate of the singular values of bivariate functions". In: *SIAM Journal on Numerical Analysis* 56.2, pp. 974–993.

Griewank, Andreas and Andrea Walther (2008). *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. 2nd ed. Philadelphia: SIAM.

Grothe, Oliver, Fabian Kächele, and Fabian Krüger (2023). "From Point Forecasts to Multivariate Probabilistic Forecasts: The Schaake Shuffle for Day-Ahead Electricity Price Forecasting". In: *Energy Economics* 120, p. 106602.

Grothe, Oliver and Jonas Rieger (2024). "Decomposition and graphical correspondence analysis of checkerboard copulas". In: *Dependence Modeling* 12.1, p. 20240006.

Gudendorf, Gordon and Johan Segers (2010). "Extreme-Value Copulas". In: *Copula Theory and Its Applications*. Ed. by Piotr Jaworski et al. Berlin: Springer, pp. 127–145.

Hendrycks, Dan and Kevin Gimpel (2016). "Gaussian Error Linear Units (GELUs)". In: *arXiv preprint 1606.08415*.

Hensman, James, Nicolò Fusi, and Neil D. Lawrence (2013). "Gaussian Processes for Big Data". In: *Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence (UAI)*. Bellevue, WA, pp. 282–290.

Hersbach, Hans (2000). "Decomposition of the continuous ranked probability score for ensemble prediction systems". In: *Weather and Forecasting* 15.5, pp. 559–570.

Hofert, Marius et al. (2018). *Elements of Copula Modeling with R*. Cham: Springer.

Hofert, Marius et al. (2024). *copula: Multivariate Dependence with Copulas*. R package. URL: `https://CRAN.R-project.org/package=copula` (visited on 01/06/2026).

Hornik, Kurt, Maxwell Stinchcombe, and Halbert White (1989). "Multilayer Feedforward Networks are Universal Approximators". In: *Neural Networks* 2.5, pp. 359–366.

Hyndman, Rob J. and Yanan Fan (1996). "Sample Quantiles in Statistical Packages". In: *The American Statistician* 50.4, pp. 361–365.

Hyvärinen, Aapo (2005). "Estimation of Non-Normalized Statistical Models by Score Matching". In: *Journal of Machine Learning Research* 6, pp. 695–709.

Idel, Martin and Michael M Wolf (2015). "Sinkhorn normal form for unitary matrices". In: *Linear Algebra and its Applications* 471, pp. 76–84.

Islam, Saidul et al. (2023). "A comprehensive survey on applications of transformers for deep learning tasks". In: *Expert Systems with Applications*, p. 122666.

Jain, Achin et al. (2018). "Learning and control using Gaussian processes". In: *2018 ACM/IEEE 9th international conference on cyber-physical systems (ICCPS)*. IEEE, pp. 140–149.

Joe, Harry (1997). *Multivariate Models and Multivariate Dependence Concepts*. London: Chapman and Hall/CRC.

– (2014). *Dependence Modeling with Copulas*. Vol. 134. Monographs on Statistics and Applied Probability. Boca Raton: Chapman and Hall/CRC.

Joe, Harry and James Jianmeng Xu (1996). *The Estimation Method of Inference Functions for Margins for Multivariate Models*. Tech. rep. 166. Vancouver: Department of Statistics, University of British Columbia.

Jones, M. Chris (1992). "Estimating densities, quantiles, quantile densities and density quantiles". In: *Annals of the Institute of Statistical Mathematics* 44.4, pp. 721–727.

Jordan, Alexander, Fabian Krüger, and Sebastian Lerch (2019). "Evaluating Probabilistic Forecasts with scoringRules". In: *Journal of Statistical Software* 90.12, pp. 1–37.

Kamthe, Sanket, Samuel Assefa, and Marc Deisenroth (2021). "Copula Flows for Synthetic Data Generation". In: *arXiv preprint 2101.00598*.

Kauermann, Göran, Christian Schellhase, and David Ruppert (2013). "Flexible copula density estimation with penalized hierarchical B-splines". In: *Scandinavian Journal of Statistics* 40.4, pp. 685–705.

Khoudraji, Abdelhaq (1995). "Contributions à l'étude des copules et à la modélisation des valeurs extrêmes bivariées". PhD thesis. Québec, Canada: Université Laval.

