Generative Machine Learning Methods for Multivariate Probabilistic Forecasting

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. Jieyu Chen

Tag der mündlichen Prüfung: 18. März 2025

Referent: Prof. Dr. Sebastian Lerch
Korreferentin: Prof. Dr. Melanie Schienle

Karlsruhe 2025

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my PhD supervisor, Dr. Sebastian Lerch. His immense support, guidance, expertise, and encouragement have been invaluable, shaping both my research and personal growth. I deeply appreciate the time and effort he dedicated to mentoring me.

I am also sincerely grateful to my Master's supervisors, Dr. Kirien Whan and Dr. Kate Saunders, who guided my path into scientific research and laid the foundation for my PhD journey. I also wish to acknowledge Prof. Dr. Jason Frank for supervising my Master's thesis, and Dr. Maurice Schmeits and colleagues at KNMI for their support during my internship, where I gained valuable experience.

I extend my gratitude to my colleagues at the Chair of Statistical Methods and Econometrics at KIT for their collaboration, camaraderie, and stimulating discussions. Special thanks to Prof. Dr. Melanie Schienle for fostering an exceptional research environment and for her support during my time at the chair. I am also thankful to Nina Horat for being a wonderful officemate, and to Dr. Benedikt Schulz and other members of Sebastian's YIG group for their insightful discussions.

I am grateful to my co-authors, Dr. Kevin Höhlein, Dr. Tomasz Serafin, Prof. Dr. Rafał Weron, Dr. Tim Janke, and Prof. Dr. Florian Steinke, for the constructive collaborations and shared insights.

Finally, I am profoundly thankful to my parents for their unconditional love and support throughout my academic journey. I also thank my other family members, especially my grandparents, for their understanding and encouragement. I am grateful to my friends, especially Team 312 Arbor Day, for their companionship. Lastly, special thanks to the mobile game Light and Night, whose five characters accompanied me and brought comfort during my challenging times.

Contents

List of Figures

List of Tables

List of Abbreviations

AE Autoencoder
CGM Conditional generative model
CRPS Continuous ranked probability score
DA Day-ahead
DM test Diebold-Mariano test
DRN Distributional regression network
DSS Dawid-Sebastiani score
ECC Ensemble copula coupling
ECMWF European Center of Medium-range Weather Forecasts
ED Energy distance
ELU Exponential linear unit
EMOS Ensemble model output statistics
EPF Electricity price forecasting
ES Energy score
GAN Generative adversarial network
GCA Gaussian copula approach
GMMN Generative moment matching network
ID Intraday
iVAE Invariant variational autoencoder
LASSO Least absolute shrinkage and selection operator
LLE Locally Linear Embedding
MAE Mean absolute error
ML Machine learning
MMD Maximum mean discrepancy
MSE Mean square error
NN Neural network
PCA Principal component analysis
QR Quantile regression
RES Renewable energy sources
RTP Realized trading potential
SCP Simultaneous coverage probability
SD Sinkhorn distance
TSO Transmission system operator
VAE Variational autoencoder

ViT Vision transformer
VS Variogram score
VWAP Volume weighted average price
WD Wasserstein distance

1 Introduction

Making predictions for the future has long been an important topic for humankind. Over past decades, the methods for predicting future quantities have evolved considerably, shifting from point forecasts, which estimate a single-value outcome, to probabilistic forecasts that represent outcomes in the form of distributions (Gneiting and Katzfuss, 2014). Probabilistic forecasting has become preferred over point predictions because it incorporates uncertainty quantifications, which provide valuable and critical information especially for decision-making. In many real-world applications, the events to be predicted often involve multiple univariate elements, such as combinations of multiple variables, time trajectories, spatial locations, or any selection of these combinations. In such cases, forecasts should not only convey uncertainty information but also capture the dependence structures between these variables. This necessitates multivariate forecasting, where understanding and modeling the multivariate dependence structure – whether cross-variable, spatial, temporal, or any combination thereof – is essential. Multivariate forecasting has been widely studied in various fields, including weather and climate forecasting (Collins, 2007; Bouallègue et al., 2016), hydrological applications (Scheuerer et al., 2017), and economic and energy forecasting (Timmermann, 2000; Pinson and Messner, 2018). One of the most prominent examples is weather forecasting, where there has been a notable evolution from deterministic point forecasts to probabilistic forecasts, including multivariate probabilistic forecasting.

Modern weather predictions rely on large-scale physics-based models where atmospheric processes are represented via partial differential equations and simulated using discrete gridded representations of the physical variables and dynamics. Over the past decades, the forecast quality of these numerical weather prediction (NWP) models has improved tremendously due to continued scientific and technological advances (Bauer et al., 2015), where one of the most prominent developments is ensemble forecasting to quantify forecast uncertainty. Nowadays, NWP models are often run in an ensemble mode by repeating runs of simulations with varying initial conditions and perturbed model physics, producing a collection of predictions for future weather states. Conceptually, the ensemble members from multiple simulations can be considered equally probable realizations of the forecast probability distribution, providing a probabilistic framework for weather forecasting. NWP models are typically simulated on gridded domains and form spatial forecast fields that contain valuable information about the predicted weather states and large-scale spatial structures, with ensemble predictions offering forecast uncertainty information. For studies on forecasts at specific locations, typically weather stations with observational data, ensemble predictions are usually derived by interpolation between grid points.

