Developments in Theory and Application of Copulas:

Variance Estimation, Economic Modeling, Forecasting and Machine Learning

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. Fabian Kächele

Tag der mündlichen Prüfung: 19. Juli 2023
Referent: Prof. Dr. Oliver Grothe
Korreferent: Prof. Dr. Melanie Schienle

Karlsruhe 2023

Danksagung

Gegen Ende meines Studiums entdeckte ich meine Freude am wissenschaftlichen Arbeiten und der Statistik. Durch Professor Dr. Oliver Grothe erhielt ich die besondere Möglichkeit dieser Leidenschaft nachzugehen und am Lehrstuhl für Analysis and Statistics zu promovieren. An erster Stelle möchte ich ihm dafür meinen tiefgreifenden Dank aussprechen. Mir ist bewusst, dass die intensive Betreuung, das entgegengebrachte Vertrauen und die weitgehenden Freiheiten während dieser Zeit keine Selbstverständlichkeit darstellen. Insbesondere möchte ich ihm außerdem für die enge Zusammenarbeit und die intensiven fachlichen Diskussionen danken. Während meiner Zeit am Lehrstuhl habe ich mich nicht nur fachlich, sondern auch persönlich weiterentwickeln können und hoffe den Führungsstil von Professor Dr. Oliver Grothe auch zukünftig in mein eigenes Arbeitsleben einbringen zu können. Des Weiteren möchte ich mich bei meiner Korreferentin und Koautorin Professor Dr. Melanie Schienle bedanken. Durch die Chance, meine Masterarbeit an ihrem Lehrstuhl zu verfassen, konnte ich mein Interesse für wissenschaftliches Arbeiten und Copulas initial entwickeln. Ich bedanke mich für ihre Geduld während unserer langjährigen Zusammenarbeit und die immer wiederkehrenden fachlichen Impulse. Ebenso möchte ich Prof Dr. Fabian Krüger für seine Rolle als Prüfer und Koautor danken. Die Zusammenarbeit hat mir großen Spaß gemacht und mir ermöglicht von seiner Expertise zu lernen. Allen genannten Mitgliedern der Prüfungskommission, sowie Professor Dr. Frank Schultmann als Vorsitzendem der Prüfungskommission möchte ich außerdem meinen Dank für das Interesse sowie die eingebrachte Zeit in die Auseinandersetzung mit meiner Arbeit aussprechen.

Diese Arbeit entstand während meiner Zeit als wissenschaftlicher Mitarbeiter am Lehrstuhl für Analytics and Statistics des Institut für Operations Research am Karlsruher Institut für Technologie. Ich danke meinen Kollegen Anika Kaplan, Mira Watermeyer, Jonas Rieger, sowie Parzival Borlinghaus für die gemeinsame Zeit. Ich habe mich in jeder Kaffeepause – einschließlich der virtuellen – sehr wohlgefühlt und unseren Austausch zu schätzen gelernt. Mein Dank gilt außerdem Marion Rihm, die mit einem offenen Ohr in jeder erdenklichen Lage stets eine tolle Unterstützung war. Des Weiteren danke ich meinem Koautor Professor Dr. Maximilian Coblenz für die vielen Diskussionen und die gute Zusammenarbeit. Für die Möglichkeit des Aufenthalts an der University of Waterloo in Kanada, die Gastfreundschaft und die eingebrachte Zeit bedanke ich mich bei meinem Gastgeber Professor Dr. Marius Hofert, sowie dem Karlsruhe House of Young Scientists (KHYS) für die großzügige Förderung der Aufenthalts. Darüber hinaus gilt mein Dank den wissenschaftlichen Hilfskräften Huidi Zhu, Tim Ortkamp, Tim Seeberger,

Nora Schneider und Yanting Liu, die ich in meiner Zeit am Lehrstuhl begleiten durfte und die mich hervorragend unterstützt haben.

Außerdem möchte ich mich von Herzen bei meiner Familie und Freunden bedanken. Insbesondere bei meinen Eltern Beatrix und Stefan, die mir immer alles ermöglicht haben und stets mit Rat und Tat zur Seite standen. Nils Koster danke ich für die zahlreichen Diskussionen und die Bereitschaft, auch für ihn nicht relevante Themen gemeinsam zu durchdenken. Zuletzt danke ich meiner Freundin Katharina für Ihr Verständnis in stressigen Phasen, Ihre Geduld und Ihre Fähigkeit mich auf andere Gedanken zu bringen. Ich bin glücklich sie an meiner Seite zu haben.

Abstract

Copulas are functions that allow disentangling the dependence structure from the univariate marginal distributions of a multivariate random variable and to model each separately. Due to increasingly larger and more complex data sets, their importance in science and application has grown exponentially in recent years. This work develops new methods and applications based on copulas as well as it presents a theoretical contribution to the estimation of the empirical copula process.

More specifically, we first consider the variability of the most widely used empirical approximation of a copula, the empirical copula. In doing so, a new estimator of the covariance of the empirical copula process in arbitrary d-dimensional boxes is developed. The presented estimator is the first that does not require bootstrap procedures and fast and precise statements about the uncertainty of the results derived from empirical copulas are possible.

Subsequently, a multivariate extension of the Lorenz curve and Gini coefficient based on a theoretically derived copula decomposition is proposed. The extension determines inequality in multivariate distributions originating from both the individual marginal distributions and their dependence structure. Unlike other extensions, the presented approach can be interpreted directly and is economically tractable.

The third part results in a multivariate improvement of day-ahead price forecasts in electricity markets. Here, univariate point forecasts are augmented with the distribution of their historical errors, while a copula approach enforces the correct dependencies between hourly prices. Thus, any simple point forecasting model can be extended to forecast complex multivariate densities without much effort. Among others, this is of great importance in the context of the current transformation of the energy system, e.g., for the risk management of energy providers.

Finally, the generation of images by randomly drawing in the latent space of an autoencoder is investigated. We compare copula models with other methodologies for modeling the latent space and demonstrate how a powerful generative autoencoder can be constructed using these statistical methods. In addition to image quality, aspects such as image manipulation of the newly presented Empirical Beta Copula Autoencoder are discussed.

The topics addressed in the thesis are illustrated using extensive examples with real-world datasets and simulation studies where applicable or necessary. Further, ready-to-use computer code is made available for all contributions.

Contents

List of Figures

List of Tables

List of Abbreviations

1-NN	1-Nearest Neighbor Test
AAE	Adversarial Auto Encoder
AE	Autoencoder
CDF	Cumulative Distribution Function
CIM	Correlation Increasing Majorization
CIT	Correlation Increasing Transfer
CRPS	Continuous Ranked Probability Score
EBCAE	Empirical Beta Copula Autoencoder
EMD	Earth Mover Distance; also called Wasserstein Distance
ES	Energy Score
EUR	Euro €
FICP	Forecast Interval Coverage Probability
FINAW	Forecast Interval Normalized Average Width
GAN	Generative Adversarial Network
GARCH	Generalized AutoRegressive Conditional Heteroscedasticity
GMM	Gaussian Mixture Model
IFM	Inference From Margins
KDE	Kernel Density Estimation
LWS	Luxembourg Wealth Study
MAE	Mean Absolute Error
MEGC	Multivariate Extension of Gini Coefficient
MEILC	Multivariate Extension of the Inverse Lorenz Curve
MMD	Mean Maximum Discrepancy
MW	Mega Watt; equals 1.000.000 W
PDBT	Pigou Dalton Bundle Transfer
PDF	Probability Density Function
QR	Quantile Regression
SOEP	Socio-Economic Panel
VAE	Variational Autoencoder
VCAE	Vine Copula Autoencoder

1. Introduction

"The world's most valuable resource is no longer oil, but data."

The Economist, May 6th 2017

The amount of data in a more and more interconnected world is constantly growing (Bell et al., 2009; Dhar, 2013). Simultaneously with the rapid advancement of methods for 'refining' this often complex and multidimensional data (Donoho, 2017; Cao, 2017), dependencies steadily gain importance in modern data analytics, statistics, and machine learning. For example, one might be interested in the interaction of prices and demand, treatment and effect, or various observations in the natural sciences. *Copulas* offer a very flexible and universal way to cope with all such kinds of dependence structures in multivariate data. This thesis contributes to the literature on copula theory and develops new methodologies based on copulas. It explores new ideas to deal with dependence on multivariate data and demonstrates how to use this rich information. To that end, it marks an important step towards modern, data-driven solutions in business and research while introducing copula theory to specialists in various fields of application.

The word 'copula' originally stems from Latin meaning 'a link, tie or bond' (Simpson, 1977) and was first mentioned in the statistical literature by Sklar (1959) to describe a function that joins multiple one-dimensional distribution functions to a single multivariate distribution function. Thus, copulas allow the modeling of each univariate marginal distribution function independently from the mutual information of the joint distribution. That said, copulas may be seen as the most general form of modeling dependence on multivariate data. Over the last few years, copulas gained much attention due to their flexibility and simplicity while dealing with multivariate data. Copula techniques are employed in a wide range of applications, such as, e.g., economics and finance (Patton, 2012; Genest et al., 2009; Cherubini et al., 2004), energy modeling (Pircalabu and Benth, 2017; He et al., 2017), natural sciences and engineering (Salvadori et al., 2016; Coblenz et al., 2020), machine learning (Janke et al., 2021; Messoudi et al., 2021; Tagasovska et al., 2019), or forecasting (Schefzik et al., 2013; Clark et al., 2004). This thesis adds further copula-based methodologies to this non-extensive list while deepening the overall understanding of copulas from a theoretical perspective.

In the following, the contributions of the thesis are shortly outlined and introduced in more detail in the paragraphs below. In the next chapter, we establish some relevant preliminaries of this work. Most important, we introduce copulas and their estimation. Further, we introduce some notation and conventions which we follow throughout this thesis. However, some additional notation is provided in most of the chapters, enabling interested readers to study the chapters independently from the rest of the thesis. In Chapter 3, we deal with a new plug-in estimator for the variance of the empirical copula. Chapter 4 utilizes copula representations to construct a multivariate extension of the well-known Gini index to measure multivariate inequality. Then, in Chapter 5, we introduce a new copula method for probabilistic forecasting of electricity prices based on univariate point forecasts. Last, Chapter 6 combines copulas with autoencoders and investigates whether modeling the latent space via copula methodologies is suitable for data generation before Chapter 7 concludes the thesis.

On a more detailed level, the contributions of each chapter are as follows. Chapter 3 is based on joined work with Oliver Grothe and Melanie Schienle. There, we consider the most common estimate of a copula, the empirical copula. Although the asymptotics of the empirical copula process is widely known, only a few estimators for the covariance of this process exist. In the chapter, we present a new plug-in estimator for the covariance of the empirical copula process. The estimator follows directly from the empirical copula processes' asymptotic behavior and uses a given sample from the empirical copula for its estimate. Notably, the presented estimator is the first in the literature that works without using bootstrap procedures. Additionally, we extend the estimator so that it can be applied for estimating covariances of arbitrary partitions in the empirical copula. Such an estimator can be used to determine the variability of any function based on the empirical copula, e.g., the survival copula or orthant probabilities, as it is the case in insurance or risk management applications. We illustrate the good finite sample properties of the new estimator for dimensions $d \geq 2$ in a simulation study and compare the performance with existing but more burdensome competitors. Further, we provide ready-to-use computer code to enable researchers and practitioners to easily work with our rather complicated estimator.

Following in Chapter 4, we propose an extension of the univariate Lorenz curve and the Gini coefficient to the multivariate case. These two measures are widely spread and used to assess inequality in any kind of distribution, e.g., in income or wealth. Our extensions allow measuring inequality in more than one variable simultaneously and are based on copulas as well. The extensions measure inequality stemming from inequality in every single variable as well as inequality stemming from the dependence structure of the variables. Simple non-parametric estimators for both instruments are derived and exemplary applied to data on individual income and wealth for various countries. In contrast to other extensions, our approach is easy to interpret and follows clear economic reasoning. The chapter is based on the joint paper with Oliver Grothe and Friedrich Schmid *published* in the *Journal of Economic Inequality* (Grothe et al., 2022a) and computer code is provided in the appendix.

Next, Chapter 5 deals with the topic of forecasting electricity prices and is based on a joint work with Oliver Grothe and Fabian Krüger *published* in *Energy Economics* (Grothe et al., 2023). There, we consider the probabilistic forecast of hourly day-ahead electricity prices, which naturally exhibit a temporal dependence structure during the day. The method we propose is based on the historical errors of 24 univariate point forecasts from each hour, while we include optional time series components to adjust for systematic errors. A set of 24 univariate day-ahead point forecasts is now supplemented by the previously learned distribution of individual errors to form the multivariate probabilistic forecast for electricity prices of the next day. Within this forecast distribution, the correct temporal dependencies across hours are handled by an appropriate copula model. We illustrate the method for five benchmark datasets, including different electricity markets in Europe and the United States. Furthermore, we demonstrate the simplicity and superior performance of the approach with an example of constructing realistic prediction intervals for the weighted sum of consecutive electricity prices as needed for pricing individual load profiles. Computer code and information to reproduce all results are also provided in the appendix.

Then, Chapter 6 is based on a joint work with Maximilian Coblenz and Oliver Grothe.In the chapter, we combine copulas with autoencoders to craft a new, powerful model for data generation and compare it to other methods paired with autoencoders. By sampling from the latent space of an autoencoder and decoding the samples to the original data space, any autoencoder can simply be turned into a generative model. To do so, it is necessary to model the autoencoders' latent space with a distribution from which samples can be obtained. This is the first study, which aims to discuss, assess, and compare various techniques, including copulas, that can be used to capture the latent space and construct a powerful generative model. Among them, the new Empirical Beta Copula Autoencoder, which combines non-parametric copula estimation with classical kernel density estimation for the marginal distributions. Furthermore, we provide insights into other aspects of the methods under investigation, such as targeted sampling or synthesizing new data with specific features as well as computer code. Last, Chapter 7 concludes the thesis and additional insights for interested readers to all chapters are given in the appendices.

2. Preliminaries on Copulas

This section shortly introduces the basic statistical concepts used in this thesis, mainly copulas. Besides providing a comprehensive introduction to copulas, it is intended to serve as a point of reference for the reader while coping with the single chapters of this thesis. In this work, we will denote random variables in capital letters and corresponding realizations in small letters. Further, vectors and matrices are written boldly. For better readability, we use parenthesis in the given order, $[\{(\cdot)\}]$, within the thesis. For a throughout theoretical treatment of copulas, we recommend the books by Nelsen (2006), Joe (2014), and Durante and Sempi (2015). The section is based on the textbooks above.

2.1. Basics

Let $\boldsymbol{X} = (X_1, \ldots, X_d)$ be a d-variate random vector, $d \geq 2$, defined on a probability space $(\Omega, \Sigma, \mathbb{P})$. The joint distribution function is given by $F(\mathbf{x}) = P(X_1 \leq x_1, \ldots, X_d \leq x_d)$ for $\mathbf{x} = (x_1, \ldots, x_d) \in \mathbb{R}^d$ and the marginal distribution functions F_k of X_k are given by $F_k(x_k) = P(X_k \leq x_k)$ for $x_k \in \mathbb{R}$ and $k \in \{1, \ldots, d\}$, which are assumed to be continuous. Sklar's theorem (Sklar 1959) then states that for every d-dimensional cumulative distribution function $F(\mathbf{x})$ with continuous, univariate margins $F_1, \ldots, F_d$ there exists a unique *copula* C such that $F(\mathbf{x})$ can be decomposed by

$$F(x_1, \ldots, x_d) = C\big\{F_1(x_1), \ldots, F_d(x_d)\big\} \tag{2.1}$$

for all $x_k \in \mathbb{R}$ and $k \in \{1, \ldots, d\}$. We see that the transformation $\big\{F_1(x_1), \ldots, F_d(x_d) = (u_1, \ldots, u_d)\big\}$ maps each value of $\mathbf{x} = (x_1, \ldots, x_d) \in \mathbb{R}$ to its corresponding quantile level

$\mathbf{u} = (u_1, \dots, u_d) \in [0,1]^d$. Consequently, a second point of view on copulas is that copulas are distribution functions with one-dimensional marginals, uniformly distributed over $[0,1]^d$. From Equation (2.1) follows, that we can disentangle $\boldsymbol{X} \in \mathbb{R}^d$ into its marginal distribution functions F_k for $k = 1, \dots, d$ and its dependence structure, which is fully governed by the copula C. Further, the expression in Equation (2.1) can be rewritten by

$$C(u_1, \dots, u_d) = F\big\{F_1^{-1}(u_1), \dots, F_d^{-1}(u_d)\big\} \tag{2.2}$$

with F_k^{-1} denoting the *quantile* or *percentile function* function for $k \in \{1, \dots, d\}$. Note that this simply corresponds to Equation (2.1) after using simple manipulations. Sklar's theorem consequently implies that any function F, which can be expressed by a copula C and corresponding univariate distribution functions F_k with $k = 1, \dots, d$, is a distribution function. We use both representations of a copula (Equation 2.1 and Equation 2.2) in this thesis, whichever is more convenient. Another more technical definition can be given by considering the properties of a copula. A copula is a function that is (1) *grounded*, i.e., $C(u_1, \dots, u_d) = 0$ if at least one $u_k = 0$ for $k \in \{1, \dots, d\}$, (2) has *uniform marginal distributions*, i.e., $C(1, \dots, 1, u_k, 1, \dots, 1) = u_k$ for all $u_k \in [0,1]$ and $k \in \{1, \dots, d\}$, and the (3) *c-volume* is non-negative. The c-volume is defined by

$$V_C = \sum_{\mathbf{v} \in [\{u_1, v_1\} \times \cdots \times \{u_d, v_d\}]} \operatorname{sgn}(\mathbf{v})\; C(\mathbf{v}) \tag{2.3}$$

where

$$\operatorname{sgn}(\mathbf{v}) = \begin{cases} 1, & \text{if } v_j = a_j \text{ for an even number of indices,} \\ -1, & \text{if } v_j = a_j \text{ for an odd number of indices,} \end{cases}$$

and $[\{u_1, v_1\} \times \cdots \times \{u_d, v_d\}]$ is the set of vertices on $[0,1]^d$, with $u_i \leq v_k$ for $k = 1, \dots, d$. Put simply, the c-volume can be interpreted as probability mass in any rectangular box of the copula, which has to be non-negative. We use this volume-computation idea later in Chapter 3 of this thesis to construct a flexible estimator for arbitrary boxes in the copula domain.

Any copula is limited pointwise perfect upper (M) and lower (W) dependence in the sense of $W(\mathbf{u}) \leq C(\mathbf{u}) \leq M(\mathbf{u})$ for all $\mathbf{u} = (u_1, \ldots, u_d) \in [0,1]^d$. By perfect lower dependence we mean the *lower Fréchet-Hoeffding bound* given by

$$W(u_1, \ldots, u_d) = \max\left\{ \sum_{k=1}^{d} u_k \ - d + 1 \ ; \ 0 \right\}, \tag{2.4}$$

which is a universal lower bound for copulas, but only a copula itself for $d = 2$. In contrary, the *upper Fréchet-Hoeffding bound* is given by

$$M(u_1, \ldots, u_d) = \min\left\{ u_1, \ldots, u_d \right\}. \tag{2.5}$$

Figure 2.1 visualizes both extremes in a two-dimensional setting, depicting scatter plots of samples from both copulas with perfect lower (negative) and upper (positive) dependence. In the middle of the figure, an example of the *independence copula* Π is depicted, which reflects stochastic independence. This copula is also called *product-copula* since it is constructed as the product of its arguments, i.e.,

$$\Pi(u_1, \ldots, u_d) = \prod_{k=1}^{d} u_k \tag{2.6}$$

for $u_k \in [0,1]$ and $k \in \{1, \ldots, d\}$. Note that the two limits W and M are beneficial in many situations, e.g., where they induce the minimal and maximal values of copula-based measurements as for the multivariate Gini coefficient developed in Chapter 4 of this thesis.

In the following, we shortly introduce some basic parametric copula models. For a throughout treatment of different copula models and families of copulas, we refer the reader to Chapter 6 in Durante and Sempi (2015). The probably most common copula is the *Gaussian Copula*, which is constructed using Equation (2.2) and the *normal*

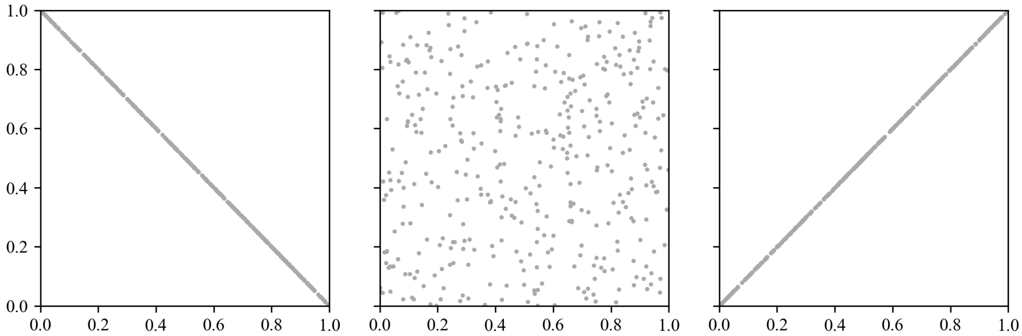


Figure 2.1.: From left to right: Samples from the lower Fréchet-Hoeffding bound W, the independence copula Π and the upper Fréchet-Hoeffding bound M.

distribution. Let Φ_ρ be the CDF of a multivariate normal distribution with standard normal margins and $d \times d$ correlation matrix $\boldsymbol{\rho}$, and ϕ^{-1} the inverse of the univariate standard normal CDF. The Gaussian copula is then given for $\mathbf{u} = (u_1, \ldots, u_d) \in [0,1]^d$ by

$$C(u_1, \ldots, u_d) \;=\; \Phi_\rho\big\{\phi^{-1}(u_1), \ldots, \phi^{-1}(u_d)\big\}. \tag{2.7}$$

while its dependence is controlled by correlation matrix $\boldsymbol{\rho}$. Similarly, the *t-copula* can be constructed using the CDF of a *multivariate Students-t* distributed variable $t_{\Sigma,v}$ with v degrees of freedom, the positive-definite scatter or dispersion matrix Σ and the inverse CDF of t-distributed variables t_v^{-1}, namely

$$C(u_1, \ldots, u_d) \;=\; t_{\Sigma,v}\big\{t_v^{-1}(u_1), \ldots, t_v^{-1}(u_d)\big\} \tag{2.8}$$

for $(u_1, \ldots, u_d) \in [0,1]^d$. Note in Figure 2.2, that similar to the Students-t distribution, the t-copula depicts more probability mass in the tails, i.e., in non-diagonal areas, compared to the Gaussian copula. Both copulas are considered as *elliptical* copula models since their base distribution is elliptical.

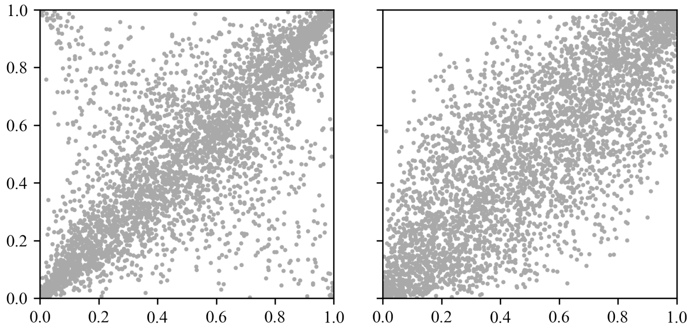


Figure 2.2.: Sample from the t-Copula with $v = 1$ degrees of freedom (left) compared to a sample from a Gaussian copula (right), both with Spearman's $\rho = 0.8$.

Another type of copulas used in this thesis are *Archimedean Copulas*. Archimedean copulas are constructed for $\mathbf{u} = (u_1, \ldots, u_d) \in [0,1]^d$ by

$$C(u_1, \ldots, u_d) \;=\; \psi\big\{\psi^{-1}(u_1) \;+\; \ldots \;+\; \psi^{-1}(u_d)\big\}, \tag{2.9}$$

where we call ψ the generator, or generator-function. The generator is a continuous, decreasing function from $[0, +\infty]$ to $[0, 1]$. It further fulfills (1) $\psi(0) = 1$ and (2) $\psi(+\infty) = 0$, while it (3) is strictly decreasing on $[0, t]$, where $t = \inf\{t > 0 : \psi(0) = 0\}$. The most common Archimedean copula models are the *Clayton copula,* the *Gumbel copula* and the *Frank copula.* Let θ be the parameter of the copula, the generator of the Clayton copula is $\psi(u) = (1+u)^{-1/\theta}$ for $\theta = (0, \infty)$, the generator of the Gumbel copula is $\psi(u) = exp(-u^{1/\theta})$ for $\theta = [1, \infty)$ and the generator of the Frank copula is $\psi(u) = -1/\theta \ln\big\{1 - (1 - e^{\theta})e^{-u}\big\}$ for $\theta = (0, \infty)$. Note that an increase in θ results in an increase of rank-based measures of dependence like Spearmans's ρ or Kendal's τ. See, e.g., Chapter 5 in Nelsen (2006) an overview of this and other copula-based dependence measures and Hofert (2008) for details and challenges in sampling from Archimedean copulas. Figure 2.3 visualizes samples from the three most common Archimedean copula models mentioned above.

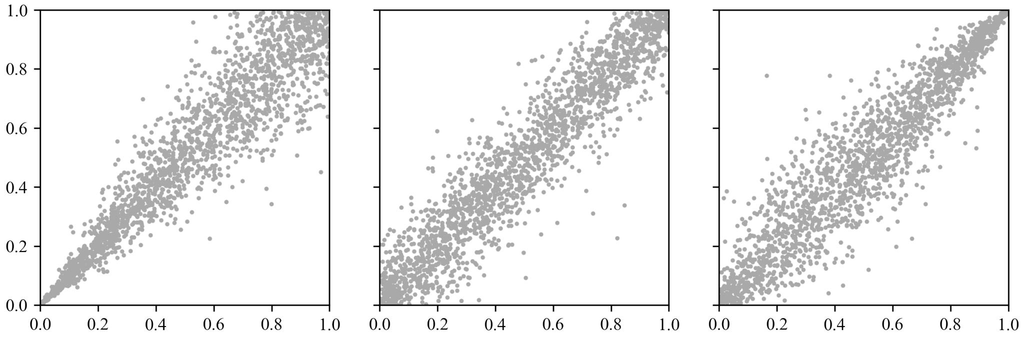


Figure 2.3.: From left to right: Sample from the Clayton copula, Frank copula and Gumbel copula, all with a dependence of Kendal's $\tau = 0.8$.

2.2. Estimation of Copulas

In this section, we consider different approaches of estimating the copula C from random vector $\boldsymbol{X} = (X_1, \dots, X_d)$ of which we observe n samples $\boldsymbol{X}_1, \dots, \boldsymbol{X}_n$. The estimation of C can be done in a fully parametric, semi-parametric, or non-parametric way, which we will address shortly in the following. For an overview on estimation, we recommend the excellent paper from Genest and Favre (2007a). In the following, we focus on the more common two-step approach while we refer to Section 10 in Joe (2014) for a treatment of the one-step, full parametric maximum-likelihood estimation.

In the two-step approach, we estimate the copula C of $\boldsymbol{X}$ based on *pseudo-observations* $\hat{\mathbf{u}}_j = (\hat{U}_{j1}, \dots, \hat{U}_{jd})$ for $j = 1, \dots, n$. In the full and semi-parametric setting, the pseudo-observations are calculated by assuming the estimated marginal distributions $\hat{F}_1, \dots, \hat{F}_d$ follow a given distribution,

$$\hat{\mathbf{u}}_j = \left\{\hat{F}_1(X_{j1}), \dots, \hat{F}_d(X_{jd})\right\} \tag{2.10}$$

where X_{jk} denotes the kth element within the jth observation $\boldsymbol{X}_j$. This approach is sometimes also called 'inference from margins' (IFM). However, if the parametric models of $\hat{F}_1, \dots, \hat{F}_d$ are wrongly specified, resulting pseudo-observations and, consequently, C, is subject to miss-specification (Genest and Favre, 2007a).

Alternatively, pseudo-observations $\hat{\mathbf{u}}_j$ for $j = 1, \ldots, n$ are estimated non-parametrically by using ranks, i.e.,

$$\hat{\mathbf{u}}_j = \left(\hat{U}_{j1}, \ldots, \hat{U}_{jd}\right) = \left\{\frac{\mathrm{rk}(X_{j1})}{n}, \ldots, \frac{\mathrm{rk}(X_{jd})}{n}\right\}, \tag{2.11}$$

where $\mathrm{rk}(\cdot)$ denotes the rank of each X_{jk} within all n observations $(X_{1k}, \ldots, X_{nk})$ for $k = 1, \ldots, d$. Note that this corresponds to replacing the unknown marginal distribution functions F_k by their empirical marginal distribution functions $\hat{F}_k(x_k) = \frac{1}{n}\sum_{j=1}^{n} \mathbf{1}\{X_k \leq x_k\}$ for $k \in \{1, \ldots, d\}$ and $x_k \in \mathbb{R}$. We use this fully rank-based approach, e.g., in Chapter 5 of this thesis for predicting day-ahead electricity prices. Further note that in applications, pseudo-observations $\hat{\mathbf{u}}_j$ are often estimated using $n+1$ in the denominator of Equation (2.11) for practical purposes. This is a slight adjustment that prevents actually having to deal with the 100% quantile.

In the second step, C is estimated based on the previously computed pseudo-observations and the estimator is called $\hat{C}$ in the following. The parametric estimation approach of C utilizes the pseudo-observations from Equation (2.10) (or from Equation 2.11 in the semi-parametric case) and employs maximum-likelihood methods to estimate copula parameters θ for a given parametric copula model (Genest et al., 1995). Alternatively, copula parameters θ can be estimated via the method of moments, which we later do in Chapter 5 using *Spearman's* ρ to parameterize a Gaussian copula. See Tsukahara (2005a) and Joe (2005) for details or Hofert et al. (2012) for estimation in the case of archimedean copulas.

For the non-parametric estimator of C, we focus on the *empirical copula* denoted by $\hat{C}_n$. It does not assume any parametric form of the copula and is defined as the empirical

CDF of pseudo-observations from Equation (2.11), i.e.,

$$\hat{C}_n(\mathbf{u}) = \frac{1}{n} \sum_{j=1}^{n} \mathbf{1}\{\hat{\mathbf{u}}_j \leq \mathbf{u}\} \tag{2.12}$$

for $\mathbf{u} \in [0,1]^d$. This definition coincides with estimating $\hat{C}_n$ directly on the ranks of the data by changing the condition in the indicator functions to ranks. Chapter 3 deals with the variance related to the asymptotic process of this simple but widely spread estimator. Further, it might be useful to work with a smoothed version of the resulting step function given by Equation (2.12) as done by using the *empirical beta copula* in Chapter 6 of this thesis.

3. Approximation of the Empirical Copula Process Revisited: A new Plug-In Estimator

The following chapter is based on joint work with Oliver Grothe and Melanie Schienle and was already presented at the *International Conference of Ordered Statistical Data 2022* in Napoli (Italy) and at the *German Probability and Statistic Days 2023* in Essen (Germany). The chapter is the most theoretically orientated in this thesis but is still equipped with an extensive simulation study and ready-to-use computer code for practitioners.

3.1. Introduction

When the true underlying copula C of data at hand is not known but of interest, the copula has to be estimated. Several estimation procedures have been introduced in the previous chapter (Chapter 2.2). In most cases, the empirical copula $\hat{C}_n$ is used as a simple estimator for the unknown copula C. The related asymptotics of the empirical copula process is known to weakly converge to a Gaussian field $\mathbb{G}_C$, as investigated by several authors (Stute, 1984; Ruschendorf, 1976; Bücher and Volgushev, 2013; Fermanian et al., 2004; Genest and Segers, 2009; Segers, 2012; Tsukahara, 2005b; Gaenssler and Stute, 1987). However, its corresponding covariance structure includes the unknown copula function and its derivatives. Therefore, estimating covariances is not easy, and existing proposals are based on bootstrap procedures. See, e.g., Bücher and Dette (2010) for an overview and a comparative simulation study.

This chapter makes two contributions toward covariance estimation in empirical copulas. First, we present a plug-in estimator for empirical copula process variances and covariances. It is constructed using the well-known theory regarding the empirical copula process. We illustrate its good finite sample properties in a simulation study and compare it with results obtained in a study from Bücher and Dette (2010) for bootstrap procedures. Secondly, we provide an expression for the exact computation and estimation of variances and covariances from arbitrary partitions in the domain of an empirical copula process, i.e., rectangles in the domain of a d-dimensional empirical copula $[0,1]^d$. The formula emerges from the introduced covariance arithmetics and the elegant approach from Cherubini and Romagnoli (2009) for the computation of copula partitions, i.e., the volume as introduced in Chapter 2. The variance and covariance of such partitions in the empirical copula process are of particular interest, e.g., for the computation of the variance of a survival copula. Additionally, ready-to-use computer code is provided in the appendix.

The remainder of the chapter proceeds as follows. Section 3.2 reiterates and completes some theoretical foundations and necessary notation from Chapter 2. Most important, the empirical copula process and its covariance structure are introduced in this section. In Section 3.3, we derive the plug-in estimator for the covariance of empirical copulas. Then, in Section 3.4, an estimator for the variance of arbitrary partitions in the d-dimensional empirical copula process is presented. Simulations demonstrate the practical ability of the derived estimator in Section 3.5 and Section 3.6 concludes the chapter.

3.2. Preliminaries and Copulas

Be $\boldsymbol{X} = (X_1, \ldots, X_d)$ a d-variate random vector with $d \geq 2$ defined on a probability space $(\Omega, \Sigma, \mathbb{P})$ of which we observe n iid copies $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$. The joint distribution function of $\boldsymbol{X}$ is given by $F(\mathbf{x}) = \Pr(X_1 \leq x_1, \ldots, X_d \leq x_d)$ and the marginal distribution functions

F_k of X_k are given by $F_k(x_k) = \Pr(X_k \leq x_k)$ for $\mathbf{x} = (x_1, \ldots, x_d) \in \mathbb{R}^d$. Throughout the chapter, we use $\rightsquigarrow$ to indicate weak convergence and $\wedge$ to mark the element-wise minimum of a vector.

As presented in Chapter 2, every d-dimensional cumulative distribution function $F(\mathbf{x})$ with continuous, univariate margins $F_1, \ldots, F_d$ can be decomposed by $F(x_1, \ldots, x_d) = C\big\{F_1(x_1), \ldots, F_d(x_d)\big\}$ for all $x_k \in \mathbb{R}$ and $k \in \{1, \ldots, d\}$, where C is the corresponding copula (Sklar, 1959). Further, remember from Chapter 2 that the simplest non-parametric estimator for the copula C is given by the *empirical copula* $\hat{C}_n$. With a slightly different notation than previously introduced, we estimate $\hat{C}_n$ based on (normalized) ranks, i.e., empirical quantile levels, of each marginal distribution (see Deheuvels 1979)

$$\hat{C}_n(\mathbf{u}) = 1/n \sum_{j=1}^{n} \prod_{k=1}^{d} \mathbf{1}\left\{\frac{r_{j,k}}{n} \leq u_k\right\}, \tag{3.1}$$

for any $\mathbf{u} = (u_1, \ldots, u_d) \in [0,1]^d$. Further, $r_{j,k}$ denotes ranks within the copies $X_{1k}, \ldots, X_{nk}$ i.e., $r_{j,k} = \sum_{i=1}^{n} \mathbf{1}\left\{X_{i,k} \leq X_{j,k}\right\}$ for $k \in \{1, \ldots, d\}$. Generally, for the empirical copula process

$$\mathbb{C}_n(\mathbf{u}) = \sqrt{n}\big\{\hat{C}_n(\mathbf{u}) - C(\mathbf{u})\big\}. \tag{3.2}$$

holds a weak convergence result for $n \to \infty$ to a continuous centered Gaussian field $\mathbb{G}_C$ on $[0,1]^d$ as stated in the following theorem.

Theorem 3.2.1. *(Segers 2012) Assume that for each $k \in \{1, \ldots, d\}$ there exist a continuous partial derivative $\partial_k C$ on the set $V_{d,k} := \{\mathbf{u} \in [0,1]^d : 0 < u_k < 1\}$, we get for $n \to \infty$:*

$$\mathbb{C}_n(\mathbf{u}) = \sqrt{n}\big\{\hat{C}_n(\mathbf{u}) - C(\mathbf{u})\big\} \rightsquigarrow \mathbb{G}_C \textit{ in } l^{\infty}([0,1]^d), \tag{3.3}$$

where

$$\mathbb{G}_C(\mathbf{u}) = \mathbb{B}(\mathbf{u}) - \sum_{k=1}^{d} \partial_k C(\mathbf{u})\mathbb{B}(\mathbf{u}^{(k)}), \tag{3.4}$$

with $\mathbf{u}^{(k)} = (1, ..., 1, u_k, 1, \; ..., 1)$ *for* $k \in \{1, \ldots, d\}$ *and* $\mathbb{B}$ *is a tight limiting Gaussian process on* $[0,1]^d$ *with*
cov$\Big\{\mathbb{B}(\mathbf{u}), \mathbb{B}(\mathbf{v})\Big\} = C(\mathbf{u} \wedge \mathbf{v}) - C(\mathbf{u})C(\mathbf{v})$ *for any* $\mathbf{u}, \mathbf{v} \in [0,1]^d$.

See Segers (2012) for further details and the proof of the theorem. Note that Bücher and Volgushev (2013) further showed that the above result also holds for some copulas under certain serial dependence. $\mathbb{G}_C(\mathbf{u})$ in Equation (3.4) can be interpreted as follows. $\mathbb{B}(\mathbf{u})$ characterizes the process if the true margins F_k were used instead of the empirical version $\hat{F}_k$, i.e., estimating empirical quantile levels by using ranks. Thus, the product of the partial derivatives $\partial_k C(\mathbf{u})$ and the 'marginalized' process $\mathbb{B}(\mathbf{u}^{(k)})$ can be referenced as resulting 'discrepancy' of not knowing the true marginal distributions F_k for $k \in \{1, \ldots, d\}$.

3.3. Plug-in Estimator

We now introduce the proposed *plug-in estimator* for the covariance of the empirical copula. For this purpose, we estimate the covariance of the empirical copula process, where the true copula C is replaced by its empirical counterpart $\hat{C}_n$. The covariance of an empirical copula $\hat{C}_n$ is then estimated by the scaled estimated covariance of the empirical copula process, i.e.,

$$\widehat{\text{cov}}\Big\{\hat{C}_n(\mathbf{u}), \hat{C}_n(\mathbf{v})\Big\} := \frac{1}{n} \cdot \widehat{\text{cov}}\Big\{\mathbb{C}_n(\mathbf{u}), \mathbb{C}_n(\mathbf{v})\Big\}. \tag{3.5}$$

The covariance of the empirical copula process and its asymptotics are outlined in the following. From the previous section, we know that the covariance of the empirical copula process $\mathbb{C}_n$ (see Equation 3.3 and Equation 3.4) can be expressed by

$$\operatorname{cov}\Big\{\mathbb{G}_C(\mathbf{u}), \mathbb{G}_C(\mathbf{v})\Big\} = \operatorname{cov}\Big\{\mathbb{B}(\mathbf{u}) - \sum_{k_1=1}^{d} \partial_{k_1} C(\mathbf{u})\mathbb{B}(\mathbf{u}^{(k_1)}), \mathbb{B}(\mathbf{v}) - \sum_{k_2=1}^{d} \partial_{k_2} C(\mathbf{v})\mathbb{B}(\mathbf{v}^{(k_2)})\Big\}, \tag{3.6}$$

with $\mathbf{u} = (u_1, \dots, u_d)$ and $\mathbf{v} = (v_1, \dots, v_d)$ for $d \in \{1, \dots, d\}$. This can now be further dissolved into:

$$\begin{aligned}\operatorname{cov}\Big\{\mathbb{G}_C(\mathbf{u}), \mathbb{G}_C(\mathbf{v})\Big\} &= C(\mathbf{u} \wedge \mathbf{v}) - C(\mathbf{u})C(\mathbf{v}) \\ &+ \sum_{k_1=1}^{d} \sum_{k_2=1}^{d} \partial_{k_1} C(\mathbf{u}) \partial_{k_2} C(\mathbf{v}) \Big\{ C(\mathbf{u}^{(k_1)} \wedge \mathbf{v}^{(k_2)}) - C(\mathbf{u}^{(k_1)})C(\mathbf{v}^{(k_2)})\Big\} \\ &+ \sum_{k_1=1}^{d} \partial_{k_1} C(\mathbf{u}) \Big\{ C(\mathbf{u}^{(k_1)} \wedge \mathbf{v}) - C(\mathbf{u}^{(k_1)})C(\mathbf{v})\Big\} \\ &+ \sum_{k_2=1}^{d} \partial_{k_2} C(\mathbf{v}) \Big\{ C(\mathbf{u} \wedge \mathbf{v}^{(k_2)}) - C(\mathbf{u})C(\mathbf{v}^{(k_2)})\Big\}.\end{aligned}$$

Step-by-step transformations of the equation above are given in the appendix to this chapter at the end of this thesis.

Note that $\operatorname{cov}\big\{\mathbb{G}_C(\mathbf{u}), \mathbb{G}_C(\mathbf{v})\big\}$ still contains the true underlying copula C and, thus, is not an estimator so far. To obtain the plug-in estimator $\widehat{\operatorname{cov}}\big\{\mathbb{C}_n(\mathbf{u}), \mathbb{C}_n(\mathbf{v})\big\}$, we estimate the desired covariance by replacing the true copula C with its empirical counterpart $\hat{C}_n$. The partial derivatives $\partial_k C(\mathbf{u})$ are also approximated by their empirical counterparts similar to Rémillard and Scaillet (2009), i.e.,

$$\widehat{\partial_k C_n(\mathbf{u})} = \frac{\hat{C}_n(u_1, \dots, u_k + h, \dots, u_d) - \hat{C}_n(u_1, \dots, u_k - h, \dots, u_d)}{2h}, \tag{3.7}$$

for small $h < 0$, where we set $h =: (1/\sqrt{n})$ and $k \in \{1, \dots, d\}$.

Consequently assuming the existence of continuous partial derivatives as defined in 3.2.1, an estimator for the asymptotic covariance of the empirical copula $\hat{C}_n$ is given by

$$\begin{aligned}\widehat{\text{cov}}\Big\{\mathbb{C}_n(\mathbf{u}), \mathbb{C}_n(\mathbf{v})\Big\} :=\ & \hat{C}_n(\mathbf{u} \wedge \mathbf{v}) - \hat{C}_n(\mathbf{u})\hat{C}_n(\mathbf{v}) \\ & + \sum_{k_1=1}^{d}\sum_{k_2=1}^{d} \widehat{\partial_{k_1} C_n(\mathbf{u})}\widehat{\partial_{k_2} C_n(\mathbf{v})}\Big\{\hat{C}_n(\mathbf{u}^{(k_1)} \wedge \mathbf{v}^{(k_2)}) - \hat{C}_n(\mathbf{u}^{(k_1)})\hat{C}_n(\mathbf{v}^{(k_2)})\Big\} \\ & + \sum_{k_2=1}^{d} \widehat{\partial_{k_2} C_n(\mathbf{v})}\Big\{\hat{C}_n(\mathbf{u} \wedge \mathbf{v}^{(k_2)}) - \hat{C}_n(\mathbf{u})\hat{C}_n(\mathbf{v}^{(k_2)})\Big\}.\end{aligned} \tag{3.8}$$

Theorem 3.3.1 below states the asymptotics of the presented estimator providing the respective weak convergence results.