Kim, Tae-Young and Sung-Bae Cho (2019). "Predicting residential energy consumption using CNN-LSTM neural networks". In: *Energy* 182, pp. 72–81.

Kingma, Diederik P. and Jimmy Ba (2015). "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun.

Kingma, Diederik P. and Prafulla Dhariwal (2018). "Glow: generative flow with invertible 1×1 convolutions". In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. NIPS'18. Red Hook: Curran Associates Inc., pp. 10236–10245.

Kirby, Brendan J (2003). *Frequency control concerns in the North American electric power system*. Tech. rep. Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States).

Klein, Nadja et al. (2022). "Multivariate conditional transformation models". In: *Scandinavian Journal of Statistics* 49.1, pp. 116–142.

Klenke, Achim (2014). *Probability Theory: A Comprehensive Course*. 2nd ed. London: Springer.

Kobyzev, Ivan, Simon J.D. Prince, and Marcus A. Brubaker (2021). "Normalizing Flows: An Introduction and Review of Current Methods". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.11, pp. 3964–3979.

Kolesárová, Anna, Gaspar Mayor, and Radko Mesiar (2015). "Quadratic constructions of copulas". In: *Information Sciences* 310, pp. 69–76.

Komorník, Jozef, Magda Komorníková, and Jana Kalická (2017). "Dependence measures for perturbations of copulas". In: *Fuzzy Sets and Systems* 324, pp. 100–116.

Kraljic, David (2023). "Towards realistic statistical models of the grid frequency". In: *IEEE Transactions on Power Systems* 38.1, pp. 256–266.

Kraskov, Alexander, Harald Stögbauer, and Peter Grassberger (2004). "Estimating mutual information". In: *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics* 69.6, p. 066138.

Krupskii, Pavel and Harry Joe (2013). "Factor copula models for multivariate data". In: *Journal of Multivariate Analysis* 120.C, pp. 85–101.

Kruse, J., D. Witthaut, and B. Schäfer (2021a). *Supplementary data: "Revealing drivers and risks for power grid frequency stability with explainable AI"*. URL: `https://doi.org/10.5281/zenodo.5118352` (visited on 01/06/2026).

Kruse, Johannes (2023). *johkruse/PIML-for-grid-frequency-modelling*. Version v0.2.0. URL: `https://doi.org/10.5281/zenodo.7688692` (visited on 01/06/2026).

Kruse, Johannes, Benjamin Schäfer, and Dirk Witthaut (2020). "Predictability of power grid frequency". In: *IEEE access* 8, pp. 149435–149446.

– (2021b). "Revealing drivers and risks for power grid frequency stability with explainable AI". In: *Patterns* 2.11, p. 100365.

Kruse, Johannes et al. (2023). "Physics-informed machine learning for power grid frequency modeling". In: *PRX energy* 2.4, p. 043003.

Kulo, Mak and Annick Poulain (Jan. 2025). *Credit Rating Transition and Default Study 2024*. Credit Policy. Annual update with transition matrices and default rates. Berlin and London: Scope Ratings GmbH and Scope Ratings UK Limited.

Kuzmenko, Viktor, Romel Salam, and Stan Uryasev (2020). "Checkerboard copula defined by sums of random variables". In: *Dependence Modeling* 8.1, pp. 70–92.

Lando, David (2004). *Credit Risk Modeling: Theory and Applications*. Princeton: Princeton University Press.

Laszkiewicz, Mike, Johannes Lederer, and Asja Fischer (2021). "Copula-Based Normalizing Flows". In: *ICML Workshop on Invertible Neural Networks, Normalizing Flows, and Explicit Likelihood Models*.

– (2022). "Marginal Tail-Adaptive Normalizing Flows". In: *Proceedings of the 39th International Conference on Machine Learning*. Vol. 162. Proceedings of Machine Learning Research. PMLR, pp. 12020–12048.

Laurent, Olivier (2020). *Multivariate Extension of the Kolmogorov–Smirnov Test*. URL: `https://github.com/o-laurent/multivariate-ks-test` (visited on 01/06/2026).

Laverny, Oskar (2020). "Empirical and Non-parametric Copula Models with the cort R Package". In: *Journal of Open Source Software* 5.56, p. 2653.

Li, X. et al. (1997). "On Approximation of Copulas". In: *Distributions with given Marginals and Moment Problems*. Ed. by Viktor Beneš and Josef Štěpán. Dordrecht: Springer Netherlands, pp. 107–116.