Despite remarkable improvements in methodology and computation, ensemble predictions continue to exhibit systematic errors such as biases, and typically fail to correctly quantify forecast uncertainty. These systematic errors can be corrected by post-processing ensemble forecasts, the application of which has become standard practice in research and operations. In recent years, the developments of modern machine learning (ML) methods for post-processing have been a focus of research interest and have shown substantial improvements over statistical approaches (Vannitsem et al., 2021; Haupt et al., 2021). While many post-processing techniques have been developed for ensemble forecast at specific locations or on the entire gridded domain, the latter is more challenging partly due to the sheer volume of ensemble datasets. The high-dimensional gridded data of ensemble spatial forecast fields particularly poses major computing challenges for downstream applications, such as hydrological or energy forecasting models, where gridded ensemble forecast fields of multiple variables serve as inputs. On the other hand, much of the research interest in post-processing has been focused on univariate methods, which are applied to correct systematic biases for a single weather variable at a certain location for a given forecast lead time step. The spatial, temporal, or inter-variable dependencies are typically lost during univariate post-processing, but accurate modelling of such multivariate dependence structures is crucial in many practical applications (Schefzik et al., 2013).

This thesis explores the use of generative machine learning methods to address these challenges and further extends the application of the technique from multivariate weather forecasting to multivariate electricity price forecasting, as the rising importance of intraday electricity trading in Europe demands improved price forecasting and tailored decision-support tools. Generative machine learning, an emerging and rapidly evolving field, has demonstrated remarkable success in generating realistic synthetic data, such as images, sounds, and texts. These models generate data through stochastic procedures parameterized by deep neural networks that learn the underlying patterns and structures in the training data. The probabilistic nature of these models makes them particularly suitable for probabilistic forecasting, especially being flexible for multivariate applications.

The methodological developments in this thesis broadly build on two classes of generative machine learning approaches: the Generative Moment Matching Networks (GMMNs) and Variational Autoencoders (VAEs). The subsequent chapters address three different questions related to multivariate probabilistic forecasting in the contexts of weather and electricity price forecasting. Chapter 2 proposes a conditional generative model for multivariate post-processing of weather forecasts at multiple station locations, with a focus on retaining spatial dependence structure of a single variable at a given forecast lead time. Chapter 3 introduces (variational) autoencoder-based approaches for dimensionality reduction in high-dimensional gridded ensemble forecast fields, while respecting the probabilistic nature of the ensemble forecasts. Chapter 4 generalizes the conditional generative model from Chapter 2 to electricity price forecasting in the intraday continuous-trading markets, with a focus on the economic evaluation of probabilistic forecasts.

The contributions of this thesis to existing literature are as follows. In Chapter 2, a novel multivariate post-processing method based on a generative neural network is proposed. In this new class of nonparametric data-driven distributional regression models, samples from the multivariate forecast distribution are directly obtained as output of a generative neural network. The generative model is trained by optimizing a proper scoring rule that measures the discrepancy between the generated and observed data, conditional on exogenous input variables. The method does not require parametric assumptions on univariate distributions or multivariate dependencies and allows for incorporating arbitrary predictors. In two case studies on multivariate temperature and wind speed forecasting at weather stations over Germany, the conditional generative model shows significant improvements over state-of-the-art methods and particularly improves the representation of spatial dependencies.

Chapter 4 adapts the conditional generative model developed in Chapter 2 to generate probabilistic path forecasts for intraday electricity prices, and presents several effective trading strategies for Germany's continuous-time intraday market. The primary goal is to forecast the volume-weighted intraday prices at multiple time periods before the delivery, framing the task as a multivariate time-series forecasting problem with temporal dependency modeling. The conditional generative neural network directly generates path samples from multivariate predictive distributions without parametric assumptions, bypassing the additional step of deriving multivariate paths from marginal predictions seen in statistical approaches. By conditioning on historical prices and auxiliary predictors, including wind data related to renewable energy resources, the model effectively learns both past market trends and exogenous information for accurately modelling temporal dependencies. The conditional generative model demonstrates competitive performance against two benchmark approaches in terms of statistical evaluation, and highlights its superior performance in an economic evaluation case study of a fixed-volume trading scenario using different trading strategies.

In Chapter 3, novel dimensionality reduction approaches specifically tailored to the format of weather forecast simulation ensembles are proposed, addressing the limitation of existing dimensionality reduction methods that are typically designed for deterministic and single-valued inputs. Two alternative frameworks are presented to obtain low-dimensional representations of ensemble forecasts while respecting their probabilistic character. The first approach derives a distribution-based representation of an input ensemble by applying standard dimensionality reduction techniques in a member-by-member fashion and merging the member representations into a joint parametric distribution model. The second approach achieves a similar representation by encoding all members jointly using a tailored variational autoencoder. Both approaches are evaluated and compared in a case study using 10 years of temperature and wind speed forecasts over Europe. Results demonstrate that the approaches preserve key spatial and statistical characteristics of the ensemble and enable probabilistic reconstructions of the forecast fields.

Chapter 2 is co-authored together with Tim Janke, Florian Steinke, Sebastian Lerch, and has been published in *Annals of Applied Statistics* (Chen et al., 2024). Chapter 3 is joint work with Kevin Höhlein and Sebastian Lerch, and has been submitted to *Environmental Data Science*. Chapter 4 is joint work with Tomasz Serafin, Melanie Schienle, Sebastian Lerch, Rafał Weron, and is in preparation for submission.