Theorem 3.3.1. *Under assumption defined in Theorem 3.2.1, we get for $n \to \infty$:*

$$\widehat{cov}\Big\{\mathbb{C}_n(\mathbf{u}), \mathbb{C}_n(\mathbf{v})\Big\} \rightsquigarrow cov\Big\{\mathbb{C}_n(\mathbf{u}), \mathbb{C}_n(\mathbf{v})\Big\}.$$

Proof. Since $\hat{C}_n(\mathbf{u}) \rightsquigarrow C(\mathbf{u})$ and $\widehat{\partial_k C_n(\mathbf{u})} \rightsquigarrow \partial_k C(\mathbf{u})$ for $k \in \{1, \ldots, d\}$ for $n \to \infty$ in $\mathbf{u} \in [0,1]^d$, each element of the sum converges towards the true value (Rémillard and Scaillet, 2009), which implies a weak convergence of the total estimator to the true value, i.e., an element-wise convergence. □

Example 3.3.1 illustrates the plug-in procedure and variance estimation in a two-dimensional copula.

Example 3.3.1. *Consider a two-dimensional empirical copula $\hat{C}_n$, $d = 2$, and derive the estimator for the variance of its empirical copula process $\mathbb{C}_n$ at $\mathbf{u} = (u_1, u_2) \in [0,1]^2$ by*

$$\begin{aligned} cov\Big\{\mathbb{G}_C(\mathbf{u}), \mathbb{G}_C(\mathbf{u})\Big\} = var\Big\{\mathbb{G}_C(\mathbf{u})\Big\} &= var\Big\{\mathbb{B}(\mathbf{u}) - \sum_{i=1}^{2} \partial_i C(\mathbf{u})\mathbb{B}(\mathbf{u}^{(i)})\Big\} \\ &= var\Big\{\mathbb{B}(\mathbf{u})\Big\} + var\Big\{\partial_1 C(\mathbf{u})\mathbb{B}(u_1, 1)\Big\} + var\Big\{\partial_2 C(\mathbf{u})\mathbb{B}(1, u_2)\Big\} \\ &\quad - 2 \cdot cov\Big\{\mathbb{B}(\mathbf{u}), \partial_1 C(\mathbf{u})\mathbb{B}(u_1, 1)\Big\} - 2 \cdot cov\Big\{\mathbb{B}(\mathbf{u}), \partial_2 C_n(\mathbf{u})\mathbb{B}(1, u_2)\Big\} \\ &\quad + 2 \cdot cov\Big\{\partial_1 C(\mathbf{u})\mathbb{B}(u_1, 1), \partial_2 C(\mathbf{u})\mathbb{B}(1, u_2)\Big\}. \end{aligned} \tag{3.9}$$

Dissolve and plug-in the empirical copula $\hat{C}_n$ to obtain the estimator.

$$\begin{aligned} \widehat{var}\Big\{\mathbb{C}_n(\mathbf{u})\Big\} = &\Big\{\hat{C}_n(u_1, u_2) - \hat{C}_n(u_1, u_2)\hat{C}_n(u_1, u_2)\Big\} \\ &+ \widehat{\partial_1 C_n(\mathbf{u})}^2 \Big\{\hat{C}_n(u_1, 1) - \hat{C}_n(u_1, 1)\hat{C}_n(u_1, 1)\Big\} \\ &+ \widehat{\partial_2 C_n(\mathbf{u})}^2 \Big\{\hat{C}_n(1, u_2) - \hat{C}_n(1, u_2)\hat{C}_n(1, u_2)\Big\} \\ &- 2\widehat{\partial_1 C_n(\mathbf{u})}\Big\{\hat{C}_n(u_1, u_2) - \hat{C}_n(u_1, u_2)\hat{C}_n(u_1, 1)\Big\} \\ &- 2\widehat{\partial_2 C_n(\mathbf{u})}\Big\{\hat{C}_n(u_1, u_2) - \hat{C}_n(u_1, u_2)\hat{C}_n(1, u_2)\Big\} \\ &+ 2\widehat{\partial_1 C_n(\mathbf{u})}\widehat{\partial_2 C_n(\mathbf{u})}\Big\{\hat{C}_n(u_1, u_2) - \hat{C}_n(u_1, 1)\hat{C}_n(1, u_2)\Big\}. \end{aligned} \tag{3.10}$$

Simulations have shown that in general covariance terms in Equation (3.9) and the partial derivatives in Equation (3.10) can have substantial size effects on the resulting estimate and should not be neglected.

Remark 3.3.1. *Calculating the variance without the product of the partial derivatives $\partial_k C(\mathbf{u})$ and the 'marginalized' process $\mathbb{B}(\mathbf{u}^{(k)})$ for $\mathbf{u} \in [0,1]^d$, i.e., assuming to know the true marginal distributions F_k for $k \in \{1, \ldots, d\}$, corresponds to calculating the variance* *of a Bernoulli distribution. More specifically, if p corresponds to the value of the underlying copula model at $\mathbf{u}$, the variance of the empirical copula process at $\mathbf{u}$ is calculated by $p(1-p)$ similar to a Bernoulli distribution with parameter p. Consequently, the simple plug-in estimator for the case with known marginal distributions can be computed from a sample with size n by estimating p via $\hat{p}(\mathbf{u}) = 1/n \sum_{i=1}^{n} \mathbf{1}_{\{\mathbf{u}_i \leq \mathbf{u}\}}$, where $\mathbf{u}_i$ is the exact quantile level vector for a observation $\boldsymbol{X}_i$ in the sample, i.e., $\mathbf{u}_i = \big\{F_1(X_{i1}), F_2(X_{i2}), \ldots, F_d(X_{id})\big\}$ (note that this corresponds to the IFM estimation approach from Chapter 2.2 with known marginal distributions). This can also be interpreted as the variance of getting exactly $\hat{p}/n$ successes (single observation of copula sample is smaller or equal to the values of $\mathbf{u}$ in each element) in n independent Bernoulli trials.*

3.4. (Co-)Variance of Copula Partitions

Based on the plug-in estimator derived above, we introduce an algorithm for estimating variances of arbitrary copula partitions, i.e., variances of the empirical copula process in any d-dimensional rectangle of the empirical copula in $[0,1]^d$. Computation of $d-$dimensional copula partitions, which are needed for the estimator, is very cumbersome, and thus, we first recapitulate an elegant algorithm to do so based on binary representations of integers. In the second step, we combine this algorithm with the plug-in estimator to estimate variances for arbitrary copula partitions. Note that resulting formulas and expressions include many terms stemming from pulling apart the occurring covariances of sums, as illustrated in an example at the end of this section. Due to this high resulting complexity, we provide ready-to-use computer code in the appendix.

According to Cherubini and Romagnoli (2009), the volume of a d-dimensional partition of a copula, defined by $S = [\mathbf{u}^{(low)}, \mathbf{u}^{(up)}] \in [0,1]^d$ with $\mathbf{u}^{(low)} \leq \mathbf{u}^{(up)}$ (element-wise), i.e., a d-dimensional rectangle, is computed by

$$V_C(S) = \sum_{i=0}^{2^d-1} (-1)^{t(i)} C\Big[g\{p(i)\}\Big].$$

Here $p(i)$ represents a d-dimensional vector of ones and zeros corresponding to the binary representation of i, resulting in $g\big\{p(i)\big\}$ a d-dimensional vector such that $g_j = u_j^{(up)}$ if $p_j = 0$ and $g_j = u_j^{(low)}$ if $p_j = 1$. The index j denotes the j-th element of the corresponding vector in $u_j^{(up)}, u_j^{(low)}$ as well as in p_j and g_j. Last, $t(i) = |p(i)|$ counts the number of elements equal to one in $p(i)$ and thereby determines the sign. For example, in dimensions $d = 2$ follows the well-known formula

$$\begin{aligned} V_C(S) &= \sum_{i=0}^{3} (-1)^{t(i)} C\Big[g\{p(i)\}\Big] \\ &= C\Big\{g(0,0)\Big\} - C\Big\{g(1,0)\Big\} - C\Big\{g(0,1)\Big\} + C\Big\{g(1,1)\Big\} \\ &= C\Big(u_1^{(up)}, u_2^{(up)}\Big) - C\Big(u_1^{(low)}, u_2^{(up)}\Big) - C\Big(u_1^{(up)}, u_2^{(low)}\Big) + C\Big(u_1^{(low)}, u_2^{(low)}\Big). \end{aligned}$$

Note that the same formula applies for the volume of the empirical copula by replacing C by $\hat{C}_n$, hence similarly for $\mathbb{C}_n$ (Equation 3.3) and $\mathbb{G}_C$ (Equation 3.4). Thus, on these d-dimensional partitions of $[0,1]^d$ the empirical copula process $\mathbb{C}_n$ and the asymptotic process approximation $\mathbb{G}_C$ are computed by

$$V_{\mathbb{C}_n}(S_1) = \sum_{i=0}^{2^d-1} (-1)^{t(i)} \mathbb{C}_n\Big[g\{p(i)\}\Big].$$

and

$$V_{\mathbb{G}_C}(S) = \sum_{i=0}^{2^d-1} (-1)^{t(i)} \mathbb{G}_C\Big[g\{p(i)\}\Big].$$

From now on, we replace the expression $g\{p(\cdot)\}$ by $\delta(i)$ for the sake of better readability.

In the next step, we combine the volume computation with the derived plug-in covariance estimator to approximate the covariance of two empirical copula partitions, i.e., the covariance of the corresponding empirical copula processes in the given partitions. Remember that both partitions are positioned within the same copula, and therefore, strictly speaking, both copula processes actually represent the same process evaluated at different points. The partitions are defined by arbitrary d-dimensional rectangles S_1 and S_2 in the domain of the empirical copula given by $S_1 = [\mathbf{u}^{(low)}, \mathbf{u}^{(up)}]$ and $S_2 = [\mathbf{v}^{(low)}, \mathbf{v}^{(up)}]$ with $u_k^{(low)}, u_k^{(up)}, v_k^{(low)}, v_k^{(up)} \in [0,1]$ and $u_k^{(low)} \leq u_k^{(up)}$, $v_k^{(low)} \leq v_k^{(up)}$ for $k = 1, \ldots, d$. Analogous to the previous section, we first consider the asymptotic covariance of the empirical copula process before applying the plug-in principle for estimation. The asymptotic covariance of the empirical copula process of copula partitions S_1 and S_2 emerges to

$$\begin{aligned}
\operatorname{cov}\Big\{V_{\mathbb{G}_C}(S_1), V_{\mathbb{G}_C}(S_2)\Big\} &= \operatorname{cov}\Bigg[\sum_{i=0}^{2^d-1}(-1)^{t(i)}\mathbb{G}_C\Big\{\delta(i)\Big\}, \sum_{h=0}^{2^d-1}(-1)^{t(h)}\mathbb{G}_C\Big\{\delta(h)\Big\}\Bigg] \\
&= \operatorname{cov}\Bigg(\sum_{i=0}^{2^d-1}(-1)^{t(i)}\Bigg[\mathbb{B}\Big\{\delta(i)\Big\} - \sum_{k_1=1}^{d}\partial_{k_1}C\Big\{\delta(i)\Big\}\mathbb{B}\Big\{\delta(i)^{(k_1)}\Big\}\Bigg], \\
&\qquad \sum_{h=0}^{2^d-1}(-1)^{t(h)}\Bigg[\mathbb{B}\Big\{\delta(h)\Big\} - \sum_{k_2=1}^{d}\partial_{k_2}C\Big\{\delta(h)\Big\}\mathbb{B}\Big\{\delta(h)^{(k_2)}\Big\}\Bigg]\Bigg).
\end{aligned}$$

Again, δ, which subsumes functions p and g, is used to address the correct value of u_i and u_h. The index $k_1, k_2 = 0$ indicates the absence of marginalization in $\mathbb{B}$, i.e., it indicates the 'full' $\mathbb{B}(u_1, u_2, \ldots, u_d)$ while $k_1, k_2 \geq 1$ addresses the marginalized input vector $\mathbb{B}(1, \ldots, 1, u_{k_1/k_2}, 1 \ldots, 1)$ for $k_1, k_2 \in \{1, \ldots, d\}$.

Next, the expression is further dissolved by decomposing the out covariance term and rearranging the resulting terms. In favour of a short notation, two additional variables

$a_{(\cdot)}$ and $t_{(\cdot)}$ are introduced and the formula results in

$$\operatorname{cov}\left\{V_{\mathbb{G}_C}(S_1),V_{\mathbb{G}_C}(S_2)\right\} = \\ \sum_{i=0}^{2^d-1}\sum_{h=0}^{2^d-1}\sum_{k_1=0}^{d}\sum_{k_2=0}^{d} t_{(i,h,k_1,k_2)}a_{(i,k_1)}a_{(h,k_2)}\operatorname{cov}\left[\mathbb{B}\left\{\delta(i)^{(k_1)}\right\},\mathbb{B}\left\{\delta(h)^{(k_2)}\right\}\right]. \tag{3.11}$$

Here, $a_{(i,k_1)}$ encodes the partial derivatives defined by

$$a_{(i,k_1)} = \left[\mathbf{1}_{\{k_1=0\}} + \mathbf{1}_{\{k_1>0\}}\frac{\partial C\left\{\delta(i)\right\}}{\partial u_{k_1}}\right], \tag{3.12}$$

with a similar expression for $a_{(h,k_2)}$ respectively. Note that indices k_1 and k_2 are extended by 0, which now indicates the absence of the partial derivative, i.e., $a_{(i,k_1)}, a_{(h,k_2)} = 1$. Secondly, $t_{(i,h,k_1,k_2)}$ determines the sign of the summand by

$$t_{(i,h,k_1,k_2)} = (-1)^{|p(i)|+|p(h)|+r_1(k_1)+r_2(k_2)},$$

with $|p(i)|$ and $|p(h)|$ counting the number of elements equal to one in $p(i)$ and $p(h)$ (d-dimensional vector of ones and zeros subsummized in $\delta(i)$), and $r_1(k_1) = \mathbf{1}_{\{k_1>0\}}$ as well as $r_2(k_2) = \mathbf{1}_{\{k_2>0\}}$, respectively.

Remark 3.4.1. *Equation (3.11) can be further decomposed in variance and covariance terms (similar to Example 3.3.1) for computing the variance of an arbitrary box in the copula's domain, i.e., if $S = S_1 = S_2$. With the help of $a_{(j,k_0)}$ defined similar as $a_{(i,k_1)}$*

and $a_{(h,k_2)}$ follows

$$
\begin{aligned}
var\Big\{V_{\mathbb{G}_C}(S)\Big\} = &\sum_{j=0}^{2^d-1}\sum_{k_0=0}^{d} a^2_{(j,k_0)} var\Big[\mathbb{B}\Big\{\delta(j)^{(k_0)}\Big\}\Big] \\
&+ 2\sum_{i=0}^{2^d-1}\sum_{h=0}^{2^d-1}\sum_{k_1=0}^{d}\sum_{k_2\neq k_1} t_{(i,h,k_1,k_2)} a_{(i,k_1)} a_{(h,k_2)} cov\Big[\mathbb{B}\Big\{\delta(i)^{(k_1)}\Big\}, \mathbb{B}\Big\{\delta(h)^{(k_2)}\Big\}\Big] \\
&+ 2\sum_{i=0}^{2^d-1}\sum_{h\neq i}\sum_{k_1=0}^{d}\sum_{k_1=k_2} t_{(i,h,k_1,k_2)} a_{(i,k_1)} a_{(h,k_2)} cov\Big[\mathbb{B}\Big\{\delta(i)^{(k_1)}\Big\}, \mathbb{B}\Big\{\delta(h)^{(k_2)}\Big\}\Big].
\end{aligned}
$$

Note that instead of aggregating the last two sums (setting k_2 to k_1), we decided to use the given notation to make them more comprehensible.

Last, for the plug-in estimator, the expression is further dissolved by decomposing the covariances and collecting the resulting terms. Then, C is estimated via $\hat{C}_n$ and the partial derivatives $\partial_i C(\mathbf{u})$ are substituted by their empirical counterparts $\widehat{\partial_i C_n}$ for $i \in \{1, \dots, d\}$. Under the assumption of existing continuous partial derivatives, we define two arbitrary empirical copula partitions S_1 and S_2 defined as above. The resulting plug-in estimator for the covariance of the empirical copula process of arbitrary rectangular partitions is given by

$$
\begin{aligned}
&\widehat{\text{cov}}\Big\{V_{\mathbb{C}_n}(S_1), V_{\mathbb{C}_n}(S_2)\Big\} \\
&\quad := \sum_{i=0}^{2^d-1}\sum_{h=0}^{2^d-1}\sum_{k_1=0}^{d}\sum_{k_2=0}^{d} t_{(i,h,k_1,k_2)} \hat{a}_{(i,k_1)} \hat{a}_{(h,k_2)} \Big[\hat{C}_n\Big\{\delta(i)^{(k_1)}\Big\} \wedge \hat{C}_n\Big\{\delta(h)^{(k_2)}\Big\} \\
&\qquad\qquad - \hat{C}_n\Big\{\delta(i)^{(k_1)}\Big\}\hat{C}_n\Big\{\delta(h)^{(k_2)}\Big\}\Big],
\end{aligned}
\tag{3.13}
$$

where $\hat{a}_{(\cdot)}$ is defined similarly to Equation (3.12), using the estimator for the partial derivative from Equation (3.7).

Corollary 3.4.1. *Analogously to the estimator given in Theorem 3.3.1, the estimator* $\widehat{cov}\big\{V_{\mathbb{C}_n}(S_1), V_{\mathbb{C}_n}(S_2)\big\}$ *converges weakly to the true value of* $cov\big\{V_{\mathbb{C}_n}(S_1), V_{\mathbb{C}_n}(S_2)\big\}$ *for* $n \to \infty$.

Proof. Follows directly from the proof of Theorem 3.3.1, i.e., the from the element-wise weak convergence of the components of the sum. □

Example 3.4.1 below (and continued in the appendix) computes the variance of a two-dimensional empirical copula process in the upper right corner of the empirical copula, i.e., the survival copula, step-by-step.

Example 3.4.1. *Consider the following example of a two-dimensional copula,* $d = 2$, *and focus on the grey marked rectangle* S *in the upper right corner of the copula domain limited by* $\mathbf{u}^{(up)} = (u_1^{(up)}, u_2^{(up)}) = (1, 1)$ *and* $\mathbf{u}^{(low)} = (u_1^{(low)}, u_2^{(low)})$ *as shown in Figure 3.1.*

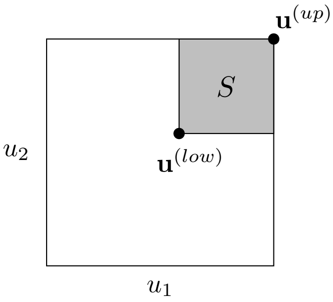


Figure 3.1.: Grey marked rectange S in the upper right corner of the copula domain $[0,1]^2$ enclosed by $\mathbf{u}^{(up)} = (u_1^{(up)}, u_2^{(up)})$ and $\mathbf{u}^{(low)} = (u_1^{(low)}, u_2^{(low)})$.

The asymptotic variance of the empirical copula process in the marked area S is estimated in accordance with the computation of the volume of the area considered.

$$
\begin{aligned}
&\widehat{var}\Big\{V_{\mathbb{C}_n}(S)\Big\} = \\
&\quad var\Big\{\mathbb{B}_{C_n}(u_1^{(up)}, u_2^{(up)}) - \widehat{\tfrac{\partial C_n(u_1^{(up)},u_2^{(up)})}{\partial u_1^{(up)}}}\mathbb{B}_{C_n}(u_1^{(up)}, 1) - \widehat{\tfrac{\partial C(u_1^{(up)},u_2^{(up)})}{\partial u_2^{(up)}}}\mathbb{B}_{C_n}(1, u_2^{(up)}) \\
&\qquad - \mathbb{B}_{C_n}(u_1^{(low)}, u_2^{(up)}) + \widehat{\tfrac{\partial C_n(u_1^{(low)},u_2^{(up)})}{\partial u_1^{(low)}}}\mathbb{B}_{C_n}(u_1^{(low)}, 1) + \widehat{\tfrac{\partial C_n(u_1^{(low)},u_2^{(up)})}{\partial u_2^{(up)}}}\mathbb{B}_{C_n}(1, u_2^{(up)}) \\
&\qquad - \mathbb{B}_{C_n}(u_1^{(up)}, u_2^{(low)}) + \widehat{\tfrac{\partial C_n(u_1^{(up)},u_2^{(low)})}{\partial u_1^{(up)}}}\mathbb{B}_{C_n}(u_1^{(up)}, 1) + \widehat{\tfrac{\partial C_n(u_1^{(up)},u_2^{(low)})}{\partial u_2^{(low)}}}\mathbb{B}_{C_n}(1, u_2^{(low)}) \\
&\qquad + \mathbb{B}_{C_n}(u_1^{(low)}, u_2^{(low)}) - \widehat{\tfrac{\partial C_n(u_1^{(low)},u_2^{(low)})}{\partial u_1^{(low)}}}\mathbb{B}_{C_n}(u_1^{(low)}, 1) - \widehat{\tfrac{\partial C_n(u_1^{(low)},u_2^{(low)})}{\partial u_2^{(low)}}}\mathbb{B}_{C_n}(1, u_2^{(low)})\Big\}
\end{aligned}
$$

Next, this expression of the variance is split up and further dissolved. The complete dissolving of the term can be found in the appendix.

3.5. Simulation Results

In the following, we present simulation results in different dimensions and sample sizes. First, we evaluate the plug-in estimator given in Section 3.3 and compare our results with the results obtained by the bootstrap approximations of Bücher and Dette (2010). Secondly, we perform simulations to estimate the variance of copula partitions for dimensions $d \geq 2$, i.e., $d = 3, 4$ and 5. Note that similar to Bücher and Dette (2010), we present the scaled covariances, i.e., the covariance of the empirical copula process.

3.5.1. Two Dimensions

For the two-dimensional simulation study, we consider the *Clayton copula* (see Chapter 2) with parameter $\theta = 1$ (corresponds to Kendal's $\tau = 1/3$). Thus, we are able to compare our results with the results of the simulation study from Bücher and Dette (2010) for various bootstrap approximations of the empirical copula process. More precisely, we report the true covariances of the limiting process $\big($first row, calculated using Equation

3.6) and the covariances of the simulated process (second row) based on a sample of size $n = 100$ on four different points in the copula domain $\{(\frac{i}{3}, \frac{j}{3}), i, j = 1, 2\}$ in Table 3.1. The simulated values, entitled *Simulated* in the table, demonstrate the quick approximation of the limiting process and serve as a benchmark for the performance of our estimator. They are obtained by calculating the variance/covariance out of 10^6 simulation runs. The last row reports the results of the proposed plug-in estimator based on the average of 1000 simulation runs, with the corresponding *mean squared error (MSE)* scaled by a factor of 10^4 below.

n=100		(1/3,1/3)	(1/3,2/3)	(2/3,1/3)	(2/3,2/3)
True	(1/3,1/3)	0.0486	0.0202	0.0202	0.0100
	(1/3,2/3)		0.0338	0.0093	0.0185
	(2/3,1/3)			0.0338	0.0185
	(2/3,2/3)				0.0508
Simulated	(1/3,1/3)	0.0489	0.0198	0.0198	0.0097
	(1/3,2/3)		0.0333	0.0089	0.0181
	(2/3,1/3)			0.0334	0.0181
	(2/3,2/3)				0.0510
Plug-in	(1/3,1/3)	0.0509 0.5488	0.0198 0.4653	0.0201 0.4841	0.0091 0.3064
	(1/3,2/3)		0.0348 0.8551	0.0087 0.1626	0.0181 0.2526
	(2/3,1/3)			0.0349 0.8665	0.0178 0.2576
	(2/3,2/3)				0.0539 0.4337

Table 3.1.: Sample covariances for the Clayton copula with $\theta = 1$ and sample size $n = 100$. Simulated values result from 10^6 simulation runs, and mean squared error values (multiplied by 10^4) for the plug-in estimator are given below the estimate (obtained from 1000 estimates).

For comparison, Table 3.2 displays the results from the multiplier bootstrap with estimated partial derivative bootstrap estimators α_{pdm}, which has proven to has the best finite-sample performance in the study of Bücher and Dette (2010) and the standard beta bootstrap α_β by Kiriliouk et al. (2021). The values of α_{pdm} are copied from Table 1 and 3 in Bücher and Dette (2010), with MSE values given below. For the α_β, we report

MSE values only, copied from Table 2.1 in Kiriliouk et al. (2021). Further, we report results for the α_{pdm}, the α_{β} and our plug-in estimator for a sample size of $n = 200$ in Appendix B.

n=100		(1/3,1/3)	(1/3,2/3)	(2/3,1/3)	(2/3,2/3)
α_{pdm}	(1/3,1/3)	0.0527 0.8887	0.0205 0.5210	0.0205 0.5222	0.0093 0.3716
	(1/3,2/3)		0.0361 1.0112	0.0092 0.1799	0.0188 0.2988
	(2/3,1/3)			0.0360 0.9899	0.0188 0.2818
	(2/3,2/3)				0.0554 0.6250
α_{β}	(1/3,1/3)	— 0.9992	— 0.3402	— 0.3473	— 0.1956
	(1/3,2/3)		— 0.7887	— 0.1294	— 0.1889
	(2/3,1/3)			— 0.7644	— 0.1821
	(2/3,2/3)				— 0.7108

Table 3.2.: Sample covariances for the Clayton copula with $\theta = 1$ and sample size $n = 100$. Mean squared error values (multiplied by 10^4) for the bootstrap estimators are given below the estimate (obtained from 1000 estimates).

In summary, we observe that the plug-in estimator yields a better result than the best bootstrap method from the study of Bücher and Dette (2010) in absolute values and measured by the MSE at all points. Comparing our plug-in estimator with bootstrap procedure based on the *empirical beta copula* (Segers et al., 2017) results in a more balanced conclusion. While α_{β} seems to have a slight advantage over the proposed plug-in estimator in some cases (but not all), the plug-in estimator performs better in others. However, α_{β} still relies on bootstrap methods.

3.5.2. Three, four, and five Dimensions

For dimensions $d \geq 2$, we stick with our example and consider a *Clayton copula* with parameter $\theta = 1$ again. For the evaluation, we consider four different rectangular boxes with positions B_1, B_2, B_3, B_4 in the d-dimensional domain of the copula $[0,1]^d$:

$$\begin{aligned}
B_1 &:= \left\{ u^{(low)} = \left(0, 0, \ldots, 0\right) \right. && \left. ,u^{(up)} = \left(\frac{1}{3}, \frac{1}{3} \ldots, \frac{1}{3}\right) \right\} \\
B_2 &:= \left\{ u^{(low)} = \left(\frac{2}{3}, \frac{2}{3}, \ldots, \frac{2}{3}\right) \right. && \left. ,u^{(up)} = \left(1, 1, \ldots, 1\right) \right\} \\
B_3 &:= \left\{ u^{(low)} = \left(\frac{1}{3}, \frac{1}{3}, \ldots, \frac{1}{3}\right) \right. && \left. ,u^{(up)} = \left(\frac{2}{3}, \frac{2}{3}, \ldots, \frac{2}{3}\right) \right\} \\
B_4 &:= \left\{ u^{(low)} = \left(\frac{1}{3}, \frac{2}{3}, \frac{1}{3}, \ldots\right) \right. && \left. ,u^{(up)} = \left(\frac{2}{3}, 1, \frac{2}{3}, \ldots\right) \right\}.
\end{aligned}$$

The first position B_1 represents the empirical copula, whereas the second position B_2 corresponds to the empirical survival copula, looking upwards. Third, B_3 is positioned in the middle of the domain, and last, B_4 is off-diagonal, more remotely positioned. The positions are intended to represent structural different areas in the copula domain. Again, values entitled *Simulated* in Table 3.3 are calculated out of 10^6 simulation runs to show the approximation of the limiting process given in the first column (*True*). Similarly, plug-in estimates, given in the third column, result from 1000 estimates with the corresponding *Mean Squared Error (MSE)* below. We further report results for the *Gaussian copula* (see Chapter 2) in Table 3.4 with Spearman's $\rho = 0.5$, which roughly corresponds to the same strength of dependence ($\tau = 1/3$) as before and covariance and correlation matrix

$$\boldsymbol{\rho} = \begin{pmatrix} 1 & \rho & \ldots & \rho & \rho \\ \rho & 1 & \ddots & \ddots & \rho \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \rho & \ddots & \ddots & 1 & \rho \\ \rho & \rho & \ldots & \rho & 1 \end{pmatrix}.$$

		True	Simulated		Plug-in	
$d\ =\ 3$			$n = 100$	$n = 1000$	$n = 100$	$n = 1000$
	B_1	0.0579	0.0616	0.0561	0.0594 0.4989	0.0593 0.1531
	B_2	0.0512	0.0524	0.0541	0.0534 0.6927	0.0528 0.1272
	B_3	0.0356	0.0384	0.0359	0.0425 1.5936	0.0384 0.1933
	B_4	0.0316	0.0328	0.0342	0.0358 1.0425	0.0336 0.1219
$d\ =\ 4$			$n = 100$	$n = 1000$	$n = 100$	$n = 1000$
	B_1	0.0570	0.0553	0.0544	0.0582 0.61739	0.0583 0.1430
	B_2	0.0431	0.0426	0.0449	0.0452 0.8338	0.0444 0.1207
	B_3	0.0186	0.0191	0.0184	0.0239 1.3881	0.0202 0.1362
	B_4	0.01690	0.0185	0.0175	0.0190 0.9750	0.0179 0.1011
$d\ =\ 5$			$n = 100$	$n = 1000$	$n = 100$	$n = 1000$
	B_1	0.0535	0.0477	0.0502	0.0545 0.7602	0.0547 0.1380
	B_2	0.0351	0.0345	0.0346	0.0365 1.0824	0.0362 0.1323
	B_3	0.0088	0.0095	0.0091	0.0115 0.9083	0.0098 0.0862
	B_4	0.0074	0.0074	0.0068	0.0091 0.6127	0.0081 0.0559

Table 3.3.: Sample covariances for the Clayton copula with $\theta = 1$ different sample sizes. MSE values (multiplied by 10^4) for the estimators are given below the mean estimate (obtained from 1000 estimates).

We observe that the plug-in estimator approximates the true values of the limiting process quite well in all boxes regardless of dimensions d. This also confirms the estimator's overall applicability for dimensions ≥ 2. Regarding the MSE, the values are comparable with the results from two dimensions and increase only slightly with growing dimensions. Such an increase is not surprising, as each region in the copula domain is becoming more sparse. Simultaneously, a significant improvement in the MSE can be observed for growing sample sizes. As the magnitude of the improvement seems to be independent of the dimensionality, we believe this is mainly driven by the improved estimation of the partial derivatives.

Even more, insights can be found if we inspect the MSE relative to the actual value of the variance in each box, i.e., the ratio of the MSE and actual value presented in Table 3.5 for the Clayton copula or Table 3.6 for the Gaussian copula. We observe that B_1

		True	Simulated		Plug-in	
$d = 3$			$n = 100$	$n = 1000$	$n = 100$	$n = 1000$
	B_1	0.0531	0.0529	0.0532	0.0556 0.4075	0.0545 0.0562
	B_2	0.0531	0.0529	0.0529	0.0554 0.3617	0.0545 0.0554
	B_3	0.0345	0.0363	0.0347	0.0410 1.4871	0.0370 0.1581
	B_4	0.0271	0.0278	0.0271	0.0311 1.0057	0.0288 0.1172
$d = 4$			$n = 100$	$n = 1000$	$n = 100$	$n = 1000$
	B_1	0.0489	0.0485	0.0486	0.0510 0.5552	0.0502 0.0602
	B_2	0.0489	0.0484	0.0491	0.0509 0.5576	0.0502 0.0570
	B_3	0.0173	0.0186	0.0176	0.0220 1.2400	0.0188 0.1221
	B_4	0.0133	0.0135	0.0133	0.0153 0.8278	0.0140 0.0853
$d = 5$			$n = 100$	$n = 1000$	$n = 100$	$n = 1000$
	B_1	0.0434	0.0426	0.0432	0.0452 0.7925	0.0444 0.0769
	B_2	0.0434	0.0428	0.0433	0.0447 0.7864	0.0445 0.0738
	B_3	0.0080	0.0087	0.0080	0.0106 0.8153	0.0087 0.0731
	B_4	0.0051	0.0054	0.0053	0.0063 0.4639	0.0057 0.0476

Table 3.4.: Sample covariances for the Gaussian copula with $\rho = 0.5$ and different sample sizes. MSE values (multiplied by 10^4) for the estimators are given below the mean estimate (obtained from 1000 estimates).

consistently yields the smallest relative MSE, closely followed by B_2 and B_4. B_3 exhibits the highest relative MSE over all boxes. We believe this order is mainly driven by the number of copula estimations, which include the copula boundary, i.e., zero and one, as an argument of the copula. In this case, the estimation is much more precise since it directly implies a value of zero or eliminates the corresponding dimension fully from the estimation. The difference between B_1 and B_2 is than solely based on the density of observations within the specific box. The number of estimations at the boundaries of the copula also explains the absence of a linear reduction in the MSE (assuming a bias of 0) while the sample size is increased. In these cases, a higher number of samples only affects some of the terms of the sum involved in the estimation; thus, the higher precision does not fully emerge. Or put into another perspective, some of the terms are already estimated very precisely (or perfectly) for small sample sizes, hence increasing the samples size does not benefit these terms and the overall estimator in the same way as it would be the case without the estimations at the copula boundary.

	$d = 3$		$d = 4$		$d = 5$	
	$n = 100$	$n = 1000$	$n = 100$	$n = 1000$	$n = 100$	$n = 1000$
B_1	8.6165	2.6442	10.8314	2.5087	14.2093	2.5794
B_2	13.5293	2.4844	19.3457	2.8005	30.8376	3.7692
B_3	44.7640	5.4298	74.6290	7.3226	103.2159	9.7955
B_4	32.9905	3.8576	57.6923	5.9822	82.7973	7.5541

Table 3.5.: Relative MSE of the plug-in estimator over boxes B_1, B_2, B_3 and B_4 for the Clayton copula with $\theta = 1$ and different sample sizes. The MSE is calculated over 1000 simulation runs and multiplied by 10^4.

Relative MSE values for the Gaussian copula are given in Table 3.6, strengthening our interpretation above. These values are also calculated as the ratio between the MSE and true value reported in Table 3.4.

	$d = 3$		$d = 4$		$d = 5$	
	$n = 100$	$n = 1000$	$n = 100$	$n = 1000$	$n = 100$	$n = 1000$
B_1	7.6742	1.0584	11.3538	1.2311	18.2604	1.7719
B_2	6.8117	1.0433	11.4029	1.1656	18.1198	1.7004
B_3	43.1043	4.5826	71.6763	7.0578	101.9125	9.1375
B_4	37.1107	4.3247	62.2406	6.4135	90.9607	9.3333

Table 3.6.: Relative MSE of the plug-in estimator over boxes B_1, B_2, B_3 and B_4 for the Gaussian copula with $\rho = 0.5$ and different sample sizes. The MSE is calculated over 1000 simulation runs and multiplied by 10^4.

3.6. Conclusion

In this chapter, we presented a new plug-in estimator for variances and covariances of the empirical copula process, i.e., the empirical copula. We further introduced a formula for calculating the estimated variance and covariance of the empirical copula process in any arbitrary rectangle partition of the d-dimensional copula domain. We illustrated its good finite sample properties in a simulation study, where the MSE seems mainly driven by the sample size used for estimation. Further, a ready-to-use computer code for the estimator is provided. For future research, the given estimator could be enhanced and, e.g., new testing methods for copulas could be developed on its basis.

4. A Multivariate Extension of the Lorenz Curve

This chapter is based on Grothe et al. (2022a), *published* in *Journal of Economic Inequality.* The article, as well as the figures, are under Creative Commons Attribution 4.0 International License, which grants permission to reuse them in this work. The chapter deals with a multivariate extension of the Lorenz curve and enables the measurement of inequality in more than one dimension, while providing a clear economic reasoning.

4.1. Introduction

The well-known Lorenz curve and Gini coefficient are still the most important tools for representation and analysis of inequality in a distribution, such as the income and wealth distribution. Both, however, are univariate instruments, i.e., they analyze the variables individually, ignoring their dependence structure. Considering the example of income and wealth, it is not possible to see the differences in the overall inequality if wealthy people coincide with high-income people compared to a more balanced, eventually compensating distribution of wealth over the income groups. Contrary to that, in this chapter, we propose extensions of both tools based on copulas to study the inequality of d variables $X_1, \ldots, X_d$ simultaneously. By that, we explicitly capture the dependence structure of these variables which gets lost if only one variable is considered at a time.

There had been some efforts to extend Lorenz curve and Gini coefficient to the multivariate case before. The earliest suggestion in this direction we know of is

Taguchi (1972a,b) who applied methods of differential geometry. Further suggestions are by Arnold (1987), Arnold and Sarabia (2018), Gajdos and Weymark (2005) and Koshevoy and Mosler (1996, 1997). We will not give an overview of these contributions because this is - at least partially - done by Arnold and Sarabia (2018). We agree with the view of the latter authors that all extensions are essentially determined by (elegant) mathematical considerations but may lack interpretability and economic reasoning.

Here, we propose direct and natural multivariate extensions of both, the (inverse) Lorenz curve and the Gini coefficient. We exploit the fact that the inverse of a variable's Lorenz curve is the distribution function of a simple monotonically increasing transformation of that variable. The multivariate inverse Lorenz curve of the d variables $X_1, \ldots, X_d$ is then defined as the joint distribution function of analogous univariate transformations of them. The resulting Lorenz curve can explicitly be expressed using copulas. As presented in Chapter 2, copulas decompose the joint distribution function of variables into marginal distribution functions and their dependence structure. Consequently, for a given vector $\boldsymbol{X} = (X_1, \ldots, X_d)$ of d variables, the copula-based multivariate Lorenz curve identifies and captures two different sources of inequality:

a. inequality contained in the individual variables X_i, measured by the univariate Lorenz curve L_i or inverse Lorenz curve L_i^{-1} for $i \in \{1, \ldots, d\}$.

b. inequality due to the dependence structure of the variables $X_1, \ldots, X_d$ which is captured by the copula of these variables.

Based on the multivariate Lorenz curve, the formulation of a multivariate Gini coefficient follows in a natural way analogously to the derivation of the univariate Gini coefficient from the univariate Lorenz curve.

The mathematics we apply in the theoretical part of the chapter is some elementary copula theory and - hopefully - is accessible to a broad readership. Later on in the chapter, we derive simple nonparametric estimators for both instruments and provide ready-to-use computer code in the appendix of this thesis. We illustrate both instruments, the multivariate Lorenz curve and Gini coefficient, on data sets consisting of individual wealth and income data of various countries. The results are intuitive and show that the two above-mentioned sources a. and b. of multivariate inequality are reflected in a reasonable and interpretable way.

The structure of the chapter is the following. Section 4.2 introduces notation and some definitions. The multivariate extension of the inverse Lorenz curve (MEILC) is introduced in Section 4.3. Various properties of the MEILC are derived. Section 4.4 presents a multivariate extension of the Gini coefficient (MEGC) related to the MEILC and considers the bivariate Gini (i.e., $d = 2$) as a special case. Nonparametric estimation of MEILC and MEGC is considered in Section 4.5. In Section 4.6 we address some aspects regarding multivariate transfers. The last section of the chapter contains the empirical applications.

4.2. Notations and Definitions

We again consider a d-variate vector of random variables $\boldsymbol{X} = (X_1, \ldots, X_d)$ defined on a probability space $(\Omega, \Sigma, \mathbb{P})$. The joint distribution function is given by $F_{\boldsymbol{X}}(x) = P(X_1 \leq x_1, \ldots, X_d \leq x_d)$ for $\mathbf{x} = (x_1, \ldots, x_d) \in \mathbb{R}^d$ and the marginal distribution functions F_i of X_i are given by $F_i(x_i) = P(X_i \leq x_i)$ for $x_i \in \mathbb{R}$ and $i \in \{1, \ldots, d\}$. Throughout this chapter we assume that $X_i \geq 0$ and $0 < \mu_i = E(X_i) < \infty$ for $i \in \{1, .., d\}$. Note the small additional $\boldsymbol{X}$ in the sub-index of F, which we added in this chapter of the thesis for a more concise notation. Further note that all variables should fit together in an economically meaningful way and should have a cardinal scale. For inequality measurement in the case of ordinal or qualitative data we refer the interested reader to Allison and Foster (2004), Gravel et al. (2021) or Kobus and Miłoś (2012) among others.

From Chapter 2, we already now that there exists a copula $C = C_{F_{\boldsymbol{X}}}$, such that $F_{\boldsymbol{X}}(x_1,..,x_d) = C\{F_1(x_1),\dots,F_d(x_d)\}$ for $(x_1,\dots,x_d) \in \mathbb{R}^d$, which is uniquely determined if the marginal distribution functions F_i are continuous. Besides our work, copulas also have been applied in the context of income analysis by Aaberge et al. (2018) recently. Remember that an important property of a copula from a random vector $\boldsymbol{X}$ is that it is invariant with respect to strictly increasing transformations of the marginal distributions and it is bounded by the Fréchet–Hoeffding bounds, i.e, $W(u_1,\dots,u_d) \leq C(u_1,\dots,u_d) \leq M(u_1,\dots,u_d)$. The upper bound M, which is called comonotonicity copula, corresponds to the dependence structure of full monotone positive dependence. An example of wealth and income with such a dependence structure would be a population of size N, where the $i-$th wealthiest individual also has the $i-$th highest income (for $i \in \{1,\dots,N\}$). The lower bound W is only a proper copula in the bivariate case and is then called the countermonotonicity copula. In a countermonotonic income/wealth example, the $i-$th wealthiest individual would have the $i-$th lowest income (for $i \in \{1,\dots,N\}$). If $\boldsymbol{X}$ is a vector of independent variables $X_1,\dots,X_d$ the corresponding copula is the independence copula Π with $\Pi(u_1,\dots,u_d) = \prod_{i=1}^d u_i$. There is a partial order $C \leq C'$ on the set of d-variate copulas given by $C(\mathbf{u}) \leq C'(\mathbf{u})$ for all $\mathbf{u} \in [0,1]^d$ (see Nelsen 2006). See Figure 2.1 in Chapter 2 for a visualization of these extreme types of dependence.