Liebscher, Eckhard (2005). "Semiparametric density estimators using copulas". In: *Communications in Statistics-Theory and Methods* 34.1, pp. 59–71.

Lin, Liyuan et al. (2025). "The checkerboard copula and dependence concepts". In: *SIAM Journal on Financial Mathematics* 16.2, pp. 426–447.

Ling, Chun Kai, Fei Fang, and J. Zico Kolter (2020). "Deep archimedean copulas". In: *Proceedings of the 34th International Conference on Neural Information Processing Systems*. NIPS '20. Red Hook: Curran Associates Inc.

Liu, Bolin, Maximilian Coblenz, and Oliver Grothe (2024a). "Predicting grid frequency short-term dynamics with Gaussian processes and sequence modeling". In: *Proceedings of the 15th ACM International Conference on Future and Sustainable Energy Systems*, pp. 535–550.

– (2024b). *Supplementary Code for the Paper "Predicting grid frequency short-term dynamics with Gaussian processes and sequence modeling"*. URL: https://github.com/bolin-liu/sequence-model-and-gaussian-process-for-frequency-prediction (visited on 01/06/2026).

– (2025a). *Supplementary Code for the Paper "Copula-based Probabilistic Prediction of Grid Frequency Dynamics"*. URL: https://github.com/bolin-liu/copula_grid_frequency_prediction (visited on 01/06/2026).

– (2025b). *Supplementary Code for the Paper "The copula via transformation representation and its estimation with normalizing flows"*. URL: https://github.com/bolin-liu/Copula-via-Transformation-Estimation-with-Normalizing-Flow (visited on 01/06/2026).

– (2026a). *Supplementary Code for the Paper "Learning Copula Densities as Separable Perturbations from Independence with Neural Networks"*. URL: https://github.com/bolin-liu/copula_density_modeling_separable_perturbations (visited on 01/06/2026).

– (2026b). *Supplementary Code for the Paper "Rank-Separable Smoothing for Checkerboard Copulas"*. URL: https://github.com/bolin-liu/checkerboard_smoothing- (visited on 01/06/2026).

Liu, Qiang, Jason D. Lee, and Michael Jordan (2016). "A kernelized stein discrepancy for goodness-of-fit tests". In: *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*. ICML'16. New York, NY, USA: JMLR.org, pp. 276–284.

Liu, Qiang and Dilin Wang (2016). "Stein variational Gradient descent: a general purpose Bayesian inference algorithm". In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. NIPS'16. Red Hook: Curran Associates Inc., pp. 2378–2386.

Longla, Martial (2024). "New copula families and mixing properties". In: *Statistical Papers* 65.7, pp. 4331–4363.

Lowin, Jeremiah L (2010). "The Fourier copula: theory & applications". In: *Available at SSRN 1804664*.

Mandolini, Giorgio Maria, Daniele Sanna, and Giorgio Zannini Quirini (2020). *Density Estimation Using Real NVP*. URL: `https://github.com/keras-team/keras-io/blob/master/examples/generative/real_nvp.py` (visited on 01/06/2026).

McDonald, Andrew, Pang-Ning Tan, and Lifeng Luo (July 2022). "COMET Flows: Towards Generative Modeling of Multivariate Extremes and Tail Dependence". In: *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*. Ed. by Lud De Raedt. Main Track. International Joint Conferences on Artificial Intelligence Organization, pp. 3328–3334.

McNeil, Alexander J., Rüdiger Frey, and Paul Embrechts (2015). *Quantitative Risk Management: Concepts, Techniques and Tools*. Princeton: Princeton University Press.

Mercer, James (1909). "Xvi. functions of positive and negative type, and their connection the theory of integral equations". In: *Philosophical Transactions of the Royal Society of London. Series A, containing papers of a mathematical or physical character* 209.441-458, pp. 415–446.

Merton, Robert C. (1974). "On the Pricing of Corporate Debt: The Risk Structure of Interest Rates". In: *The Journal of Finance* 29.2, pp. 449–470.

Mesiar, Radko and Vadoud Najjari (2014). "New families of symmetric/asymmetric copulas". In: *Fuzzy Sets and Systems* 252, pp. 99–110.