2 Generative machine learning methods for multivariate ensemble post-processing

2.1 Introduction

Most weather forecasts today are based on ensemble simulations of numerical weather prediction (NWP) models. Despite substantial improvements over the past decades (Bauer et al., 2015), these ensemble predictions continue to exhibit systematic errors such as biases, and typically fail to correctly quantify forecast uncertainty. These systematic errors can be corrected by statistical post-processing, the application of which has become standard practice in research and operations. Over the past years, a focal point of research interest has been the use of modern machine learning (ML) methods for post-processing, where random forest or neural network models enable the incorporation of arbitrary input predictors and have demonstrated superior forecast performance (Taillardat et al., 2016; Rasp and Lerch, 2018). For a general overview of recent developments, we refer to Vannitsem et al. (2021) and Haupt et al. (2021).

While much of the research interest in post-processing has been focused on univariate methods, many practical applications require accurate models of spatial, temporal, or inter-variable dependencies (Schefzik et al., 2013). Key examples include energy forecasting (Pinson and Messner, 2018; Worsnop et al., 2018), air traffic management (Chaloulos and Lygeros, 2007) and hydrological applications (Scheuerer et al., 2017). While the spatial, temporal, or inter-variable dependencies are present in the raw ensemble predictions from the NWP model, they are lost if univariate post-processing methods are applied separately in each margin (i.e., at each location, time step and for each target variable), which corresponds to implicitly assuming independence across space, time and variables.

Over the past decade, various multivariate post-processing methods have been proposed, see Schefzik and Möller (2018) and Vannitsem et al. (2021) for general overviews. The vast majority of multivariate post-processing methods follows a two-step strategy[1]. In the first step, univariate post-processing methods are applied independently in all dimensions to obtain calibrated marginal probability distributions. Samples are then generated from these marginal predictive distributions obtained after post-processing. In the second step, the univariate sample values are re-arranged according to a specific multivariate dependence template with the aim of restoring the multivariate dependencies that are lost in the first step. From a mathematical perspective, this can be interpreted as the application of a (parametric or non-parametric) copula, i.e., a multivariate cumulative distribution function (CDF) with standard uniform marginal distributions (Nelsen, 2006).

In most popular copula-based approaches to multivariate post-processing, the copula C is chosen to be either the parametric Gaussian copula (in the Gaussian copula approach, GCA; Möller et al., 2013; Pinson and Girard, 2012), or a non-parametric empirical copula induced by a pre-specified dependence template based on the raw ensemble predictions (in the ensemble copula coupling (ECC; Schefzik et al., 2013) approach) or past observations (in the Schaake shuffle approach, Clark et al., 2004). More advanced variants of these methods which aim to better incorporate structure in forecast error autocorrelations (Bouallègue et al., 2016) or optimize the selection of the dependence template based on similarity (Schefzik, 2016; Scheuerer et al., 2017) have been proposed over the past years. Several comparative studies of multivariate post-processing methods based on simulated and real-world data are available (Wilks, 2015; Lerch et al., 2020; Perrone et al., 2020; Lakatos et al., 2023). Overall, findings from these studies indicate that there is no consistently best approach across all settings, and that the observed differences in

[1] There exist examples of directly fitting specific multivariate probability distributions, which are mostly used in low-dimensional settings or if a specific structure can be chosen for the application at hand, with examples focusing on spatial (Feldmann et al., 2015), temporal (Muschinski et al., 2022) and inter-variable (Schuhen et al., 2012; Baran and Möller, 2015; Lang et al., 2019) dependencies being available. In addition, there are alternative approaches such as member-by-member post-processing (Van Schaeybroeck and Vannitsem, 2015) which inherently preserve dependencies present in the raw ensemble predictions.

predictive performance depend on the misspecifications of the raw ensemble predictions, but tend to be minor and not statistically significant.

All these state-of-the-art two-step methods for multivariate post-processing share common key limitations. Perhaps most importantly, there is no straightforward way to include additional predictors beyond ensemble forecasts or past observations of the target variable in the second step of imposing the dependence template onto the samples from the univariate post-processed forecast distributions. Incorporating additional predictors has proven to be a key aspect in the substantial improvements in predictive performance observed for ML-based univariate post-processing models (Taillardat et al., 2016; Rasp and Lerch, 2018; Vannitsem et al., 2021), and it seems reasonable to expect similar beneficial effects for modeling multivariate dependencies. Furthermore, the number of samples that can be obtained from the multivariate forecast distribution in the two-step post-processing methods is limited by the number of ensemble predictions (in the ECC approach), or the number of suitable past observations (in case of the Schaake shuffle) which might be disadvantageous for reliably representing and predicting multivariate extreme events.

To overcome these challenges, we propose a novel nonparametric multivariate post-processing method based on generative ML approaches. In this new class of data-driven multivariate distributional regression models, samples from the multivariate forecast distribution are directly obtained as output of a generative deep neural network which allows for incorporating arbitrary input predictors including NWP-based ensemble predictions and additional exogenous variables. Our generative model circumvents the two-step structure of the state-of-the-art multivariate post-processing approaches and aims to simultaneously correct systematic errors in the marginal distributions and the multivariate dependence structure without requiring parametric assumptions.