The univariate Lorenz curve for $i \in \{1,\dots,d\}$ is given by

$$L_i(u_i) = \frac{1}{\mu_i}\int_0^{u_i} F_i^{-1}(t)dt \text{ for } u_i \in [0,1], \tag{4.1}$$

see Gastwirth (1971). Each L_i is a continuous, weakly increasing and weakly convex function. It has all the properties of a distribution function if we extend L_i by 1 for

$u_i > 1$ and by 0 for $u_i < 0$. The inverse of L_i is defined by

$$L_i^{-1}(u_i) = \begin{cases} inf\{t|L_i(t) \geq u_i\}, & \text{for } u_i \in \ ]0,1] \\ sup\{t|L_i(t) = u_i\}, & \text{for } u_i = 0 \end{cases}$$

and L_i^{-1} is continuous, weakly increasing and weakly concave on $[0,1]$. It has all the properties of a distribution function if we extend L_i^{-1} by 1 for $u_i > 1$ and by 0 for $u_i < 0$. Note that there might be a point mass at zero.

Consider individual income in a population. The usual interpretation of the Lorenz curve L of this variable is that, e.g., for $p \in [0,1]$, $L(p)$ denotes the proportion of total income that corresponds to the bottom $p \cdot 100\%$ of the individuals. The interpretation of the inverse Lorenz curve is that, for $q \in [0,1]$, $L_i^{-1}(q)$ indicates the maximum percentage of the population with a combined cumulative share of $q \cdot 100\%$ of the total income (the maximum ensures starting with the bottom income individual here). Obviously, both curves describe the inequality in an equivalent way. It is worth mentioning that Max Otto Lorenz (Lorenz, 1905) originally proposed in his chapter what we now call the inverse Lorenz curve.

Using X_i, F_i and L_i as defined above we now define the following random variables X_i^* by

$$X_i^* = L_i\big\{F_i(X_i)\big\} \text{ for } i \in \{1,\ldots,d\}. \tag{4.2}$$

Note, the difference between X_i and X_i^*. In applications X_i has a dimension (such as income or wealth). X_i^*, however, is a fraction (i.e., a number between 0 and 1). If, e.g., X_i denotes again individual income in a population then X_i^* is the corresponding joint fraction of the total income of that part of the population having individual incomes smaller or equal to X_i. The d-variate vector $\boldsymbol{X}^*$ is defined by $\boldsymbol{X}^* = (X_1^*,\ldots,X_d^*)$. The marginal distribution function for X_i^* is given by the inverse Lorenz curve of X_i, i.e.,

$$F_{X_i^*}(u_i) = P\Big[L_i\{F_i(X_i)\} \leq u_i\Big] = L_i^{-1}(u_i)$$

for $u_i \in [0,1]$ and $i \in \{1,\ldots,d\}$. The joint distribution function of $\boldsymbol{X}^*$ is given by

$$\begin{aligned} F_{\boldsymbol{X}^*}(u_1,\ldots,u_d) &= P(X_1^* \leq u_1,\ldots,X_d^* \leq u_d) && (4.3)\\ &= C\Big\{L_1^{-1}(u_1),\ldots,L_d^{-1}(u_d)\Big\} \text{ for } u_i \in [0,1] \text{ and } i \in \{1,\ldots,d\}. && (4.4)\end{aligned}$$

Note, that the copula of $\boldsymbol{X}$ is identical to the copula of $\boldsymbol{X}^*$, since X_i^* is a monotonically increasing function of X_i for $i \in \{1,\ldots,d\}$.

The univariate Gini coefficient is defined as a normalization of the area enclosed by the Lorenz curve and the diagonal of the unit square. It equals one minus twice the area under the Lorenz curve (Kakwani, 1977; Gastwirth, 1972)

$$G = 1 - 2\int_{[0,1]} L(u)du.$$

Considering that $1 - \int L(u)du = \int L^{-1}(u)du$ and L^{-1} is the *cdf* of X^*, it follows that $\int L(u)du = E(X^*)$ and the univariate Gini coefficient may be expressed as

$$G = 1 - 2E(X^*)$$

as well as

$$G = 2\int_{[0,1]} L^{-1}(u)du - 1$$

when using the inverse Lorenz curve L^{-1} and considerations above. Notation and definitions introduced in this section are used to define a multivariate extension of the univariate Lorenz curve (see Section 4.3) and a multivariate extension of the univariate Gini coefficient (see Section 4.4).

4.3. A Multivariate Extension of the Lorenz Curve Based on Copulas (MEILC)

As mentioned in the introduction, inequality in a d-variate random vector $\boldsymbol{X} = (X_1, \dots, X_d)$ has two different sources:

a. inequality in the individual variables X_i, which is measured by the corresponding Lorenz curves $L_i(u_i)$ or inverse Lorenz curves $L_i^{-1}(u_i)$ for $i \in \{1, \dots, d\}$ and $u_i \in [0, 1]$.

b. inequality contained in the dependence structure of the vector $\boldsymbol{X} = (X_1, \dots, X_d)$ which is represented by the copula C of $\boldsymbol{X}$.

To illustrate the effect of b. on the joint inequality in $\boldsymbol{X} = (X_1, \dots, X_d)$ in more detail, we look at a very simple example for the bivariate case, i.e., $d = 2$, and a population of five individuals, where X_1 and X_2 might again stand for individual income and wealth, respectively.

		Individual 1	Individual 2	Individual 3	Individual 4	Individual 5
Example 1	$\begin{pmatrix}X_1\\X_2\end{pmatrix}$	$\begin{pmatrix}1\\1\end{pmatrix}$	$\begin{pmatrix}2\\2\end{pmatrix}$	$\begin{pmatrix}3\\3\end{pmatrix}$	$\begin{pmatrix}4\\4\end{pmatrix}$	$\begin{pmatrix}5\\5\end{pmatrix}$
Example 2	$\begin{pmatrix}X_1\\X_2\end{pmatrix}$	$\begin{pmatrix}1\\3\end{pmatrix}$	$\begin{pmatrix}2\\2\end{pmatrix}$	$\begin{pmatrix}3\\5\end{pmatrix}$	$\begin{pmatrix}4\\1\end{pmatrix}$	$\begin{pmatrix}5\\4\end{pmatrix}$
Example 3	$\begin{pmatrix}X_1\\X_2\end{pmatrix}$	$\begin{pmatrix}1\\5\end{pmatrix}$	$\begin{pmatrix}2\\4\end{pmatrix}$	$\begin{pmatrix}3\\3\end{pmatrix}$	$\begin{pmatrix}4\\2\end{pmatrix}$	$\begin{pmatrix}5\\1\end{pmatrix}$

It can be seen that the marginal distributions of X_1 and X_2 over the five individuals are the same in these examples. We think that it is quite obvious to see that the inequality is largest in Example 1 and smallest in Example 3. Example 2 is somewhere in between.

The differences in joint inequality in these examples are due to different dependence structures between the variables. In terms of copulas, the dependence structure in

Example 1 corresponds to the comonotoncity copula M, the upper bound in the set of bivariate copulas. In contrast, Example 3 corresponds to the countermonotonicity copula W which is the lower bound in the set of bivariate copulas. Thus, in Example 1, the high income individuals are also the wealthiest, whereas in Example 3, income and wealth kind of compensate each other. Example 2 might stem from the independence copula Π. We conclude from this example that joint inequality in a vector $\boldsymbol{X} = (X_1, \dots, X_d)$ is increasing in C in the partial order as defined in Section 4.2 of this chapter.

Having the example in mind, we now define a multivariate extension of univariate (inverse) Lorenz curves considering the dependence structure of the variables. It will turn out that the multivariate Lorenz curve of random vector $\boldsymbol{X}$ is the joint distribution function (compare to Equation 4.3) of the random vector $\boldsymbol{X^*}$ as defined in Equation (4.2) in Section 4.2.

Definition 4.3.1. *Multivariate extension of the inverse Lorenz curve (MEILC) and Lorenz order*
Using the notation of Section 4.2, let

1. $\mathbb{L}^{-1}_{C,L_1^{-1},\dots,L_d^{-1}}(u_1, \dots, u_d) = C\left\{L_1^{-1}(u_1), \dots, L_d^{-1}(u_d)\right\}$ *for* $(u_1, \dots, u_d) \in [0,1]^d$.
2. *For a second vector* $\tilde{\boldsymbol{X}} = (\tilde{X}_1, \dots, \tilde{X}_d)$ *with copula* $\tilde{C}$ *and inverse Lorenz curves* $\tilde{L_i}^{-1}(u_i)$ *of* $\tilde{X}_i$ *for* $i \in \{1, \dots, d\}$, *we define the multivariate ordering* $\tilde{\boldsymbol{X}} \succeq \boldsymbol{X}$ *if and only if*
$\mathbb{L}^{-1}_{\tilde{C},\tilde{L_1}^{-1},\dots,\tilde{L_d}^{-1}}(\mathbf{u}) \geq \mathbb{L}^{-1}_{C,L_1^{-1},\dots,L_d^{-1}}(\mathbf{u})$ *for all* $\mathbf{u} = (u_1, \dots, u_d) \in [0,1]^d$.

The extension of the inverse Lorenz curve (MEILC) $\mathbb{L}^{-1}(\mathbf{u}) = \mathbb{L}^{-1}_{C,L_1^{-1},\dots,L_d^{-1}}(\mathbf{u})$ has a nice interpretation in terms of the X_i^* for $i \in \{1, \dots, d\}$ and $\mathbf{u} = (u_1, \dots, u_d) \in [0,1]^d$. Since $\mathbb{L}^{-1}(u_1, \dots, u_d)$ is the joint distribution function of $\boldsymbol{X}^* = (X_1^*, \dots, X_d^*)$ we see that $\mathbb{L}^{-1}(u_1, \dots, u_d)$ is the population fraction for which $X_1^* \leq u_1, \dots, X_d^* \leq u_d$ and therefore the fraction with a cumulative share of the features smaller or equal to $u_1, \dots, u_d$. E.g., for $d = 2$ if u_1 denotes a share of the cumulative income in a population and u_2 a share of wealth, than $\mathbb{L}^{-1}(u_1, u_2)$ is the corresponding fraction of people collectively having not more than shares u_1 and u_2 of the total income and wealth, respectively. Note that the interpetation of the MEILC coincides with the interpretation of the upper parts of the *Lorenz zonoid* introduced by Koshevoy and Mosler (1996). However, while calculating zonoids from data is computationally intensive, the copula approach results in simple formulas, also allowing a straight forward extension of the Gini coefficient later in the chapter.

We analyze some of the (formal) properties of the MEILC. Here, we are in particular interested in how $\mathbb{L}^{-1}(u_1, \ldots, u_d)$ behaves, when ceterus paribus either marginal inequalities or the dependence structures are changed. Later in the chapter, in Section 4.6, we discuss some implied properties of the MEILC such as the reaction to transfers in empirical data.

1. Obviously $\mathbb{L}^{-1}(u_1, \ldots, u_d)$ is a function from $[0,1]^d$ to $[0,1]$. Furthermore for every $C, L_1^{-1}(u_1), \ldots, L_d^{-1}(u_d)$ and $\mathbf{u} = (u_1, \ldots, u_d) \in [0,1]^d$ we have
   $\mathbb{L}^{-1}_{min}(\mathbf{u}) \leq \mathbb{L}^{-1}_{C,L_1^{-1},\ldots,L_d^{-1}}(\mathbf{u}) \leq \mathbb{L}^{-1}_{max}(\mathbf{u})$
   where
   $\mathbb{L}^{-1}_{min}(\mathbf{u}) = W(u_1, \ldots, u_d) = max\{0, \sum_i^d u_i - (d-1)\}$
   and
   $\mathbb{L}^{-1}_{max}(\mathbf{u}) = M(1, \ldots, 1) = min\{1, \ldots, 1\} = 1.$
   These boundaries follow directly from the Fréchet–Hoeffding bounds (see Chapter 2). Regarding the margins, note that the arguments of the lower bound refer to minimal marginal inequality, i.e., $L_i^{-1}(u_i) = u_i$ for $i \in \{1 \ldots d\}$, whereas the arguments of the upper bound refer to maximal marginal inequality, i.e., $L_i^{-1}(u_i) \equiv 1$ for $i \in \{1 \ldots d\}$. Thus, e.g., the upper bound corresponds to the case of maximal marginal inequality as well as maximal dependence between the variables and reflects thus the case of maximal multivariate inequality.

2. If $X_1, \ldots, X_d$ are independent, i.e., $C = \Pi$ we have
   $\mathbb{L}^{-1}_{\Pi,L_1^{-1},\ldots,L_d^{-1}}(\mathbf{u}) = \prod_{i=1}^d L_i^{-1}(u_i)$ for $\mathbf{u} = (u_1, \ldots, u_d) \in [0,1]^d$.
   So the MEILC is the product of the univariate Lorenz curves in such cases.

3. If $u_d = 1$ we know $\mathbb{L}^{-1}_{C,L_1^{-1},\ldots,L_{d-1}^{-1},L_d^{-1}}(u_1,\ldots,u_{d-1},1) = \mathbb{L}^{-1}_{C,L_1^{-1},\ldots,L_{d-1}^{-1}}(u_1,\ldots,u_{d-1})$ for $(u_1,\ldots,u_{d-1}) \in [0,1]^{d-1}$ and similar formulas hold for $i \in \{1,\ldots,d-1\}$ and for more general index sets $I \subset \{1,\ldots,d\}$.

   This marginalization is quite intuitive, since setting $u_i = 1$ in the MEILC refers to the fraction of the population having less or equal than the maximum value of X_i. Therefore, the i-th dimension is not restrictive anymore, while the other dimensions still are.

   By further marginalizing, we can see that for $d = 1$, we get the univariate (inverse) Lorenz curve as a margin, e.g., $\mathbb{L}^{-1}_{C,L_1^{-1}}(u_1) = L_1^{-1}(u_1)$ for $u_1 \in [0,1]$.

4. If for $\mathbf{u} = (u_1,\ldots,u_d) \in [0,1]^d$ at least one $L_i^{-1}(u_i)$ is zero, then $\mathbb{L}^{-1}(\mathbf{u})$ is zero. But note that there might be point masses at zero in some or even all of the X_i. The MEILC therefore does not necessarily start at zero since a point mass of X_i at zero would imply $L_i^{-1}(0) > 0$.

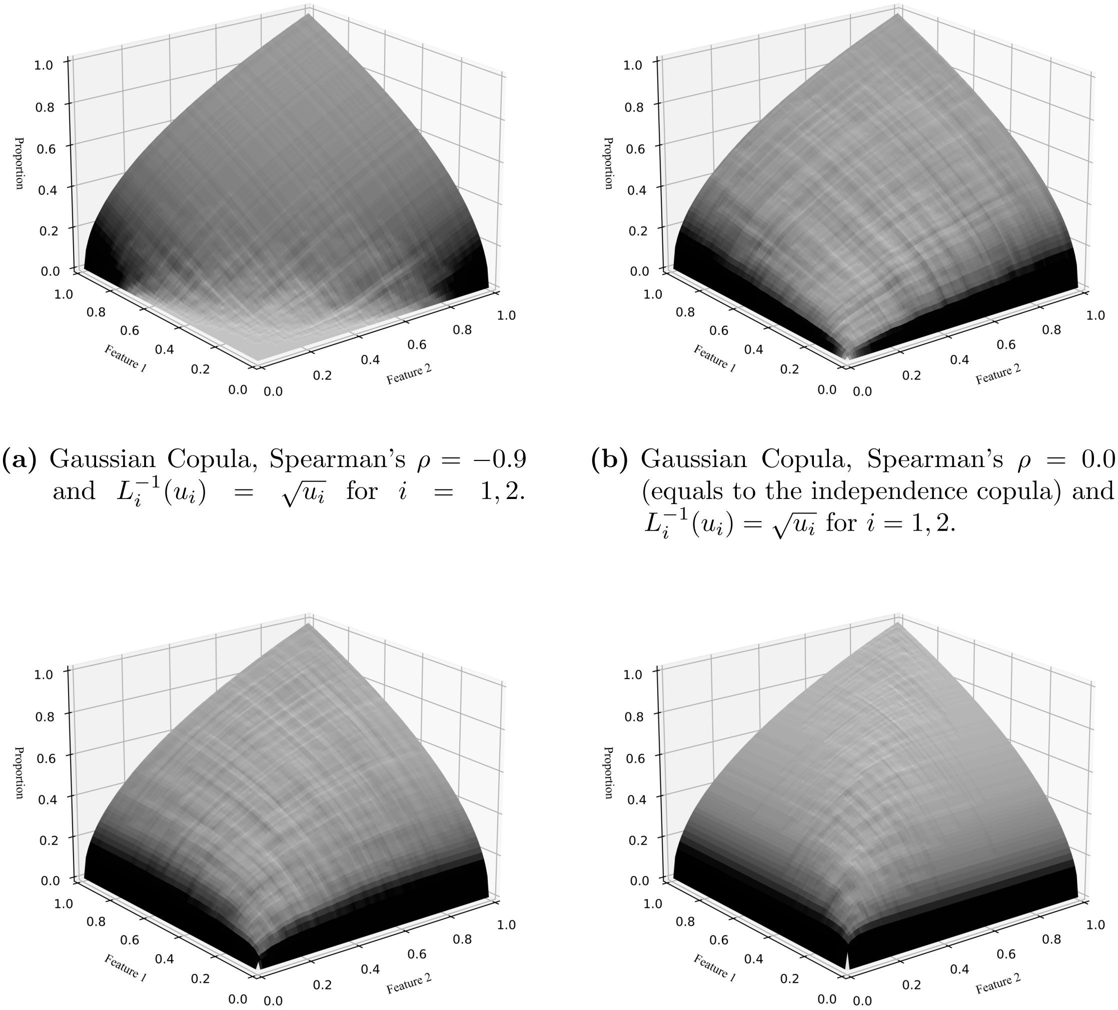
5. Response of $\mathbb{L}^{-1}(\mathbf{u})$ to changing $L_i^{-1}(u_i)$ for fixed $\mathbf{u} = (u_1,\ldots,u_d) \in [0,1]^d$: Higher values of $L_i^{-1}(u_i)$ lead to higher values of $\mathbb{L}^{-1}(\mathbf{u})$, ceteris paribus. This follows directly from the definition of $\mathbb{L}^{-1}(\mathbf{u})$ and general properties of every copula. An increased inequality in one dimension, therefore, leads to an increased total inequality without any further changes.

6. Response of $\mathbb{L}^{-1}(\mathbf{u})$ to changes in the dependence structure of the variables, i.e., to changes of C, when the $L_i^{-1}(u_i)$ do not change: Consider two copulas C_A and C_B with $C_A(\mathbf{u}) \leq C_B(\mathbf{u})$ for all $\mathbf{u} \in [0,1]^d$. Here, referring to the example of income and wealth, in B the wealthy would tend to belong more to the high-income part of the society than in A. It then follows the corresponding multivariate Lorenz order from Definition 4.3.1, i.e., $\mathbb{L}_A^{-1}(\mathbf{u}) \leq \mathbb{L}_B^{-1}(\mathbf{u})$. Generally, the order properties of the involved copulas transfer directly to the multivariate Lorenz order. Since

the copula order is, however, a partial order, not all changes in the dependence structure lead to ordered Lorenz curves.

We illustrate the MEILC for some bivariate examples in Figures 4.1 and 4.2. We consider two different types of marginal Lorenz curves in all examples, $L_i(u_i) = u_i^2$ (this corresponds to values X_i which are uniformly distributed over a finite interval $[0, b]$ with $b > 0$) or $L_i(u_i) = u_i^{10/9}$ (which is close to the Lorenz curve of minimal inequality). Note that the corresponding marginal inverse Lorenz curves are $L_i^{-1}(u_i) = \sqrt{u_i}$ and $L_i^{-1}(u_i) = u_i^{0.9}$, respectively. In Figure 4.1, we consider Gaussian dependence structures of the variables and vary Spearman's ρ from negative to positive dependence, starting from the case of strong negative dependence (a) to the case of independent margins (b), small positive dependence (c) and strong positive dependence (d). As expected, the surface of the MEILC becomes more domed for increasing strength of dependence. Recall that a point on the surface $\mathbb{L}^{-1}(u_1, u_2)$ at (u_1, u_2) reflects the maximum share of the society having together less than shares u_1 and u_2 of the total variable sums of X_1 and X_2, respectively. Thus, it refers to the share of individuals being at the bottom in both variables. A more domed surface therefore reflects a larger inequality.

In Figure 4.2, we illustrate the effect of the copula family and a case of unequal marginal inverse Lorenz curves. Panels (a) and (b) both refer to cases with a rank correlation of X_1 and X_2 of $\rho = 0.8$ but different asymmetric dependence structures, i.e., copulas. The Clayton copula (a) has a stronger dependence between small values, while the Gumbel copula (b) has strongest dependence between large values. Consequently, we see that the surface of the MEILC in the Clayton case is more domed for pairs of small values than in the Gumbel case (b). Panel (c) depicts the case of independence where the margins are now different. It can be seen that surface interpolates between the margins. Again, the surface gets more domed, if the dependence is increased, e.g., by using a Clayton copula with $\rho = 0.8$ (d). This is done by visualizing $\mathbb{L}^{-1}(\mathbf{u})$ for data with Gaussian and Archimedean copulas (see Chapter 2), different dependence parameters and marginal distributions.



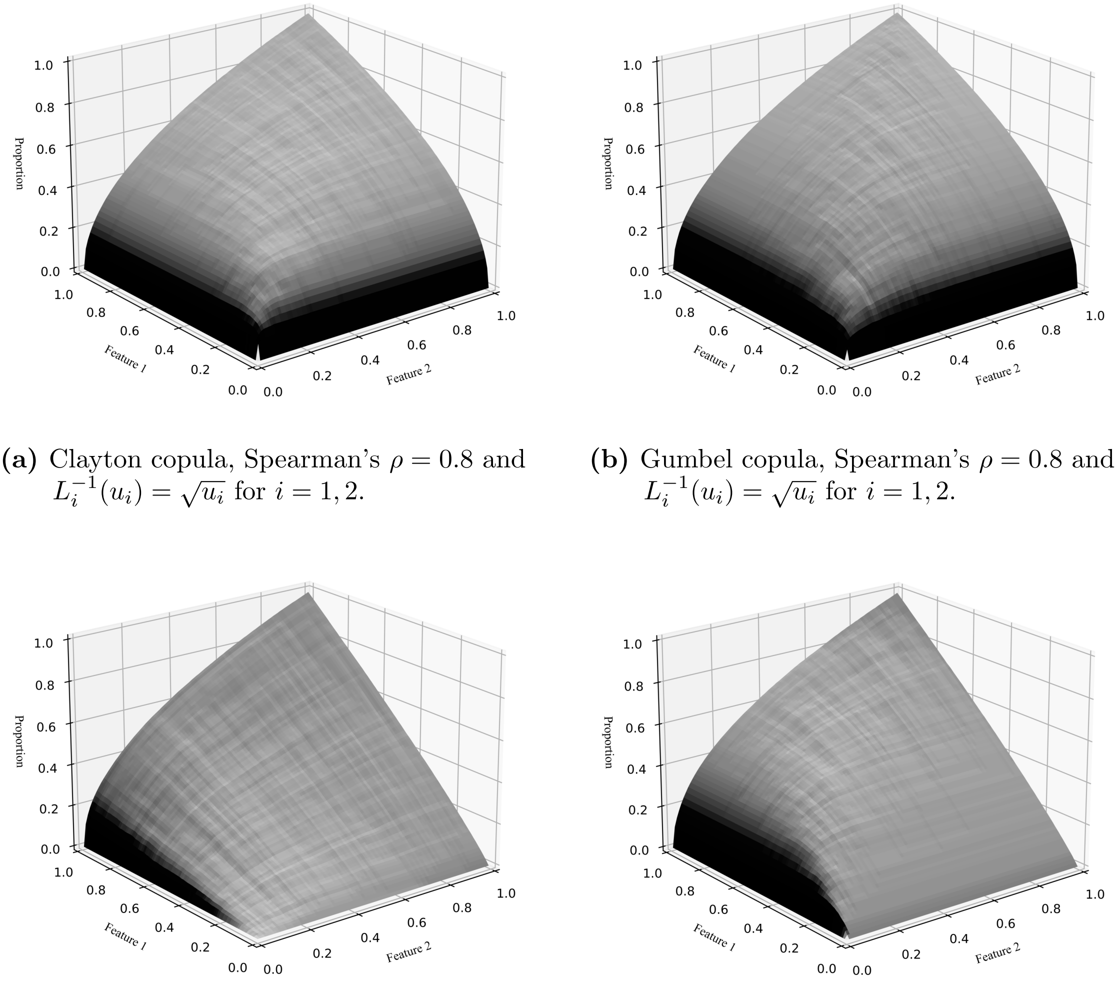
(a) Gaussian Copula, Spearman's $\rho = -0.9$ and $L_i^{-1}(u_i) = \sqrt{u_i}$ for $i = 1, 2$.

(b) Gaussian Copula, Spearman's $\rho = 0.0$ (equals to the independence copula) and $L_i^{-1}(u_i) = \sqrt{u_i}$ for $i = 1, 2$.

(c) Gaussian Copula, Spearman's $\rho = 0.5$ and $L_i^{-1}(u_i) = \sqrt{u_i}$ for $i = 1, 2$.

(d) Gaussian Copula, Spearman's $\rho = 0.9$ and $L_i^{-1}(u_i) = \sqrt{u_i}$ for $i = 1, 2$.

Figure 4.1.: Graphs of $\mathbb{L}^{-1}(u_1, u_2)$ based on Gaussian copulas with different values for the dependence parameter Spearman's ρ. The surface of $\mathbb{L}^{-1}(u_1, u_2)$ gets more domed with increasing parameter ρ.



(a) Clayton copula, Spearman's $\rho = 0.8$ and $L_i^{-1}(u_i) = \sqrt{u_i}$ for $i = 1, 2$.

(b) Gumbel copula, Spearman's $\rho = 0.8$ and $L_i^{-1}(u_i) = \sqrt{u_i}$ for $i = 1, 2$.

(c) Independence copula Π and $L_1^{-1}(u_1) = u_1^{0.9}$ and and $L_2^{-1}(u_2) = \sqrt{u_2}$.

(d) Clayton copula, Spearman's $\rho = 0.8$ and $L_1^{-1}(u_1) = u_1^{0.9}$ and $L_2^{-1}(u_2) = \sqrt{u_2}$.

Figure 4.2.: Graphs of $\mathbb{L}^{-1}(u_1, u_2)$ based on Archimedean copulas with different values for the dependence parameter Spearman's ρ and marginal distributions. Panels (a) and (b) illustrate the effect of different asymmetric dependence structures, while (c) and (d) illustrate effects of margins and dependence structure.

Remark 4.3.1. *It might be surprising that our extension of the Lorenz curve is based on its inverse and not on the Lorenz curve itself. The inverse Lorenz curve draws proportions of the people on the y-axis and the variable of interest, e.g., share of total income, on the x-axis. Proportion of people is thus the value of the inverse function, while the variable of interest is the argument. Having only one variable of interest, the choice between Lorenz curve or inverse Lorenz curve seems arbitrary. Considering $d > 1$ variables of interest, however, it seems conceptually more natural to add these variables as further arguments of the inverse Lorenz curve. Furthermore, the resulting extension is easily interpretable.*

Alternatively, starting from the Lorenz curve, a seemingly intuitive idea like

$$(u_1, \dots, u_d) \longmapsto C\big\{L_1(u_1), \dots, L_d(u_d)\big\}.$$

behaves contradictorily. If inequality in the X_i rises than the above definition indicates a decreasing value. If inequality accounted in C increases an increasing value is indicated. Obviously this is contradictory.

A possible adjustment would be to look at

$$(u_1, \dots, u_d) \longmapsto 1 - C\big\{1 - L_1(u_1), \dots, 1 - L_d(u_d)\big\}$$

This object reacts in the expected directions in all cases, but it lacks a reasonable and convincing interpretation.

Remark 4.3.2. *We are quite aware that there might be further reasonable ways of combining a copula with Lorenz curves L_i or its inverses. The survival copula $\overline{C}$ which corresponds to copula C (see Nelsen 2006 p.33) might also be a useful tool for the definition of a multivariate extension of the Lorenz curve, but we have not derived any details.*

4.4. A Multivariate Extension of the Gini Coefficient (MEGC) Related to the Multivariate Extension of the Lorenz Curve

In the univariate setting, i.e., $d = 1$, it is well known that the Lorenz order is only a weak order. Indeed, Lorenz curves may intersect and consequently X_1 and $\tilde{X}_1$ with intersecting Lorenz curves cannot be ordered with respect to inequality. A numerical measure of inequality, such as the Gini coefficient, is called for.

In the multivariate setting, it can be seen that the order defined in Section 4.4 is also a weak order and a related numerical measure of inequality is required, too. Using the notation of Section 4.2 and 4.3 we define a Gini coefficient related to the MEILC as follows.

Definition 4.4.1. *Multivariate extension of the Gini coefficient (MEGC)*
The MEGC is defined as:

$$G_{C,L_1^{-1},\dots,L_d^{-1}} = \frac{\int_{[0,1]^d} \mathbb{L}^{-1}_{C,L_1^{-1},\dots,L_d^{-1}}(\mathbf{u})d\mathbf{u} - \int_{[0,1]^d} \mathbb{L}^{-1}_{min}(\mathbf{u})d\mathbf{u}}{\int_{[0,1]^d} \mathbb{L}^{-1}_{max}(\mathbf{u})d\mathbf{u} - \int_{[0,1]^d} \mathbb{L}^{-1}_{min}(\mathbf{u})d\mathbf{u}}. \tag{4.5}$$

Using $\int_{[0,1]^d} \mathbb{L}^{-1}_{max}(\mathbf{u})d\mathbf{u} = 1$ *and* $\int_{[0,1]^d} \mathbb{L}^{-1}_{min}(\mathbf{u})d\mathbf{u} = \frac{1}{(d+1)!}$ *(see Nelsen 2006) we can rewrite,*

$$G_{C,L_1^{-1},\dots,L_d^{-1}} = \frac{(d+1)! \int_{[0,1]^d} \mathbb{L}^{-1}_{C,L_1^{-1},\dots,L_d^{-1}}(\mathbf{u})d\mathbf{u} - 1}{(d+1)! - 1}.$$

Note, the similarity of the above definition to the univariate Gini coefficient. The latter is two times the area between the inverse Lorenz curve and the diagonal of the unit square, where the diagonal stands for the inverse Lorenz curve of minimal inequality. The factor two results from normalization to the unit interval. In our multivariate definition, we measure the volume enclosed by the actual Lorenz curve and the curve of minimal inequality (see numerator of Equation 4.5) and rescale the result to be between 0 and 1 (see denominator of that equation). Consequently, setting $d = 1$ yields,

$$G_{C,L_1^{-1}} = 2\int_{[0,1]} L_1^{-1}(u_1)du_1 - 1 = G_1$$

which is the Gini coefficient in the univariate case (see, e.g., Section 4.2).

For $d = 2$ we have $0 \leq G_{C,L_1^{-1},L_2^{-1}} \leq 1$, where $G_{C,L_1^{-1},L_2^{-1}} = 0$ implies that $C = W$ and $G_{C,L_1^{-1},L_2^{-1}} = 1$ implies that $C = M$. For $d \geq 3$ we have $0 < G_{C,L_1^{-1},\dots,L_d^{-1}} \leq 1$. This is due to the fact, that W is not a copula for $d \geq 3$. Further $G_{C,L_1^{-1},\dots,L_d^{-1}} = 1$ implies that $C = M$.

Example 4.4.1. *Consider the case of independent $X_1, \dots, X_d$, where cov$(X_i, X_j) = 0$, for $i, j \in \{1, \dots, d\}$ and $i \neq j$. With $C = \Pi$ and $\mathbf{u} = (u_1, \dots, u_d)$ it follows from*

$$\begin{aligned}\int_{[0,1]^d} \mathbb{L}^{-1}_{\Pi,L_1^{-1},\dots,L_d^{-1}}(\mathbf{u})d\mathbf{u} &= \Big\{1 - E(X_1^*)\Big\}\Big\{1 - E(X_2^*)\Big\}\dots\Big\{1 - E(X_d^*)\Big\} \\ &= \left(\frac{1}{2}\right)^d \prod_{j=1}^{d}(1 + G_j)\end{aligned}$$

and Definition 4.4.1 that the MEGC can be written as

$$G_{C,L_1^{-1},\dots,L_d^{-1}} = \frac{(1+d)!(\frac{1}{2})^d \prod_{j=1}^{d}(1+G_j) - 1}{(d+1)! - 1}.$$

In the special case $d = 2$, we obtain

$$G_{1,2} = \frac{1}{10}\left(1 + 3G_1 + 3G_2 + 3G_1G_2\right). \tag{4.6}$$

Focusing on the two dimensional case, the MEGC is decomposable into the marginal Gini coefficients and a term resulting from the dependence structure.

Theorem 4.4.1. *Decomposition of two dimensional MEGC*

The two dimensional MEGC can be expressed as

$$G_{C,L_1^{-1},L_2^{-1}} = G_{1,2} = \frac{6}{5}E(X_1^*X_2^*) + \frac{3}{5}G_1 + \frac{3}{5}G_2 - \frac{1}{5}$$

Proof. Note, that $cov(X_1^*, X_2^*) = \int_0^1 \int_0^1 \left[C\{L_1^{-1}(u_1), L_2^{-1}(u_2)\} - L_1^{-1}(u_1)L_2^{-1}(u_2) \right] du_1 du_2$ (Nelsen, 2006) and remember that the *cdf* of X_i^* is L_i^{-1}, for $i = 1, 2$. So $\int L_1^{-1}(u_1)du_1 = 1 - \int L_1(u_1)du_1$, with $\int L_1(u_1)du_1 = E(X_1^*)$. It follows

$$\begin{aligned}\int_{[0,1]^2} \mathbb{L}^{-1}_{C,L_1^{-1},L_2^{-1}}(u_1, u_2)du_1du_2 &= cov(X_1^*, X_2^*) + \int_0^1 L_1^{-1}(u_1)du_1 \int_0^1 L_2^{-1}(u_2)du_2 \\ &= cov(X_1^*, X_2^*) + \{1 - E(X_1^*)\}\{1 - E(X_2^*)\} \\ &= cov(X_1^*, X_2^*) + 1 - E(X_1^*) - E(X_2^*) + E(X_1^*)E(X_2^*) \\ &= E(X_1^*X_2^*) + \frac{1}{2}G_1 + \frac{1}{2}G_2\end{aligned}$$

and therefore

$$G_{1,2} = \frac{6\int_{[0,1]^2} \mathbb{L}^{-1}_{C,L_1^{-1},L_2^{-1}}(u_1, u_2)du_1du_2 - 1}{5} = \frac{6}{5}E(X_1^*X_2^*) + \frac{3}{5}G_1 + \frac{3}{5}G_2 - \frac{1}{5}$$

□

It follows from Theorem 4.4.1 that upper and lower bounds for $G_{1,2}$ are given by

$$\begin{aligned}\tfrac{3}{5}G_1 + \tfrac{3}{5}G_2 - \tfrac{1}{5} \quad \leq G_{1,2} \quad &= \tfrac{6}{5}E(X_1^*X_2^*) + \tfrac{3}{5}G_1 + \tfrac{3}{5}G_2 - \tfrac{1}{5} \\ &\leq \tfrac{6}{5}min\{E(X_1^*), E(X_2^*)\} + \tfrac{3}{5}G_1 + \tfrac{3}{5}G_2 - \tfrac{1}{5} \\ &= \tfrac{6}{5}min\{\tfrac{1}{2} - \tfrac{1}{2}G_1, \tfrac{1}{2} - \tfrac{1}{2}G_2\} + \tfrac{3}{5}G_1 + \tfrac{3}{5}G_2 - \tfrac{1}{5} \\ &= \tfrac{6}{5}(\tfrac{1}{2} - \tfrac{1}{2}max\{G_1, G_2\}) + \tfrac{3}{5}G_1 + \tfrac{3}{5}G_2 - \tfrac{1}{5} \\ &= \tfrac{2}{5} - \tfrac{3}{5}max\{G_1, G_2\} + \tfrac{3}{5}G_1 + \tfrac{3}{5}G_2\end{aligned}$$

Note, that the sum of weights is 1 in the lower and upper bound.

Table 4.1 shows the univariate Gini coefficients and the MEGC for the examples using the Gaussian copula from Figure 4.1 and Figure 4.2.

Table 4.1.: Univariate Gini coefficients and corresponding MEGC

Copula type	**(Fig. No.)**	**MEGC**
Gaussian $\rho = -0.9$	(1a)	0.25
Gaussian $\rho = -0.5$	-	0.29
Gaussian $\rho = 0.0$	(1b)	0.33
Gaussian $\rho = 0.5$	(1c)	0.39
Gaussian $\rho = 0.9$	(1d)	0.43
Gumbel $\rho = 0.8$	(2a)	0.42
Clayton $\rho = 0.8$	(2b)	0.41
Gaussian $\rho = 0.0$	(2c)	0.22
Clayton $\rho = 0.8$	(2d)	0.30

Values of MEGC for the examples from Figure 4.1 and Figure 4.2. Note that the univariate Gini coefficients are 0.33 in all cases, except for the last two where the marginal Gini coefficient of the first variable equals 0.05. Further notice that the MEGC, unlike a convex combination, is not necessarily enclosed by the marginal univariate Gini coefficients.

As expected, the MEGC increases with increasing strength of the dependence between X_1 and X_2. For the example of wealth and income, the influence of the dependence structure on the MEGC is positive if a rich person tends to belong to the group of high income individuals and is negative if a rich person is more likely to belong to the individuals with low income. Note, that also values $G_{1,2} > max\{G_1, G_2\}$ and $G_{1,2} < min\{G_1, G_2\}$ are possible to correctly capture the influence of the dependence on the inequality. This is in contrast to a convex combination of G_1 and G_2. Consider for example the first eight cases in Table 4.1, where we have $G_1 = G_2 = 1/3$. If the MEGC would be bounded by these values to be equal to $1/3$ in all cases, the different dependence structures would not be reflected.

4.5. Nonparametric Estimation of the Multivariate Lorenz Curve (MEILC) and the Corresponding Multivariate Gini Coefficient (MEGC)

We assume that we have observations $\boldsymbol{X}_1, \dots, \boldsymbol{X}_n$ on $\boldsymbol{X} = (X_1, \dots, X_d)$, where $\boldsymbol{X}_j = (X_{j1}, X_{j2}, \dots, X_{jd})$ for $j \in \{1, \dots, n\}$. We only consider the case where $n > d$, where we have more observations than dimensions. If F_i and L_i would be known for $i \in \{1, \dots, d\}$ we could easily derive observations $\boldsymbol{X_1^*, \dots, X_n^*}$ on $\boldsymbol{X}^*$ with $X_{ji}^* = L_i\big\{F_i(X_{ji})\big\}$ for $i \in \{1, \dots, d\}$ and $j \in \{1, \dots, n\}$. However, F_i and L_i are unknown in practical applications and have to be estimated using $\boldsymbol{X}_1, \dots, \boldsymbol{X}_n$.

We estimate F_i for $i \in \{1, \dots, d\}$ by its empirical counterpart

$$\hat{F}_{ni}(x) = \frac{1}{n} \sum_{j=1}^{n} \mathbf{1}_{\{X_{ji} \leq x\}} \text{ for } x \in \mathbb{R}.$$

L_i is usually estimated by

$$\hat{L}_{ni}\left(u = \frac{k}{n}\right) = \frac{\sum_{j=1}^{k} X_{[j:n]i}}{\sum_{j=1}^{n} X_{ji}} \qquad \text{for } k = 0, 1, \dots, n \text{ and } i \in \{1, \dots, d\},$$

where $X_{[1:n]i} \leq X_{[2:n]i} \leq \dots \leq X_{[n:n]i}$ is the increasingly ordered sequence of X_{ji} and linear interpolation between $\hat{L}_{ni}(u = \frac{k}{n})$ and $\hat{L}_{ni}(u = \frac{k-1}{n})$ for $k = 1, 2, \dots, n$. This is tantamount to the compact formula

$$\hat{L}_{ni}(u_i) = \frac{\int_0^{u_i} \hat{F}_{in}^{-1}(t)dt}{\frac{1}{n}\sum_{j=1}^{n} X_{ji}} \qquad \text{for } i \in \{1, \dots, d\} \text{ and } u_i \in [0, 1].$$

It is now possible to estimate the observations X^*_{ji} by what we suggest to call "pseudo-observations" of X^*_{ji} with

$$\hat{X}^*_{ji,n} = \hat{L}_{ni}\{\hat{F}_{ni}(X_{ji})\} = \frac{\sum_{l:X_{li}\leq X_{ji}} X_{li}}{\sum_{l=1}^{n} X_{li}} \text{ for } i \in \{1,\dots,d\} \text{ and } j \in \{1,\dots,n\} \qquad (4.7)$$

and obtain the corresponding vector $\hat{\boldsymbol{X}}^*_{j,n} = (\hat{X}^*_{j1,n}, \hat{X}^*_{j2,n} \dots, \hat{X}^*_{jd,n})$ for $j \in \{1,\dots,n\}$.

4.5.1. Estimation of the MEILC

It was pointed out that the MEILC is given by

$$\mathbf{u} = (u_1,\dots,u_d) \longmapsto C\Big\{L_1^{-1}(u_1),\dots,L_d^{-1}(u_d)\Big\} \qquad \text{for } \mathbf{u} \in [0,1]^d$$

and that it is the joint distribution function of the vector $\boldsymbol{X}^* = (X^*_1,\dots,X^*_d)$. Therefore, the MEILC is estimated by the empirical distribution function based on $\hat{\boldsymbol{X}}^*_{j,n}$ for $j \in \{1,\dots,n\}$, i.e.,

$$\hat{\mathbb{L}}^{-1}_{C,L_1^{-1},\dots,L_d^{-1},n}(u_1,\dots,u_d) = \frac{1}{n}\sum_{j=1}^{n}\prod_{i=1}^{d}\mathbf{1}_{\{\hat{X}^*_{ji,n}\leq u_i\}}.$$

4.5.2. Estimation of the Multivariate Gini Coefficient (MEGC)

In order to estimate $G_{C,L_1^{-1},\dots,L_d^{-1}}$ we have to estimate the integral

$$I_{C,L_1^{-1},\dots,L_d^{-1}} = \int_{[0,1]^d} \mathbb{L}^{-1}_{C,L_1^{-1},\dots,L_d^{-1}}(\mathbf{u})d\mathbf{u}$$

for $\mathbf{u} = (u_1, \ldots, u_d)$, which is usually done by

$$\begin{aligned}\hat{I}_{C,L_1^{-1},\ldots,L_d^{-1},n} = \int_{[0,1]^d} \hat{\mathbb{L}}^{-1}_{C,L_1^{-1},\ldots,L_d^{-1},n}(\mathbf{u})d\mathbf{u} &= \frac{1}{n}\sum_{j=1}^{n}\int_{[0,1]^d}\prod_{i=1}^{d}\mathbf{1}_{\{\hat{X}^*_{ji,n}\leq u_i\}}du_i \\ &= \frac{1}{n}\sum_{j=1}^{n}\prod_{i=1}^{d}\int_0^1 \mathbf{1}_{\{\hat{X}^*_{ji,n}\leq u_i\}}du_i \\ &= \frac{1}{n}\sum_{j=1}^{n}\prod_{i=1}^{d}(1-\hat{X}^*_{ji,n})\end{aligned}$$

After normalizing we obtain the estimator

$$\hat{G}_{C,L_1^{-1},\ldots,L_d^{-1},n} = \frac{(d+1)!\frac{1}{n}\sum_{j=1}^{n}\prod_{i=1}^{d}(1-\hat{X}^*_{ji,n})-1}{(d+1)!-1} \tag{4.8}$$

4.6. Considerations on Transfers

In this section we want to discuss some considerations on transfers and their effect on the MEILC and MEGC. We are aware that this is a very wide and complex topic, so we can not cover it in all its aspects. However, we at least want to share first considerations and encourage further research on this topic.