Morillas, Patricia Mariela (2005). "A method to obtain new copulas from a given one". In: *Metrika* 61, pp. 169–184.

Mukhopadhyay, Subhadeep and Emanuel Parzen (2020). "Nonparametric universal copula modeling". In: *Applied Stochastic Models in Business and Industry* 36.1, pp. 77–94.

Murphy, Kevin P. (2012). *Machine Learning: A Probabilistic Perspective*. Cambridge: MIT Press.

Naaman, Michael (2021). "On the tight constant in the multivariate Dvoretzky–Kiefer–Wolfowitz inequality". In: *Statistics & Probability Letters* 173, p. 109088.

Nagler, Thomas (2018). "kdecopula: An R Package for the Kernel Estimation of Bivariate Copula Densities". In: *Journal of Statistical Software* 84.7, pp. 1–22.

– (2024). *kdevine: Multivariate Kernel Density Estimation with Vine Copulas*. R Package. URL: `https://CRAN.R-project.org/package=kdevine` (visited on 01/06/2026).

Nagler, Thomas and Claudia Czado (2016). "Evading the curse of dimensionality in nonparametric density estimation with simplified vine copulas". In: *Journal of Multivariate Analysis* 151, pp. 69–89.

Nagler, Thomas, Christian Schellhase, and Claudia Czado (2017). "Nonparametric estimation of simplified vine copula models: comparison of methods". In: *Dependence Modeling* 5.1, pp. 99–120.

Nagler, Thomas et al. (2024). *VineCopula: Statistical Inference of Vine Copulas*. R package. URL: `https://CRAN.R-project.org/package=VineCopula` (visited on 01/06/2026).

Nelsen, Roger B. (2006). *An Introduction to Copulas*. 2nd ed. New York: Springer.

Ngounou Bakam, Yves I. and Denys Pommeret (2025). "Nonparametric estimation of copulas and copula densities by orthogonal projections". In: *Econometrics and Statistics* 36, pp. 90–118.

Ngounou Bakam, Yves Ismaël and Denys Pommeret (2024). "Smooth test for equality of copulas". In: *Electronic Journal of Statistics* 18.1, pp. 895–941.

Nielsen, Michael A and Isaac L Chuang (2010). *Quantum computation and quantum information*. Cambridge: Cambridge University Press.

Oates, Chris J., Mark Girolami, and Nicolas Chopin (2017). "Control Functionals for Monte Carlo Integration". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 79.3, pp. 695–718.

Ourahou, Meriem et al. (2020). "Review on smart grid control and reliability in presence of renewable energies: Challenges and prospects". In: *Mathematics and computers in simulation* 167, pp. 19–31.

Papamakarios, George, Theo Pavlakou, and Iain Murray (2017). "Masked autoregressive flow for density estimation". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Red Hook: Curran Associates Inc., pp. 2335–2344.

Papamakarios, George et al. (2021). "Normalizing flows for probabilistic modeling and inference". In: *Journal of Machine Learning Research* 22.57, pp. 1–64.

Papapantoleon, Antonis (2015). "Computation of copulas by Fourier methods". In: *Innovations in Quantitative Risk Management: TU München, September 2013*. Springer International Publishing Cham, pp. 347–354.

Parzen, Emanuel (1962). "On estimation of a probability density function and mode". In: *The Annals of Mathematical Statistics* 33.3, pp. 1065–1076.

Prakash, Vivek et al. (2022). "Frequency response support assessment from uncertain wind generation". In: *International Journal of Electrical Power & Energy Systems* 134, p. 107465.

Prechelt, Lutz (1998). "Automatic early stopping using cross validation: quantifying the criteria". In: *Neural Networks* 11.4, pp. 761–767.

Provost, Serge B and Yishan Zang (2024). "Nonparametric copula density estimation methodologies". In: *Mathematics* 12.3, p. 398.

R Core Team (2024). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL: `https://www.R-project.org/` (visited on 01/06/2026).

Ramachandran, Prajit, Barret Zoph, and Quoc V. Le (2017). "Searching for Activation Functions". In: *arXiv preprint 1710.05941*.

Ranganath, Rajesh et al. (2016). "Operator variational inference". In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. NIPS'16. Red Hook: Curran Associates Inc., pp. 496–504.

Rasmussen, Carl Edward and Christopher K. I. Williams (2006). *Gaussian processes for machine learning*. Vol. 1. Cambridge: MIT Press.