There has recently been an increase in research activity on generative machine learning methods for weather modeling, in particular in the context of downscaling precipitation forecasts (Leinonen et al., 2021; Harris et al., 2022; Hess et al., 2022; Price and Rasp, 2022) and nowcasting (Ravuri et al., 2021), where several approaches based on generative adversarial networks (GANs) have been proposed. While GANs are particularly useful for generating realistic images and have shown some success in a post-processing application to total cloud cover forecasts (Dai and Hemri, 2021), their training is often challenging (Gui et al., 2021). We consider a conceptually simpler class of scoring rule-based generative models (Li et al., 2015; Dziugaite et al., 2015), extending recent work in probabilistic model averaging in energy forecasting (Janke and Steinke, 2020).[2] In our approach, a conditional generative model based on a neural network is trained by optimizing a suitable multivariate proper scoring rule which measures the discrepancy between the generated and true data, and replaces the GAN discriminator. By conditioning the generative process on exogenous input variables, we enable the generative model to incorporate arbitrary input predictors beyond random noise only and to flexibly learn nonlinear relations to both the univariate forecast distributions and the multivariate dependence structure.

Using two case studies on spatial dependencies of temperature and wind speed forecasts at weather stations over Germany, we compare our conditional generative model with state-of-the-art two-step approaches to multivariate post-processing based on ECC and GCA. For the univariate post-processing step, we consider models only based on ensemble predictions of the target variable as well as state-of-the-art neural network models with additional predictors. This will allow for specifically investigating the benefits of incorporating additional information in the marginal distribution or the full multivariate forecast.

The remainder of the paper is organized as follows. Section 2.2 introduces the datasets and Section 2.3 reviews the standard two-step post-processing methods. Our conditional generative model is described in Section 2.4, followed by the main results presented in Section 2.5. Section 2.6 concludes with a discussion. Python and R code with implementations of all methods is available in the Supplemental Material and online (`https://github.com/jieyu97/mvpp`).

[2] For theoretical results on generative models based on scoring rule optimization, see Pacchiardi et al. (2024).

2.2 Data

Our study focuses on forecasts of 2-m temperature and 10-m wind speed with a forecast lead time of 48 hours. The dataset for temperature[3] is based on the one used in Rasp and Lerch (2018), while the dataset for wind speed[4] was compiled for this study. Both datasets are based on forecasts from the 50-member ensemble of the European Center of Medium-range Weather Forecasts (ECMWF) initialized at 00 UTC every day, which were obtained from the THORPEX Interactive Grand Global Ensemble (TIGGE) database (Bougeault et al., 2010) on a 0.5° × 0.5° grid over Europe. Following the procedure outlined in Rasp and Lerch (2018), the ensemble forecasts of all meteorological predictor variables are interpolated to weather station locations over Germany. Stations with larger fractions of missing data and with altitudes above 1 000 m are omitted to avoid outliers due to substantially different topographical properties when considering spatial dependencies. This results in a total of 419 stations in the temperature dataset and 198 stations in the wind speed dataset, the locations of which are shown in Figure 2.1. Corresponding observations were obtained from the Climate Data Center[5] of the German weather service.

In addition to ensemble forecasts of the target variables (temperature and wind speed, respectively), ensemble forecasts of several auxiliary predictor variables based on the selection in Rasp and Lerch (2018) are available. Ensemble forecasts of all meteorological predictor variables are reduced to their mean and standard deviation. For the wind speed dataset, we also explicitly compute the wind speeds at different pressure levels from the corresponding wind components. In addition, we use a sine-transformed value of the day of the year[6], and relevant information about the station coordinates, altitudes, and orography (altitude of the model grid point) as additional input predictors for the post-processing models. Table 2.1 provides an overview of all available predictors in both datasets.

[3] The dataset is available from `https://doi.org/10.6084/m9.figshare.19453580`.

[4] The dataset is available from `https://doi.org/10.6084/m9.figshare.19453622`.

[5] `https://www.dwd.de/DE/klimaumwelt/cdc/cdc_node.html`

[6] The numerical value of the day of the year ranging from 1 to 366, t is transformed via $\texttt{doy} = \sin\left(\frac{t-105}{366} \cdot 2\pi\right)$.

Table 2.1: Available predictors for temperature and wind speed post-processing. The variables ws, ws_pl850 and ws_pl500 are derived from the corresponding U and V wind components and are not included in the temperature dataset. For wind speed forecasts, the orography (orog) information is missing.

Variable	Description
Meteorological variables	
t2m	2-m temperature
d2m	2-m dewpoint temperature
cape	Convective available potential energy
sp	Surface pressure
tcc	Total cloud cover
q_pl850	Specific humidity at 850 hPa
u_pl850	U component of wind at 850 hPa
v_pl850	V component of wind at 850 hPa
ws_pl850	Wind speed at 850 hPa
u_pl500	U component of wind at 500 hPa
v_pl500	V component of wind at 500 hPa
gh_pl500	Geopotential height at 500 hPa
ws_pl500	Wind speed at 500 hPa
u10	10-m U component of wind
v10	10-m V component of wind
ws	10-m wind speed
sshf	Sensible heat flux
slhf	Latent heat flux
ssr	Shortwave radiation flux
str	Longwave radiation flux
Other predictors	
lon	Longitude of station
lat	Latitude of station
alt	Altitude of station
orog	Altitude of model grid point
doy	Sine-transformed value of the day of the year