First, we define the *Correlation Increasing Transformation (CIT)* introduced by Tsui (1998) into the inequality literature and further considered by many authors, e.g. Epstein and Tanny (1980); Atkinson and Bourguignon (1982); Decancq (2012); Gravel and Moyes (2012) or lately Faure and Gravel (2021).

Definition 4.6.1. *Correlation Increasing Transformation (CIT)*
We are considering two possible distributions or allocations A and B of d variables among a fixed number of individuals. Let, e.g., t_A denote the d-dimensional vector of variables of individual t in allocation scenario A, with analogue expressions for other individuals and

distributions. We say that distribution B is obtained from distribution A by a Correlation Increasing Transformation (CIT), if for two individuals t and z with d-dimensional attribute vectors $t_B, z_B \in \mathbb{R}^d$ we have the reallocation

$$t_B := max\{t_A, z_A\} \text{ and } z_B := min\{t_A, z_A\}, \tag{4.9}$$

while the variable vectors of all other individuals stay unchanged, i.e., $m_B = m_A$ for all other individuals $m \notin \{t, z\}$. Here, max/min denote the element-wise maximum/minimum.

Note, that within our framework, this corresponds to only swapping $\hat{X}^*$ values between two individuals while all others $\hat{X}^*$ values remain unchanged. A distribution B is called a *Correlation Increasing Majorization (CIM)* of distribution A if it is obtained by a finite sequence of CIT's from A.

The CIT naturally affects the order of multivariate Lorenz curves (MEILC) as summarized in the following proposition 4.6.1.

Proposition 4.6.1. *Any Correlation Increasing Transformation (CIT) or Correlation Increasing Majorization (CIM) from a distribution A towards a distribution B, implies the multivariate Lorenz order $B \succeq A$ from Definition 4.3.1.*

Keeping in mind, that a CIT only exchanges values of $\hat{X}$ of two individuals (with the same effect to $\hat{X}^*$) and that $\hat{\mathbb{L}}^{-1}$ is the joint distribution function of $\hat{\boldsymbol{X}}^* = (\hat{X}_1^*, \ldots, \hat{X}_d^*)$ for $k \in \{1, \ldots, d\}$, the proposition follows directly from Epstein and Tanny (1980).

Consequently, any CIT has also direct implications towards the MEGC.

Proposition 4.6.2. *Any Correlation Increasing Transformation (CIT) or Correlation Increasing Majorization (CIM), induces a higher multivariate Gini coefficient MEGC. An analogous statement applies to the decreasing counterparts of the operations, which induce lower MEGCs.*

Proposition 4.6.2 follows directly from Proposition 4.6.1 and Definition 4.4.1.

More intuitively, a CIT does only have an impact on the dependence structure, i.e., the copula, as it only swaps the coupling of realizations in the margins. Then, a CIT results per definition in more concordant dependence structure and consequently to an increased multivariate Gini coefficient.

Further, we want to elaborate the topic with the help of an example motivated by an anonymous referee. We look at a population of three individuals and consider the bivariate case, i.e., $d = 2$, where X_1 and X_2 might again stand for the individuals' income and wealth, respectively.

$$\begin{array}{lcccc}
 & & \text{Individual 1} & \text{Individual 2} & \text{Individual 3} \\
\text{Society 1} & \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} & \begin{pmatrix} 3 \\ 3 \end{pmatrix} & \begin{pmatrix} 4 \\ 4 \end{pmatrix} & \begin{pmatrix} 6 \\ 6 \end{pmatrix} \\
\text{Society 2} & \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} & \begin{pmatrix} 5 \\ 3 \end{pmatrix} & \begin{pmatrix} 2 \\ 4 \end{pmatrix} & \begin{pmatrix} 6 \\ 6 \end{pmatrix} \\
\text{Society 3} & \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} & \begin{pmatrix} 4 \\ 3 \end{pmatrix} & \begin{pmatrix} 3 \\ 4 \end{pmatrix} & \begin{pmatrix} 6 \\ 6 \end{pmatrix}
\end{array}$$

Society 1 is obtained from Society 3 by a simple CIT between Individual 1 and Individual 2. We therefore expect the MEGC of Society 1 to be higher than that of Society 3. For Society 2 the transfer is more complicated. Society 2 is obtained from Society 1 by a transfer of two units X_1 from Individual 2 to Individual 1. Reversely, this is equal to first applying a CIT to Society 2 (increases inequality) and then a transfer of one unit X_1 from Individual 2 to Individual 1, which is typically considered to reduce inequality. In this case we can not directly rank the distributions by means of their multivariate inequality from looking at the transfers. However, it is possible to rank the distributions by calculating the MEGC, resulting in $MEGC = 0.121$ for the first society, $MEGC = 0.098$ for the second society and $MEGC = 0.084$ for the third society. As expected, we see that Society 1 is more unequal than Society 3. Furthermore, we now can include Society 2 in the ordering.

More general, Pigou-Dalton transfers are widely known to reduce inequality in the univariate case (Dalton, 1920). However, the extension to the multivariate case is not straightforward and multiple suggestions have been made, see e.g. Basili et al. (2017); Bosmans et al. (2009); Banerjee (2014). The problem at hand in the multivariate case is that both, the marginal distributions and the dependence structure, can be changed at the same time, even in opposite directions, e.g., decreasing inequality in the margins while increasing in the dependence structure. Thus it might not be so clear to define types of pure basis transfers between two individuals that act always in the same direction with respect to margins and dependence structure. Further complicating matters, individuals who are not directly included in the transfer can be effected and general statements are very difficult to make. The following example illustrates the above and hopefully encourages further research.

Example 4.6.1. *Multivariate Pigou-Dalton-Bundle-Transfers (PDBT) are defined as non-negative transfer from one unambiguously richer individual to a poorer individual in each attribute. The amounts or the proportions of the transfers need not to be the same for all attributes, i.e., it is possible to transfer only one attribute (see Fleurbaey and Trannoy 2003; de la Vega et al. 2010). Consider the two societies below, where Society 2 is obtained from a PDBT of* 1.1 *units of* X_1 *from Individual 1 to Individual 4.*

		Individual 1	Individual 2	Individual 3	Individual 4	MEGC
Society 1	$\begin{pmatrix} X_1 \\ X_2 \end{pmatrix}$	$\begin{pmatrix} 5 \\ 4 \end{pmatrix}$	$\begin{pmatrix} 4 \\ 5 \end{pmatrix}$	$\begin{pmatrix} 3 \\ 2 \end{pmatrix}$	$\begin{pmatrix} 2 \\ 3 \end{pmatrix}$	0.131
Society 2	$\begin{pmatrix} X_1 \\ X_2 \end{pmatrix}$	$\begin{pmatrix} 3.9 \\ 4 \end{pmatrix}$	$\begin{pmatrix} 4 \\ 5 \end{pmatrix}$	$\begin{pmatrix} 3 \\ 2 \end{pmatrix}$	$\begin{pmatrix} 3.1 \\ 3 \end{pmatrix}$	0.141

The transfer from richer Individual 1 to Individual 4 obviously reduces the inequality in X_1*, but also affects Individuals 2 and 3 leading to another result in the multivariate case. Individual 2 now is the richest in both dimensions whereas Individual 3 is the poorest. Therefore, the transfer increased the dependence, more specific the rank-dependence, within the society which increases inequality in this case leading to a higher MEGC.*

Example 4.6.1 displays the complexity of the topic. Although a transfer seems to reduce inequality at the first glance, it may have an opposing efficacy within the dependence structure of the whole society. In our example, the PDBT results in a clear richest and poorest individual, decreasing the balancing effect of the dependence structure.

For more general considerations on transfers we refer to Epstein and Tanny (1980); Atkinson and Bourguignon (1982); Decancq (2012), or most lately Faure and Gravel (2021).

4.7. Analysis of Income and Wealth Inequality Using the MEILC and the MEGC

In the following section we demonstrate a possible application of the MEILC and the corresponding MEGC. The first example in Section 4.7.1 uses data for Germany (SOEP, 2019) and is implemented in Python 3.8 (Van Rossum and Drake, 2009). Section 4.7.2 is implemented via the LISSY R-Interface (Luxembourg Wealth Study (LWS) Database, 2020) and examines the MEGC of 13 additional countries.

4.7.1. MEILC and the MEGC for Germany 2017

We analyze the joint inequality of income and wealth in Germany based on the data provided by the Socio-Economic Panel (SOEP) for 2017 (SOEP, 2019). Detailed information about the survey and the methods used in the SOEP are provided by Goebel et al. (2019) and Wagner et al. (2007). The analysis is based on the variable *'i11102'* from the *pequi* dataset for income and *'n0111a'* from the *hwealth* dataset for wealth. Entries with negative values in one of the variables are dropped as suggested by many authors (see e.g., Rehm et al. 2014; Harvey et al. 2017; Saez and Zucman 2016; Formby et al. 1989). The data set refers to households, whereas inequality numbers are usually reported at the individual level. Income and wealth numbers in the data set are therefore broken down to the individual level. To this end, the values (income and wealth) are equivalised with respect of the number of household members by multiplying with

1/(household members)$^{0.5}$, see, e.g., OECD (distribution database, 2017). In a next step, each adjusted pair of income and wealth is replicated K times, where K is the product of the household members (e.g., to get 5 individual entries from a 5 person household) and the integer part of a weight reported in variable *'w11102'*. The variable *'w11102'* corrects for differences in the socio-economic distribution between households in the panel and all households of the country. Additionally, data errors are eliminated by excluding individuals with values lying more than 30 standard deviations off the mean.

The resulting data is presented in Figure 4.3. For better visualization, the figure only shows data with income and wealth below EUR 0.1 million and EUR 1 million, respectively. This corresponds to more than 98.0% of the data. The empirical copula of the full data set is shown in Figure 4.4 and the resulting MEILC in Figure 4.5. For the univariate Gini coefficients, we compute $G_1 = 0.29$ for income and $G_2 = 0.65$ for wealth. Thus, inequality in wealth is considerably higher than in income. This difference is also observable in the shape of the margins L^{-1}_{income} and L^{-1}_{wealth} of the MEILC in Figure 4.5. The multivariate Gini coefficient of wealth and income yields $G_{1,2} = 0.47$. Due to the moderate positive dependence of wealth and income (Spearman's ρ is $\rho = 0.56$, here) this is slightly higher than it would be for independent variables (compare to Equation 4.6 in Section 4.4).

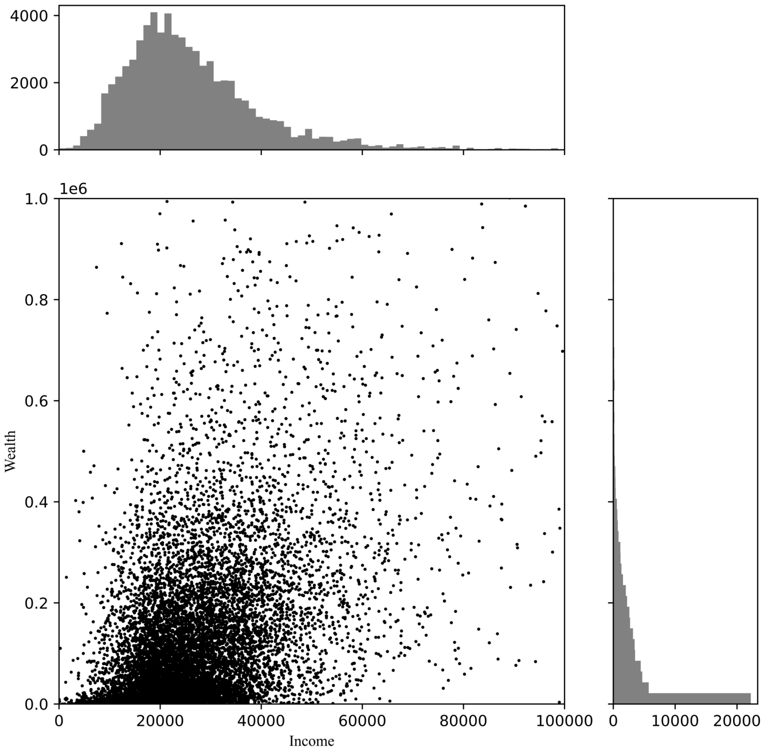


Figure 4.3.: Scatter-plot and corresponding histograms of income and wealth in Germany 2017 based on the SOEP dataset in Euro €.

4.7.2. MEGC of Income and Wealth for Other Countries

Table 4.2 summarizes the Gini coefficients for wealth G_1, income G_2 and the MEGC $G_{1,2}$ for several countries based on the Luxembourg Wealth Study (LWS) database (Luxembourg Wealth Study (LWS) Database, 2020). For an extensive documentation of the cross-national wealth database, see LIS (2019a) and LIS (2019b). In the analysis, the variables *disposable household income ('dhi')* and *disposable net worth ('dnw')* from the latest available data sets are used (if too many of the values are missing, we use the *total current income ('hitotal')* variable instead of *'dhi'*). The data processing is done analogously to the SOEP data set in the last section. Again, households with negative values in one or both variables are excluded. Both household variables are again broken down to individual levels. First we equalize by multiplying by 1/(household members)$^{0.5}$

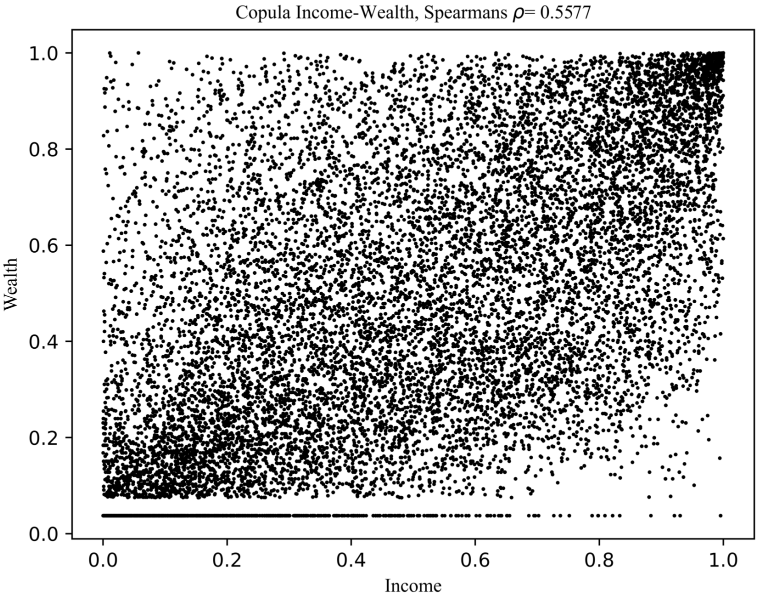


Figure 4.4.: Empirical copula of income and wealth in Germany 2017 based on the SOEP dataset.

and then we replicate each income and wealth pair in the sample according to the number of household members times the integer part of panel adjustment weights. In this database, the number of household members is stored in the variable *'nhhmem'*, while the adjustment weights are stored in *'hpopwgt'*.

Turning to Table 4.2, all reported numbers are plausible, and inequality is larger for wealth than income in all cases. With regard to wealth, South Africa, as well as the United States, have the highest inequality. South Africa also shows the largest inequality in income distribution. Considering all countries, dependencies between income and wealth are positive and mainly moderate. The reported numbers of Spearman's rho are often below 0.5. For this reason, all reported MEGC numbers of multivariate inequality lie well between the univariate Gini coefficients. The highest MEGC is reported for South Africa, followed by the United States. The lowest MEGC numbers are reported for Slovakia and Finland. An interesting example of the effect of the dependence structure

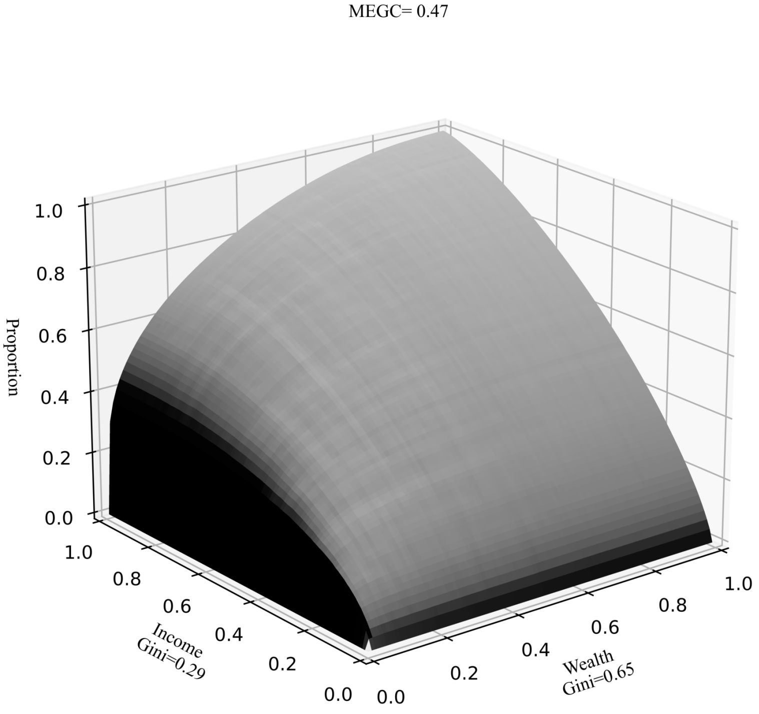


Figure 4.5.: MEILC of income and wealth in Germany 2017 based on the SOEP dataset.

on inequality is the pair of Italy and Slovenia. While marginal inequalities in wealth and income are slightly lower in Italy than in Slovenia, the stronger dependency of these variables in Italy (i.e., the rich tend to coincide with the higher earners) results in a higher MEGC in Italy than in Slovenia. As an alternative, a graphical illustration of the dominance structure within the three aspects, two univariate Gini coefficients, and the MEGG, we provide a Hasse-diagram in Figure 4.6. There, the concordant ordering of all three aspects between the two countries results in a connecting edge within the graph.

Note that the values for the univariate Gini coefficients can differ from other publications for various reasons. First, we do not apply any top or bottom coding of the data and exclude all individuals with negative values in the variables. Second, we floor the provided weights to the next integer because of computational reasons. Third, we only use complete

Table 4.2.: Gini coefficient, MEGC and and Spearman's ρ on wealth and income

Country	Gini on income	Gini on wealth	MEGC	Spearman's ρ
Australia	0.33	0.62	0.46	0.28
Austria[1]	0.28	0.67	0.47	0.41
Canada	0.32	0.66	0.48	0.41
Finland	0.25	0.60	0.43	0.41
Germany[2]	0.29	0.65	0.47	0.56
Greece[1]	0.32	0.55	0.45	0.41
Italy	0.34	0.58	0.48	0.56
Luxembourg[1]	0.39	0.63	0.51	0.54
Slovakia[1]	0.34	0.51	0.44	0.40
Slovenia[1]	0.36	0.59	0.46	0.29
South Africa	0.61	0.85	0.71	0.43
Spain[1]	0.38	0.60	0.50	0.45
United Kingdom	0.35	0.60	0.48	0.55
United States	0.45	0.80	0.61	0.63

Gini coefficient, MEGC and and Spearman's ρ on wealth and income for multiple countries based on the Luxembourg Wealth Study (LWS) Database (2020) database. [1]*hitotal* instead of *'dhi'* from LWS dataset used because of missing values. [2] From SOEP data, Section 4.7.1.

cases of the datasets. This means we consider a person in our calculation only if all information (*dhi, dnw, hpopwgt, nhhm*) of the case is available. Furthermore, we treated both variables in the same way, especially when adjusting for the household size. See Sierminska and Smeeding (2005) for a brief discussion on the topic. Last, we used only the data provided by the LWS database and did not supplement it with data from other sources.

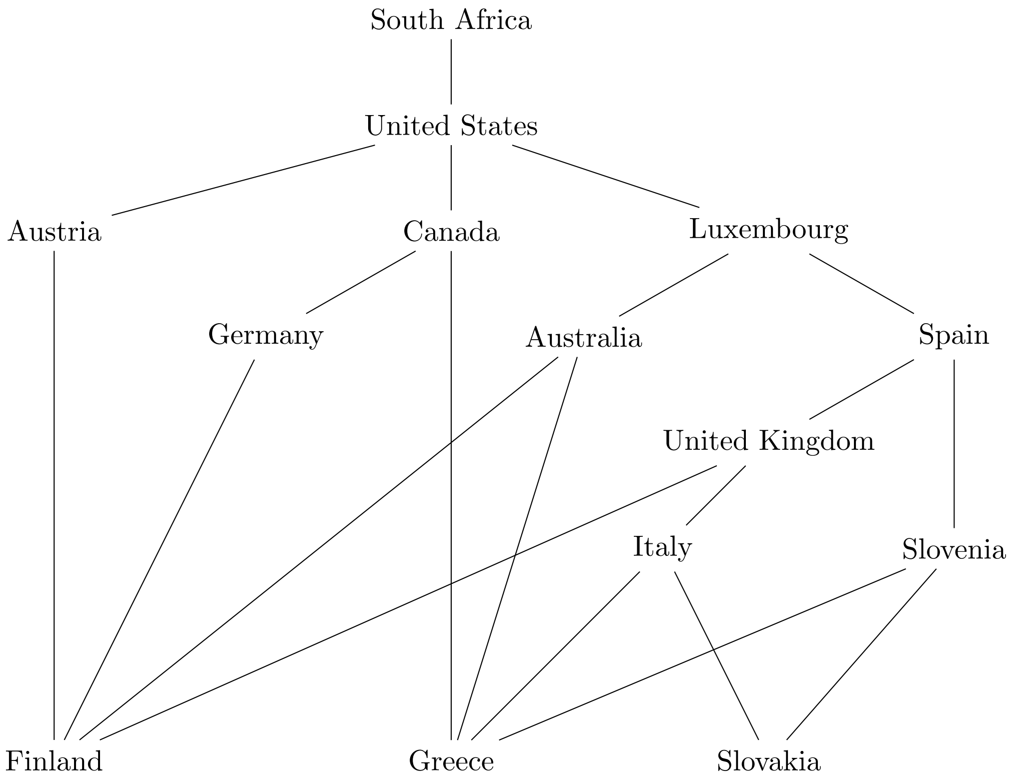


Figure 4.6.: Illustration of the incomplete ranking resulting from table 4.2 in terms of a Hasse diagram. Dominance in all three Gini coefficients leads to an edge between countries, whereas inequality decreases from the top down.

5. From Point Forecasts to Multivariate Probabilistic Forecasts: The Schaake Shuffle for Day-Ahead Electricity Price Forecasting

This chapter is based on a joint publication with Oliver Grothe and Fabian Krüger in *Energy Economics* (Grothe et al., 2023) dealing with the forecast of electricity prices. Permission to reuse the article, as well as the figures, in this work is granted by the copyright holder. Another paper published during my time as a Ph.D. student related to energy markets estimates in-feeds of the biggest European offshore wind farms (Grothe et al., 2022b). However, it is not part of this thesis.

5.1. Introduction

Day-ahead electricity price forecasting is a critical element in the decision-making of energy companies. Accordingly, an active applied research literature is concerned with developing and comparing price forecasting methods. The majority of this literature addresses point forecasts, for which a wide range of methods have been proposed. These efforts, which have recently been reviewed by Hong et al. (2020) and Lago et al. (2021), have given rise to a rich toolkit that includes time series models, regularized regression techniques, deep learning models, as well as strategies for parameter estimation, variable selection, hyperparameter selection and forecast combination.

While univariate point forecasts are an important first step, they are typically limited in two respects. First, they usually do not address forecast uncertainty, which is an important concern for economic decision making. The studies surveyed by Nowotarski and Weron (2018) and Ziel and Steinert (2018) address this aspect by considering probabilistic forecasting in energy markets. While growing, the corresponding literature and range of approaches are still more limited than for point forecasting. Second, univariate forecasts are concerned with a single measurement unit (such as a given time period or a given location), whereas economic decisions often require joint forecasts for several time periods or locations. In applications, the two aspects of forecast uncertainty and multivariate dependence may well interact. For example, managing daily price risks requires information on both the uncertainty in hourly prices and information on dependencies of prices across hours. Consider the sum of the prices from two consecutive hours. If the price of the first hour is higher than predicted, the price of the second predicted hour is likely to be higher as well. Assuming independence of these prices would underestimate the uncertainty of the sum, leading to sub-optimal economic decisions. Dependence modeling thus plays a vital role in uncertainty estimation.

This chapter of the thesis considers a generic and easy-to-implement method for constructing multivariate probabilistic forecasts based on univariate point forecasts. The method uses past forecast errors to learn about forecast uncertainty and dependencies across measurement units. If necessary, we first fit simple time series models to the univariate point forecast errors, thus accounting for possible predictability in the errors' conditional mean and variance. Such methods have been proven successful in forecasting electricity prices, e.g., by Weron and Misiorek (2008), Jan et al. (2022), Bibi et al. (2021) or Garcia et al. (2005). In contrast to these authors, however, we use the methods for modeling the errors of given point forecasting models instead of designing stand-alone forecasting models. For dependence, we apply straightforward empirical copula methods.

Being based on forecast errors, our approach leverages the rich literature on univariate point forecasting (thus avoiding to re-invent the wheel) while at the same time addressing forecast uncertainty and multivariate dependence. Our framework is inspired by the literature on post-processing multivariate ensemble forecasts in meteorology (Clark et al., 2004; Schefzik et al., 2013; Vannitsem et al., 2021). Here, the point forecast errors of numerical weather prediction models are used to learn about forecast uncertainty and (possibly) about multivariate dependencies across variables, locations, or time points. Related post-processing approaches have been considered in macroeconomics; in particular, Clark et al. (2020) consider a Bayesian approach for modeling forecast errors across multiple time points.

Given the rich literature on point forecasting and the wide availability of data for learning structures in past point forecasting errors, post-processing seems particularly promising for forecasting energy prices or demand. Nevertheless, the literature on post-processing of energy forecasts is relatively sparse. In a univariate context, studies such as Marcjasz et al. (2020) and Kath and Ziel (2021) discuss approaches to estimate forecast uncertainty from past point forecast errors. Phipps et al. (2022, 2020) and Ludwig et al. (2022) consider post-processing univariate or multivariate weather forecasts in a situation where these are used as an input to energy forecasting models. In a multivariate setting, Muniain and Ziel (2020) deal with bivariate probabilistic price forecasting in off-peak and peak time series from the German-Austrian day-ahead price using a residual based approach that is conceptually similar to post-processing. Janke and Steinke (2020) apply an implicit generative model to generate a multivariate forecast distribution for energy prices from an ensemble of univariate point forecasts. Furthermore, Chai et al. (2019) employ a Gaussian copula to generate scenarios on the basis of an ensemble from extreme learning machines. Compared to the latter two studies, our use of empirical copula methods is considerably simpler to implement and more easily comparable to the large meteorological literature. Other uses of copula methods to model multivariate dependence in energy contexts include Toubeau et al.

(2019) employing empirical copulas to create scenarios of load, renewable generation, and prices; Pinson and Girard (2012) introducing multivariate verification tools for sets of scenarios; Manner et al. (2016) modeling electricity price spikes in two markets; and Pircalabu and Benth (2017) investigating the joint behavior of electricity prices in interconnected markets. Furthermore, existing multivariate probabilistic techniques for day-ahead electricity prices include functional models, where the daily profile is modeled as a functional observation, allowing the use of additional information of the data (cf. Shah and Lisi, 2015); vector autoregressive models, where the dependence between univariate time series is taken into account (cf. Panagiotelis and Smith, 2008; Shah et al., 2021); and combinations of functional models and stochastic methodologies (cf. Lisi and Shah, 2020; Chen and Li, 2017; Shah et al., 2020). Furthermore, Arpinoa et al. (2021) and Kou et al. (2015) exploit multivariate Gaussian Processes to model the multivariate density of prices. However, these are either standalone methodologies (thus do not leverage the rich literature on univariate point forecasts), assume a fixed, parametric distribution of errors (e.g., Gaussian), or do not take the dependence structure between errors into account. In contrast to these approaches, our methodology simply post-processes univariate point forecasts, resulting in a multivariate distribution. Our approach can be used in a fully parametric, non-parametric, or semi-parametric manner, incorporating a plausible dependence structure of error terms.

We next present a more specific summary of our approach, which is illustrated in Figure 5.1. Full details on the methodology are provided in Section 5.3 of this chapter. We consider a vector of 24 day-ahead point forecasts, each of which refers to one hour h of the following day t stemming from any given point forecasting model visualized by the grey box. From now on, we stick to this hourly example to simplify the presentation, but quarter-hourly or minute-by-minute forecasts could be handled analogously. The prediction is subject to an error defined by $\epsilon_{t,h} = y_{t,h} - \hat{y}_{t,h}$, where $y_{t,h}$ denotes the true price and $\hat{y}_{t,h}$ the prediction for each day t and hour $h = 1, \dots, 24$. Conditional on the information set $\mathcal{F}_{t-1}$, available at day $t-1$, $\epsilon_{t,h}$ is a random variable with unknown, conditional distribution function that we model as $G_{t,h}(x) := F_h(\frac{x-\mu_{t,h}}{\sigma_{t,h}})$. Here, F_h reflects an hour specific shape of the distribution while $\sigma_{t,h}$ and $\mu_{t,h}$ are scale and location parameters.

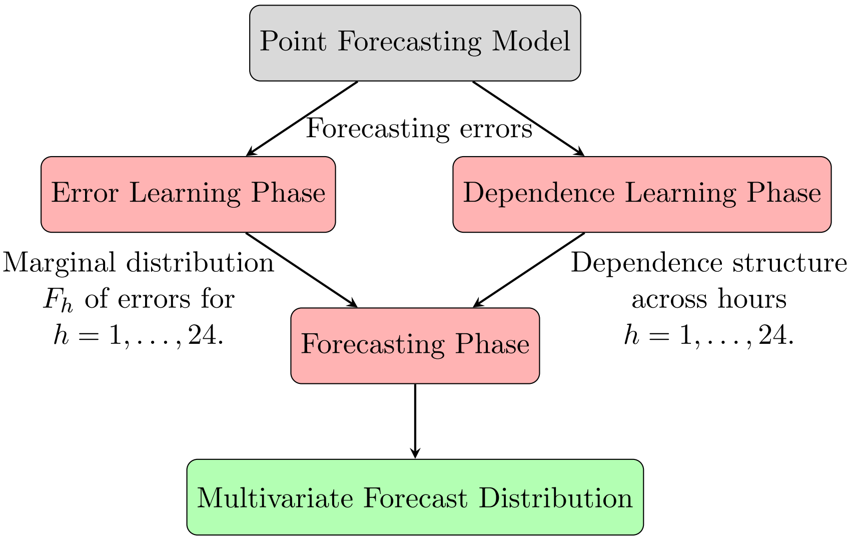

Point Forecasting Model
Forecasting errors
Error Learning Phase
Dependence Learning Phase
Marginal distribution F_h of errors for $h = 1, \ldots, 24$.
Dependence structure across hours $h = 1, \ldots, 24$.
Forecasting Phase
Multivariate Forecast Distribution

Figure 5.1.: Schematic representation of the proposed method. The text next to each arrow represents the result of the corresponding phase. More details are given in the main text around the figure.

Next, these errors are used in the following phases of the proposed method, colored red in Figure 5.1. In what we call the *Error Learning Phase*, the error distributions F_h, as well as the scale and location parameters are estimated. We allow a potentially time-varying conditional mean $\mu_{t,h} = \mathbb{E}[\epsilon_{t,h}|\mathcal{F}_{t-1}]$ and variance $\sigma^2_{t,h} = \mathbb{V}[\epsilon_{t,h}|\mathcal{F}_{t-1}]$. The nonzero conditional mean $\mu_{t,h}$ is motivated by possible misspecification of the point forecasting model, leading to predictable forecast errors. Time variation in the conditional variance $\sigma^2_{t,h}$ accommodates for differences in price uncertainty over time. If predictable structure in the data is apparent, $\mu_{t,h}$ and $\sigma^2_{t,h}$ can be tracked by time series models, e.g., by estimating the models' parameters in rolling windows of past forecast errors, separately for each hour. Alternatively, with no predictable structure apparent, we simply set $G_{t,h}(x) := F_h(x)$, so that the error distribution remains constant over time. We call the latter the 'raw'-error approach.

In the *Dependence Learning Phase*, the procedure aims to learn the dependence of the given point forecast errors. Therefore, we estimate the joint distribution of the 24-dimensional vector of day-ahead forecasting errors $\epsilon_{t,h}$. We disentangle univariate error distributions and their dependence by inferring the copula from the corresponding uniformly standardized values, i.e., $\hat{u}_{t,h} := \hat{G}_{t,h}(\epsilon_{t,h})$, where $\hat{G}_{t,h}$ has been obtained from the *Error Learning Phase.* Finally, the learned error distribution and dependence structure are passed on and combined in the *Forecasting Phase* to construct a forecast distribution consisting of m simulated time paths of day-ahead electricity prices, with each path consisting of 24 hourly prices. This resulting simulated distribution (colored green in Figure 5.1) encodes both forecast uncertainty for each individual hour, and the temporal dependence across the day's 24 hours. For example, if the realized price of the first hour exceeds the point forecast, the model typically places a high probability on a positive forecast error in the second hour as well. We thus obtain the full univariate forecast distribution (and not only an interval) for each hour $h = 1, \ldots, 24$ of the day, as well as joint behavior of prices across hours. Among others, the forecast distribution can be used to assess the likelihood of price-related events (such as the event of a price spike) and to measure the uncertainty of summary random variables (such as the sum of hourly prices). We consider an example for the price of a Standard Load Profile (SLP), reflecting a typical daily pattern of energy consumption, and show that the proposed method is able to capture the dependence structure of hourly prices, leading to a realistic assessment of price risks. Furthermore, we highlight the importance of the dependence structure in a multivariate probabilistic forecast by comparing our results to their naïve counterpart that assumes temporal independence of the prices.

In summary, the contributions of the chapter are:

1. We provide a methodology to leverage any given point forecasting model using a post-processing approach.
2. The methodology addresses forecast uncertainty by providing a probabilistic forecast of the variable of interest.

3. Dependence structures across time are taken into account by using copula techniques.
4. Rigorous assessment of the proposed method in five different benchmark data sets, as well as in a load forecasting example.

The remainder of the chapter is structured as follows. Section 5.2 presents the used electricity price data sets and day-ahead load data that we consider in our application. In Section 5.3 we explain our proposed method in detail and give a toy example. Section 5.4 provides a case study for multivariate probabilistic electricity price forecasting in five markets and an example for the price of a Standard Load Profile in Germany. Further, we apply our method on a highly seasonal load forecasting time series to show the benefit of the time series component. Last, Section 5.5 summarizes our work and points toward future research.

5.2. Data and Point Forecasts

5.2.1. Data Sets

The method we consider is based on a data set of past point forecasts and corresponding realizations. In our illustrations, we utilize point forecasts of energy prices and the corresponding realized prices. As discussed in the introduction, a wide range of point forecasting models and data sources are available for this purpose. Here we consider the five data sets recently proposed as replicable and transparent benchmarks by Lago et al. (2021). We access the data via the python library *epftoolbox*.[1] The

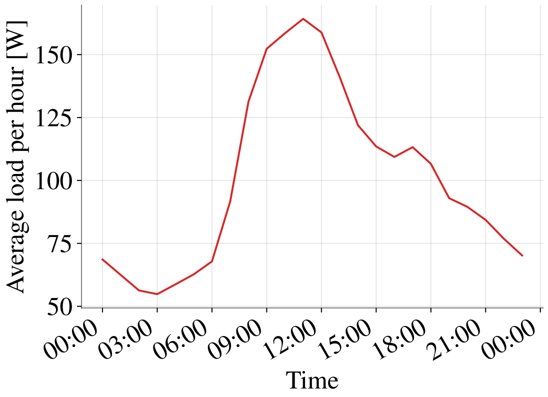


Figure 5.2.: Load profile (SLP G0), averaged over all seasons, weekdays, and hours in [kW] for a normalized consumption of 1.000kWh/year.

[1] https://epftoolbox.readthedocs.io/en/latest/modules/cite.html

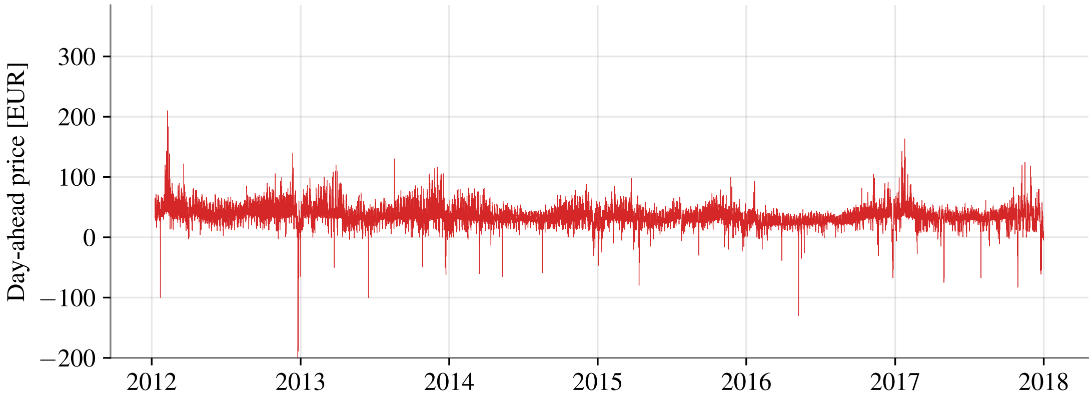


Figure 5.3.: Day-ahead electricity prices in EUR/MW from 4 January 2012 to 31 December 2017 in the EPEX-DE dataset (Lago et al., 2021).

data covers five day-ahead electricity markets, each spanning a history of six years. Namely, the EPEX-DE dataset covers the German electricity market; the PJM the Pennsylvania-New Jersey-Maryland market; the EPEX-BE the day-ahead electricity market in Belgium; the EPEX-FR the day-ahead electricity market in France; and the NORD POLE the European power market of the Nordic countries. Note that the python library only provides an interface to easily access the data, which is all openly available at the ENTSO-E transparency platform.[2] In addition to observations of day-ahead prices, two time series of influential exogenous variables are included, which differ for each market. To illustrate the data, Figure 5.3 displays the realized day-ahead electricity prices in EUR per MW from 4 January 2012 to 31 December 2017 in the data set for the German electricity market (EPEX-DE).

Our method builds on a series of point forecasts. For the five data sets above we use point forecasts stemming from the the LEAR model by Lago et al. (2021), which we implement with a two-year calibration window.[3] The LEAR model is a parameter-rich

[2] `https://transparency.entsoe.eu/`

[3] Forecast averaging across calibration windows of different lengths could possibly yield improved results (Hubicka et al. 2019; Marcjasz et al. 2018); here we consider the simpler choice of a single calibration window in order to retain focus on the proposed method.

ARX model (that is, an autoregressive model with a large number of 'exogenous' regressors or features) estimated using LASSO for feature selection. In total, 247 features are considered, including day-ahead prices, day-ahead forecasts of the two exogenous variables, and historical day-ahead forecasts of the exogenous variables, all stemming from previous days and weeks. Additionally, a dummy variable for the day of the week is included. Finally, all variables are preprocessed with the arc hyperbolic sine (asinh) variance stabilizing transformation. Details on the data and the LEAR model can be found in Lago et al. (2021, Section 4.2).

In our forecast evaluation analysis, we further consider a *Standard Load Profile (SLP)* that provides a practically relevant example of intra-day seasonality in energy demand. We use this profile to construct a single daily energy price from 24 hourly prices. This setup allows us to assess whether our multivariate probabilistic predictions of hourly prices enable a realistic estimation of the uncertainty in daily prices. In practice, German energy suppliers use SLPs to model the consumption patterns of electricity customers without registered power metering. The SLPs replace the non-existent load profile of these customers with a forecast of electricity consumption during every quarter-hour. We utilize SLP G0 (see Figure 5.2), which represents the average of all industrial SLPs in Germany and is provided by the German Association of Energy and Water Industries (Bundesverband der Energie- und Wasserwirtschaft e.V., 2021).

We also demonstrate the application of our method for the related task of load forecasting. To this end, we use openly available day-ahead load forecasts from the ENTSO-E transparency platform for the German market from 2016 to 2020. Load forecasting is essential for the planning and operation of energy suppliers, system operators, and other market participants (Weron, 2006, Chapter 3). Figure 5.4 shows the time series of errors that results from the forecasts provided on the platform. Upon closer inspection, the error series displays strong seasonal patterns and therefore calls for appropriate time series modeling techniques for post-processing.

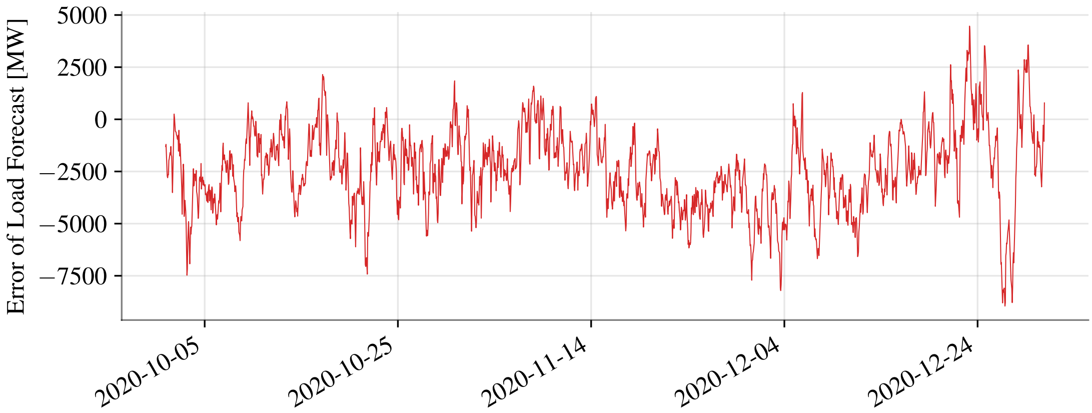


Figure 5.4.: Load forecast errors for Germany from 1 October 2020 to 31 December 2020.