Robbins, Herbert and Sutton Monro (1951). "A Stochastic Approximation Method". In: *The Annals of Mathematical Statistics* 22.3, pp. 400–407.

Rodrıguez-Lallena, José Antonio and Manuel Úbeda-Flores (2004). "A new class of bivariate copulas". In: *Statistics & probability letters* 66.3, pp. 315–325.

Rosenblatt, Murray (1952). "Remarks on a multivariate transformation". In: *The Annals of Mathematical Statistics* 23.3, pp. 470–472.

Ross, Brian C (2014). "Mutual information between discrete and continuous data sets". In: *PloS one* 9.2, e87357.

Ross, Nathan (2011). "Fundamentals of Stein's Method". In: *Probability Surveys* 8, pp. 210–293.

Rousseeuw, Peter J (1987). "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis". In: *Journal of computational and applied mathematics* 20, pp. 53–65.

Rumelhart, David E., Geoffrey E. Hinton, and Ronald J. Williams (1986). "Learning Representations by Back-Propagating Errors". In: *Nature* 323, pp. 533–536.

Rüschendorf, Ludger (1976). "Asymptotic Distributions of Multivariate Rank Order Statistics". In: *The Annals of Statistics* 4.5, pp. 912–923.

– (2009). "On the distributional transform, Sklar's theorem, and the empirical copula process". In: *Journal of Statistical Planning and Inference* 139.11, pp. 3921–3927.

Saha, Sajeeb, MI Saleem, and TK Roy (2023). "Impact of high penetration of renewable energy sources on grid frequency behaviour". In: *International Journal of Electrical Power & Energy Systems* 145, p. 108701.

Saminger-Platz, Susanne et al. (2021). "New results on perturbation-based copulas". In: *Dependence Modeling* 9.1, pp. 347–373.

– (2024). "On comprehensive families of copulas involving the three basic copulas and transformations thereof". In: *Dependence Modeling* 12.1, p. 20240007.

Sancetta, Alessio and Stephen Satchell (2004). "The Bernstein copula and its applications to modeling and approximations of multivariate distributions". In: *Econometric theory* 20.3, pp. 535–562.

Sarmiento, Carlos, Carlos Valencia, and Raha Akhavan-Tabatabaei (2018). "Copula autoregressive methodology for the simulation of wind speed and direction time series". In: *Journal of Wind Engineering and Industrial Aerodynamics* 174, pp. 188–199.

Schäfer, Benjamin et al. (2017). "Escape routes, weak links, and desynchronization in fluctuation-driven networks". In: *Physical Review E* 95.6, p. 060203.

Schefzik, Roman, Thordis L. Thorarinsdottir, and Tilmann Gneiting (2013). "Uncertainty Quantification in Complex Simulation Models Using Ensemble Copula Coupling". In: *Statistical Science* 28.4, pp. 616–640.

Schmidt, Erhard (1908). "Zur Theorie der linearen und nichtlinearen Integralgleichungen. III. Teil: Über die Auflösung der nichtlinearen Integralgleichung und die Verzweigung ihrer Lösungen". In: *Mathematische Annalen* 65.3, pp. 370–399.

Schwab, Christoph and Radu Alexandru Todor (2006). "Karhunen–Loève approximation of random fields by generalized fast multipole methods". In: *Journal of Computational Physics* 217.1, pp. 100–122.

Segers, Johan, Masaaki Sibuya, and Hideatsu Tsukahara (2017). "The empirical beta copula". In: *Journal of Multivariate Analysis* 155, pp. 35–51.

Shen, Xiaojing, Yunmin Zhu, and Lixin Song (2008). "Linear B-spline copulas with applications to nonparametric estimation of copulas". In: *Computational Statistics & Data Analysis* 52.7, pp. 3806–3819.

Si, Shijing et al. (2020). "Scalable Control Variates for Monte Carlo Methods via Stochastic Optimization". In: *arXiv preprint 2006.07487*.

Silverman, Bernard W. (1986). *Density Estimation for Statistics and Data Analysis*. London: Chapman & Hall.

Simonoff, Jeffrey S. (1996). *Smoothing Methods in Statistics*. Springer Series in Statistics. New York: Springer.