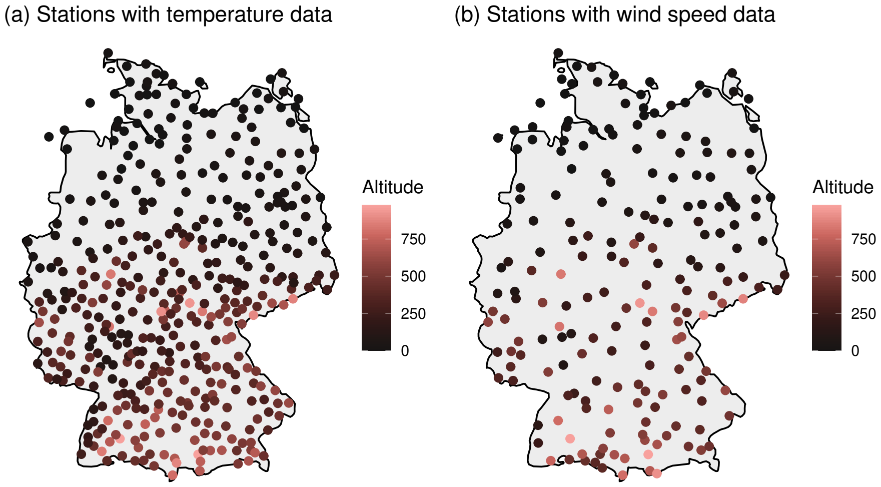

(a) Stations with temperature data
(b) Stations with wind speed data
Altitude
750
500
250
0
Altitude
750
500
250
0

Figure 2.1: Locations of weather stations with (a) temperature and (b) wind speed observations.

For both datasets, a total of 10 years of daily forecast and observation data from 2007–2016 are available. Following Rasp and Lerch (2018), we use data from 2007–2014 as training set and 2015 as validation set for choosing hyperparameters and model specifications. Data from 2016 serves as out-of-sample test dataset and is not used during model training.

2.3 Benchmark methods for multivariate ensemble post-processing

This section introduces state-of-the-art approaches to multivariate post-processing which serve as benchmark methods for our generative machine learning method that will be introduced in Section 2.4. We focus on the two-step approaches based on a combination of univariate post-processing models and copulas. In a first step, univariate post-processing is applied to ensemble forecasts for each margin (i.e., location, forecast horizon or target variable), and in a second step, the multivariate dependence structure is imposed upon the univariately post-processed forecast using a suitable copula. Sklar's theorem (Sklar, 1959) provides the theoretical underpinning in that a multivariate CDF H (our target) can be decomposed into a copula function C modeling the dependence structures (this is what needs to be specified) and its marginal univariate CDFs $F_1, \ldots, F_D$ (here obtained via univariate post-processing) via

$$H(z_1, \ldots, z_D) = C(F_1(z_1), \ldots, F_D(z_D))$$

for $z_1, \ldots, z_D \in \mathbb{R}$.

In the following, we first introduce methods for univariate post-processing and then discuss the use of copula functions for restoring multivariate dependencies. Our notation follows that of Lerch et al. (2020) and we denote the unprocessed D-dimensional ensemble forecasts of a weather variable with M members by $\boldsymbol{X}_1, ..., \boldsymbol{X}_M \in \mathbb{R}^D$, where $\boldsymbol{X}_m = \left(X_m^{(1)}, ..., X_m^{(D)}\right)$ for $m = 1, ..., M$. The corresponding observation is $\boldsymbol{y} = \left(y^{(1)}, ..., y^{(D)}\right) \in \mathbb{R}^D$, and we use a generic index $d = 1, ..., D$ to summarize the margins (in our case the locations when modeling spatial dependencies).

2.3.1 Univariate post-processing

Over the past years, a large variety of univariate post-processing methods have been proposed. In particular, the development of modern methods from ML for post-processing has been a focus of recent research interest. We refer to Vannitsem et al. (2021) for a general overview and to Schulz and Lerch (2022b) for a recent comparison in the context of wind gust forecasting. We restrict our attention to post-processing methods within the parametric distributional regression framework proposed by Gneiting et al. (2005). To specifically investigate the effect of improved marginal predictions, we consider a simple ensemble model output statistics approach (Gneiting et al., 2005) and a state-of-the-art method based on neural networks (Rasp and Lerch, 2018) which allows for incorporating additional predictor information and flexibly model nonlinear relations to forecast distribution parameters.

Within the distributional regression framework, the conditional probability distribution of the (univariate) variable of interest y, given ensemble predictions $X_1, ..., X_M$ of the target variable, is modeled by a parametric distribution, $F_{\boldsymbol{\theta}}$, the parameters of which depend on the ensemble predictions via a link function g, i.e.,

$$\boldsymbol{\theta} = g(X_1, ..., X_M).$$

Note that within the current subsection, we suppress the index of the dimension to simplify notation.

Ensemble model output statistics (EMOS)

Following Gneiting et al. (2005), the standard EMOS model for temperature (t2m) assumes a Gaussian predictive distribution $\mathcal{N}$ with mean μ and standard deviation σ, i.e.,

$$y|X_1^{\text{t2m}}, ..., X_M^{\text{t2m}} \sim \mathcal{N}(\mu, \sigma),$$

and uses ensemble forecasts of the target variable, $X_1^{\text{t2m}}, ..., X_M^{\text{t2m}}$, as sole predictors. The distribution parameters are linked to the ensemble mean and standard deviation (sd) via

$$\mu = a_0 + a_1 \cdot \text{mean}(X_1^{\text{t2m}}, ..., X_M^{\text{t2m}}) \quad \text{and} \quad \sigma = b_0 + b_1 \cdot \text{sd}(X_1^{\text{t2m}}, ..., X_M^{\text{t2m}}). \tag{2.1}$$

The EMOS coefficients a_0, a_1, b_0, b_1 are determined by minimizing the continuous ranked probability score (CRPS; Matheson and Winkler, 1976, see Section 2.5.1 for details) on the training dataset. We here implement local EMOS models by estimating separate sets of coefficients for different stations, which makes the models locally adaptive and typically leads to better performance compared with a global model that is jointly estimated for all stations, provided that a sufficient amount of training data is available (Lerch and Baran, 2017). While rolling training windows consisting of only the most recent days have often been used for the estimation of EMOS models, we use a static training period given by the entire training dataset, following common practice in the operational use of post-processing models. Furthermore, several studies suggest that the benefits of using long archives of training data often outweigh potential changes in the underlying NWP model or the meteorological conditions (e.g., Lang et al., 2020).