5.2.2. Sample Choices for Forecasting Scheme

Training of our model is based on two phases. First, the univariate error distribution is learned (*Error Learning Phase*). Second, the joint behavior of realized and standardized errors is modeled by their copula (*Dependence Learning Phase*). Both phases require calibration windows for learning and we implement the proposed method using backward-looking rolling windows. The windows should be long enough to provide stable estimates but also short enough to adapt quickly to changing market developments. Here, we use about one year (364 days) of raw errors $\epsilon_{t,h}$ for the Error Learning Phase. Then, the raw errors are standardized with the help of an appropriate time series model, and the most recent 90 standardized errors are used to infer the distribution of standardized errors F_h. Alternatively, when the raw errors do not show temporal structures, the 24 distributions $F_h, h = 1, \ldots, 24$ are inferred directly from the most recent 90 raw errors without filtering them through a time series model. We call the latter 'raw-error' approach. Finally, the Dependence Learning Phase estimates multivariate dependence by computing the copula between the 24 distributions F_h based on a windows of the last 90 filtered or raw errors, respectively. While the trade-off between too short or too long window sizes needs to be resolved individually for each use case, our experiments indicate that the above lengths are a good baseline setting for hourly prices.

After the next day's distribution of observations is predicted, the windows are rolled over by one day and the estimates of the error distributions and dependency structures are updated. The initial rolling windows used for each data set are shown in Table 5.1 along with the overall resulting evaluation periods. Note that the different lengths of the evaluation period and different times result from the used benchmark data as proposed by Lago et al. (2021) and are not explicitly chosen by us.

Dataset	Error Learning	Dependence Learning	Evaluation Period
EPEX-DE	05.01.2015-03.01.2016	06.10.2015-03.01.2016	04.01.2016-31.12.2017
PJM	29.12.2015-26.12.2016	28.09.2016-26.12.2016	27.12.2016-24.12.2018
EPEX-BE	05.01.2014-03.01.2015	06.10.2014-03.01.2015	04.01.2015-31.12.2016
EPEX-FR	05.01.2014-03.01.2015	06.10.2014-03.01.2015	04.01.2015-31.12.2016
NORD POLE	29.12.2015-26.12.2016	28.09.2016-26.12.2016	27.12.2016-24.12.2018

Table 5.1.: Sample choices in the empirical analysis. We show the initial backward-looking windows for the learning phases, which are then rolled over until the end of the evaluation periods (rightmost column).

5.3. Methods

The method we propose is inspired by Clark et al. (2004) and Schefzik et al. (2013) who consider multivariate dependencies in weather forecasting models, e.g., across locations or across different weather variables. Intuitively, we learn the univariate forecast distributions of 24 hours of the day (*Error Learning Phase*) and their dependence structure (*Dependence Learning Phase*) in history and use this information to construct multivariate forecasts in the future (*Forecasting Phase*). This algorithm can be used on top of any arbitrary univariate point forecasting model. We next detail the three steps of the method. An algorithmic description is given below for the Schaake-NP/P and in the appendix for Schaake-Raw. For ease of presentation, we omit time indices in this section; for example, we denote the price for hour h of day t by y_h instead of $y_{t,h}$.

Algorithm 1: Schaake shuffle with time series model (Schaake-NP/Schaake-P)

```
Result: Simulated multivariate forecast distribution for day-ahead electricity prices $\widehat{\mathbf{y}}_t^*$
Input: History of point forecasts $\hat{y}$ for day-ahead electricity prices
for each day in Error Learning Phase do
    for each hour h do
        Fit time series model to errors in learning data $\epsilon_h = y_h - \hat{y}_h$
        Standardize realized forecast error by $\hat{z}_h = \frac{\epsilon_h - \hat{\mu}_h}{\hat{\sigma}_h}$ with $\hat{\mu}_h$ and $\hat{\sigma}_h$ obtained from time series
          model and save last 90 of the resulting standardized error $\hat{z}_h$ in $\hat{F}_h$
    end
end
for each day in Dependence Learning Phase do
    for each hour h do
        if parametric margins==True then
            Calculate and save realized quantile $\hat{u}_h$ of $\hat{z}_h$ in specified distribution, e.g., $N(0,1)$, by
              $\hat{u}_h = \Phi(\hat{z}_h)$
        else
            Calculate and save realized quantile $\hat{u}_h$ of $\hat{z}_h$ in empirical distribution $\hat{F}_h$ by $\hat{u}_h = \hat{F}_h(z_h)$
        end
    end
end
for each day t in Forecasting Phase do
    for each hour h do
        Create $m$ univariate samples representing each $\frac{1}{m+1}^{th}$ quantile
        if parametric margins==True then
            use specified distribution, e.g., $N(0,1)$ and its inverse CDF $\Phi^{-1}$:
            for each ensemble member i do
                $\tilde{y}_h^i = (\hat{y}_h + \hat{\mu}_h) + \Phi^{-1}\{i/(m+1)\} \times \hat{\sigma}_h$
            end
        else
            use empirical CDF $\hat{F}_h$ and its inverse CDF $\hat{F}_h^{-1}$:
            for each ensemble member i do
                $\tilde{y}_h^i = (\hat{y}_h + \hat{\mu}_h) + \hat{F}_h^{-1}\{i/(m+1)\} \times \hat{\sigma}_h$
            end
        end
    end
    if parametric dependence==True then
        Fit specified parametric copula model $C$ to saved realized quantiles $\hat{u}$
        Sample $m$ times from parametric copula model $C$ and create rank matrix $\hat{\mathbf{R}}$
    else
        Derive empirical copula $\hat{C}$ and rank matrix $\hat{\mathbf{R}}$ out of saved, realized quantiles $\hat{u}$
    end
    Pair up univariate forecast ensembles according to rank matrix $\hat{\mathbf{R}}$ (Formula (5.6) to obtain final
      forecast ensemble $\widehat{\mathbf{y}}_t^*$
    Update Error Learning Phase and Dependence Learning Phase
end
```

5.3.1. Error Learning Phase

In the *Error Learning Phase*, we estimate the conditional distributions $G_{t,h}(x) := F_h(\frac{x-\mu_{t,h}}{\sigma_{t,h}})$ of the forecast errors $\epsilon_{t,h}$, where the indices represent the day (t) and hour (h). The conditional mean and variance of the error, $\mu_{t,h}$ and $\sigma^2_{t,h}$, can be estimated via a time series model. This is recommended if, e.g., the point prediction has high autocorrelation, seasonal effects, or heterogeneous variance patterns. The estimates $\hat{\mu}_{t,h}$ and $\hat{\sigma}_{t,h}$ then result from the forecast of the time series model. The required degree of complexity of the model partly depends on the point forecasting model that generated this sequence of forecast errors. In an idealized setup, one would not require a model for the expected forecast error $\mu_{t,h}$, as the latter is equal to zero if the forecast model is correctly specified (see e.g. Pesaran and Weale, 2006, Section 2.3). However, due to complex seasonality and persistence patterns, this goal is hard to achieve for practical energy price forecasts (see e.g. Maciejowska et al., 2021). Time variation in $\sigma^2_{t,h}$ reflects different market phases and seems unrelated to the accuracy of the point forecasting model.[4] If no time series model is applied, we set $G_{t,h}(x) = F_h(x)$, i.e., we assume that the distribution of $y_{t,h}|\mathcal{F}_{t-1}$ does not change over time. For background on time series analysis, we refer to the textbooks by Lütkepohl and Krätzig (2004) and Brockwell and Davis (2016). The time series specification used in our example can be found in Appendix C.

To complete the model specification, we must estimate or fix the distribution F_h of the standardized residuals $\hat{z}_{t,h} = \frac{\epsilon_{t,h}-\hat{\mu}_{t,h}}{\hat{\sigma}_{t,h}}$. We consider two versions: A parametric variant, where we suppose that $\hat{z}_{t,h}$ follows a standard normal distribution, and a non-parametric

[4] To see this point, consider a stylized example where the true conditional distribution of $y_{t,h}$ given $\mathcal{F}_{t-1}$ has mean five and variance $1 + \mathbf{1}(t > 100)$, i.e., the conditional variance equals one for time periods $t \leq 100$ and 2 for $t > 100$. In this example, the optimal mean forecast of $y_{t,h}|\mathcal{F}_{t-1}$ is given by 5, and the conditional variance of $\varepsilon_{t,h} = y_{t,h} - 5$ is heteroskedastic even though the mean forecast is optimal.

variant, where we estimate F_h via the empirical distribution function. We hence set

$$\hat{F}_h(z) = \frac{1}{n}\sum_{t=1}^{n} \mathbf{1}(\hat{z}_{t,h} \leq z), \tag{5.1}$$

where $z \in \mathbb{R}$, and n is the length of the error learning phase.

5.3.2. Dependence Learning Phase

The *Dependence Learning Phase* aims to learn the joint distribution of all $\epsilon_{t,h}$. Since we have already estimated the marginal distributions $G_{t,h}(x)$ of each hour h, we next estimate their dependence structure using copula techniques. To motivate our use of copulas, note that we need the joint distribution of the 24-day ahead errors $\epsilon_{t,h}$ of the point forecast model for $h = 1, \dots, 24$. This joint distribution is represented by the multivariate distribution function

$$G_{\epsilon_\mathbf{t}}(\theta) = P(\epsilon_{t,1} \leq \theta_1, \dots, \epsilon_{t,24} \leq \theta_{24}) \tag{5.2}$$

with $\epsilon_\mathbf{t} = (\epsilon_{t,1}, \dots, \epsilon_{t,24})$, where $\theta = (\theta_1, \dots, \theta_{24}) \in \mathbb{R}^{24}$ is a vector of threshold values. For example, for a vector of zero thresholds, $\theta_0 = (0, \dots, 0)$, $G_{\epsilon_\mathbf{t}}(\theta_0)$ yields the probability that all 24 forecast errors are negative (or zero). In general, the distribution function $G_{\epsilon_\mathbf{t}}$ is a complicated object that depends on both the marginal distribution of each element $\epsilon_{t,h}$ as well as the dependence structure across the 24 elements. However, copula methods allow us to specify the 24 elements' marginal distributions and their dependence separately, i.e., a copula C constructs a multivariate CDF from 24 marginal CDFs. A simple example of a copula is given in the case of independence, where $C\big\{G_{t,1}(\theta_1), \dots, G_{t,24}(\theta_{24})\big\} = \prod_{h=1}^{24} G_{t,h}(\theta_h)$, i.e., the copula function simply multiplies all of its arguments. In practice, independence of the hourly prices is clearly unrealistic and more flexible models of dependence are required. We can achieve this by choosing and estimating an appropriate copula function C. In the parametric case, we require additional parameters that determine the function C. For example, we could use a

Gaussian copula which is parameterized by correlation matrix $\boldsymbol{\rho}$ (see Chapter 2 for details on estimation and selected parametric copula models). The latter matrix has $24 \times 23/2 = 276$ unique elements so that the Gaussian copula allows for considerable flexibility in modeling dependence at the cost of parameter estimation uncertainty. In order to estimate the Gaussian copula, we use the rank correlation estimator (see, e.g., Genest and Favre 2007b).

In contrast to parametric copulas, non-parametric copulas do not assume a specific functional form for C as already introduced in Chapter 2.2. The use of non-parametric copulas in the context of multivariate forecast preprocessing was introduced by Clark et al. (2004), who named the reordering idea after Dr. J. Schaake, a member of the National Weather Service Office of Hydrologic Development. Schefzik et al. (2013) established the connection to empirical copulas. Our exposition of non-parametric copulas in this chapter loosely follows Schefzik et al. (2013, Section 3.3) for a more comprehensible presentation in the present application. More details on non-parametric copula estimation can be found in Chapter 2 and Chapter 3 of this thesis. Now consider a training sample covering m days of past point forecast errors for every hour h, i.e.,

$$\{(\varepsilon_{t,1}, \ldots, \varepsilon_{t,24}) : t = 1, \ldots, m\}. \tag{5.3}$$

We standardize these forecast errors by considering the corresponding quantile levels $\hat{u}_{t,h} = \hat{G}_{t,h}(\varepsilon_{t,h}) = \hat{F}_h(\hat{z}_t) \in [0, 1]$.[5] Assume for simplicity that there are no ties, i.e., the training sample contains $m \times 24$ unique values of $\hat{u}_{t,h}$. We denote the rank of $\hat{u}_{t,h}$ within $\hat{u}_{1,h}, \ldots, \hat{u}_{m,h}$ by $\mathrm{rk}(\hat{u}_{t,h})$. That is, for the day $\underline{t}$ at which the smallest quantile level is observed, we have $\mathrm{rk}(\hat{u}_{\underline{t},h}) = 1$. Similarly, the day $\bar{t}$ with the largest quantile level yields $\mathrm{rk}(\hat{u}_{\bar{t},h}) = m$. The empirical copula – an estimator of the unknown copula C, based on a

[5] In the case of parametric (standard normal) margins, we have $\hat{u}_{t,h} = \Phi(\hat{z}_{t,h})$, where Φ is the CDF of the standard normal distribution.

training sample of length m – is then given by

$$\hat{C}_m(i_1/m, \ldots, i_{24}/m) = \frac{1}{m} \sum_{t=1}^{m} \mathbf{1}\left\{\mathrm{rk}(\hat{u}_{t,1}) \leq i_1, \ldots, \mathrm{rk}(\hat{u}_{t,24}) \leq i_{24}\right\}, \tag{5.4}$$

for integers $0 \leq i_1, \ldots, i_{24} \leq m$. For example, suppose that $i_1 = \ldots = i_{24} = 10$. In this case, the above function yields the empirical frequency of the event that all 24 quantile levels are simultaneously smaller or equal than their tenth smallest observed value. Clearly, setting $i_1 = \ldots = i_{24} = 0$ yields a function value of zero, and setting $i_1 = \ldots = i_{24} = m$ yields a function value of one. Note how the here given representation of the empirical copula coincides with the empirical copula based on ranks (defined in Chapter 2), which is similar except the scaling factor of m and n, respectively.

5.3.3. Forecasting Phase

We next describe how to draw multivariate forecasts for the next day t. From the previous two steps, we have obtained estimates of the univariate distributions $G_{t,h}(x)$ and the copula C of standardized residuals $\hat{z}_{t,h}$. We now combine these two pieces to construct the desired multivariate distribution. Therefore, we first construct probabilistic *univariate* forecast distributions and then combine these distributions with the help of the learned copula representation.

Specifically, we begin by constructing a probabilistic *univariate* forecast distribution of size m, for each hour h in the next day, by

$$\tilde{y}^i_{t,h} = \underbrace{(\hat{y}_{t,h} + \hat{\mu}_{t,h})}_{\text{bias-corrected point forecast}} + \underbrace{\hat{F}_h^{-1}\{i/(m+1)\} \times \hat{\sigma}_{t,h}}_{\text{predicted quantiles of forecast error}} \tag{5.5}$$

for $i = 1, \ldots, m$, where $\hat{F}_h^{-1}$ is the inverse cumulative distribution function of the standardized residuals. In the case of parametric (Gaussian) marginal distributions, we have that $\hat{F}_h^{-1} = \Phi^{-1}$ is the quantile function of the standard normal CDF. Thus, Equation (5.5) generates a stylized sample by computing m equally spaced quantiles of the 24 univariate forecast distributions. In the 'raw-error' case without a time series model, we simply set $\hat{\mu}_{t,h} \equiv 0$ and $\hat{\sigma}_{t,h} \equiv 1$ in the formula above.

Up to now, the 24 ensembles represent the 24 marginal forecast densities for the hours but lack the correct dependence structure. Therefore, the resulting univariate ensembles per hour h, consisting of forecasts $\tilde{y}^i_{t,h},\ i \in \{1, \ldots, m\}$ have to be paired according to the learned copula to encode the desired dependence structure. A discrete representation of the copula is needed for this purpose. Such a representation is given by a suitable rank matrix $\hat{\mathbf{R}} = (r_{k,h})_{\ k\in\{1,\ldots,m\};\ h\in\{1,\ldots,24\}}$, containing the pairing of univariate ensembles. For the parametric copula approach, we derive the rank matrix from a random sample of size m from the fitted parametric copula model. I.e., we assign ranks to m random samples from the learned copula in each dimension starting with one for the smallest value. For the non-parametric approach, the rank matrix $\hat{\mathbf{R}}$ is calculated from the points of the empirical copula in the Dependence Learning Phase (see Equation 5.4).

From $\hat{\mathbf{R}}$ we derive the $m \times 24$ matrix $\widehat{\mathbf{y}}^*_t$ that contains our final multivariate forecast distribution reflecting the correct dependence structure. That is, each row of $\widehat{\mathbf{y}}^*_t$ represents 24 hourly prices that could plausibly be observed together. Specifically, to obtain $\widehat{\mathbf{y}}^*_t$ we sort the points $\tilde{y}^i_{t,h},\ i \in \{1, \ldots, m\}$ from Equation (5.5) such that the ranks in each row of $\widehat{\mathbf{y}}^*_t$ correspond to the k^{th} row of the rank matrix, i.e,

$$(\widehat{\mathbf{y}}^*_t)_k := \{\tilde{y}^{r_{k,1}}_{t,1}, \ldots, \tilde{y}^{r_{k,24}}_{t,24}\} \tag{5.6}$$

for $k \in \{1, \ldots, m\}$ and $(\cdot)_k$ denoting the k^{th} row of a matrix. Note that $\text{rk}(\tilde{y}^i_{t,h}) = i$ by construction, i.e., $\tilde{y}^i_{t,h}$ has rank i among the m univariate forecast draws for hour h (see Equation 5.5). We illustrate this approach in Example 5.3.1, which complements the meteorological example of Clark et al. (2004, Section 3b).

Example 5.3.1. *To illustrate the method (and in particular, the Forecasting Phase), we next consider a toy example based on a Dependence Learning Phase of length* $m = 7$

and four (instead of 24*) hours of the day. First, Table 5.2 illustrates the four univariate forecast distributions. Equation (5.5) together with* $m = 7$ *implies that we represent each distribution by a set of quantiles at levels* $(1/8, \ldots, 7/8)$*. The differences in the four univariate distributions (e.g., the lower median price forecast at 00:00 compared to 12:00) reflect intra-daily seasonality in energy prices. This seasonality is typically already reflected by the point forecasting model that we consider as an input to our method.*

Quantile level	12.5%	25%	37.5%	50%	62.5%	75%	87.5%
00:00	6.1	16.1	23.6	30.3	37.0	44.5	54.5
06:00	21.7	31.6	39.0	45.7	52.3	59.7	69.6
12:00	27.2	37.0	44.4	50.9	57.5	64.8	74.6
18:00	26.7	36.5	43.9	50.5	57.0	64.4	74.2

Table 5.2.: Univariate forecast distributions in the toy example, separately for four hours h (rows).

Table 5.3 illustrates an estimated rank matrix $\hat{\mathbf{R}}$*. For example, the table's first row shows the ranks for the first day of the Dependence Learning Phase. Prices were fairly low on that day: Among others, the price at 00:00 was the lowest recorded in the training sample compared to the prices at 00:00 for the other* $m - 1 = 6$ *days in the training sample. The table also shows strong positive dependence on the prices across hours, as reflected by the positive correlation between the ranks.*

Finally, Table 5.4 shows the multivariate forecast distribution that results from combining the univariate distributions in Table 5.2 with the rank structure from Table 5.3. For example, the first draw of the multivariate forecast distribution consists of the lowest quantile for the price at 00:00 (given by 6.1*), the second-lowest quantile for the price at 06:00 (given by* 31.6*), the lowest quantile for the price at 12:00 (given by* 27.2*), and the second-lowest quantile for the price at 18:00 (given by* 36.5*).*

	00:00	06:00	12:00	18:00
$t = 1$	1	2	1	2
$t = 2$	4	3	3	5
$t = 3$	5	4	7	7
$t = 4$	2	1	2	1
$t = 5$	3	5	5	6
$t = 6$	7	7	6	4
$t = 7$	6	6	4	3

Table 5.3.: Rank matrix $\hat{R}$ in the toy example.

Forecast Draw #	00:00	06:00	12:00	18:00
1	6.1	31.6	27.2	36.5
2	30.3	39.0	44.4	57.0
3	37.0	45.7	74.6	74.2
4	16.1	21.7	37.0	26.7
5	23.6	52.3	57.5	64.4
6	54.5	69.6	64.8	50.5
7	44.5	59.7	50.9	43.9

Table 5.4.: Final multivariate forecast distribution in the toy example.

Observe that the number of elements is the same in Tables 5.2, 5.3 and 5.4. At the same time, the interpretation differs across the three tables: While Table 5.2 represents quantile levels per time point in each row, rows in Table 5.3 represent ranks, and forecast draws in Table 5.4. This observation mirrors the simple yet clever construction behind the Schaake shuffle: Based on a dependence training sample of a given size (here, $m = 7$), it constructs a multivariate forecast sample of the same size. As a consequence, larger training samples yield larger and potentially more informative forecast distributions. On the other hand, short samples are quicker to adapt to possible structural breaks in dependence patters. In our empirical analysis below, we consider $m = 90$.

5.4. Results

5.4.1. Forecast Evaluation

For the evaluation of our forecast distributions, we follow the principle of Gneiting et al. (2007), according to which a probabilistic forecast should maximize sharpness subject to calibration. Calibration means that, ideally, each observation should resemble a random draw from the predictive distribution. On the other hand, sharpness requires that the predictive distribution be as concentrated as possible. *Proper scoring rules* (Gneiting and Katzfuss, 2014; Gneiting and Raftery, 2007) assess sharpness and calibration simultaneously by assigning numerical predictive performance measures. A scoring rule is a function $S(F, y) \to \mathbb{R}$, where F is the forecast distribution and y the observed outcome. We consider scoring rules in negative orientation, i.e., a smaller score value indicates a better forecast. A scoring rule is called proper if stating the true forecast distribution yields the best expected score. We consider the *Continuous Ranked Probability Score (CRPS)* to evaluate the marginal forecast distributions. Specifically, we evaluate the forecast distribution for each of the 24 horizons and then compute the average score. Furthermore, we use the *Energy Score (ES)* to evaluate the multivariate forecast distribution. We further apply the Diebold and Mariano (1995) test jointly to all hourly series (i.e., to the mean score across all 24 hours) to assess the statistical significance of differences in forecast performance (see Ziel and Weron, 2018). Finally, we check the calibration of multivariate forecasts graphically with *Average Rank Histograms* (Thorarinsdottir et al., 2016). We provide details on these evaluation methods in the appendix.

Note that we do not use the popular *Mean Absolute Error (MAE)* criterion because it refers to point forecasts (rather than forecast distributions). However, the CRPS criterion we consider reduces to the MAE in the special case where the forecast distribution collapses to a single point (see Gneiting and Katzfuss, 2014). For a discussion of evaluation metrics in probabilistic forecasting for electricity prices, we refer to Nowotarski and Weron (2018).

5.4.2. Electricity Price Forecasting

Our probabilistic forecasts are based on the rolling window scheme explained in Section 5.2.2. We examine several settings of the proposed method to investigate the impact of different modeling choices on forecast performance. Additionally, we compare all settings to their simplified (independence) counterparts ignoring the dependence structure.

Settings

We investigated several settings of the proposed method. All price forecasts used as inputs stem from the benchmark LEAR model by Lago et al. (2021) introduced in Section 5.2. The variants we consider differ in their specification of the margin (standardizing forecast errors via a time series model versus using raw forecast errors; using a non-parametric versus Gaussian marginal distribution) and the copula (non-parametric versus Gaussian). Thus, we are able to systematically identify drivers of the method's performance and to disentangle the effect of different modeling approaches in the Error and Dependence Learning Phase. For ease of presentation, we do not consider all possible configurations but focus on the settings listed in Table 5.5: The *Schaake-NP* variant uses non-parametric margins and a non-parametric copula, as well as an AR(1)-GARCH(1,1) time series model for error standardization. As detailed in the appendix, this model combines an autoregressive (AR) specification for the mean with a generalized autoregressive conditional heteroskedasticity (GARCH) specification for the variance. The *Schaake-P* variant uses Gaussian margins, a Gaussian copula, and the same AR(1)-GARCH(1,1) for error standardization. Last, the *Schaake-Raw* variant is based on raw forecast errors, i.e., without time series standardization, but is otherwise identical to the *Schaake-NP* variant. As a simple evaluation of copula modeling performance, we compare each variant with its independence counterpart that uses the same marginal distributions of forecast errors for each hour h, but assumes them to be independent across hours (*I-NP, I-P*, and *I-Raw*). The AR-GARCH calculation is done using the 'rugarch' package (Ghalanos, 2020) in R (R Core Team, 2013), while everything else is implemented in Python 3 (Van Rossum and Drake, 2009). Since the method proposed here is a post-processing approach, the runtime of the overall estimation is mainly driven by the underlying point forecasting model. Last, we provide ready-to-use computer code to reproduce our results in the appendix.

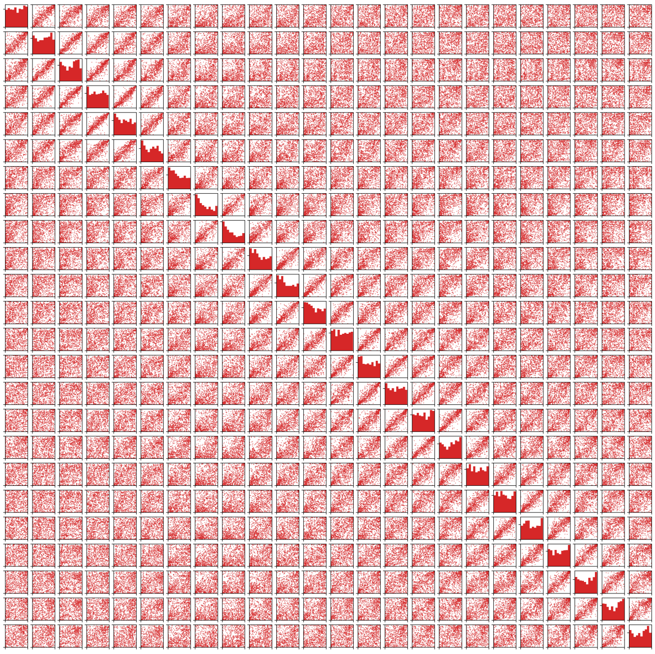
Setting	AR-GARCH	Dependence modeling	Marginal distributions
Schaake-NP	Yes	Empirical copula	Non-parametric
Schaake-P	Yes	Gaussian copula	Parametric
Schaake-Raw	No	Empirical copula	Non-parametric
I-NP	Yes	Independence	Non-parametric
I-P	Yes	Independence	Parametric
I-Raw	No	Independence	Non-parametric

Table 5.5.: Settings considered for electricity price forecasting. Note that the names and abbreviations are specifically chosen for this work for ease of presentation.

Assessment

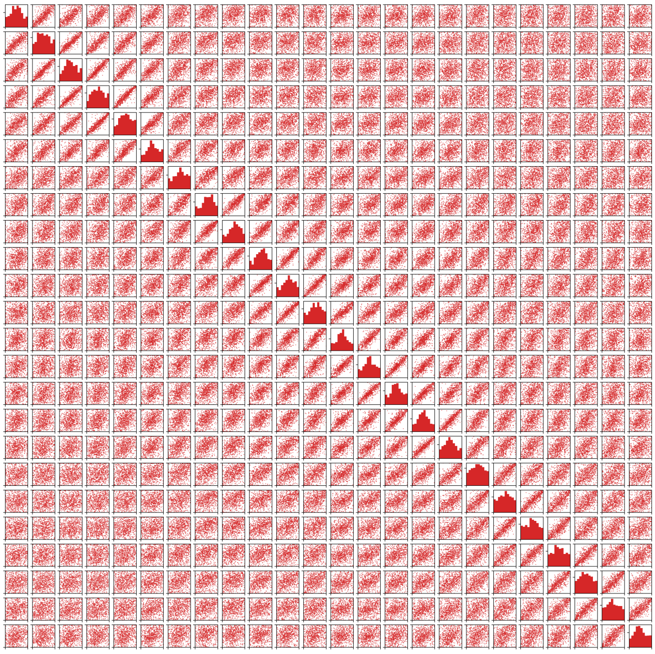
Figures 5.5 (a), 5.5 (b), and 5.5 (c) provide illustrative calibration checks for the EPEX-DE data set. The univariate verification rank histograms for all 24 hours are displayed on the diagonal and the bivariate scatterplots of realized quantiles on the off-diagonals. Each point in the figure corresponds to one day in the forecasting phase. Figure 5.5 (a) displays the result for Schaake-NP, while panel (b) displays the case of its parametric counterpart Schaake-P, and panel (c) refers to the case of raw errors (Schaake-Raw). The off-diagonal elements in Figures 5.5 (a) to (c) indicate a decreasing dependence for hours that are further apart, which seems plausible. The verification rank histograms (diagonal elements) suggest good calibration of the univariate forecast distribution for the non-parametric case (Figure 5.5 a), as indicated by uniform histograms. Univariate calibration seems to be slightly worse in the parametric case (diagonal elements of Figure 5.5 (b), where most of the histograms are hump-shaped. The margins of the non-parametric raw-error approach in Figure 5.5 (c) also seem to be well-calibrated. Thus, the results suggest better calibration for the two non-parametric versions. Furthermore, satisfactory calibration of the raw error approach suggests that error post-processing via a time series model is not of much importance in this case study.

Figure 5.6 shows the Average Rank Histograms for various forecast distributions. The non-parametric Schaake methods in panels (a) and (c) display slight signs of miscalibration (i.e., non-uniform histograms) which however, seem rather unsystematic. The histogram for the parametric case in panel (b) is hump-shaped, which indicates under-dispersion and is in line with our observation in Figure 5.5 (b). The three naïve independence counterparts in panels (d)-(f) clearly lack calibration, as indicated by

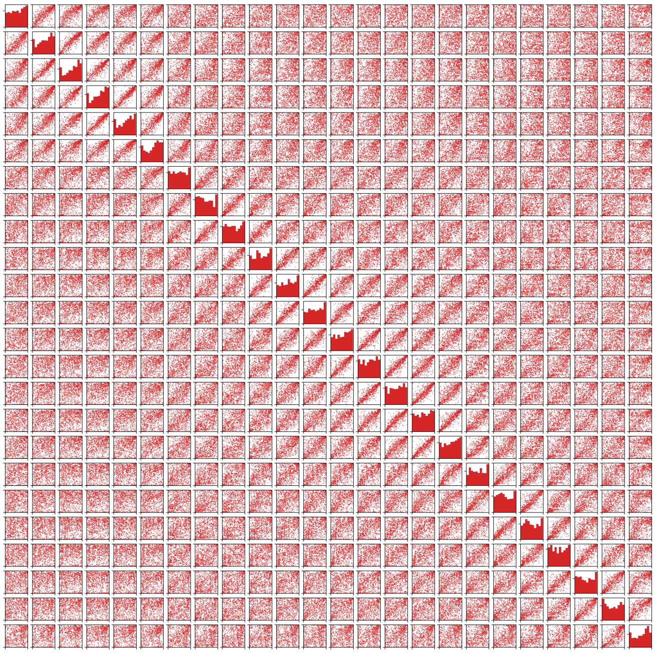


(a) Schaake-NP

Figure 5.5.: Verification rank histograms for the EPEX-DE data set. Diagonals: Univariate rank verification histograms. Off-diagonals: Bivariate scatterplots of realized quantiles.



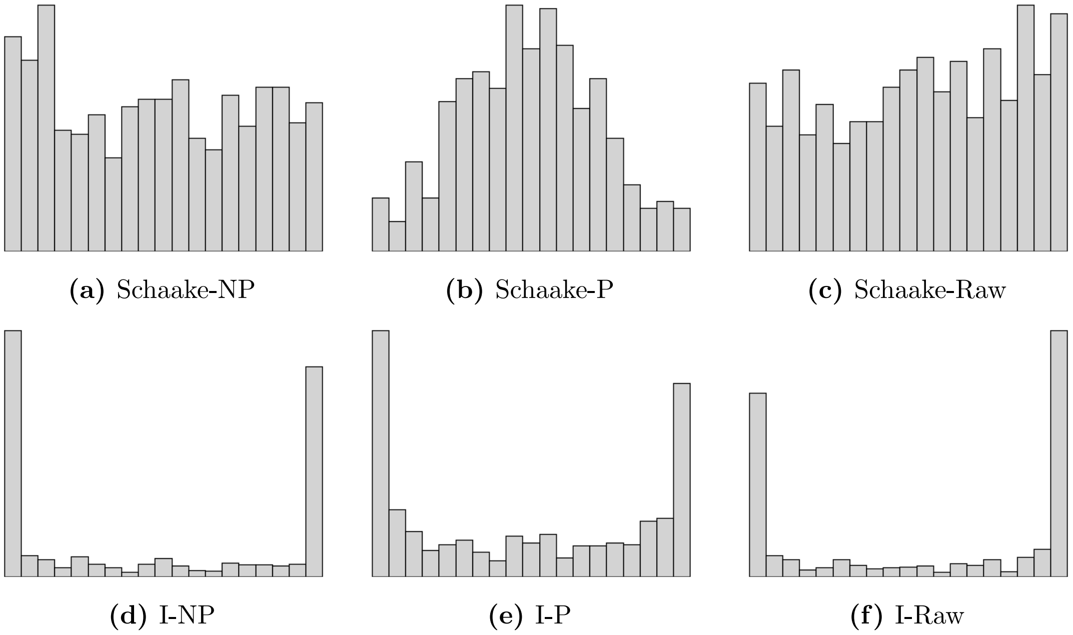
(a) Schaake-P



(c) Schaake-Raw

Figure 5.5, continued.

distinct U-shaped patterns of the Average Rank Histograms.



(a) Schaake-NP **(b)** Schaake-P **(c)** Schaake-Raw

(d) I-NP **(e)** I-P **(f)** I-Raw

Figure 5.6.: Average Rank Histograms for the EPEX-DE dataset.

Finally, Table 5.6 summarises the Energy Score and CRPS for all settings and the corresponding baselines in the five markets. We observe the same values for the copula based approaches and their independence counterparts for the CRPS as they share the same marginal distributions. In terms of the Energy Score, the three copula based approaches consistently outperform their independence counterparts. Furthermore, we note a better performance for both non-parametric versions (Schaake-NP/Schaake-Raw) than for the parametric error distributions (Schaake-P). This result is in line with the lack of calibration of the Schaake-P method documented earlier. Remarkably, the performance of the raw error approach (Schaake-Raw) is not systematically worse than that of the time series based non-parametric variant (Schaake-NP). For the PJM, and FR data sets, both variants perform similarly, and differences in score performance (Diebold-Mariano tests) are typically insignificant at the 5% level. For the BE and DE data set, which are potentially more challenging to predict (Lago et al., 2021, Section 3.4 and 3.6), Schaake-Raw outperforms Schaake-NP. This may be due to increased robustness of Schaake-Raw in an unstable data environment. On the contrary, in the Nord Pole data, Schaake-NP outperforms Schaake-Raw. Overall, the competitive performance of the raw error approach indicates that there is little need of systematic error post-processing. This suggests that the forecast models provided by Lago et al. (2021) yield a good fit to the data, leaving little unmodeled heterogeneity in the models' forecast errors. The need for error post-processing may be more pronounced in a setup with less sophisticated forecasting models. We consider such a case in the context of load forecasting below.

Setting	Energy Score	CRPS	p-values	
			(a)	(b)
EPEX-DE				
Schaake-NP	18.769	3.138		
Schaake-P	19.841	3.421	<0.001	<0.001
Schaake-Raw	18.216	3.035	<0.001	<0.001
I-NP	19.227	3.138	<0.001	
I-P	20.075	3.421	<0.001	<0.001
I-Raw	18.663	3.035	0.333	<0.001
Pennsylvania-New Jersey-Maryland (PJM)				
Schaake-NP	14.594	2.394		
Schaake-P	15.318	2.581	<0.001	<0.001
Schaake-Raw	14.791	2.436	0.162	0.047
I-NP	15.133	2.394	<0.001	
I-P	15.687	2.581	<0.001	<0.001
I-Raw	15.426	2.436	<0.001	0.047
EPEX-BE				
Schaake-NP	35.13	5.503		
Schaake-P	38.64	6.134	<0.001	<0.001
Schaake-Raw	33.488	5.311	0.012	0.001
I-NP	36.423	5.503	<0.001	
I-P	38.868	6.134	<0.001	<0.001
I-Raw	34.642	5.311	0.444	0.001
EPEX-FR				
Schaake-NP	22.553	3.307		
Schaake-P	23.578	3.515	0.001	<0.001
Schaake-Raw	21.203	3.243	0.097	0.119
I-NP	23.088	3.307	<0.001	
I-P	23.935	3.515	<0.001	<0.001
I-Raw	21.633	3.243	0.259	0.119
NORD POLE				
Schaake-NP	9.729	1.571		
Schaake-P	10.437	1.72	<0.001	<0.001
Schaake-Raw	9.904	1.626	0.007	<0.001
I-NP	10.179	1.571	<0.001	
I-P	10.657	1.72	<0.001	<0.001
I-Raw	10.189	1.626	<0.001	<0.001

Table 5.6.: Energy Score and CRPS for several variants of proposed method and five benchmark data sets. (a) p-value of Diebold-Mariano tests for equal Energy Score, compared to Schaake-NP. (b) p-value of Diebold-Mariano tests for equal CRPS, compared to Schaake-NP. Blank cells indicate that scores are equal by construction, so that the test is undefined.

5.4.3. Load Profile Price Prediction Intervals

We illustrate the importance of the correct dependence structure in forecasting with the example of the Standard Load Profile (SLP) G0 from the Bundesverband der Energie- und Wasserwirtschaft e.V. (2021). To do so, we compare daily price forecasts for the SLP G0 from the Schaake-NP and Schaake-Raw variants with their independence counterparts (I-NP/I-Raw) over the time period from 04/01/2016 to 31/12/2017. We average the quarter-hourly SLP across seasons and aggregate it to full hours. The true price for the SLP of each day is then calculated with realized prices from the EPEX-DE dataset. Further, we added a *Quantile Regression (QR)* which uses the 24 univariate point predictions of the day as explanatory variables with a small L_1-regularization constant of 0.1; a *Kernel Density Estimation (KDE)* over the last 90 price realizations of the full daily profile; and an *AR(1)-GARCH(1,1)* time series model also fitted to the last 90 price realizations of the full daily profile as additional benchmarks (in line with the choice for the Dependence Learning Phase). The L_1-regularization in the QR is necessary to avoid exploding regression coefficients and, hence, unrealistic interval forecasting results. See Li and Zhu (2008) and Koenker et al. (1994) for theoretical introductions to regularization in quantile regression. Finally, we evaluate how often this price realizes within the 93.33% quantile of the respective forecast distribution (*Forecast Interval Coverage Probability, FICP)* and the normalized width of the interval *(Forecast Interval Normalized Average Width, FINAW)*. See the appendix for details on these measures.

As expected, the positive dependence between consecutive hours leads to wider prediction intervals for the SLP compared to the approaches not considering the dependence. Figure 5.7 illustrates this aspect for a three-month subsample of the data (for better visibility), whereas Figure C.1 in the appendix presents the entire sample period. This is also reflected in the FINAW values reported in Table 5.7. We observe that I-Raw results in the shortest intervals (smallest FINAW), followed by I-NP; however, both methods fail to achieve the desired level of coverage. Notably, only Schaake-Raw and Schaake-NP combine good coverage results with small values of FINAW. In particular, 91.35% of the

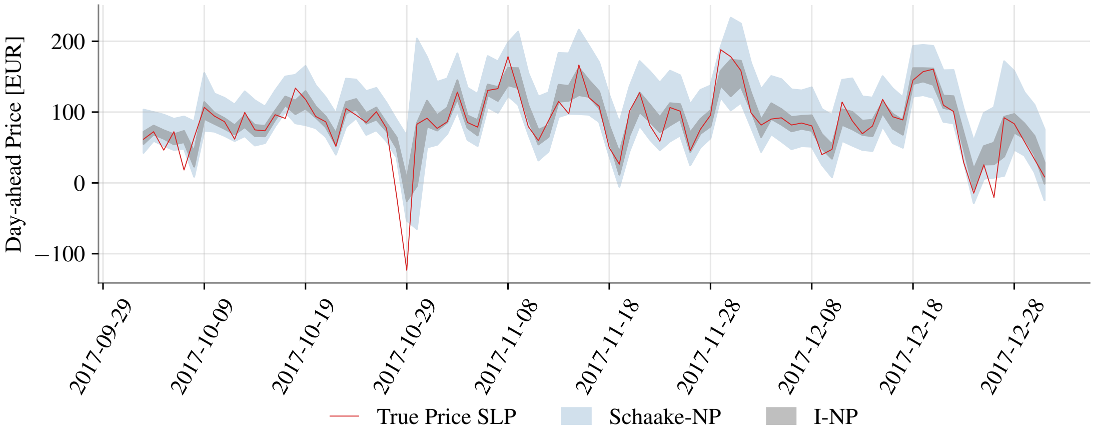


Figure 5.7.: Prediction intervals and realized prices for SLP G0 from 3 October 2017 to 31 December 2017 (last 90 days of 2017).

realized prices are within the projected 93.33% prediction interval of the Schaake-NP model. For the Schaake-Raw model, the corresponding share is 91.49%. Hence both models achieve coverage rates that are very close to nominal coverage, indicating good calibration. By contrast, the corresponding coverage rates of 53.71% and 56.19% for the independence assumption show that economic uncertainty cannot be well approximated without considering the dependence structure. As a simple alternative method, one could compute a prediction interval from past price realizations. In the present case, using this 'direct' method with a sample of 90 past prices also yields a satisfactory coverage rate of 88.59% but much wider intervals, with a FINAW value of 0.2533. Similarly, the reported KDE and AR(1)-GARCH(1,1) methods in Table 5.7 provide good coverage but exhibit wider forecasting intervals. In contrast, QR results in much smaller forecasting intervals, but does not provide sufficient coverage. Importantly, note that the latter three methods require an aggregated time series which is specific to the particular load profile considered here. By contrast, the Schaake methods can be adapted to any given load profile as they yield forecasts on an hourly resolution.

	FICP	FINAW
Schaake-NP	91.35%	0.1122
I-NP	53.71%	0.0372
Schaake-Raw	91.49%	0.1068
I-Raw	56.19%	0.0333
QR	80.77%	0.0697
KDE	90.93%	0.2789
AR(1)-GARCH(1,1)	90.11%	0.1907

Table 5.7.: Forecast Interval Coverage Probability *FICP* (nominal level: 93.33%) and Forecast Interval Normalized Average Width *FINAW* for realized SLP prices over the time period from 04/01/2016 to 31/12/2017.

5.4.4. Forecasting Intraday Load Data

We finally give a short example of other operational scenarios where our proposed method can be applied. Instead of hourly prices, this example uses intra-day load data of the entso-e transparency platform for Germany from 2016 to 2020.[6] We average the day-ahead quarter-hourly load forecasts ([6.1.B]) and actual loads ([6.1.A]) over each hour, hence resulting in 24 hourly forecast error observations, in line with the setup of the price data considered earlier. We choose this data set to substantiate the use of time series models as an optional building block within the proposed method. In contrast to the point forecast errors for the price data above, point forecast errors for the load data contain a time series structure with a marked seasonal component (see, e.g., the analysis in Maciejowska et al. (2021)). Without claiming to be optimal, we tackle this stylized fact by using a *Seasonal Autoregressive S-AR(1)(1,7)* time series with one AR parameter and one seasonal AR parameter with a lag of one week, i.e., 7 days for each hourly error time series (Schaake-NP). See Hyndman and Athanasopoulos 2021, Section 9.9 for an introduction to seasonal autoregressive models. We display the results for the Energy Score and CRPS for the years 2019 and 2020 in Table 5.8.