Sinkhorn, Richard and Paul Knopp (1967). "Concerning nonnegative matrices and doubly stochastic matrices". In: *Pacific Journal of Mathematics* 21.2, pp. 343–348.

Sklar, Abe (1959). "Fonctions de répartition à n dimensions et leurs marges". In: *Publications de l'Institut de Statistique de l'Université de Paris (ISUP)* 8, pp. 229–231.

Smith, Michael Stanley (2023). "Implicit Copulas: An Overview". In: *Econometrics and Statistics* 28, pp. 81–104.

Smith, Michael Stanley and Nadja Klein (2021). "Bayesian inference for regression copulas". In: *Journal of Business & Economic Statistics* 39.3, pp. 712–728.

Smith, Michael Stanley and Worapree Maneesoonthorn (2018). "Inversion copulas from nonlinear state space models with an application to inflation forecasting". In: *International Journal of Forecasting* 34.3, pp. 389–407.

Stein, Charles (1972). "A bound for the error in the normal approximation to the distribution of a sum of dependent random variables". In: *Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability, Volume 2: Probability Theory*. University of California Press, pp. 583–602.

Tabak, Esteban G. and Cristina V. Turner (2013). "A family of nonparametric density estimation algorithms". In: *Communications on Pure and Applied Mathematics* 66.2, pp. 145–164.

Teshima, Takeshi et al. (2020). "Coupling-based invertible neural networks are universal diffeomorphism approximators". In: *Advances in Neural Information Processing Systems*. Vol. 33, pp. 3362–3373.

U.S. Census Bureau (2024). *American Community Survey (ACS) 1-Year Estimates: Table B19037: Age of Householder by Household Income in the Past 12 Months (in 2023 Inflation-Adjusted Dollars)*. ACS 1-Year Detailed Table, 2023. URL: `https://data.census.gov/table/ACSDT1Y2023.B19037` (visited on 01/06/2026).

Vaswani, Ashish et al. (2017). "Attention is All You Need". In: *Advances in Neural Information Processing Systems*. Vol. 30.

Virtanen, Pauli et al. (2020). "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17, pp. 261–272.

Watson, Matthew et al. (2024). *KerasHub*. URL: `https://github.com/keras-team/keras-hub` (visited on 01/06/2026).

Weiß, Gregor and Marcus Scheffer (2012). "Smooth nonparametric bernstein vine copulas". In: *arXiv preprint 1210.2043*.

Wen, Kuangyu and Ximing Wu (2020). "Transformation-Kernel Estimation of Copula Densities". In: *Journal of Business & Economic Statistics* 38.1, pp. 148–164.

White, Lawrence H (2005). "The Federal Reserve System's influence on research in monetary economics". In: *Econ Journal Watch* 2.2, p. 325.

Wiese, Magnus, Robert Knobloch, and Ralf Korn (2019). "Copula & Marginal Flows: Disentangling the Marginal from its Joint". In: *arXiv preprint 1907.03361*.

Wood, Allen J., Bruce F. Wollenberg, and Gerald B. Sheblé (2014). *Power Generation, Operation, and Control*. 3rd ed. Hoboken: Wiley.

Woolson, Robert F (2005). "Wilcoxon signed-rank test". In: *Encyclopedia of biostatistics* 8.

Xing, Eric et al. (2002). "Distance metric learning with application to clustering with side-information". In: *Advances in neural information processing systems* 15.

Yamut, Tolga and Burcu Hudaverdi (2023). "Classification with Bernstein copula as discrimination function". In: *Communications in Statistics-Simulation and Computation*, pp. 1–17.

Yang, Yandong et al. (2018). "Power load probability density forecasting using Gaussian process quantile regression". In: *Applied Energy* 213, pp. 499–509.

Yu, Yong et al. (2019). "A review of recurrent neural networks: LSTM cells and network architectures". In: *Neural computation* 31.7, pp. 1235–1270.

Zanetta, Francesco and Sam Allen (2024). *Scoringrules: a python library for probabilistic forecast evaluation*. URL: `https://github.com/frazane/scoringrules` (visited on 01/06/2026).

Zerveas, George et al. (2021). "A transformer-based framework for multivariate time series representation learning". In: *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pp. 2114–2124.

Zhu, Haibin (2006). "An empirical comparison of credit spreads between the bond market and the credit default swap market". In: *Journal of financial services research* 29.3, pp. 211–235.