For wind speed, we proceed analogously, but follow Thorarinsdottir and Gneiting (2010) and utilize a Gaussian distribution left-truncated at 0. This ensures that no probability mass is assigned to negative wind speed values. Forecasts of wind speed (ws) are used as sole input predictors, i.e.,

$$y|X_1^{\text{ws}}, ..., X_M^{\text{ws}} \sim \mathcal{N}_{[0,\infty)}(\mu, \sigma),$$

where $\mathcal{N}_{[0,\infty)}(\mu, \sigma)$ denotes a truncated Gaussian distribution with cumulative distribution function

$$F(z) = \Phi\left(\frac{\mu}{\sigma}\right)^{-1} \Phi\left(\frac{z-\mu}{\sigma}\right)$$

for $z > 0$ and 0 otherwise. By contrast to the situation for temperature, the choice of a parametric distribution is less clear for wind speed, and a large variety of distributions have been considered (Lerch and Thorarinsdottir, 2013; Baran and Lerch, 2015; Scheuerer and Möller, 2015; Pantillon et al., 2018; Baran et al., 2021), including weighted mixtures of distributions (Sloughter et al., 2010; Baran and Lerch, 2016). However, the differences across parametric distributions are usually only minor and are unlikely to effect our results and conclusions. To link the parameters of the truncated Gaussian distribution to the ensemble predictions, we proceed as in (2.1), i.e.,

$$\mu = a_0 + a_1 \cdot \text{mean}(X_1^{\text{ws}}, ..., X_M^{\text{ws}}) \quad \text{and} \quad \sigma = b_0 + b_1 \cdot \text{sd}(X_1^{\text{ws}}, ..., X_M^{\text{ws}}). \tag{2.2}$$

Neural network methods for univariate post-processing

Many standard approaches to post-processing, including the EMOS method introduced above, share a common limitation in that incorporating additional predictors beyond forecasts of the target variable is challenging. To do so, it would be necessary to manually specify the exact functional form of the dependencies between the distribution parameters and all available input predictors in equations (2.1) and (2.2). Over the past years, a variety of ML methods have been developed to address this issue (Vannitsem et al., 2021). Rasp and Lerch (2018) propose a neural network (NN) approach, where the distribution parameters are obtained as the output of a NN which allows for learning arbitrary nonlinear relations between input predictors and distribution parameters in an automated, data-driven manner. We refer to this approach as the distributional regression network (DRN).

In our implementations, we follow Rasp and Lerch (2018) and use a NN with one hidden layer. All available predictors except for the date information are normalized to the range $[0, 1]$ using a min-max scaler and are then used as inputs to the NN which returns the distribution parameters μ and σ as outputs. A single NN model is estimated jointly for all stations, using the CRPS as a custom loss function. The model predictions are made locally adaptive by the use of embeddings of the station identifiers, a technique that was originally proposed in natural language processing (Pennington et al., 2014). Our model architecture and implementation choices directly follow those of Rasp and Lerch (2018) for temperature, where we employ a Gaussian predictive distribution. For wind speed, we use a truncated Gaussian predictive distribution as in the corresponding EMOS model, and apply a softplus activation,

$$\text{softplus}(z) = \log(1 + \exp(z)),$$

to the output layer to ensure positivity of the distribution parameters which helps to avoid numerical issues.

The results presented in Rasp and Lerch (2018) indicate that the DRN approach to post-processing leads to substantial improvements over state-of-the-art benchmark methods, and subsequent research has generalized the methodology to other target variables and characterizations of the forecast distribution (Bremnes, 2020; Scheuerer et al., 2020; Chapman et al., 2022; Schulz and Lerch, 2022b). For our purposes here it will be interesting to investigate whether the improvements in the univariate predictive performance directly extends to improved multivariate predictions when coupling the univariate DRN models with the re-ordering techniques introduced below.

2.3.2 Multivariate extensions using copulas

Univariate post-processing methods are intended to correct systematic errors in the marginal distributions. However, multivariate dependencies are lost when univariate post-processing is applied separately for each margin (e.g., each station in our application), and need to be restored. A variety of methods for restoring multivariate dependencies via copula functions have been proposed over the past years. We here limit our discussion to the popular ensemble copula coupling and Gaussian copula approach, and refer to Schefzik et al. (2013); Wilks (2015) and Lerch et al. (2020) for overviews and comparisons. In our descriptions, we follow Lerch et al. (2020) and refer to their Section 2 for further mathematical details and references.

Ensemble copula coupling

Given univariately post-processed marginal distributions, a sample of the same size as the raw ensemble, M, is drawn from each predictive marginal distribution. While several sampling schemes have been proposed (Schefzik et al., 2013; Hu et al., 2016), we only consider the use of equidistant quantiles at levels $\frac{1}{M+1}, ..., \frac{M}{M+1}$. Ensemble copula coupling (ECC) is based on the assumption that the ensemble forecasts are informative about the true multivariate dependence structure, and it makes use of the rank order structure of the raw ensemble member forecasts to rearrange the sampled values, with possible ties resolved at random. This can be interpreted as a non-parametric, empirical copula approach, which we refer to as ECC-Q.