[6] `https://transparency.entsoe.eu/dashboard/show`

Setting	Energy Score	CRPS	p-values	
			(a)	(b)
Schaake-NP	5854.619	1047.883		
Schaake-Raw	13412.822	2614.415	< 0.001	< 0.001
I-NP	6082.442	1047.883	< 0.001	
I-Raw	13734.781	2614.415	< 0.001	< 0.001

Table 5.8.: Energy-Score and CRPS for forecasting German load data from 01 January 2019 to 30 December 2020. (a) p-value of Diebold-Mariano tests for equal Energy Score, compared to Schaake-NP. (b) p-value of Diebold-Mariano tests for equal CRPS, compared to Schaake-NP. Blank cells indicate that scores are equal by construction, so that the test is undefined.

The table indicates that raw error approaches are clearly inferior to S-AR based error post-processing. Thus, in contrast to the case of price data considered above, there appears to be a clear need for error post-processing in this example. We conjecture that this result is driven by the lower sophistication of the (unknown) load forecasting model, as compared to the forecasting models provided by Lago et al. (2021) that we considered for the energy price data.

5.5. Conclusion

The work in this chapter proposes a post-processing method to create multivariate forecast distributions for day-ahead electricity prices out of any point forecasting model. The method is motivated by two main aspects: First, many sophisticated point forecasting models have been introduced in the literature, so that we may take the availability of a ‘good’ point forecasting model as given. Second, many economic decisions require probabilistic multivariate forecasts, which are hardly available at present.

Our method exploits several time series of univariate point forecast errors and creates a multivariate forecast distribution that inherits their dependence structure. In a case study on energy price forecasting based on benchmark models of Lago et al. (2021), a simple raw-error variant of the method performs well. This result indicates that the point forecast errors contain little exploitable structure so using a time series model for these errors is not necessary. In an additional case study on load forecasting, the point forecast errors contain pronounced seasonal patterns, and removing these patterns via a time series model leads to clear gains in forecasting performance. Throughout our empirical analysis, simple non-parametric techniques (for estimating the marginal distribution and the copula) outperform parametric ones based on normality assumptions.

6. Sampling from the Latent Space in Autoencoders: A Simple way Towards Generative Models?

This chapter is based on joint work with Maximilian Coblenz and Oliver Grothe. It investigates methodologies to model the latent space of an autoencoder turning it into a generative model.

6.1. Introduction

Generating realistic sample points of various data formats has been of growing interest in recent years. Thus, new algorithms such as *Autoencoders (AEs)* and *Generative Adversarial Networks (GANs)* Goodfellow et al. (2014) have emerged. GANs use a discriminant model, penalizing the creation of unrealistic data from a generator and learning from this feedback. On the other hand, AEs try to find a low-dimensional representation of the high-dimensional input data and reconstruct from it the original data. To turn an AE into a generative model, the latent low-dimensional distribution is modeled, samples are drawn, and thereupon new data points in the original space are constructed with the decoder. Based on that, *Variational Autoencoders (VAEs)* have evolved, optimizing for a Gaussian distribution in the latent space Kingma and Welling (2014). Adversarial autoencoders (AAEs) utilize elements of both types of generative models, where a discriminant model penalizes the distance of the encoded data from a prior (Gaussian) distribution (Makhzani et al., 2016). However, such strong (and simplifying) distributional assumptions as in the VAE or AAE can have a negative impact on performance, leading to a rich literature coping with the challenge of reducing the gap between approximate and true posterior distributions (e.g., Rezende and Mohamed 2015; Tomczak and Welling 2018; Kingma et al. 2016; Gregor et al. 2015; Cremer et al. 2018; Marino et al. 2018; Takahashi et al. 2019). We argue that imposing restrictions on the distribution should be avoided and that more flexible approaches for modeling the latent space seem beneficial.

Recently, Tagasovska et al. (2019) presented the *Vine Copula Autoencoder (VCAE)* to overcome the mentioned problems. Their approach comprises two building blocks, an autoencoder and a vine copula which models the dependence structure in latent space. By that, they were able to create realistic, new images with samples from the fitted vine copula model in the latent space. In this work, we want to elaborate on this idea and compare various methods to model the latent space of an autoencoder to turn it into a generative model. To this end, we analyze, amongst others, the usage of *Gaussian mixture models (GMM)* as done by Ghosh et al. (2020), the vine copula approach by Tagasovska et al. (2019), and simple multivariate *Kernel Density Estimates.* Additionally, we introduce a new, non-parametric copula approach, the *Empirical Beta Copula Autoencoder (EBCAE)* based on a special form of the already known empirical copula. To assess the ability to turn a standard autoencoder into a powerful generative model, we inspect resulting images, check the models for their ability to generalize and compare additional features. We also check if these methods may be a simple alternative to more complex models, such as normalization flows (see Rezende and Mohamed 2015). More specifically, we use the well-known Real NVP (Dinh et al., 2017) as a benchmark but do not elaborate on these in detail. Note that in contrast to other methods (e.g., Oring et al. 2021 or Berthelot et al. 2019), the overall approach does not restrict or change the training of the autoencoder in any form, enabling it to find the best low-dimensional representation of the data.

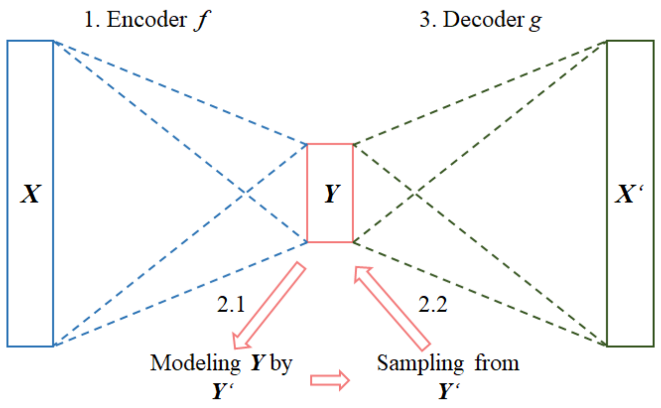


Figure 6.1.: Function scheme of simple generative autoencoders. 1. An encoder f encodes the data $\boldsymbol{X}$ to a low dimensional representation $\boldsymbol{Y}$. 2.1 $\boldsymbol{Y}$ is modeled by $\boldsymbol{Y}'$, 2.2 Generate new synthetic samples of the latent space by sampling from $\boldsymbol{Y}'$. 3. Decode the new samples with the decoder g.

All models considered in this work are constructed in three steps, visualized in Figure 6.1. First, an autoencoder, consisting of an encoder f and a decoder g, is trained to find a low-dimensional representation of the data $\boldsymbol{X}$. Second, the data in the latent space $\boldsymbol{Y}$ is used to learn the best fitting representation $\boldsymbol{Y}'$ of it. This is where the examined models differ from each other by using different methods to model the latent space. Finally, we sample from the learned representation of the latent space and feed the samples into the decoder part of the autoencoder, creating new synthetic data samples.

Generative models are a vivid part of the machine learning literature. For example, new GAN developments Varshney et al. (2021); Karras et al. (2021); Lee et al. (2021); Hudson and Zitnick (2021), developments in the field of autoencoders, Larsen et al. (2016); Yoon et al. (2021); Zhang et al. (2020); Shen et al. (2020) or developments in variational autoencoders Sohn et al. (2015); Havtorn et al. (2021); Masrani et al. (2019); Xu et al. (2019) are emerging. We again want to emphasize that for the models we consider, no prior is needed, nor the optimization approach is changed, i.e., the latent space is modeled after the training of the autoencoder (post-doc). Thus, the presented approach could be transferred to other, more sophisticated, state-of-the-art autoencoders, as hinted in Ghosh et al. (2020). The general idea of creating new data by sampling in the latent space of a generative model has already been used by, e.g., Tagasovska et al. (2019); Dai and Wipf (2019); Brehmer and Cranmer (2020) or Ghosh et al. (2020), but to the best of our knowledge, no analysis and comparison of such methods have been made so far. Closely related, more and more researchers specifically address the latent space of generative models Mishne et al. (2019); Fajtl et al. (2020); Moor et al. (2020); Oring et al. (2021); Hofert et al. (2021) in their work.

This work does not propose a new 'black-box algorithm' for generating data (although we present the new EBCAE) but analyses challenges and possible answers on how autoencoders can be turned into generative models by using well-understood tools of data modeling. We show that this idea generally works with various approaches but that it is hard to find a trade-off between out-of-bound sampling and creating new pictures. We debate further properties of the used methods as targeted sampling and synthesizing images. Our conclusion is intended to point out relevant aspects to the user and discusses the advantages and disadvantages of the models examined.

The remainder of the chapter is structured as follows. Section 6.2 introduces various methods for modeling the latent space. Besides traditional approaches, copula-based methods are introduced. Section 6.3 describes the implementation, evaluation, and results of the experiments carried out. In Section 6.4 we discuss the results and conclude the chapter.

6.2. Modeling the Latent Space

In this section, we want to introduce and reflect on different methods to model the latent space in an autoencoder. All methods aim to fit the low-dimensional data $\boldsymbol{Y}$ as best as possible to be able to create new sample points in the latent space, which leads to new realistic images after passing the decoder. We first recap more 'traditional' statistical tools, followed by copulas as an intuitive and flexible tool for modeling high-dimensional data. We briefly explain how each approach can be used to model data in the latent space and how to obtain samples thereof. Note that we do not introduce our benchmark models, namely the standard plain vanilla *VAE* and the *Real NVP*, and refer to the original papers instead (Kingma and Welling, 2014; Dinh et al., 2017).

6.2.1. Traditional Modeling Methods

We classify the *multivariate Gaussian distribution*, a *Kernel Density Estimation (KDE)*, and a *Gaussian Mixture Model (GMM)* as traditional modeling methods and give a rather short treatment of each below. They are well known and can be studied in various statistics textbooks such as Hastie et al. (2001) or Bishop (2006).

Multivariate Gaussian

The probably simplest method is to assume the data in the latent space to follow a multivariate Gaussian distribution. Thus, we estimate the covariance matrix $\hat{\Sigma}$ and mean vector $\hat{\mu}$ of $\boldsymbol{Y}$. In the second step, we draw samples thereof and pass them through the decoder to generate new images.

GMM

The *Gaussian Mixture Model (GMM)* aims to model the density of the data by mixing M multivariate Gaussian distributions. Thus, the resulting density of the Gaussian mixture model has the form

$$f(y) = \sum_{m=1}^{M} \alpha_m \phi(y; \mu_m, \Sigma_m) \qquad (6.1)$$

where α_m denotes the mixing parameter and ϕ the density of the multivariate normal distribution with mean vector μ_m and covariance matrix Σ_m. The model is usually fit by maximum likelihood using the EM algorithm. By combining several Gaussian distributions, it is more flexible than estimating only one Gaussian distribution as above. A GMM can be seen as some kind of kernel method (Hastie et al., 2001), having a rather wide kernel. In the extreme case, i.e., where m equals the number of points the density is estimated on, a Gaussian distribution with zero variance is centered over each point. Kernel density estimation is introduced in the following.

KDE

Kernel Density Estimation is a well-known non-parametric tool for density estimation. Put simply, a KDE places a density around each of the N data points y_i, $i \in \{1, \dots, N\}$ of $\boldsymbol{Y}$. The estimated density at point y_0 is constructed by

$$f(y_0) = \frac{1}{N\lambda} \sum_{i=1}^{N} K_\lambda(y_0, y_i) \tag{6.2}$$

with bandwidth λ and used kernel K. The kernel density estimation can be performed in univariate data as well as in multivariate data. Note that the choice of bandwidth and kernel can affect the resulting estimated density. In this work, we rely on the most commonly used kernel, the Gaussian Kernel, and a bandwidth fitted via *Silverman's rule of thumb* (Silverman, 1986) for the univariate KDEs, while we use a grid search with 10-fold cross-validation in the multivariate case.

We use kernel density estimation in multiple fashions. First, we use a multivariate KDE to model the density of the data in the latent space itself. In the case of a Gaussian kernel, the density at point y_0 can be estimated

$$f(y_0) = \frac{1}{N\sqrt{\Sigma}2\pi} \sum_{i=1}^{N} e^{-1/2(y_0-y_i)'\Sigma^{-1}(y_0-y_i)} \tag{6.3}$$

where Σ represents the covariance matrix of the kernel, i.e., the matrix of bandwidths, and N the total number of observations in $\boldsymbol{Y}$ again. Second, we ignore the dependence structure between margins and estimate the univariate densities of each dimension in the latent space by a KDE. In this way, we are able to find out whether explicitly modeling the dependence structure is necessary or not. We call that approach the *Independent modeling approach.* Last, we use univariate KDEs for modeling the marginal distributions of each dimension in the latent space and use them in the copula models described below.

6.2.2. Copula Based Models

In the following, we introduce two copula-based methods to model the latent space of the autoencoder: the *vine copula* and the *empirical beta copula* approach. *Copulas* have been subject to an increasing interest in the *Machine Learning* community over the last decades, see, e.g., Dimitriev and Zhou (2021); Janke et al. (2021); Messoudi et al. (2021); Ma et al. (2021); Letizia and Tonello (2020); Liu (2019); Kulkarni et al. (2018); Tran et al. (2015). Sklar's theorem, introduced in Chapter 2 of this work, allows us to construct multivariate distributions with the same dependence structure but different margins or multivariate distributions with the same margins but different couplings/pairings, i.e., dependence structures. Thus, using copula techniques seems natural to model high-dimensional distributions like the latent space of an autoencoder. The simplest estimator for the copula of data at hand is given by the empirical copula (see Chapter 2), which one of the methods we propose is based on. Next, we focus on two copula-based methods for modeling the latent space $\boldsymbol{Y}$ of an autoencoder.

Vine Copula Autoencoder

Although a variety of two-dimensional copula models exist, the amount of multivariate (parametric) copula models is somewhat limited. *Vine copulas* offer a solution to this problem and decompose the multivariate density as a cascade of bivariate building blocks organized in a hierarchical structure. This decomposition is not unique, and it influences the estimation procedure of the model. Here, we use *regular-vine (r-vine)* models Czado (2019); Joe (2014). A r-vine is built of a sequence of linked trees $T_i = (V_i, E_i)$, with nodes V_i and edges E_i for $i = 1, \dots, d-1$. A $d-$dimensional vine tree structure $V = (T_1, ..., T_{d-1})$ is a sequence of $T - 1$ trees if (see Czado 2019):

1. Each tree $T_j = (N_i, E_i)$ is connected, i.e. for all nodes $a, b \in T_i, i = 1, ..., d-1$, there exists a path $n_1, ..., n_k \subset N_j$ with $a = n_1, b = n_k$.
2. T_1 is a tree with node set $N_1 = \{1, ..., d\}$ and edge set E_1.
3. For $i \geq 2$, T_j is a tree with node set $N_i = E_{i-1}$ and edge set E_i .
4. For $i = 2, ..., d-1$ and $\{a, b\} \in E_i$ it must hold that $|a \cap b| = 1$.

An example of a five-dimensional vine tree structure is given below in Figure 6.2. Note that the structure has to be estimated and multiple structures are possible. See Czado (2019); Joe (2014); Bedford and Cooke (2002) for details on vine copula estimation.

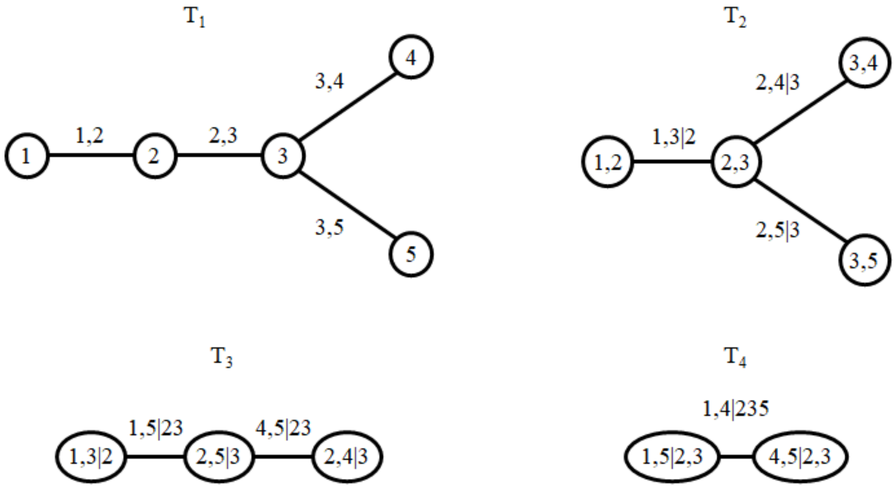


Figure 6.2.: Example of a vine copula tree structure $T_1 - T_4$ for five dimensions.

A d-dimensional copula density can be written as the product of its bivariate building blocks:

$$c(u_1, \ldots, u_d) = \prod_{i=1}^{d-1} \prod_{e \in E_i} c_{a_e b_e; D_e}(u_{a_e|D_e}, u_{b_e|D_e}) \tag{6.4}$$

with conditioning set D_e and conditional probabilities, e.g., $u_{a_e|D_e} = \mathbb{P}(U_{a_e} \leq u_{a_e}|D_e)$. For each copula, encoding the dependence of two conditional variables, any bivariate copula model including non-parametric modeling approaches (as done by Tagasovska et al. 2019) can be chosen. However, the construction and estimation of vine copulas is rather complicated. Hence, assuming independence for seemingly unimportant building blocks, so-called truncation is regularly applied. Because of this, truncated vine copula models do not capture the complete dependence structure of the data, and their usage is not underpinned by asymptotic theory. We refer to Czado (2019); Czado and Nagler (2022) and Aas (2016) for reviews of vine copula models.

Empirical Beta Copula Autoencoder

The *empirical beta copula* (Segers et al., 2017) avoids choosing a single, parametric multivariate copula model due to its non-parametric nature. Further, it offers an easy way to model the full, non-truncated multivariate distribution based on the univariate ranks of the joint distribution and, thus, seems to be a reasonable choice to model the latent space. The empirical beta copula is closely related to the empirical copula and is a crucial element of the Empirical-Beta-Copula Autoencoder. It is solely based on the ranks $r_{i,j}$ of the original data $\boldsymbol{Y}$ and can be interpreted as a continuous counterpart of the empirical copula introduced in Chapter 2. It is defined by

$$C^{\beta}(\mathbf{u}) = \frac{1}{n} \sum_{i=1}^{n} \prod_{j=1}^{d} F_{n,r_{i,j}}(u_j) \tag{6.5}$$

for $\mathbf{u} = (u_1, \ldots, u_d) \in [0,1]^d$, where

$$\begin{aligned} F_{n,r_{i,j}}(u_j) &= P(U_{(r_{i,j})} \leq u_j) && (6.6) \\ &= \sum_{p=r_{i,j}}^{n} \binom{n}{p} u_j^p (1-u_j)^{(n-p)} && (6.7) \end{aligned}$$

is the cumulative distribution function of a *beta distribution*, i.e., $\mathbb{B}(r_{i,j}, r_{i,j} + n - 1)$. As $r_{i,j}$ is the rank of the i^{th} element in dimension j, $U_{(r_{i,j})}$ represents the ${r_{i,j}}^{\text{th}}$ order statistic of n i.i.d. uniformly distributed random variables on $[0,1]$. For example, if the rank of the i^{th} element in dimension j is 5, $U_{(r_{i,j})} = U_{(5)}$ denotes the 5^{th} order statistic on n i.i.d. uniformly distributed random variables. Example 6.2.1 illustrates this computation.

Example 6.2.1. *Consider six data points, each having three dimensions. The rank matrix is now computed by replacing the values of each data point with its corresponding rank per dimension (see Table 6.1 and Table 6.2).*

Table 6.1.: Example data with six data points (rows) spanning over Three dimensions (columns).

i, j		
0.1	0.4	0.2
0.5	0.5	0.3
0.2	0.1	0.5
0.7	0.3	0.1
0.9	0.8	0.4
0.8	0.6	0.6

→

Table 6.2.: Resulting rank matrix **R**. Remember, ranks are computed separately per dimension.

$rk_{i,j}$		
1	3	2
3	4	3
2	1	5
4	2	1
6	6	4
4	5	6

The ranks are used in the following to parameterize the empirical beta copula.

The intuition behind the empirical beta copula is as follows: Recall that the marginal distributions of a copula are uniformly distributed on $[0,1]$ and, hence, the k^{th} smallest value of scaled ranks $r_{i,j}/n$ corresponds to the k^{th} order statistic $U_{(k)}$. Such order statistics are known to follow a *beta distribution* $\mathbb{B}(k, k+n-1)$ (David and Nagaraja, 2003). Consequently, the mathematical idea of the empirical beta copula is to replace each indicator function of the empirical copula with the cumulative distribution function of the corresponding rank $r_{i,j}$.

Synthetic samples in the latent space y' are created by reversing the modeling path. First, m random samples from the copula model $\mathbf{u}_i = (u_{i1}, \dots, u_{id})$ for $i \in \{1, \dots, m\}$ are drawn. Then, the copula samples are transformed back to the natural scale of the data by the inverse probability integral transform of the marginal distributions, i.e., $y'_{ij} = \hat{F}_j(u_{ij})$, where $\hat{F}_j$ is the estimated marginal distribution and u_j the j^{th} element of the copula sample for $j = 1, \dots, d$. Algorithm 2 below summarizes the procedure.

Theorem 6.2.1 gives the asymptotic behavior of the empirical beta copula.

Theorem 6.2.1 (Asymptotics of the empirical beta copula)**.** *Let the copula C have continuous first-order partial derivatives $\dot{C}_j = \delta C(\mathbf{u})/\delta u_j$ for each $j \in \{1, \dots, d\}$ on the set $I_j = \{\mathbf{u} \in [0,1]^d : 0 < u_j < 1\}$. The corresponding empirical copula is denoted as $\mathbb{C}_n$, with empirical copula process $\mathbb{G}_n = \sqrt{n}\Big(\mathbb{C}_n(\mathbf{u}) - C(\mathbf{u})\Big)$ and empirical beta copula $\mathbb{C}_n^\beta$ with empirical beta copula process $\mathbb{G}_n^\beta = \sqrt{n}\Big(\mathbb{C}_n^\beta(\mathbf{u}) - C(\mathbf{u})\Big)$. Suppose $\mathbb{G}_n \rightsquigarrow \mathbb{G}$ for $n \to \infty$ to $\mathbb{G}$ in $l^\infty([0,1]^d)$, where $\mathbb{G}$ is a limiting process having continuous trajectories almost surely. Then, in $l^\infty([0,1]^d)$*

$$\mathbb{G}_n^\beta = \mathbb{G}_n + o_p(1), n \longrightarrow \infty.$$

Proof. See Segers et al. 2017 Section 3. □

In short, Theorem 6.2.1 states that the empirical beta copula has the same large-sample distribution as the empirical copula and, thus, converges to the true copula (compare to Theorem 3.2.1 in Chapter 3). However, the empirical beta copula performs better for small samples. Segers et al. 2017 demonstrate that the empirical beta copula outperforms the empirical copula both in terms of bias and variance.

Algorithm 2: Sampling from Empirical Beta Copula

```
Input: Sample $\boldsymbol{Y} \subset \mathbb{R}^{n \times d}$, new sample size $m$
begin
    Compute rank matrix $\mathbf{R}^{n \times d}$ out of $\boldsymbol{Y}$
    Estimate marginal distributions of $\boldsymbol{Y}$ with univatiate KDE leading to
      $\widehat{F}_1^{-1}, \ldots, \widehat{F}_d^{-1}$
    for $i \leq m$ do
        Draw random from $I \in [1, \ldots, n]$
        for $j \leq d$ do
            Draw $u_{I,j} \sim \mathbb{B}(r_{Ij}, n + 1 - R_{Ij})$
        Set $\mathbf{u}_i = (u_{I1}, \ldots, u_{Id})$
        Rescale margins by $y_i' = \widehat{F}_1^{-1}(u_{i1}), \ldots, \widehat{F}_d^{-1}(u_{id})$.
Output: New sample $\boldsymbol{Y'} = (y_1', \ldots, y_m')$ of size m
```

6.3. Experiments

In this section, we present the results of our experiments. We use the same architecture for the autoencoder in all experiments for one dataset but replace the modeling technique for the latent space for all algorithms. The architecture, as well as implementation details, are given in the appendix. We further include a standard VAE and the Real NVP normalization flow approach modeling the latent space in our experiments to serve as a benchmark. Note that we purposely did not include any GAN in the experiments since the presented results should be independent of the trained generative models to ensure comparability of results.

Methodology

We train an autoencoder consisting of two neural nets, an *encoder* f, and a *decoder* g. The encoder f maps data $\boldsymbol{X}$ from the original space to a lower-dimensional space, while the decoder g reconstructs this low-dimensional data $\boldsymbol{Y}$ from the low-dimensional latent space to the original space (see Figure 6.1). We train both neural nets in a way that the reconstruction loss is minimized, i.e., that the reconstructed data $\boldsymbol{X}' = g\big\{f(\boldsymbol{X})\big\}$ is as similar to the original data $\boldsymbol{X}$ as possible. In the second step, we model the latent space $\boldsymbol{Y}$ data with a multivariate Gaussian distribution, a Gaussian mixture model, Kernel density estimates, the two presented copula methods and the Real NVP. Thus, we fit models with different flexibility and complexity while keeping the training process of the autoencoder untouched. Last, new samples are generated by decoding random samples from the learned model in the latent space. Note that such an approach is only reasonable when the underlying autoencoder has learned a relevant and interesting representation of the data and the latent space is smooth. We demonstrate this in the appendix.

Datasets

We conduct experiments on one small-scale, one medium, and one large-scale dataset. The small-scale *MNIST* dataset (LeCun et al., 2010) includes binary images of digits, while the medium-scale *SVHN* dataset (Netzer et al., 2011) contains images of house numbers in Google Street View pictures. The large-scale *CelebA* dataset (Liu et al., 2015) consists of celebrity images covering 40 different face attributes. We split data into a train set and a test set of 2000 samples which is a commonly used size for evaluation (Tagasovska et al., 2019; Xu et al., 2018). Note that the data sets cover different dimensionalities in the latent space, allowing for a throughout assessment of the methods under investigation.

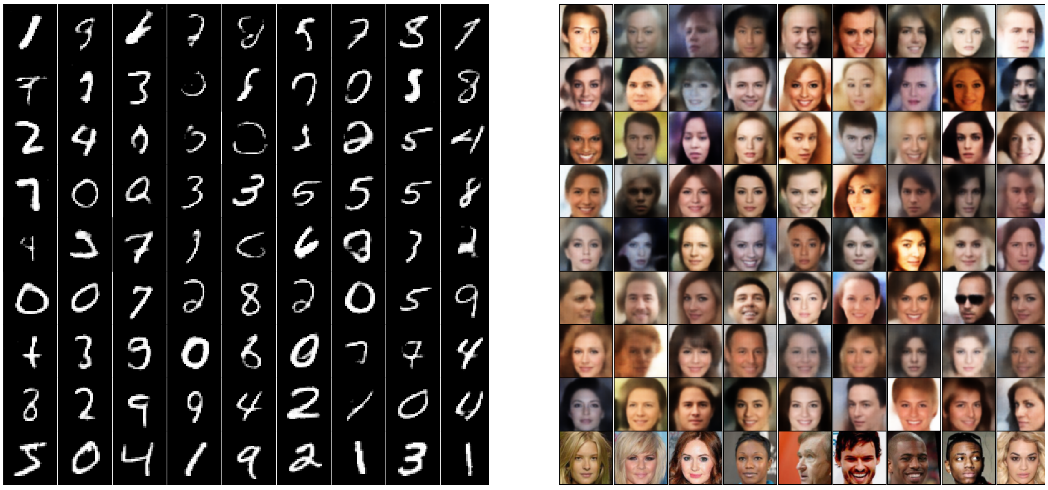
Evaluation

Evaluation of results is performed in several ways. First, we visually compare random pictures generated by the models. Second, we evaluate the results with the framework proposed by Xu et al. 2018, since a log-likelihood evaluation is known to be incapable of assessing the quality (Theis et al., 2016) and unsuitable for non-parametric models. Based on their results, we choose five metrics in our experiments: The *earth mover distance (EMD)*, also known as *Wasserstein distance* (Vallender, 1974); the *mean maximum discrepancy (MMD)* (Gretton et al., 2007); the *1-nearest neighbor-based* *two-sample test (1NN)*, a special case of the classifier two-sample test (Lopez-Paz and Oquab, 2017); the *Inception Score* (Salimans et al., 2016); and the *Frêchet inception distance* (Heusel et al., 2017). In line with Tagasovska et al. (2019) and as proposed by Xu et al. (2018), we further apply the EMD, MMD, and 1NN over feature mappings in the convolution space over ResNet-34 features. For all metrics except the Inception Score, lower values are better. For more details on the metrics, we refer to Xu et al. (2018). Next, we evaluate the ability to generate new, realistic pictures by the different latent space modeling techniques. Therefore, we compare new samples with their nearest neighbor in the latent space stemming from the original data. This shows us whether the learned distribution covers the whole latent space, or stays too close to known examples, i.e., the model does not generalize enough. Finally, we compare other features of the tested models, such as their ability of targeted sampling and of recombining attributes.

Results

Figure 6.3a and Figure 6.3b show images generated from each method for MNIST and CelebA. The GMM model is composed of 10 elements, and the KDE is constructed using a Gaussian kernel with a bandwidth fitted via a grid search and 10-fold cross-validation. The specification of the Real NVPs are given in the appendix.

For the MNIST dataset, we observe the best results for the EBCAE (row 6) and KDE (row 3), while the other methods seem to struggle a bit. For the CelebA, our visual observations are slightly different. All methods produce images that are clearly recognizable as faces. However, the Gaussian samples in row 1 and independent margins in row 2 create pictures with some unrealistic artefacts, blurry backgrounds, or odd colors. This is also the case for the GMM in row 4 and VCAE in row 5, but less severe. We believe that this comes from samples of an empty area in the latent space, i.e., where none of the original input pictures were projected to. In contrast to that, the samples in the latent space of the KDE, EBCAE, and Real NVP stay within these natural bounds, producing good results after passing the decoder (rows 3, 6, 8). Recall that all methods



(a) MNIST samples

(b) CelebA samples

Figure 6.3.: Comparison of synthetic samples of different autoencoder models. **1st row:** Fitted normal distribution, **2nd row:** Independent margins, **3rd row:** KDE-AE, **4th row:** GMM, **5th row:** VCAE, **6th row:** EBCAE, **7th row:** VAE, **8th row:** Real NVP, **Last row:** original pictures.

use the same autoencoder and only differ by means of sampling in the latent space. From our observations, we also conclude that the autoencoder for the CelebA dataset is less sensitive toward modeling errors in the latent space since all pictures are clearly recognizable as faces. In contrast, for the MNIST dataset, not all images clearly show numbers. Similar results for SVHN are presented in the appendix of the thesis.

The numerical results computed from 2000 random samples displayed in Figure 6.4 prove that dependence truly matters within the latent space. Simultaneously, the KDE, GMM, and EBCAE perform consistently well over all metrics, delivering comparable results to the more complex Real NVP. Especially the EBCAE outperforms the other methods, whereas the VCAE, Gauss model and VAE usually cluster in the middle.

We further report results over the number of samples in the latent space in the appendix. This, at first sight, unusual perspective visualizes the capability to reach good

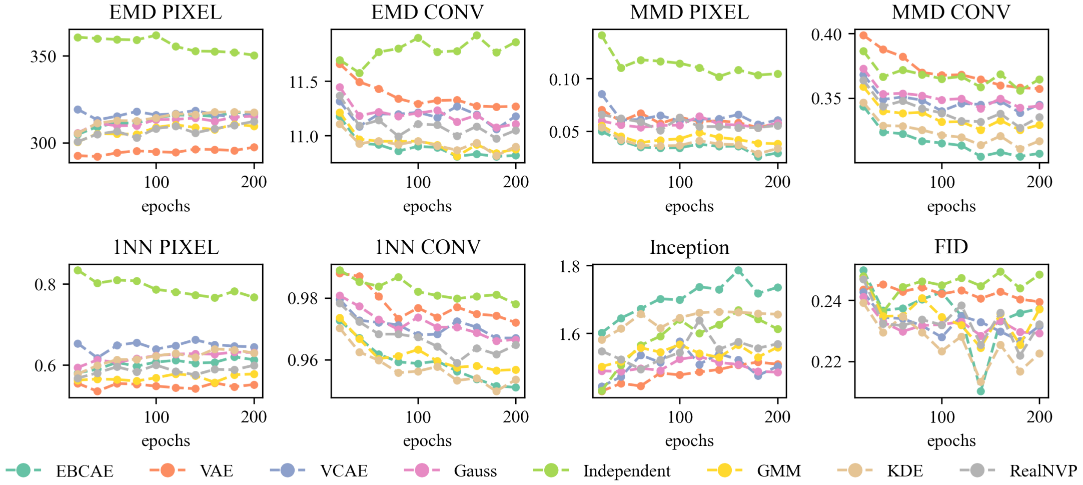
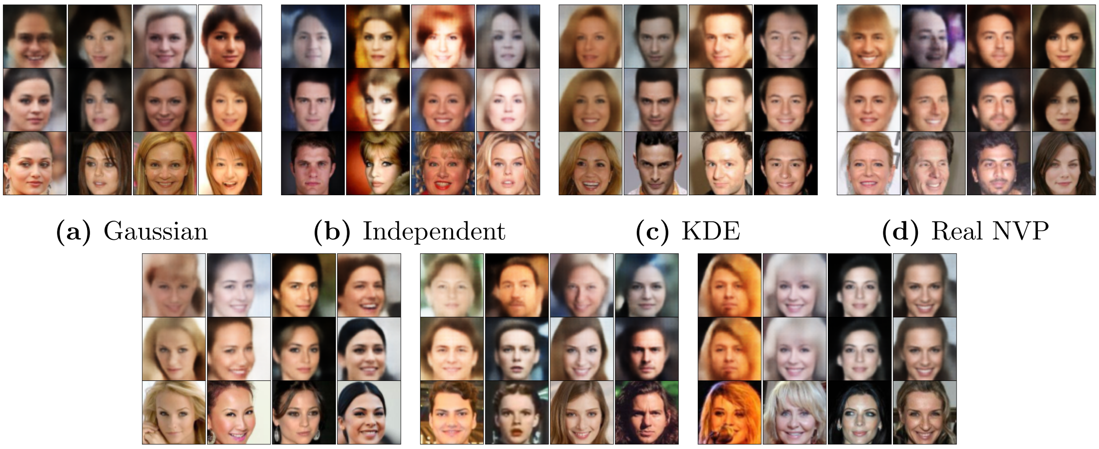


Figure 6.4.: Performance metrics of generative models on **CelebA**, reported over epochs computed from 2000 random samples. Note that they only differ in the latent space sampling and share the same autoencoder.

performance even for small sample sizes in latent space. In a small-sample regime, it is crucial to assess how fast a method adapts to data in the latent space and models it correctly. We see that all methods perform well for small sample sizes, i.e., $n = 200$. Similar experiments for MNIST and SVHN can be found in the appendix.

Next, we evaluate the different modeling techniques in their ability to generate new, realistic images. For this, we focus on pictures from the CelebA dataset in Figure 6.5. First, we create new, random samples with the respective method (top row) and then compare these with their decoded nearest neighbor in the latent space (middle row). The bottom row displays the latent space nearest neighbor in the original data space before applying the autoencoder. By doing so, we are able to disentangle two effects. First, the effect from purely encoding-decoding an image and, second, the effect of modeling the latent space. Thus, we can check whether new images are significantly different from the input, i.e., whether the distribution modeling the latent space merely reproduces images or generalizes to some extent.

We observe that the samples from GMM, VCAE and the here-used Real NVP substantially differ from their nearest neighbors. However, again they sometimes exhibit unrealistic colors and blurry backgrounds. The samples created from KDE and EBCAE look much more similar to their nearest neighbors in the latent space, indicating that these methods do not generalize to the extent of the other methods. However, their samples do not include unrealistic colors or features and seem to avoid sampling from areas where no data point of the original data is present. Thus, they stay in ’natural bounds’. Note that this effect apparently is not reflected in the numerical evaluation metrics. We, therefore, recommend that, in addition to a quantitative evaluation, a qualitative evaluation of the resulting images should always be performed.



(a) Gaussian **(b)** Independent **(c)** KDE **(d)** Real NVP

(e) GMM **(f)** VCAE **(g)** EBCAE

Figure 6.5.: Resulting examples of the six investigated modeling methods after decoding. **Top row:** New examples. **Middle row:** Nearest neighbor of training data Y in latent space after decoding. **Bottom row:** Original input picture of nearest neighbor in latent space.

To further underpin this point, Figure 6.6 shows 2-dimensional TSNE-Embeddings (see, e.g.,van der Maaten and Hinton 2008) of the latent space for all six versions of the autoencoder (MNIST). Black points indicate original input data and colored points are synthetic samples from the corresponding method. We see that the KDE, as well as the EBCAE, stay close to the original space. The samples from the GMM and Real NVP also seem to closely mimic the original data, whereas the other methods fail to do so. This visualization confirms our previous conjecture that some algorithms tend to sample from 'empty' areas in the latent space, leading to unrealistic results.

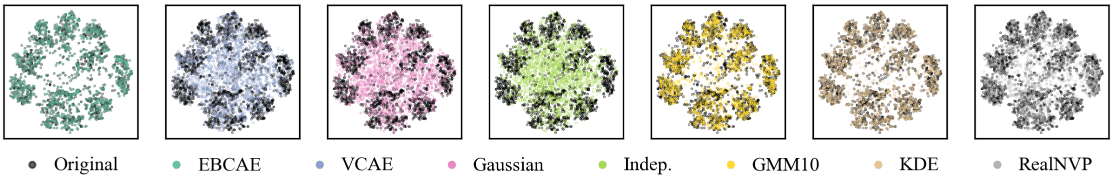


Figure 6.6.: TSNE embeddings of samples in the latent space of the **MNIST** dataset. Points from the original input training data $\boldsymbol{Y}$ are given in black, whereas new, synthetic samples $\boldsymbol{Y}'$ stemming from the different modeling methods are colored.

We also report computing times for learning and sampling of the different models for MNIST and CelebA in Table 6.3. Unsurprisingly, the more straightforward methods such as Gauss, Independence, KDE, and GMM, exhibit the lowest sampling times. The Real NVP shows the highest learning time as a neural network is fitted. However, we expect the difference to be much smaller once trained on an appropriate GPU. The times also reflect the complexities of the methods in the latent space dimensions.

Last, we discuss other features of the tested methods, such as targeted sampling and recombination. In contrast to the other techniques, the KDE and EBCAE allow for targeted sampling. Thus, we can generate new images with any desired characteristic directly, e.g., only ones in a data set of images of numbers. In the case of the KDE, this simply works by sampling from the estimated density of the corresponding sub-group. In the case of the EBCAE, we randomly choose among rows in the rank matrix of original samples that share the desired specific

Table 6.3.: Modeling and sampling time in the **CelebA** and **MNIST** dataset of 2000 artificial samples based on a latent space of size $n = 2000$ in [s].

	CelebA	**CelebA**	**MNIST**	**MNIST**
Method	Learn	Sample	Learn	Sample
Gauss	<0.01	0.01	0.002	0.002
Indep.	4.10	0.07	0.393	0.003
KDE	75.25	0.01	13.958	0.001
GMM	1.35	0.03	0.115	0.004
VCAE	306.97	148.48	10.345	4.590
EBCAE	3.41	59.36	0.328	5.738
Real NVP	2541.19	3.69	341.608	0.477

attribute. Other approaches are also possible, however, they need further tweaks to the model, training, or sampling as the *conditional variational autoencoder* (Sohn et al., 2015).

The second feature we discuss is recombination. By using copula-based models (VCAE and EBCAE), we can facilitate the decomposition idea and split the latent space in its dependence structure and margins, i.e., we combine the dependence structure of images with a specific attribute with the marginal distributions of images with different attributes. Therefore, copula-based methods allow controlling the attributes of created samples to some extent. Our experiments suggest that the dependence structure provides the basic properties of an image, while the marginal distributions are responsible for details (see, e.g., Figure 6.7). However, we want to point out that it is not generally clear what information is embedded in the dependence structure and what information is in the marginal distributions of latent space. This might also depend on the autoencoder and the dataset at hand. Thad said using such a decomposition enables higher flexibility and hopefully fuels new methodological developments in this field.

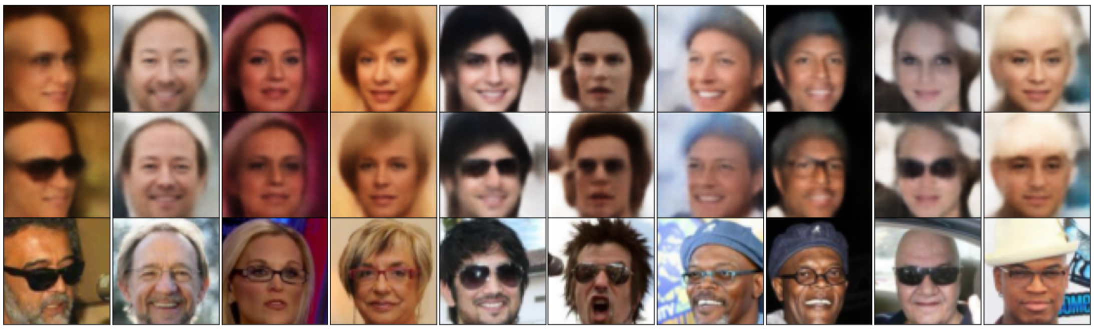


Figure 6.7.: **Top row:** Samples from the EBCAE constructed with the latent space dependence structure from samples with glasses and latent space marginal distributions from samples without glasses. **Middle row:** Nearest neighbor of training data $\boldsymbol{Y}$ in latent space after decoding. **Bottom row:** Original input picture of nearest neighbor in latent space.

6.4. Discussion

In this section, we want to discuss our experiments' results and express some further thoughts. So, is sampling from the latent space a simple way towards generative models? We observed that sampling from the latent space is indeed a viable approach to turning an autoencoder into a generative model. The main advantage is that the autoencoder is not restricted in any way (so it is able to find the best low-dimensional representation of the data). We also observe that the studied methods can achieve competitive results comparable to the more complex Real NVP approach and hence may be interesting for future research applications in more advanced autoencoders. Simultaneously, each modeling approach in this setting comes with its own restrictions, advantages, and problems.