A widely used alternative non-parametric approach is the Schaake shuffle (Clark et al., 2004) which proceeds as ECC, but reorders the sampled quantiles based on the rank order structure of past observations instead of the ensemble forecasts. As noted in the introduction, comparative studies have often found similar predictive performances between the Schaake shuffle and ECC (e.g., Lakatos et al., 2023). For the datasets at hand, initial tests indicated a slightly worse performance compared to ECC-Q (not shown), and we thus only use ECC-Q as a non-parametric benchmark approach to retain focus.

Gaussian copula approach

In contrast to ECC-Q, the Gaussian copula approach (GCA; Pinson and Girard, 2012; Möller et al., 2013) is based on a parametric Gaussian copula. In a first step of the application of GCA, a set of past observations is transformed into latent standard Gaussian observations via

$$\tilde{y}^{(d)} = \Phi^{-1}\left(F_\theta^{(d)}\left(y^{(d)}\right)\right)$$

for all dimensions $d = 1, ..., D$, where $F_\theta^{(d)}$ denotes the corresponding forecast distributions obtained via univariate post-processing. In a next step, multivariate random samples $\boldsymbol{Z}_1, ..., \boldsymbol{Z}_M$ are randomly drawn from a D-dimensional Gaussian distribution $\mathcal{N}^{(D)}(\mathbf{0}, \Sigma)$ with a mean vector of 0 and an empirical correlation matrix Σ based on the observations transformed into a latent Gaussian space in the first step. The final post-processed GCA forecasts are then obtained via

$$X_m^{\mathrm{GCA}\,(d)} = \left(F_\theta^{(d)}\right)^{-1}\left(\Phi\left(Z_m^{(d)}\right)\right)$$

for $m = 1, ..., M$ and $d = 1, ..., D$.

In addition to the assumption of a parametric copula, the main difference of GCA to ECC-Q is given by the use of past observations to determine the dependence template. While the number of GCA ensemble members is not limited by the size of the raw ensemble, we here only consider $M = 50$ to ensure comparability across methods.

To summarize, in the following we will consider four two-step approaches based on the available combinations of methods for univariate post-processing (EMOS, DRN) and copula-based modeling of multivariate dependencies (ECC-Q, GCA) for both target variables (temperature and wind speed) which will serve as benchmarks for our generative ML approach. These benchmark methods will be abbreviated by EMOS+ECC, EMOS+GCA, DRN+ECC and DRN+GCA, respectively.

2.4 Generative models for multivariate distributional regression

We propose a novel approach to multivariate ensemble post-processing using generative machine learning techniques. Moving beyond the previous two-step strategy of separately modeling marginal distributions and multivariate dependence structure, this new class of data-driven multivariate distributional regression models allows for obtaining multivariate probabilistic forecasts directly as output of a NN. The proposed generative models provide a non-parametric way to generate post-processed multivariate samples without any distributional assumptions on the marginal distributions or the multivariate dependencies. By allowing for incorporating additional predictors beyond forecasts of the target variable and for generating an arbitrary number of samples, the proposed generative models further address key limitations of state-of-the-art two-step approaches to multivariate post-processing.

2.4.1 Deep generative models

Implicit generative models aim to provide a representation of the probability distribution of a target variable by defining a stochastic procedure to generate samples $\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_{n_{\text{out}}}$ from the distribution of interest (Mohamed and Lakshminarayanan, 2017). The only input to the generative model

$$\boldsymbol{Y}_i = g_{\boldsymbol{\theta}}(\boldsymbol{Z}_i) \tag{2.3}$$

are samples $\boldsymbol{Z}_1, \ldots \boldsymbol{Z}_{n_{\text{out}}}$ from a simple base distribution, e.g., a standard multivariate normal distribution $\boldsymbol{Z} \sim N(\boldsymbol{0}, \boldsymbol{I})$. The learnable map g is typically parameterized by a deep NN with parameters $\boldsymbol{\theta}$.

As implicit generative models do not provide a tractable density of the target distribution, classic parameter estimation procedures such as maximum likelihood estimation are infeasible. Generative adversarial networks (Goodfellow et al., 2014) sidestep this problem by specifying an additional classification model, the discriminator, which is trained to discriminate between the true data and the generated samples. The generator model is then trained to maximize the misclassification rate of the discriminator. In the context of ensemble post-processing, Dai and Hemri (2021) propose a GAN-based model for generating spatially coherent maps of total cloud cover forecasts.