We witness a trade-off between the ability to generalize, i.e., to create genuinely new pictures, and sample quality, i.e., to avoid unrealistic colors or artefacts. In cases where new data points are sampled in the neighborhood to existing points (as in the KDE or EBCAE), the newly generated data stays in somehow natural bounds and provides realistic, but not completely new, decoded samples. On the other hand, modeling the latent space too generically leads to bad-quality images. We believe this is similar to leaving the feasible set of an optimization problem or sampling from a wrong prior. While being close to actual points of the original latent space, new samples stay within the feasible set. By moving away from these points, the risk of sampling from an unfeasible region and thus creating unrealistic new samples increases. Recombination via a copula-based approach of marginal distributions and dependence structures offers the possibility to detect new feasible areas of the latent space for the creation of realistic images. Also, interpolating by building convex combinations of two points in the latent space seems reasonable. However, we can not guarantee proper interpolation results without further restrictions during training. Further, we observe that the performance metrics do not reflect the mentioned trade-off. Therefore, we strongly recommend not only checking quantitative results but also finding and analyzing the nearest neighbor in the original data to detect the pure reproduction of pictures. This also reveals that the development of additional evaluation metrics could be beneficial.

A closely related issue is the choice of a parametric vs. a non-parametric modeling method in the latent space. Parametric methods can place probability mass in the latent space, where no data point of the original input data was observed. Thus, parametric methods are able to generate (truly) new data, subject to their assumption. However, if the parametric assumption is wrong, the model creates samples from 'forbidden' areas in the latent space leading to unrealistic images. Despite this, carefully chosen parametric models can be beneficial, and even a log-likelihood is computable (although we do not use it for training). Non-parametric methods altogether avoid this human decision and possible source of error but are closely bound to the empirical distribution of the given input data. Consequently, such methods can miss important areas of the latent space but create more realistic images. Furthermore, adjusting parameters of the non-parametric models, such as increasing bandwidths or lowering truncation levels, offer possibilities to overcome these limitations slowly.

Besides the major points above, the EBCAE and KDE offer an easy way of targeted sampling without additional training effort. This can be beneficial for various applications and is not as straightforward with other methods. Lastly, the investigated methods differ in their runtime. While vine copula learning and sampling is very time-intensive for high dimensions, the EBCAE is much faster but still outperformed by the competitors. For the non-copula methods, the GMM is really fast in both datasets while still capturing the dependence structure to some extent. In contrast to that, the Real NVP needs more time for training but is rather quick in generating new samples.

To sum up, we can confirm that there are indeed simple methods to turn a plain autoencoder into a generative model. We conclude that the optimal method to do so depends on the goals of the user. Besides runtime considerations, the specific application of the autoencoder matters. For example, if one is interested in targeted sampling, EBCAE or KDE should be applied. Recombination experiments call for a copula-based approach, whereas in all cases, the trade-off between generalization and out-of-bound sampling should be considered. Lastly, during our research, we found that future, more theory-driven work most likely could establish the structural link between copulas and normalization flows via the Rosenblatt transformation Rosenblatt (1952).

7. Conclusion

This thesis widened the theoretical understanding of copulas and contributed to new applications of copula theory. After introducing some necessary theory and basic notation in the beginning, we analyzed the asymptotic behavior of the empirical copula, i.e., especially the variance thereof in detail (see Chapter 3). We introduced a new estimator and showed its superiority over existing ones in extensive simulation studies. In this part of the thesis, we also demonstrated how the same estimating procedure could be applied to arbitrary rectangular boxes out of the d-dimensional copula domain and therefore opening the door for future developments and applications.

Next, we applied the concept of copulas to extend the well-known idea of the Lorenz curve to arbitrary dimensions. We argue that this multivariate tool, including the further introduced multivariate extension of the Gini coefficient, is a more complete measure of inequality compared to the univariate Lorenz curve since it allows multiple variables to be included in the assessment. Last, we demonstrate its applicability in a small case study using income and wealth to evaluate the monetary inequality in different countries, including Germany (see Chapter 4).

The third contribution in this thesis helps to assess risks and prices in day-ahead electricity markets more precisely by providing a multivariate forecast density for electricity prices (see Chapter 5). It is simple and straightforward to apply as it can be based on any existing point forecasting model and does not need much computational effort. Within this powerful post-processing method, we propose to combine our rank-based copula approach with time series modeling and achieve good results even for non-optimal point forecasting methods. A thorough assessment of several different configurations of the method in various electricity markets demonstrates the practical abilities of the approach. Such methods are increasingly important for future energy markets contributing to a stable and reliable energy supply.

Last, we deal with the vast area of generative modeling by using autoencoders, a classical field in machine learning (see Chapter 6). We compare different methodologies for modeling the latent space in autoencoders, including the use of copula techniques, which have not been done in the literature so far. In the chapter, we further propose the usage of the empirical beta copula and assess it against various other methods. We conclude that no 'best' method exists, and the user should choose the method according to its needs.

In summary, the thesis provides new solutions for dealing with dependence in different application areas and therefore is an essential step in an increasingly data-driven world. However, the thesis left some work related to the chapters for future research. In particular, it would be interesting to aim for further improvement of the estimator in Chapter 3 by deriving more precise estimates of the partial copula derivatives. Further, the proposed forecasting algorithm for electricity prices could be evaluated in other applications with a 24-hourly structure, e.g., solar, wind, or load forecasting. We anticipate that only a few adjustments to the proposed methods may be necessary, but leave this to future research. Finally, in light of the last chapter in this thesis, the link between copulas and normalization flows for generative modeling seems interesting and worth further investigation.

Appendices

A. Computer Code and Data

In this appendix, we provide the links to the computer code and data of each chapter. All code is written using Python and published online to make it accessible to other researchers.

Code for Chapter 3

Ready-to-use computer code (Python) for Chapter 3 is available under the following URL:
`https://github.com/FabianKaechele/Copula-Covariance-Estimation`.

Code and Data for Chapter 4

Python computer code for the chapter is publicly available in the following repository:
`https://github.com/FabianKaechele/Multivariate_Extension_Lorenz_Gini`

Raw datasets analyzed in Section 4.7 of the Chapter are available from SOEP (2019) and Luxembourg Wealth Study (LWS) Database (2020), but restrictions apply to the availability of these data, which were used under license for the current study, and so are not publicly available. Information on how to obtain it and reproduce the analysis is available from the corresponding author on request.

Code and Data for Chapter 5

Computer code and all information including data to reproduce the experiments of the chapter is available at the following link:
`https://github.com/FabianKaechele/Energy-Schaake`

Code and Data for Chapter 6

Computer code and all information to reproduce the experiments of the chapter is available at the following link: `https://github.com/FabianKaechele/SamplingFromAutoencoders`

The used datasets are publicly available for download via the *pytorch* interface (link to the documentation):
`https://pytorch.org/vision/stable/datasets.html`

B. Appendix to Chapter 3

Complete Derivation of Covariance Term

$$
\begin{aligned}
\mathrm{cov}\Big\{\mathbb{G}_C(\mathbf{u}), \mathbb{G}_C(\mathbf{v})\Big\} &= \mathrm{cov}\Big\{\mathbb{B}(\mathbf{u}) - \sum_{k_1=1}^{d} \partial_{k_1} C(\mathbf{u})\mathbb{B}(\mathbf{u}^{(k_1)}), \mathbb{B}(\mathbf{v}) - \sum_{k_2=1}^{d} \partial_{k_2} C(\mathbf{v})\mathbb{B}(\mathbf{v}^{(k_2)})\Big\} \\
&= \mathrm{cov}\Big\{\mathbb{B}(\mathbf{u}), \mathbb{B}(\mathbf{v})\Big\} \\
&\quad + \mathrm{cov}\Big\{- \sum_{k_1=1}^{d} \partial_{k_1} C(\mathbf{u})\mathbb{B}(\mathbf{u}^{(k_1)}), - \sum_{k_1=1}^{d} \partial_{k_1} C(\mathbf{u})\mathbb{B}(\mathbf{v}^{(k_2)})\Big\} \\
&\quad + \mathrm{cov}\Big\{\mathbb{B}(\mathbf{u}), - \sum_{k_2=1}^{d} \partial_{k_2} C(\mathbf{v})\mathbb{B}(\mathbf{v}^{(k_2)})\Big\} \\
&\quad + \mathrm{cov}\Big\{- \sum_{k_1=1}^{d} \partial_{k_1} C(\mathbf{u})\mathbb{B}(\mathbf{u}^{(k_1)}), \mathbb{B}(\mathbf{v}),\Big\} \\
&= C(\mathbf{u} \wedge \mathbf{v}) - C(\mathbf{u})C(\mathbf{v}) \\
&\quad + \sum_{k_1=1}^{d} \sum_{k_2=1}^{d} \partial_{k_1} C(\mathbf{u}) \partial_{k_2} C(\mathbf{v}) \mathrm{cov}\Big\{\mathbb{B}(\mathbf{u}^{(k_1)}), \mathbb{B}(\mathbf{v}^{(k_2)})\Big\} \\
&\quad + \sum_{k_2=1}^{d} \partial_{k_2} C(\mathbf{v}) \mathrm{cov}\Big\{\mathbb{B}(\mathbf{u}), \mathbb{B}(\mathbf{v}^{(k_2)})\Big\} \\
&\quad + \sum_{k_1=1}^{d} \partial_{k_1} C(\mathbf{u}) \mathrm{cov}\Big\{\mathbb{B}(\mathbf{u}^{(k_1)}), \mathbb{B}(\mathbf{v})\Big\}
\end{aligned}
$$

$$
\begin{aligned}
&= C(\mathbf{u} \wedge \mathbf{v}) - C(\mathbf{u})C(\mathbf{v}) \\
&\quad + \sum_{k_1=1}^{d} \sum_{k_2=1}^{d} \partial_{k_1} C(\mathbf{u}) \partial_{k_2} C(\mathbf{v}) \Big\{ C(\mathbf{u}^{(k_1)} \wedge \mathbf{v}^{(k_2)}) - C(\mathbf{u}^{(k_1)}) C(\mathbf{v}^{(k_2)} \Big\} \\
&\quad + \sum_{k_2=1}^{d} \partial_{k_2} C(\mathbf{v}) \Big\{ C(\mathbf{u} \wedge \mathbf{v}^{(k_2)}) - C(\mathbf{u}) C(\mathbf{v}^{(k_2)} \Big\} \\
&\quad + \sum_{k_1=1}^{d} \partial_{k_1} C(\mathbf{u}) \Big\{ C(\mathbf{u}^{(k_1)} \wedge \mathbf{v}) - C(\mathbf{u}^{(k_1)}) C(\mathbf{v}) \Big\}.
\end{aligned}
$$

Example 3.4.1 continued

We now dissolve the resulting variance term for area S and use the following notations for the derivatives and volumes of the copulas:

$$
\begin{array}{llll}
\widehat{\dfrac{\partial C_n(u_1^{(up)}, u_2^{(up)})}{\partial u_1^{(up)}}} := a_0 & \widehat{\dfrac{\partial C_n(u_1^{(up)}, u_2^{(up)})}{\partial u_2^{(up)}}} & := & b_0 \\
\widehat{\dfrac{\partial C_n(u_1^{(low)}, u_2^{(up)})}{\partial u_1^{(low)}}} := a_1 & \widehat{\dfrac{\partial C_n(u_1^{(low)}, u_2^{(up)})}{\partial u_2^{(up)}}} & := & b_1 \\
\widehat{\dfrac{\partial C_n(u_1^{(up)}, u_2^{(low)})}{\partial u_1^{(up)}}} := a_2 & \widehat{\dfrac{\partial C_n(u_1^{(up)}, u_2^{(low)})}{\partial u_2^{(low)}}} & := & b_2 \\
\widehat{\dfrac{\partial C_n(u_1^{(low)}, u_2^{(low)})}{\partial u_1^{(low)}}} := a_3 & \widehat{\dfrac{\partial C_n(u_1^{(low)}, u_2^{(low)})}{\partial u_2^{(low)}}} & := & b_3
\end{array}
$$

$$
\begin{array}{rlrl}
\mathbb{B}_{C_n}(u_1^{(up)}, u_2^{(up)}) := & x & \mathbb{B}_{C_n}(u_1^{(up)}, 1) & := \ d \\
\mathbb{B}_{C_n}(1, u_2^{(up)}) := & e & \mathbb{B}_{C_n}(u_1^{(low)}, u_2^{(up)}) & := \ y \\
\mathbb{B}_{C_n}(u_1^{(low)}, 1) := & f & \mathbb{B}_{C_n}(u_1^{(up)}, u_2^{(low)}) & := \ z \\
\mathbb{B}_{C_n}(1, u_2^{(low)}) := & g & \mathbb{B}_{C_n}(u_1^{(low)}, u_2^{(low)}) & := \ \alpha
\end{array}
$$

Splitting term within the variance expression in its components leads to:

$$
\begin{aligned}
\widehat{\text{var}}\Big\{V_{\mathbb{C}_n}(S)\Big\} = {} & \text{var}(x) + a_0^2\text{var}(d) + b_0^2\text{var}(e) + \text{var}(y) + a_1^2\text{var}(f) + b_1^2\text{var}(e) + \text{var}(z) \\
& + a_2^2\text{var}(d) + b_2^2\text{var}(g) + \text{var}(\alpha) + a_3^2\text{var}(f) + b_3^2\text{var}(g) \\
& - 2a_0\text{cov}(x,d) - 2b_0\text{cov}(x,e) - 2\text{cov}(x,y) + 2a_1\text{cov}(x,f) + 2b_1\text{cov}(x,e) \\
& - 2\text{cov}(x,z) + 2a_2\text{cov}(x,d) + 2b_2\text{cov}(x,g) + 2\text{cov}(x,\alpha) - 2a_3\text{cov}(x,f) \\
& - 2b_3\text{cov}(x,g) \\
& + 2a_0b_0\text{cov}(d,e) + 2a_0\text{cov}(d,y) - 2a_0a_1\text{cov}(d,f) - 2a_0b_1\text{cov}(d,e) \\
& + 2a_0\text{cov}(d,z) - 2a_0a_2\text{var}(d) - 2a_0b_2\text{cov}(d,g) - 2a_0\text{cov}(d,\alpha) \\
& + 2a_0a_3\text{cov}(d,f) + 2a_0b_3\text{cov}(d,g) \\
& + 2b_0\text{cov}(e,y) - 2b_0a_1\text{cov}(e,f) - 2b_0b_1\text{var}(e) + 2b_0\text{cov}(e,z) \\
& - 2b_0a_2\text{cov}(e,d) - 2b_0b_2\text{cov}(e,g) - 2b_0\text{cov}(e,\alpha) + 2b_0a_3\text{cov}(e,f) \\
& + 2b_0b_3\text{cov}(e,g) \\
& - 2a_1\text{cov}(y,f) - 2b_1\text{cov}(y,e) + 2\text{cov}(y,z) - 2a_2\text{cov}(y,d) \\
& - 2b_2\text{cov}(y,g) - 2\text{cov}(y,\alpha) + 2a_3\text{cov}(y,f) + 2b_3\text{cov}(y,g) \\
& + 2a_1b_1\text{cov}(f,e) - 2a_1\text{cov}(f,z) + 2a_1a_2\text{cov}(f,d) + 2a_1b_2\text{cov}(f,g) \\
& + 2a_1\text{cov}(f,\alpha) - 2a_1a_3\text{var}(f) - 2a_1b_3\text{cov}(f,g) \\
& - 2b_1\text{cov}(e,z) + 2b_1a_2\text{cov}(e,d) + 2b_1b_2\text{cov}(e,g) + 2b_1\text{cov}(e,\alpha) \\
& - 2b_1a_3\text{cov}(e,f) - 2b_1b_3\text{cov}(e,g) \\
& - 2a_2\text{cov}(z,d) - 2b_2\text{cov}(z,g) - 2\text{cov}(z,\alpha) + 2a_3\text{cov}(z,f)
\end{aligned}
$$

$$
\begin{aligned}
&+ 2b_3\mathrm{cov}(z, g) \\
&+ 2a_2b_2\mathrm{cov}(d, g) + 2a_2\mathrm{cov}(d, \alpha) - 2a_2a_3\mathrm{cov}(d, f) - 2a_2b_3\mathrm{cov}(d, g) \\
&+ 2b_2\mathrm{cov}(g, \alpha) - 2b_2a_3\mathrm{cov}(g, f) - 2b_2b_3\mathrm{var}(g) \\
&- 2a_3\mathrm{cov}(\alpha, f) - 2b_3\mathrm{cov}(\alpha, g) \\
&+ 2a_3b_3\mathrm{cov}(f, g).
\end{aligned}
$$

At this point, the covariance terms can be dissolved. Therefore, we use the super-scripted ^{c} to indicate the corresponding copula, e.g., $x^c := \hat{C}_n(u_1^{(up)}, u_2^{(up)})$, and obtain the estimator by

$$
\begin{aligned}
\widehat{\mathrm{var}}\left\{V_{\mathbb{C}_n}(S)\right\} = {}& x^c(1 - x^c) + a_0^2d^c(1 - d^c) + b_0^2e^c(1 - e^c) + y^c(1 - y^c) + a_1^2f^c(1 - f^c) \\
&+ b_1^2e^c(1 - e^c) + z^c(1 - z^c) + a_2^2d^c(1 - d^c) + b_2^2(g^c(1 - g^c) + \alpha^c(1 - \alpha^c) \\
&+ a_3^2f^c(1 - f^c) - b_3^2g^c(1 - g^c) \\
&- 2a_0(x^c - x^cd^c) - 2b_0(x^c - x^ce^c) - 2(y^c - x^cy^c) + 2a_1(y^c - x^cf^c) \\
&+ 2b_1(x^c - x^ce^c) - 2(z^c - x^cz^c) + 2a_2(x^c - x^cd^c) + 2b_2(z^c - x^cg^c) \\
&+ 2(\alpha^c - x^c\alpha^c) - 2a_3(y^c - x^cf^c) - 2b_3(z^c - x^cg^c) \\
&+ 2a_0b_0(x^c - d^ce^c) + 2a_0(y^c - d^cy^c) - 2a_0a_1(f^c - f^cd^c) - 2a_0b_1(x^c - d^ce^c) \\
&+ 2a_0(z^c - d^cz^c) - 2a_0a_2d(1 - d^c) - 2a_0b_2(z^c - d^cg^c) - 2a_0(\alpha^c - d^c\alpha^c) \\
&+ 2a_0a_3(f^c - f^cd^c) + 2a_0b_3(z^c - d^cg^c) \\
&+ 2b_0(y^c - e^cy^c) - 2b_0a_1(y^c - e^cf^c) - 2b_0b_1e(1 - e^c) + 2b_0(z^c - e^cz^c) \\
&- 2b_0a_2(x^c - d^ce^c) - 2b_0b_2(g^c - e^cg^c) - 2b_0(\alpha^c - e^c\alpha^c) + 2b_0a_3(y^c - e^cf^c) \\
&+ 2b_0b_3(g^c - e^cg^c) \\
&- 2a_1(y^c - y^cf^c) - 2b_1(y^c - e^cy^c) + 2(k^c - y^cz^c) - 2a_2(y^c - d^cy^c) \\
&- 2b_2(\alpha^c - y^cg^c) - 2(\alpha^c - y^c\alpha^c) + 2a_3(y^c - y^cf^c) + 2b_3(\alpha^c - y^cg^c) \\
&+ 2a_1b_1(y^c - e^cf^c) - 2a_1(z^c - f^cz^c) + 2a_1a_2(f^c - f^cd^c) + 2a_1b_2(\alpha^c - f^cg^c) \\
&+ 2a_1(\alpha^c - f^c\alpha^c) - 2a_1a_3f^c(1 - f^c) - 2a_1b_3(\alpha^c - f^cg^c)
\end{aligned}
$$

$$
\begin{aligned}
&- 2b_1(z^c - e^c z^c) + 2b_1 a_2(x^c - d^c e^c) + 2b_1 b_2(g^c - e^c g^c) + 2b_1(\alpha^c - e^c \alpha^c) \\
&- 2b_1 a_3(y^c - e^c f^c) - 2b_1 b_3(g^c - e^c g^c) \\
&- 2a_2(z^c - d^c z^c) - 2b_2(z^c - g^c z^c) - 2(\alpha^c - z^c \alpha^c) + 2a_3(z^c - f^c z^c) \\
&+ 2b_3(z^c - g^c z^c) \\
&+ 2a_2 b_2(z^c - d^c g^c) + 2a_2(\alpha^c - d^c \alpha^c) - 2a_2 a_3(f^c - d^c f^c) - 2a_2 b_3(z^c - d^c g^c) \\
&+ 2b_2(\alpha^c - g^c \alpha^c) - 2b_2 a_3(\alpha^c - f^c g^c) - 2b_2 b_3 g^c(1 - g^c) \\
&- 2a_3(\alpha^c - f^c \alpha^c) - 2b_3(\alpha^c - g^c \alpha^c) \\
&+ 2a_3 b_3(\alpha^c - f^c g^c).
\end{aligned}
$$

Simulation Results for Sample Size $n = 200$

We report additional simulation results to Table 3.1 for a sample size of $n = 200$ in the table below. The values of α_{pdm} again are copied from Table 2 and 4 in Bücher and Dette (2010), with MSE values given below. For the α_β, we report MSE values only, copied from Table 2.1 in Kiriliouk et al. (2021). Similar to a sample size of $n = 100$ we observe a better finite sample performance of the plug-in estimator in all cases compared to the α_{pdm}. In comparison with the α_β both estimators yield comparable results.

n=100		(1/3,1/3)	(1/3,2/3)	(2/3,1/3)	(2/3,2/3)
Plug-in	(1/3,1/3)	0.05043 0.3993	0.01978 0.2271	0.01986 0.2500	0.00907 0.1499
	(1/3,2/3)		0.035 0.4391	0.00854 0.0969	0.01797 0.1314
	(2/3,1/3)			0.03439 0.4732	0.01837 0.1397
	(2/3,2/3)				0.05311 0.22568
α_{pdm}	(1/3,1/3)	0.0513 0.4595	0.0203 0.2673	0.0201 0.2798	0.0092 0.1961
	(1/3,2/3)		0.0356 0.5211	0.0087 0.1069	0.0184 0.1577
	(2/3,1/3)			0.0355 0.5092	0.0185 0.1681
	(2/3,2/3)				0.0537 0.2992
α_{β}	(1/3,1/3)	– 0.6205	– 0.2427	– 0.2383	– 0.1547
	(1/3,2/3)		– 0.4933	– 0.0857	– 0.1366
	(2/3,1/3)			– 0.4898	– 0.1376
	(2/3,2/3)				– 0.4183

Table B.1.: Sample covariances for the Clayton copula with $\theta = 1$ and sample size $n = 200$. Mean squared error values (multiplied by 10^4) for the plug-in estimator are given below the estimate (obtained from 1000 estimates).

C. Appendix to Chapter 5

Algorithmic Description of Proposed Method

Here we present algorithmic descriptions of the proposed method for the raw-error version without a time series model. For ease of presentation, we omit time indices in this section; for example, we denote the price for hour h of day t by y_h instead of $y_{t,h}$.

Algorithm 3: Schaake shuffle without time series model (Schaake-Raw)

```
Result: Simulated multivariate forecast distribution for day ahead electricity prices $\widehat{\mathbf{y}}_t^*$
Input: History of point forecasts $\hat{y}$ for day ahead electricity prices
for each day in Error Learning Phase do
    for each hour h do
        Save last 90 errors in learning data $\epsilon_h = y_h - \hat{y}_h$
    end
end
for each day in Dependence Learning Phase do
    for each hour h do
        Calculate and save realized quantile $\hat{u}_h$ of $\epsilon_h$ in empirical distribution $\hat{F}_h$ by $\hat{u}_h = \hat{F}_h(\epsilon_h)$
    end
end
for each day t in Forecasting Phase do
    for each hour h do
        for each ensemble member i do
            $\tilde{y}_h^i = (\hat{y}_h + 0) + \hat{F}_h^{-1}\{i/(m+1)\} \times 1$
        end
    end
    Derive empirical copula $\hat{C}$ and rank matrix $\hat{\mathbf{R}}$ out of saved, realized quantiles $\hat{u}$
    Pair up univariate forecast ensembles according to rank matrix $\hat{\mathbf{R}}$ (Formula 5.6) to obtain final
      forecast ensemble $\widehat{\mathbf{y}}_t^*$
    Update Error Learning Phase and Dependence Learning Phase
end
```

Time Series Model for Error Standardization

Our goal is to estimate the conditional distributions $G_{t,h}(x) := F_h(\frac{x-\mu_{t,h}}{\sigma_{t,h}})$ of the forecast errors $\epsilon_{t,h}$ via a time series model, where the indices represent the day (t) and hour (h). In our case, we utilize an AR(1)-GARCH(1,1) model but complexity may vary according to the point forecasting model that generated this sequence of forecast errors.

The AR(1)-GARCH(1,1) model of point forecast errors $\epsilon_{t,h}$ for a given hour h is given by

$$\epsilon_{t,h} = a_{0,h} + a_{1,h}\epsilon_{t-1,h} + \sigma_{t,h}z_{t,h} \tag{C.1}$$

$$\sigma^2_{t,h} = b_{0,h} + b_{1,h}\sigma_{t-1,h}z^2_{t-1,h} + b_{2,h}\sigma^2_{t-1,h}. \tag{C.2}$$

The coefficients $a_{0,h},\ a_{1,h},\ a_{2,h},\ b_{0,h},\ b_{1,h},\ b_{2,h}$ are the model parameters and $z_{t,h}$ is the standardized residual that follows the distribution function F_h (with zero mean and unit variance) and is independent over time. From the perspective of day $t-1$, $\epsilon_{t,h}$ is a random variable with expected conditional standard deviation $\hat{\sigma}_{t,h}$ and mean $\hat{\mu}_{t,h} := \hat{a}_{0,h} + \hat{a}_{1,h}\epsilon_{t-1,h}$.

The forecasts $\hat{\mu}_{t,h}$ and $\hat{\sigma}_{t,h}$ of the time series model are now used to obtain the standardized residuals $\hat{z}_{t,h} = \frac{\epsilon_{t,h}-\hat{\mu}_{t,h}}{\hat{\sigma}_{t,h}}$. Note that we use a separate AR-GARCH model for each hour h, resulting in 24 independent models. See Brockwell and Davis (2016, Section 7) and Lütkepohl and Krätzig (2004, Section 5) for textbook treatments of GARCH and related models.

Details on Forecast Evaluation

Assessment of Calibration

We assess univariate calibration by examining *verification rank histograms* (see e.g. Gneiting et al., 2008), separately for each hour $h = 1, \dots, 24$. To describe the verification rank histogram for hour h, let $R_{t,h} = 1 + \sum_{i=1}^{90} \mathbf{1}(\tilde{y}^i_{t,h} < y_{t,h})$ denote the rank of the realization $y_{t,h}$ in the merged sample $\{y_{t,h}, \{\tilde{y}^i_{t,h}\}_{i=1}^{90}\}$ that contains the realization and the forecast sample of size 90. If the forecast for hour h is calibrated, i.e. if observed prices behave like random draws from the (hour-specific) forecast distribution, then the distribution of $\{R_{t,h}\}_{t=1}^{T}$ is uniform. By contrast, a U-shaped distribution indicates that observed prices are often higher or lower than expected by the forecast distribution, implying that the latter is too narrow (or overconfident).

To assess multivariate calibration, we use the *Average Rank Histogram* (Thorarinsdottir et al., 2016) which has a relatively simple interpretation that is similar to the univariate case. The Average Rank Histogram is based on the average rank across hours, given by $\text{AvgRank}_t = (\sum_{h=1}^{24} R_{t,h})/24$. If the forecast is calibrated, the latter series is uniformly distributed.

CRPS

To assess the univariate forecasts, we use the *Continuous Ranked Probability Score (CRPS)*, which we average over time and across all 24 hours h. For a univariate forecast distribution F (represented by its cumulative distribution function, or CDF) and an outcome y, the CRPS is defined by

$$\begin{aligned} CRPS(F, y) &= \int \big\{F(x) - \mathbf{1}(x \geq y)\big\}^2 dx \\ &= \mathbb{E}_F|Y - y| - \frac{1}{2}\mathbb{E}_F|Y - Y'|, \end{aligned} \tag{C.3}$$

where Y and Y' are independent random variables distributed according to F, and $\mathbf{1}(x \geq y)$ denotes the indicator function which equals one if $x \geq y$ and zero otherwise. We aggregate the $T \times 24$ score values (for each day and hour) by computing their mean.

Energy Score

The multivariate extension of the CRPS, the *Energy Score (ES)*. For a multivariate forecast distribution F, the ES is given by

$$ES(F, y) = \mathbb{E}_F ||Y - y|| - \frac{1}{2}\mathbb{E}_F ||Y - Y'||.$$

For an ensemble forecast F_{ens} with members $\hat{y}^{(1)}, \ldots, \hat{y}^{(m)}$ it reduces to

$$ES(F_{ens}, \mathbf{y}) = \frac{1}{m}\sum_{k=1}^{m} ||\hat{y}^{(k)} - y|| - \frac{1}{2m^2}\sum_{l=1}^{m}\sum_{k=1}^{m} ||\hat{y}^{(l)} - \hat{y}^{(k)}||, \tag{C.4}$$

see Gneiting et al. (2008). We average the ES across the T days of the evaluation sample.

Diebold-Mariano-Test

We apply the Diebold and Mariano (1995) test for testing whether differences in forecast performance are statistically significant. The test considers the null hypothesis that $\mathbb{E}(S_{1,t} - S_{2,t}) = 0$, where S_{1t} and S_{2t} denote the scores of forecasting methods 1 and 2 on day t. In our case, the scores correspond to either the CRPS or the Energy Score. In case of the CRPS we follow Ziel and Weron (2018) and apply the DM test jointly to all hourly series, by using the daily mean score across all 24 hourly series. The Diebold-Mariano test is a t-test with test statistic given by

$$\frac{\bar{\delta}_{12}}{\hat{\sigma}(\bar{\delta}_{12})}$$

where $\bar{\delta}_{12} = T^{-1}\sum_{t=1}^{T}(S_{1,t} - S_{2,t})$ is the mean score difference, and $\hat{\sigma}(\bar{\delta}_{12})$ is its estimated standard deviation. The test statistic is standard normally distributed under the null hypothesis. Since we focus on one-day ahead forecasts, we set $\hat{\sigma}(\bar{\delta}_{12})$ equal to the sample standard deviation.

Assessment of Interval Forecasts

To asses interval forecasts in Section 5.4.3, we use the *Forecast Interval Coverage Probability (FICP)*, and the *Forecast Interval Normalized Average Width (FINAW)*, see e.g. Quan et al. 2014).

Denote the prediction interval for forecast case t by $[L_t^\alpha, U_t^\alpha]$, where $\alpha \in [0,1]$ is the desired nominal coverage probability (in our case, $\alpha = 93.33\%$). FICP measures the actual coverage rate for a sequence of interval forecasts corresponding to observations $t = 1, \ldots, T$:

$$FICP \;=\; \frac{1}{T}\sum_{t=1}^{T}\psi_t^{(\alpha)}$$

where $\psi_t^{(\alpha)}$ is an indicator for the coverage behavior, with $\psi_t^{(\alpha)} = 1$ if the realized value y_t of the predictand lies between the forecasted lower bound $L_t^{(\alpha)}$ and upper bound $U_t^{(\alpha)}$, and $\psi_t^{(\alpha)} = 0$ otherwise. FICP should be at least as large as its nominal value of α. If FICP is too small, the intervals are overconfident, in that they cover too few of the realized observations. While 'too high' values of FICP (exceeding nominal coverage α) are not problematic by themselves, they typically come at the expense of wider prediction intervals.

FINAW measures the normalized width of the interval defined by the upper bound $U_t^{(\alpha)}$ and lower bound $L_t^{(\alpha)}$. It is defined by

$$FINAW \;=\; \frac{1}{T \cdot R}\sum_{t=1}^{n}(U_t^{(\alpha)} - L_t^{(\alpha)})$$

where R is the maximum minus the minimum of y_t, i.e., $R = \max_t\{y_t\} - \min_t\{y_t\}$. Here a smaller value corresponds to a sharper prediction interval. However, the actual coverage (FICP) must be respected by the forecast as well.

Prediction Intervals in Load Profile Example

Figure C.1 displays prediction intervals in the load profile example of Section 5.4.3 for the full sample period. It complements Figure 5.7 in the main text that zooms in on the last 90 days of the sample period.

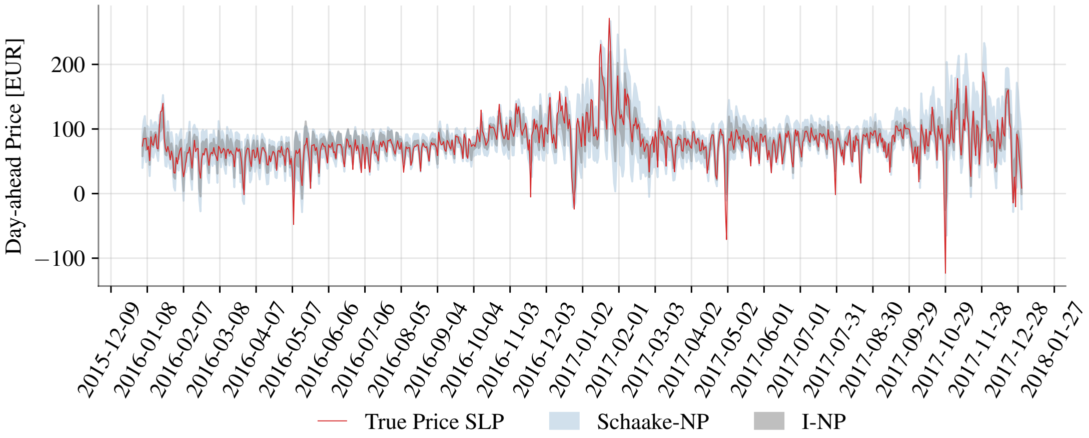


Figure C.1.: Prediction intervals and realized prices for SLP G0 from 4 January 2016 to 31 December 2017.

D. Appendix to Chapter 6

Architectures of Autoencoders and VAE

We use the same architecture for EBCAE, VCAE, and VAE as described below. All models were trained by minimizing the Binary Cross Entropy loss.

MNIST

Encoder:

$$\begin{aligned} x \in R^{32\times32} &\to Conv_{32} && \to BN \to ReLu \\ &\to Conv_{64} && \to BN \to ReLu \\ &\to Conv_{128} && \to BN \to ReLu \\ & && \to FC_{10} \end{aligned}$$

Decoder:

$$\begin{aligned} y \in R^{10} \to FC_{100} &\to ConvT_{128} && \to BN \to ReLu \\ &\to ConvT_{64} && \to BN \to ReLu \\ &\to ConvT_{128} && \to BN \to ReLu \\ & && \to FC_{1} \end{aligned}$$

For all (de)convolutional layers, we used 4 × 4 filters, a stride of 2, and a padding of 1. BN denotes batch normalization, $ReLU$ rectified linear units, and FC fully connected layers. Last, $Conv_k$ denotes the convolution with k filters.

SVHN

In contrast to the MNIST dataset, images in SVHN are colored. We do not use any preprocessing in this dataset.

Encoder:

$$\begin{aligned}
x \in R^{3\times 32\times 32} \rightarrow Conv_{64} && \rightarrow BN \rightarrow ReLu \\
\rightarrow Conv_{128} && \rightarrow BN \rightarrow ReLu \\
\rightarrow Conv_{256} && \rightarrow BN \rightarrow ReLu \\
&& \rightarrow FC_{100} \rightarrow FC_{20}
\end{aligned}$$

Decoder:

$$\begin{aligned}
y \in R^{20} \rightarrow FC_{100} \rightarrow ConvT_{256} && \rightarrow BN \rightarrow ReLu \\
\rightarrow ConvT_{128} && \rightarrow BN \rightarrow ReLu \\
\rightarrow ConvT_{64} && \rightarrow BN \rightarrow ReLu \\
\rightarrow ConvT_{32} && \rightarrow BN \rightarrow ReLu \\
&& \rightarrow FC_{1}
\end{aligned}$$

Notations are the same as described above.

CelebA

In contrast to the MNIST dataset, images in CelebA are colored, Further, we first took central crops of 140 × 140 and resize the images to a resolution 64 × 64.

Encoder:

$$
\begin{aligned}
x \in R^{3\times 64\times 64} &\to Conv_{64} && \to BN \to LeakyReLu \\
&\to Conv_{128} && \to BN \to LeakyReLu \\
&\to Conv_{256} && \to BN \to LeakyReLu \\
&\to Conv_{512} && \to BN \to LeakyReLu \\
& && \to FC_{100} \to FC_{100}
\end{aligned}
$$

Decoder:

$$
\begin{aligned}
y \in R^{100} \to FC_{100} &\to Conv_{512} && \to BN \to ReLu \\
&\to ConvT_{256} && \to BN \to ReLu \\
&\to ConvT_{128} && \to BN \to ReLu \\
&\to ConvT_{64} && \to BN \to ReLu \\
&\to ConvT_{32} && \to BN \to ReLu \\
& && \to FC_{1}
\end{aligned}
$$

LeakyReLU uses a negative slope of 0.2, and padding was set to 0 for the last convolutional layer of the encoder and the first of the decoder. All other notations are the same as described above.

Implementation of Real NVP

In our study, we used a Real NVP (see Dinh et al. 2017) to model the latent space of the autoencoder and serve as a benchmark. For all data sets, we use a spatial checkerboard masking, where the mask has value of 1 if the sum of coordinates is odd, and 0 otherwise. For the MNIST data set, we use 4 coupling layers with 2 hidden layers each and 256 features per hidden layer. Similary for the SVHN data set we also use four coupling layers with two hidden layers each and 256 hidden layer features. Lastly, for the CelebA data set we use four coupling layers with two hidden layers each and 1024 hidden layer features. For all data sets, we applied a learning rate of 0.0001 and learn for 2000 epochs.

Additional Experiments

Numerical Assessments

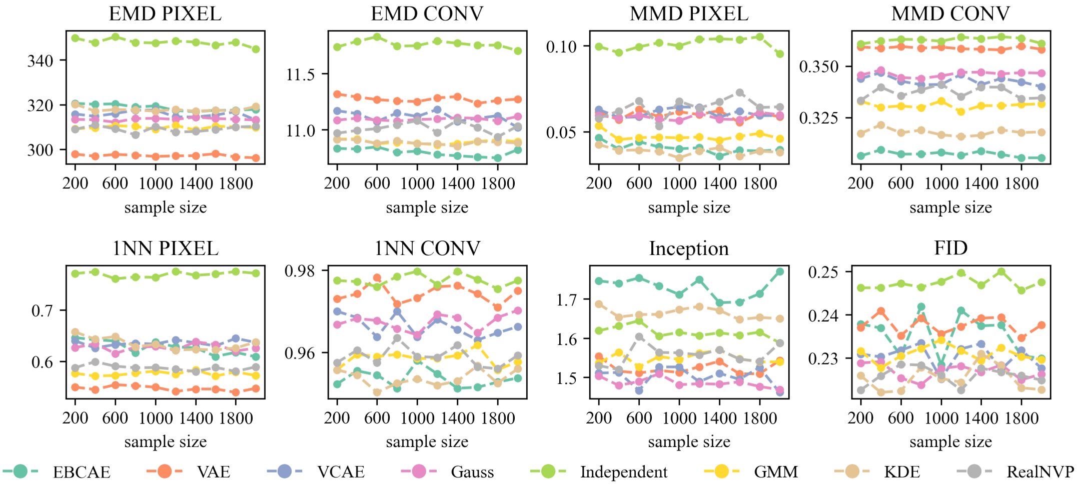


Figure D.1.: Performance metrics of generative models on **CelebA**, reported over latent space sample size. Note that they only differ in the latent space sampling and share the same autoencoder.

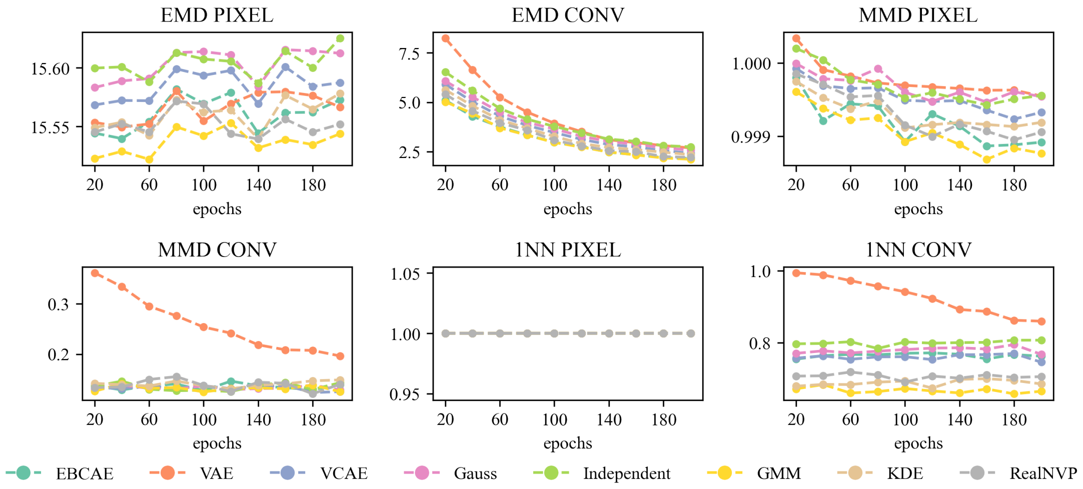


Figure D.2.: Performance metrics of generative models on **MNIST**, reported over epochs computed from 2000 random samples. Note that they only differ in the latent space sampling and share the same autoencoder.

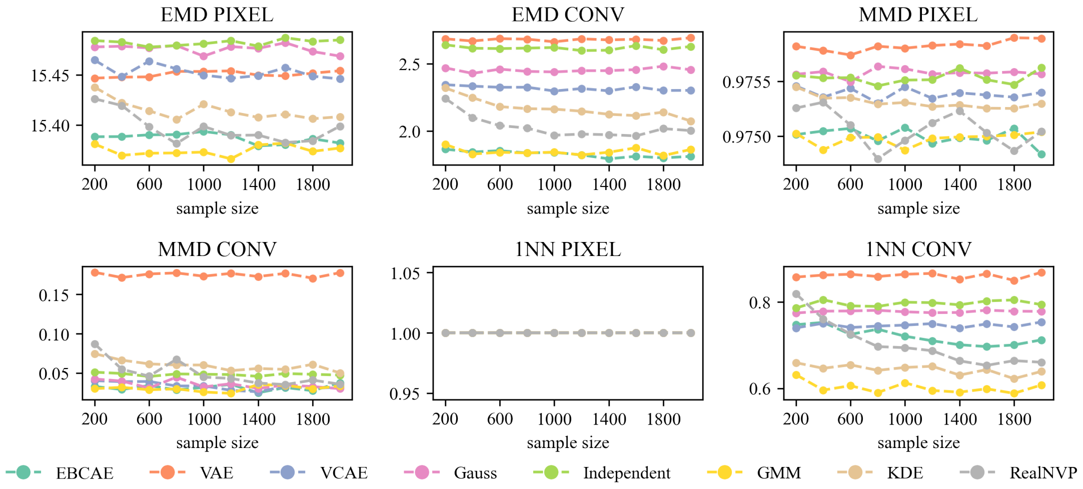


Figure D.3.: Performance metrics of generative models on **MNIST**, reported over latent space sample size. Note that they only differ in the latent space sampling and share the same autoencoder.

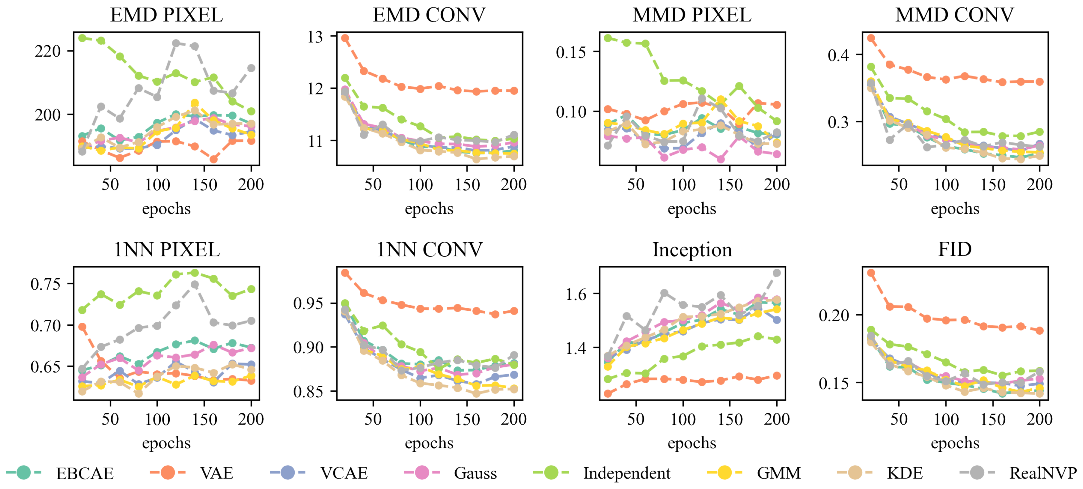


Figure D.4.: Performance metrics of generative models on **SVHN**, reported over epochs computed from 2000 random samples. Note that they only differ in the latent space sampling and share the same autoencoder.

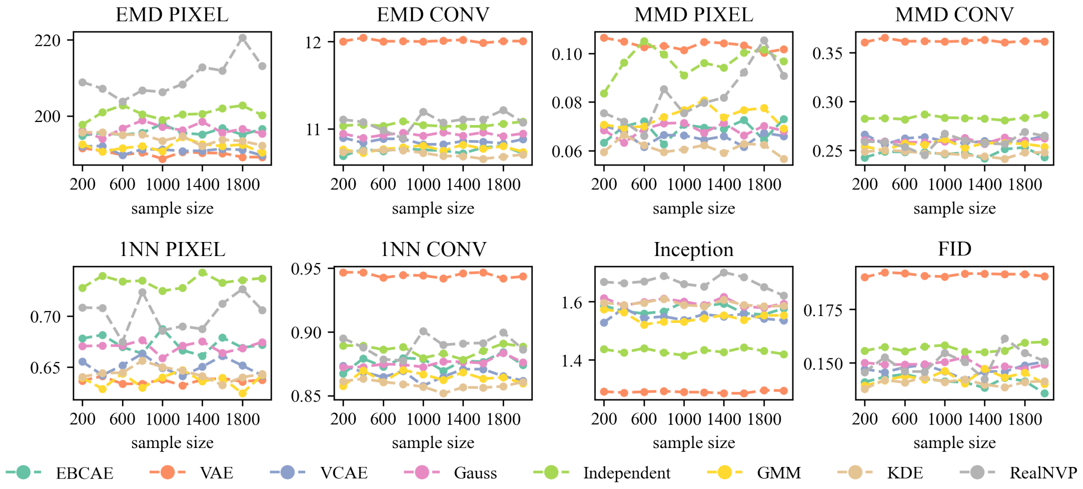


Figure D.5.: Performance metrics of generative models on **SVHN**, reported over latent space sample size. Note that they only differ in the latent space sampling and share the same autoencoder.

Samples from SVHN

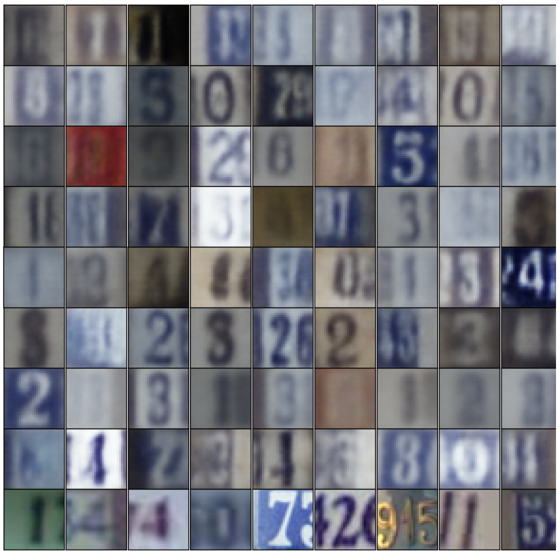


Figure D.6.: SVHN samples

Figure D.7.: Comparison of synthetic samples of different Autoencoder models. **1st row:** Fitted normal distribution, **2nd row:** Independent margins, **3rd row:** KDE-AE, **4th row:** GMM, **5th row:** VCAE, **6th row:** EBCAE, **7th row:** VAE, **8th row:** Real NVP, **Last row:** original pictures.

Image Interpolation of the Autoencoder

We show that our used autoencoder learned a relevant and smooth representation of the data by interpolation in the latent space and, thus, modeling the latent space is reasonable. For example, consider two images A and B with latent variables $y_{A,1}, ..., x_{A,100}$ and $y_{B,1}, ..., y_{B,100}$. We now interpolate linearly in each dimension between these two values and feed the resulting interpolation to the decoder to get the interpolated images.

Each row in Figure D.8 shows a clear linear progression in ten steps from the first face on the left to the final face on the right. For example, in the last row, we see a female

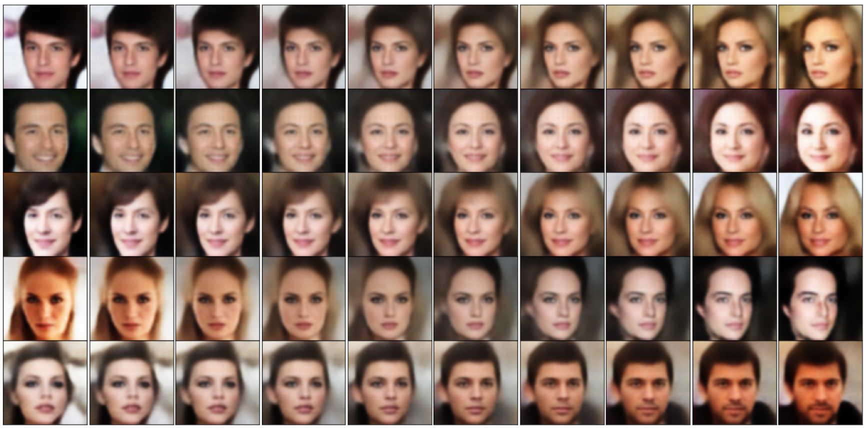


Figure D.8.: Interpolation in the latent space of samples of the autoencoder.

with blonde hair slowly transforming into a male with a beard. The transition is smooth, and no sharp changes or random images occur in-between.

Bibliography

Aaberge, R., A. B. Atkinson, and S. Königs (2018): "From classes to copulas: wages, capital, and top incomes," *The Journal of Economic Inequality*, 16, 295–320.

Aas, K. (2016): "Pair-Copula Constructions for Financial Applications: A Review," *Econometrics*, 4.

Allison, R. and J. E. Foster (2004): "Measuring health inequality using qualitative data," *Journal of Health Economics*, 23, 505–524.

Arnold, B. (1987): *Majorization and the Lorenz Order: A Brief Introduction*, Lecture Notes in Statistics, Springer-Verlag New York.

Arnold, B. C. and J. M. Sarabia (2018): "Analytic Expressions for Multivariate Lorenz Surfaces," *Sankhya A: The Indian Journal of Statistics*, 80, 84–111.

Arpinoa, G., H. Shihc, W. Bruinsmad, and E. P. Martins (2021): "Gaussian Processes for Probabilistic Electricity Price Forecasting," Preprint, `https://gabrielarpino.github.io/files/MEC_forecast-4.pdf`; accessed: 2022-03-22.

Atkinson, A. B. and F. Bourguignon (1982): "The Comparison of Multi-Dimensioned Distributions of Economic Status," *The Review of Economic Studies*, 49, 183–201.

Banerjee, A. K. (2014): "A multidimensional Lorenz dominance relation," *Social Choice and Welfare*, 42, 171–191.

Basili, M., P. Casaca, A. Chateauneuf, and M. Franzini (2017): “Multidimensional Pigou–Dalton transfers and social evaluation functions,” *Theory and Decision*, 83, 573–590.

Bedford, T. and R. M. Cooke (2002): “Vines: A New Graphical Model for Dependent Random Variables,” *The Annals of Statistics*, 30, 1031–1068.

Bell, G., T. Hey, and A. Szalay (2009): “Beyond the Data Deluge,” *Science*, 323, 1297–1298.

Berthelot, D., C. Raffel, A. Roy, and I. Goodfellow (2019): “Understanding and Improving Interpolation in Autoencoders via an Adversarial Regularizer,” in *International Conference on Learning Representations*, OpenReview.net.

Bibi, N., I. Shah, A. Alsubie, S. Ali, and S. A. Lone (2021): “Electricity Spot Prices Forecasting Based on Ensemble Learning,” *IEEE Access*, 9, 150984–150992.

Bishop, C. M. (2006): *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Berlin & Heidelberg: Springer-Verlag.

Bosmans, K., L. Lauwers, and E. Ooghe (2009): “A consistent multidimensional Pigou–Dalton transfer principle,” *Journal of Economic Theory*, 144, 1358–1371.

Brehmer, J. and K. Cranmer (2020): “Flows for simultaneous manifold learning and density estimation,” in *Advances in Neural Information Processing Systems*, ed. by H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Curran Associates, Inc., vol. 33, 442–453.

Brockwell, P. J. and R. A. Davis (2016): *Introduction to time series and forecasting*, Springer, 3 ed.

Bundesverband der Energie- und Wasserwirtschaft e.V. (2021): “Standardlastprofile Strom,” `https://www.bdew.de/energie/standardlastprofile-strom/`; accessed: 2021-04-15.

Bücher, A. and H. Dette (2010): "A note on bootstrap approximations for the empirical copula process," *Statistics & Probability Letters*, 80, 1925–1932.

Bücher, A. and S. Volgushev (2013): "Empirical and sequential empirical copula processes under serial dependence," *Journal of Multivariate Analysis*, 119, 61 – 70.

Cao, L. (2017): "Data Science: A Comprehensive Overview," *ACM Comput. Surv.*, 50.

Chai, S., Z. Xu, and Y. Jia (2019): "Conditional Density Forecast of Electricity Price Based on Ensemble ELM and Logistic EMOS," *IEEE Transactions on Smart Grid*, 10, 3031–3043.

Chen, Y. and B. Li (2017): "An Adaptive Functional Autoregressive Forecast Model to Predict Electricity Price Curves," *Journal of Business & Economic Statistics*, 35, 371–388.

Cherubini, U., E. Luciano, and W. Vecchiato (2004): *Copula Methods in Finance*, John Wiley & Sons, Ltd, 289–293.

Cherubini, U. and S. Romagnoli (2009): "Computing the Volume of n-Dimensional Copulas," *Applied Mathematical Finance*, 16, 307–314.

Clark, M., S. Gangopadhyay, L. Hay, B. Rajagopalan, and R. Wilby (2004): "The Schaake Shuffle: A Method for Reconstructing Space–Time Variability in Forecasted Precipitation and Temperature Fields," *Journal of Hydrometeorology*, 5, 243 – 262.

Clark, T. E., M. W. McCracken, and E. Mertens (2020): "Modeling time-varying uncertainty of multiple-horizon forecast errors," *Review of Economics and Statistics*, 102, 17–33.

Coblenz, M., S. Holz, H.-J. Bauer, O. Grothe, and R. Koch (2020): "Modelling fuel injector spray characteristics in jet engines by using vine copulas," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 69, 863–886.

CREMER, C., X. LI, AND D. DUVENAUD (2018): "Inference Suboptimality in Variational Autoencoders," in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, ed. by J. G. Dy and A. Krause, PMLR, vol. 80 of *Proceedings of Machine Learning Research*, 1086–1094.

CZADO, C. (2019): *Analyzing Dependent Data with Vine Copulas: A Practical Guide With R*, Springer International Publishing.

CZADO, C. AND T. NAGLER (2022): "Vine Copula Based Modeling," *Annual Review of Statistics and Its Application*, 9, 453–477.

DAI, B. AND D. P. WIPF (2019): "Diagnosing and Enhancing VAE Models," in *International Conference on Learning Representations*, OpenReview.net.

DALTON, H. (1920): "The Measurement of the Inequality of Incomes," *The Economic Journal*, 30, 348–361.

DAVID, H. A. AND H. N. NAGARAJA (2003): *Order Statistics*, John Wiley and Sons.

DE LA VEGA, C. L., A. URRUTIA, AND A. SARACHU (2010): "Characterizing multidimensional inequality measures which fulfill the Pigou–Dalton bundle principle," *Social Choice and Welfare*, 35, 319–329.

DECANCQ, K. (2012): "Elementary multivariate rearrangements and stochastic dominance on a Fréchet class," *Journal of Economic Theory*, 147, 1450–1459.

DEHEUVELS, P. (1979): "La fonction de dépendance empirique et ses propriétés. Un test non paramétrique d'indépendance," *Bulletins de l'Académie Royale de Belgique*, 65, 274–292.

DHAR, V. (2013): "Data science and prediction," *Communications of the ACM*, 56, 64–73.

DIEBOLD, F. X. AND R. S. MARIANO (1995): "Comparing Predictive Accuracy," *Journal of Business & Economic Statistics*, 13, 253–263.

DIMITRIEV, A. AND M. ZHOU (2021): "CARMS: Categorical-Antithetic-REINFORCE Multi-Sample Gradient Estimator," in *Advances in Neural Information Processing Systems*, ed. by A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan.

DINH, L., J. SOHL-DICKSTEIN, AND S. BENGIO (2017): "Density estimation using Real NVP," in *International Conference on Learning Representations*, OpenReview.net.

DISTRIBUTION DATABASE, O. I. (2017): "OECD Project on Distribution of Household Incomes," `https://www.oecd.org/social/income-distribution-database.htm`; accessed: 2023-03-22.

DONOHO, D. (2017): "50 Years of Data Science," *Journal of Computational and Graphical Statistics*, 26, 745–766.

DURANTE, F. AND C. SEMPI (2015): *Principles of copula theory*, CRC Press LLC.

EPSTEIN, L. G. AND S. TANNY (1980): "Increasing Generalized Correlation: A Definition and Some Economic Consequences," *Canadian Journal of Economics*, 13, 16–34.

FAJTL, J., V. ARGYRIOU, D. MONEKOSSO, AND P. REMAGNINO (2020): "Latent Bernoulli Autoencoder," in *Proceedings of the 37th International Conference on Machine Learning*, ed. by H. D. III and A. Singh, PMLR, vol. 119 of *Proceedings of Machine Learning Research*, 2964–2974.

FAURE, M. AND N. GRAVEL (2021): "REDUCING INEQUALITIES AMONG UNEQUALS," *International Economic Review*, 62, 357–404.

FERMANIAN, J.-D., D. RADULOVIC, AND M. WEGKAMP (2004): "Weak convergence of empirical copula processes," *Bernoulli*, 10, 847–860.

FLEURBAEY, M. AND A. TRANNOY (2003): "The impossibility of a Paretian egalitarian," *Social Choice and Welfare*, 21, 243–263.

FORMBY, J. P., T. G. SEAKS, AND W. J. SMITH (1989): "On the Measurement and Trend of Inequality: A Reconsideration," *The American Economic Review*, 79, 256–264.

GAENSSLER, P. AND W. STUTE (1987): *Copula processes*, Basel: Birkhäuser Basel, 49–57.

GAJDOS, T. AND J. WEYMARK (2005): "Multidimensional generalized Gini indices," *Economic Theory*, 26, 471–496.

GARCIA, R., J. CONTRERAS, M. VAN AKKEREN, AND J. GARCIA (2005): "A GARCH forecasting model to predict day-ahead electricity prices," *IEEE Transactions on Power Systems*, 20, 867–874.

GASTWIRTH, J. L. (1971): "A General Definition of the Lorenz Curve," *Econometrica*, 39, 1037–39.

——— (1972): "The Estimation of the Lorenz Curve and Gini Index," *The Review of Economics and Statistics*, 54, 306–316.

GENEST, C. AND A.-C. FAVRE (2007a): "Everything You Always Wanted to Know about Copula Modeling but Were Afraid to Ask," *Journal of Hydrologic Engineering*, 12, 347–368.

——— (2007b): "Everything You Always Wanted to Know about Copula Modeling but Were Afraid to Ask," *Journal of Hydrologic Engineering*, 12, 347–368.

GENEST, C., M. GENDRON, AND M. BOURDEAU-BRIEN (2009): "The Advent of Copulas in Finance," *The European Journal of Finance*, 15, 609–618.

GENEST, C., K. GHOUDI, AND L.-P. RIVEST (1995): "A Semiparametric Estimation Procedure of Dependence Parameters in Multivariate Families of Distributions," *Biometrika*, 82, 543–552.

GENEST, C. AND J. SEGERS (2009): "On the covariance of the asymptotic empirical copula process," *J. Multivar. Anal.*, 101, 1837–1845.

Ghalanos, A. (2020): *rugarch: Univariate GARCH models.*, r package version 1.4-4.

Ghosh, P., M. S. M. Sajjadi, A. Vergari, M. J. Black, and B. Schölkopf (2020): "From Variational to Deterministic Autoencoders," in *International Conference on Learning Representations*, OpenReview.net.

Gneiting, T., F. Balabdaoui, and A. E. Raftery (2007): "Probabilistic forecasts, calibration and sharpness," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69, 243–268.

Gneiting, T. and M. Katzfuss (2014): "Probabilistic Forecasting," *Annual Review of Statistics and Its Application*, 1, 125–151.

Gneiting, T. and A. E. Raftery (2007): "Strictly Proper Scoring Rules, Prediction, and Estimation," *Journal of the American Statistical Association*, 102, 359–378.

Gneiting, T., L. I. Stanberry, E. P. Grimit, L. Held, and N. A. Johnson (2008): "Assessing probabilistic forecasts of multivariate quantities, with an application to ensemble predictions of surface winds," *TEST*, 17, 211–235.

Goebel, J., M. M. Grabka, S. Liebig, M. Kroh, D. Richter, C. Schröder, and J. Schupp (2019): "The German Socio-Economic Panel (SOEP)," *Jahrbücher für Nationalökonomie und Statistik*, 239, 345 – 360.

Goodfellow, I., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio (2014): "Generative Adversarial Nets," in *Advances in Neural Information Processing Systems*, ed. by Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, Curran Associates, Inc., vol. 27.

Gravel, N., B. Magdalou, and P. Moyes (2021): "Ranking distributions of an ordinal variable," *Economic Theory*, 71.

Gravel, N. and P. Moyes (2012): "Ethically robust comparisons of bidimensional distributions with an ordinal attribute," *Journal of Economic Theory*, 147, 1384–1426.

Gregor, K., I. Danihelka, A. Graves, D. Rezende, and D. Wierstra (2015): "DRAW: A Recurrent Neural Network For Image Generation," in *Proceedings of the 32nd International Conference on Machine Learning*, ed. by F. Bach and D. Blei, Lille, France: PMLR, vol. 37 of *Proceedings of Machine Learning Research*, 1462–1471.

Gretton, A., K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola (2007): "A Kernel Method for the Two-Sample-Problem," in *Advances in Neural Information Processing Systems*, ed. by B. Schölkopf, J. Platt, and T. Hoffman, MIT Press, vol. 19.

Grothe, O., F. Kächele, and F. Krüger (2023): "From point forecasts to multivariate probabilistic forecasts: The Schaake shuffle for day-ahead electricity price forecasting," *Energy Economics*, 106602.

Grothe, O., F. Kächele, and F. Schmid (2022a): "A multivariate extension of the Lorenz curve based on copulas and a related multivariate Gini coefficient," *The Journal of Economic Inequality*, 20, 727–748.

Grothe, O., F. Kächele, and M. Watermeyer (2022b): "Analyzing Europe's Biggest Offshore Wind Farms: A Data Set with 40 Years of Hourly Wind Speeds and Electricity Production," *Energies*, 15.

Harvey, L. A., J. O. Mierau, and J. Rockey (2017): "Inequality in an Equal Society," LWS Working papers 26, LIS Cross-National Data Center in Luxembourg.

Hastie, T., R. Tibshirani, and J. Friedman (2001): *The Elements of Statistical Learning*, Springer Series in Statistics, New York, NY, USA: Springer New York Inc.

Havtorn, J. D. D., J. Frellsen, S. Hauberg, and L. Maaløe (2021): "Hierarchical VAEs Know What They Don't Know," in *Proceedings of the 38th International Conference on Machine Learning*, ed. by M. Meila and T. Zhang, PMLR, vol. 139 of *Proceedings of Machine Learning Research*, 4117–4128.

He, Y., J. Deng, and H. Li (2017): "Short-Term Power Load Forecasting with Deep Belief Network and Copula Models," in *2017 9th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, vol. 1, 191–194.

Heusel, M., H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter (2017): "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium," in *Advances in Neural Information Processing Systems*, ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Curran Associates, Inc., vol. 30.

Hofert, M. (2008): "Sampling Archimedean copulas," *Computational Statistics & Data Analysis*, 52, 5163–5174.

Hofert, M., M. Mächler, and A. J. McNeil (2012): "Likelihood inference for Archimedean copulas in high dimensions under known margins," *Journal of Multivariate Analysis*, 110, 133–150, special Issue on Copula Modeling and Dependence.

Hofert, M., A. Prasad, and M. Zhu (2021): "Quasi-Random Sampling for Multivariate Distributions via Generative Neural Networks," *Journal of Computational and Graphical Statistics*, 30, 647–670.

Hong, T., P. Pinson, Y. Wang, R. Weron, D. Yang, and H. Zareipour (2020): "Energy Forecasting: A Review and Outlook," *IEEE Open Access Journal of Power and Energy*, 7, 376–388.

Hubicka, K., G. Marcjasz, and R. Weron (2019): "A Note on Averaging Day-Ahead Electricity Price Forecasts Across Calibration Windows," *IEEE Transactions on Sustainable Energy*, 10, 321–323.

Hudson, D. A. and L. Zitnick (2021): "Generative Adversarial Transformers," in *Proceedings of the 38th International Conference on Machine Learning*, ed. by M. Meila and T. Zhang, PMLR, vol. 139 of *Proceedings of Machine Learning Research*, 4487–4499.

Hyndman, R. and G. Athanasopoulos (2021): *Forecasting: principles and practice*, OTexts: Melbourne, Australia, 3 ed., `OTexts.com/fpp3`; accessed: 2021-12-12.

Jan, F., I. Shah, and S. Ali (2022): "Short-Term Electricity Prices Forecasting Using Functional Time Series Analysis," *Energies*, 15.

Janke, T., M. Ghanmi, and F. Steinke (2021): "Implicit Generative Copulas," in *Thirty-Fifth Conference on Neural Information Processing Systems.*

Janke, T. and F. Steinke (2020): "Probabilistic multivariate electricity price forecasting using implicit generative ensemble post-processing," *2020 International Conference on Probabilistic Methods Applied to Power Systems (PMAPS).*

Joe, H. (2005): "Asymptotic efficiency of the two-stage estimation method for copula-based models," *Journal of Multivariate Analysis*, 94, 401–419.

——— (2014): *Dependence modeling with copulas*, Chapman & Hall/CRC.

Kakwani, N. C. (1977): "Applications of Lorenz Curves in Economic Analysis," *Econometrica*, 45, 719–727.

Karras, T., M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, and T. Aila (2021): "Alias-Free Generative Adversarial Networks," in *Thirty-Fifth Conference on Neural Information Processing Systems.*

Kath, C. and F. Ziel (2021): "Conformal prediction interval estimation and applications to day-ahead and intraday power markets," *International Journal of Forecasting*, 37, 777–799.

Kingma, D. P., T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling (2016): "Improved Variational Inference with Inverse Autoregressive Flow," in *Advances in Neural Information Processing Systems*, ed. by D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Curran Associates, Inc., vol. 29.

KINGMA, D. P. AND M. WELLING (2014): "Auto-Encoding Variational Bayes," in *International Conference on Learning Representations*, OpenReview.net.

KIRILIOUK, A., J. SEGERS, AND H. TSUKAHARA (2021): *Resampling Procedures with Empirical Beta Copulas*, Singapore: Springer Singapore, 27–53.

KOBUS, M. AND P. MIŁOŚ (2012): "Inequality decomposition by population subgroups for ordinal data," *Journal of Health Economics*, 31, 15–21.

KOENKER, R., P. NG, AND S. PORTNOY (1994): "Quantile smoothing splines," *Biometrika*, 81, 673–680.

KOSHEVOY, G. AND K. MOSLER (1996): "The Lorenz Zonoid of a Multivariate Distribution," *Journal of the American Statistical Association*, 91, 873–882.

——— (1997): "Multivariate Gini Indices," *Journal of Multivariate Analysis*, 60, 252 – 276.

KOU, P., D. LIANG, L. GAO, AND J. LOU (2015): "Probabilistic electricity price forecasting with variational heteroscedastic Gaussian process and active learning," *Energy Conversion and Management*, 89, 298–308.

KULKARNI, V., N. TAGASOVSKA, T. VATTER, AND B. GARBINATO (2018): "Generative Models for Simulating Mobility Trajectories," *ArXiv*.

LAGO, J., G. MARCJASZ, B. DE SCHUTTER, AND R. WERON (2021): "Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark," *Applied Energy*, 293, 116983.

LARSEN, A. B. L., S. K. SØNDERBY, H. LAROCHELLE, AND O. WINTHER (2016): "Autoencoding beyond pixels using a learned similarity metric," in *Proceedings of The 33rd International Conference on Machine Learning*, ed. by M. F. Balcan and K. Q. Weinberger, New York, New York, USA: PMLR, vol. 48 of *Proceedings of Machine Learning Research*, 1558–1566.

LeCun, Y., C. Cortes, and C. Burges (2010): “MNIST handwritten digit database,” *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2.

Lee, J., H. Kim, Y. Hong, and H. W. Chung (2021): “Self-Diagnosing GAN: Diagnosing Underrepresented Samples in Generative Adversarial Networks,” in *Thirty-Fifth Conference on Neural Information Processing Systems.*

Letizia, N. A. and A. M. Tonello (2020): “Segmented Generative Networks: Data Generation in the Uniform Probability Space,” *IEEE Transactions on Neural Networks and Learning Systems*, 1–10.

Li, Y. and J. Zhu (2008): “L1-Norm Quantile Regression,” *Journal of Computational and Graphical Statistics*, 17, 163–185.

LIS (2019a): *METIS Metadata Information System*, `http://www.lisdatacenter.org/frontend#/home`; accessed 2020-10-01.

——— (2019b): *THE LWS USER GUIDE 2019 Template*, `http://www.lisdatacenter.org/wp-content/uploads/files/data-lws-guide.pdf`; accessed: 2019-10-01.

Lisi, F. and I. Shah (2020): “Forecasting next-day electricity demand and prices based on functional models,” *Energy Systems*, 11.

Liu, W. (2019): “Copula Multi-label Learning,” in *Advances in Neural Information Processing Systems*, ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Curran Associates, Inc., vol. 32.

Liu, Z., P. Luo, X. Wang, and X. Tang (2015): “Deep Learning Face Attributes in the Wild,” in *Proceedings of International Conference on Computer Vision (ICCV).*

Lopez-Paz, D. and M. Oquab (2017): “Revisiting Classifier Two-Sample Tests,” in *International Conference on Learning Representations*, OpenReview.net.

Lorenz, M. O. (1905): “Methods of Measuring the Concentration of Wealth,” *Publications of the American Statistical Association*, 9, 209–219.

LUDWIG, N., S. ARORA, AND J. W. TAYLOR (2022): "Probabilistic load forecasting using post-processed weather ensemble predictions," *Journal of the Operational Research Society*, 0, 1–13.

Luxembourg Wealth Study (LWS) Database (2020): Multiple countries; 2012-2017), Luxembourg: LIS.

LÜTKEPOHL, H. AND M. KRÄTZIG (2004): *Applied Time Series Econometrics*, Themes in Modern Econometrics, Cambridge University Press.

MA, J., B. CHANG, X. ZHANG, AND Q. MEI (2021): "CopulaGNN: Towards Integrating Representational and Correlational Roles of Graphs in Graph Neural Networks," in *International Conference on Learning Representations*, OpenReview.net.

MACIEJOWSKA, K., W. NITKA, AND T. WERON (2021): "Enhancing load, wind and solar generation for day-ahead forecasting of electricity prices," *Energy Economics*, 99, 105273.

MAKHZANI, A., J. SHLENS, N. JAITLY, AND I. GOODFELLOW (2016): "Adversarial Autoencoders," in *International Conference on Learning Representations*, OpenReview.net.

MANNER, H., D. TÜRK, AND M. EICHLER (2016): "Modeling and forecasting multivariate electricity price spikes," *Energy Economics*, 60, 255 – 265.

MARCJASZ, G., T. SERAFIN, AND R. WERON (2018): "Selection of Calibration Windows for Day-Ahead Electricity Price Forecasting," *Energies*, 11.

MARCJASZ, G., B. UNIEJEWSKI, AND R. WERON (2020): "Probabilistic electricity price forecasting with NARX networks: Combine point or probabilistic forecasts?" *International Journal of Forecasting*, 36, 466–479.

MARINO, J., Y. YUE, AND S. MANDT (2018): "Iterative Amortized Inference," in *Proceedings of the 35th International Conference on Machine Learning*, ed. by J. Dy and A. Krause, PMLR, vol. 80 of *Proceedings of Machine Learning Research*, 3403–3412.

MASRANI, V., T. A. LE, AND F. WOOD (2019): "The Thermodynamic Variational Objective," in *Advances in Neural Information Processing Systems*, ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Curran Associates, Inc., vol. 32.

MESSOUDI, S., S. DESTERCKE, AND S. ROUSSEAU (2021): "Copula-based conformal prediction for multi-target regression," *Pattern Recognit.*, 120, 108101.

MISHNE, G., U. SHAHAM, A. CLONINGER, AND I. COHEN (2019): "Diffusion nets," *Applied and Computational Harmonic Analysis*, 47, 259–285.

MOOR, M., M. HORN, B. RIECK, AND K. BORGWARDT (2020): "Topological Autoencoders," in *Proceedings of the 37th International Conference on Machine Learning*, ed. by H. D. III and A. Singh, PMLR, vol. 119 of *Proceedings of Machine Learning Research*, 7045–7054.

MUNIAIN, P. AND F. ZIEL (2020): "Probabilistic forecasting in day-ahead electricity markets: Simulating peak and off-peak prices," *International Journal of Forecasting*, 36, 1193–1210.

NELSEN, R. B. (2006): *An Introduction to Copulas*, Springer Science+Business Media, Inc.

NETZER, Y., T. WANG, A. COATES, A. BISSACCO, B. WU, AND A. Y. NG (2011): "Reading Digits in Natural Images with Unsupervised Feature Learning," in *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011.*

NOWOTARSKI, J. AND R. WERON (2018): "Recent advances in electricity price forecasting: A review of probabilistic forecasting," *Renewable and Sustainable Energy Reviews*, 81, 1548 – 1568.

ORING, A., Z. YAKHINI, AND Y. HEL-OR (2021): "Autoencoder Image Interpolation by Shaping the Latent Space," in *Proceedings of the 38th International Conference on*

Machine Learning, ed. by M. Meila and T. Zhang, PMLR, vol. 139 of *Proceedings of Machine Learning Research*, 8281–8290.

Panagiotelis, A. and M. Smith (2008): "Bayesian density forecasting of intraday electricity prices using multivariate skew t distributions," *International Journal of Forecasting*, 24, 710–727.

Patton, A. J. (2012): "A review of copula models for economic time series," *Journal of Multivariate Analysis*, 110, 4–18, Special Issue on Copula Modeling and Dependence.

Pesaran, M. H. and M. Weale (2006): "Survey expectations," in *Handbook of Economic Forecasting*, Elsevier, vol. 1, 715–776.

Phipps, K., S. Lerch, M. Andersson, R. Mikut, V. Hagenmeyer, and N. Ludwig (2022): "Evaluating ensemble post-processing for wind power forecasts," *Wind Energy*.

Phipps, K., N. Ludwig, V. Hagenmeyer, and R. Mikut (2020): "Potential of Ensemble Copula Coupling for wind power forecasting," in *Proceedings-30. Workshop Computational Intelligence: Berlin, 26.-27. November 2020*, KIT Scientific Publishing, 87–109.

Pinson, P. and R. Girard (2012): "Evaluating the quality of scenarios of short-term wind power generation," *Applied Energy*, 96, 12 – 20.

Pircalabu, A. and F. Benth (2017): "A regime-switching copula approach to modeling day-ahead prices in coupled electricity markets," *Energy Economics*, 68, 283 – 302.

Quan, H., D. Srinivasan, and A. Khosravi (2014): "Uncertainty handling using neural network-based prediction intervals for electrical load forecasting," *Energy*, 73, 916–925.

R Core Team (2013): *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria.

Rehm, M., K. D. Schmid, and D. Wang (2014): “Why has Inequality in Germany not Risen Further After 2005?” IMK Working Paper 137-2014, IMK at the Hans Boeckler Foundation, Macroeconomic Policy Institute.

Rezende, D. and S. Mohamed (2015): “Variational Inference with Normalizing Flows,” in *Proceedings of the 32nd International Conference on Machine Learning*, ed. by F. Bach and D. Blei, Lille, France: PMLR, vol. 37 of *Proceedings of Machine Learning Research*, 1530–1538.

Rosenblatt, M. (1952): “Remarks on a Multivariate Transformation,” *The Annals of Mathematical Statistics*, 23, 470 – 472.

Ruschendorf, L. (1976): “Asymptotic Distributions of Multivariate Rank Order Statistics,” *Ann. Statist.*, 4, 912–923.

Rémillard, B. and O. Scaillet (2009): “Testing for equality between two copulas,” *Journal of Multivariate Analysis*, 100, 377 – 386.

Saez, E. and G. Zucman (2016): “Wealth Inequality in the United States since 1913: Evidence from Capitalized Income Tax Data,” *The Quarterly Journal of Economics*, 131, qjw004.

Salimans, T., I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, and X. Chen (2016): “Improved Techniques for Training GANs,” in *Advances in Neural Information Processing Systems*, ed. by D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Curran Associates, Inc., vol. 29.

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

Schefzik, R., T. L. Thorarinsdottir, and T. Gneiting (2013): “Uncertainty Quantification in Complex Simulation Models Using Ensemble Copula Coupling,” *Statistical Science*, 28, 616–640.

SEGERS, J. (2012): “Asymptotics of empirical copula processes under non-restrictive smoothness assumptions,” *Bernoulli*, 18, 764 – 782.

SEGERS, J., M. SIBUYA, AND H. TSUKAHARA (2017): “The empirical beta copula,” *Journal of Multivariate Analysis*, 155, 35–51.

SHAH, I., S. AKBAR, T. SABA, S. ALI, AND A. REHMAN (2021): “Short-Term Forecasting for the Electricity Spot Prices With Extreme Values Treatment,” *IEEE Access*, 9, 105451–105462.

SHAH, I., H. BIBI, S. ALI, L. WANG, AND Z. YUE (2020): “Forecasting One-Day-Ahead Electricity Prices for Italian Electricity Market Using Parametric and Nonparametric Approaches,” *IEEE Access*, 8, 123104–123113.

SHAH, I. AND F. LISI (2015): “Day-ahead electricity demand forecasting with nonparametric functional models,” in *12th International Conference on the European Energy Market (EEM)*, 1–5.

SHEN, T., J. MUELLER, R. BARZILAY, AND T. S. JAAKKOLA (2020): “Educating Text Autoencoders: Latent Representation Guidance via Denoising,” in *Proceedings of the 37th International Conference on Machine Learning*, ed. by H. D. III and A. Singh, PMLR, Proceedings of Machine Learning Research, 8719–8729.

SIERMINSKA, E. AND T. SMEEDING (2005): “Measurement Issues: Equivalence Scales, Accounting Framework, and Reference Unit,” *prepared for the LWS Workshop: “Construction and Usage of Comparable Microdata on Wealth: the LWS”*.

SILVERMAN, B. W. (1986): *Density estimation for statistics and data analysis*, Chapman and Hall London.

SIMPSON, D. P. (1977): *Cassell’s Latin Dictionary : Latin-English, English-Latin.*, SLondon : Cassell ; New York : Macmillan.

SKLAR, A. (1959): “Fonctions de répartition à n dimensions et leurs marges,” *Publications de l’Institut de Statistique de l’Université de Paris*, 8, 229–231.

SOEP (2019): *Socio-Economic Panel (SOEP), data for years 1984-2017, version 34.*

Sohn, K., H. Lee, and X. Yan (2015): "Learning Structured Output Representation using Deep Conditional Generative Models," in *Advances in Neural Information Processing Systems*, ed. by C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Curran Associates, Inc., vol. 28.

Stute, W. (1984): "The Oscillation Behavior of Empirical Processes: The Multivariate Case," *The Annals of Probability*, 12, 361 – 379.

Tagasovska, N., D. Ackerer, and T. Vatter (2019): "Copulas as High-Dimensional Generative Models: Vine Copula Autoencoders," in *Advances in Neural Information Processing Systems*, ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Curran Associates, Inc., vol. 32.

Taguchi, T. (1972a): "n the two-dimensional concentration surface and extensions of concentration coefficient and pareto distribution to the two dimensional case—I," *Annals of the Institute of Statistical Mathematics*, 24, 355–381.

——— (1972b): "On the two-dimensional concentration surface and extensions of concentration coefficient and pareto distribution to the two dimensional case—II," *Annals of the Institute of Statistical Mathematics*, 24, 599–612.

Takahashi, H., T. Iwata, Y. Yamanaka, M. Yamada, and S. Yagi (2019): "Variational Autoencoder with Implicit Optimal Priors," AAAI Press, AAAI'19/IAAI'19/EAAI'19.

Theis, L., A. van den Oord, and M. Bethge (2016): "A note on the evaluation of generative models," in *International Conference on Learning Representations*, OpenReview.net.

Thorarinsdottir, T. L., M. Scheuerer, and C. Heinz (2016): "Assessing the Calibration of High-Dimensional Ensemble Forecasts Using Rank Histograms," *Journal of Computational and Graphical Statistics*, 25, 105–122.

TOMCZAK, J. AND M. WELLING (2018): "VAE with a VampPrior," in *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, ed. by A. Storkey and F. Perez-Cruz, PMLR, vol. 84 of *Proceedings of Machine Learning Research*, 1214–1223.

TOUBEAU, J.-F., J. BOTTIEAU, F. VALLÉE, AND Z. DE GRÈVE (2019): "Deep Learning-Based Multivariate Probabilistic Forecasting for Short-Term Scheduling in Power Markets," *IEEE Transactions on Power Systems*, 34, 1203–1215.

TRAN, D., D. BLEI, AND E. M. AIROLDI (2015): "Copula variational inference," in *Advances in Neural Information Processing Systems*, ed. by C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Curran Associates, Inc., vol. 28.

TSUI, K.-Y. (1998): "Multidimensional inequality and multidimensional generalized entropy measures: An axiomatic derivation," *Social Choice and Welfare*, 16, 145–157.

TSUKAHARA, H. (2005a): "Semiparametric estimation in copula models," *Canadian Journal of Statistics*, 33, 357–375.

——— (2005b): "Semiparametric Estimation in Copula Models," *The Canadian Journal of Statistics / La Revue Canadienne de Statistique*, 33, 357–375.

VALLENDER, S. S. (1974): "Calculation of the Wasserstein Distance Between Probability Distributions on the Line," *Theory of Probability and Its Applications*, 18, 435–435.

VAN DER MAATEN, L. AND G. HINTON (2008): "Visualizing Data using t-SNE," *Journal of Machine Learning Research*, 9, 2579–2605.

VAN ROSSUM, G. AND F. L. DRAKE (2009): *Python 3 Reference Manual*, Scotts Valley, CA: CreateSpace.

VANNITSEM, S., J. B. BREMNES, J. DEMAEYER, G. R. EVANS, J. FLOWERDEW, S. HEMRI, S. LERCH, N. ROBERTS, S. THEIS, A. ATENCIA, ET AL. (2021): "Statistical Postprocessing for Weather Forecasts: Review, Challenges, and Avenues in a Big Data World," *Bulletin of the American Meteorological Society*, 102, E681–E699.

Varshney, S., V. K. Verma, S. P. K, L. Carin, and P. Rai (2021): "CAM-GAN: Continual Adaptation Modules for Generative Adversarial Networks," in *Thirty-Fifth Conference on Neural Information Processing Systems.*

Wagner, G. G., J. R. Frick, and J. Schupp (2007): "The German Socio-Economic Panel Study (SOEP): Scope, Evolution and Enhancements," SOEPpapers on Multidisciplinary Panel Data Research 1, DIW Berlin, The German Socio-Economic Panel (SOEP).

Weron, R. (2006): *Modeling and Forecasting Electricity Loads and Prices*, John Wiley & Sons, Ltd.

Weron, R. and A. Misiorek (2008): "Forecasting spot electricity prices: A comparison of parametric and semiparametric time series models," *International Journal of Forecasting*, 24, 744–763.

Xu, H., W. Chen, J. Lai, Z. Li, Y. Zhao, and D. Pei (2019): "On the Necessity and Effectiveness of Learning the Prior of Variational Auto-Encoder," *ArXiv.*

Xu, Q., G. Huang, Y. Yuan, C. Guo, Y. Sun, F. Wu, and K. Q. Weinberger (2018): "An empirical study on evaluation metrics of generative adversarial networks," *ArXiv.*

Yoon, S., Y.-K. Noh, and F. Park (2021): "Autoencoding Under Normalization Constraints," in *Proceedings of the 38th International Conference on Machine Learning*, ed. by M. Meila and T. Zhang, PMLR, vol. 139 of *Proceedings of Machine Learning Research*, 12087–12097.

Zhang, Z., R. Zhang, Z. Li, Y. Bengio, and L. Paull (2020): "Perceptual Generative Autoencoders," in *Proceedings of the 37th International Conference on Machine Learning*, ed. by H. D. III and A. Singh, PMLR, vol. 119 of *Proceedings of Machine Learning Research*, 11298–11306.

ZIEL, F. AND R. STEINERT (2018): "Probabilistic mid- and long-term electricity price forecasting," *Renewable and Sustainable Energy Reviews*, 94, 251–266.

ZIEL, F. AND R. WERON (2018): "Day-ahead electricity price forecasting with high-dimensional structures: Univariate vs. multivariate modeling frameworks," *Energy Economics*, 70, 396 – 420.