While impressive results have been achieved by GANs, in particular in the context of image processing and computer vision, the adversarial training process is often considered to be complex and unstable (Gui et al., 2021). Therefore, several alternative approaches for training generative models have been developed. From a statistical perspective, generative moment matching networks (GMMNs; Li et al., 2015; Dziugaite et al., 2015) are a particularly interesting class of generative models. GMMNs replace the discriminator with the maximum mean discrepancy (MMD; Gretton et al., 2012), a kernel-based two-sample test statistic which can be used to measure distances on the space of probability distributions. The training objective is then to minimize a Monte Carlo estimate of the MMD. However, GMMNs only approximate the unconditional data distribution $P(\boldsymbol{Y})$ while we are interested in the conditional distribution $P(\boldsymbol{Y}|\boldsymbol{X})$, i.e., we aim for a conditional generative model of the form

$$\boldsymbol{Y}_i = g_{\boldsymbol{\theta}}(\boldsymbol{X}, \boldsymbol{Z}_i). \tag{2.4}$$

Our approach builds on recent work on probabilistic model averaging in energy forecasting (Janke and Steinke, 2020) and uses the energy score (ES; Gneiting and Raftery, 2007, see Section 2.5.1 for details) as a loss function to train a conditional generative model. The ES is a multivariate strictly proper scoring rule which is derived from the energy distance (Székely, 2003), which in turn is a special case of the MMD (Sejdinovic et al., 2013). Training a conditional generative model based on ES optimization allows for generating multivariate post-processed forecasts that simultaneously correct systematic biases and dispersion errors as well as systematic errors in the multivariate dependence structure.

2.4.2 Notation

We now introduce the notation that will be used for describing the model architecture and training process. As input predictors at every weather station location $d = 1, ..., D$, we have NWP forecasts of K different weather variables available (see Table 2.1), each in the form of an ensemble of size M, i.e., $X_{k,m}^{(d)}$ is the value of the mth ensemble member for variable a_k at location d, with $m = 1, ..., M$, $k = 1, ..., K$, and $d = 1, ..., D$. The ensemble weather forecasts are reduced to the corresponding ensemble mean $\mu(\boldsymbol{X}_k^{(d)}) = \frac{1}{M}\sum_m X_{k,m}^{(d)}$ and standard deviation $\sigma(\boldsymbol{X}_k^{(d)})$. In the following, we will use

$$\bar{\boldsymbol{X}}^{(d)} = \left[\mu\left(\boldsymbol{X}_1^{(d)}\right), \ldots, \mu\left(\boldsymbol{X}_K^{(d)}\right)\right]^T \quad \text{and} \quad \boldsymbol{s}^{(d)} = \left[\sigma\left(\boldsymbol{X}_1^{(d)}\right), \ldots, \sigma\left(\boldsymbol{X}_K^{(d)}\right)\right]^T$$

to denote vectors of size K which contain the mean ensemble predictions of all variables at location d and the corresponding standard deviations, respectively. As additional static input predictors, we use the vector $\mathbf{loc}^{(d)} = [\mathrm{lat}^{(d)}, \mathrm{lon}^{(d)}, \mathrm{alt}^{(d)}, \mathrm{orog}^{(d)}]^T$ with location-specific information, as well as the (scalar) sine-encoded day of the year, doy. We will denote a single sample from the D_{latent}-dimensional noise distribution by $\boldsymbol{z}_i = [z_1, \ldots, z_{D_{\text{latent}}}]^T \sim \mathcal{N}^{D_{\text{latent}}}(\boldsymbol{0}, \boldsymbol{I})$, and denote a single final output sample from the D-dimensional target distribution by $\hat{\boldsymbol{y}}_i, i = 1, ..., n_{\text{out}}$.

2.4.3 Model architecture and training

A schematic overview of our conditional generative model (CGM) is provided in Figure 2.2. The same basic model structure is used for both temperature and wind speed prediction,

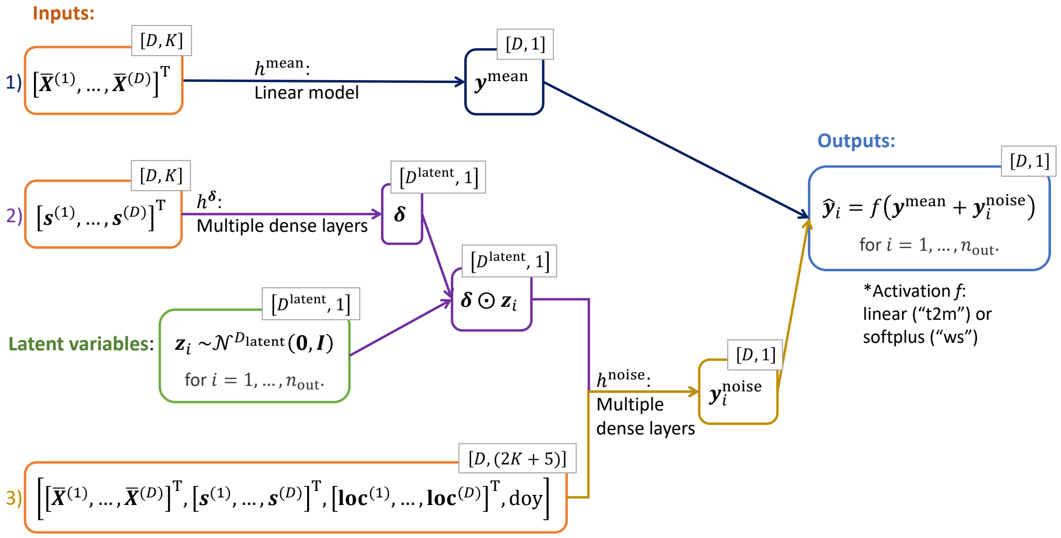

Inputs:
[D, K]
1) $[\bar{\boldsymbol{X}}^{(1)}, \dots, \bar{\boldsymbol{X}}^{(D)}]^{\mathrm{T}}$
h^{mean}:
Linear model
[D, 1]
$\boldsymbol{y}^{\mathrm{mean}}$
Outputs:
[D, 1]
$\hat{\boldsymbol{y}}_i = f(\boldsymbol{y}^{\mathrm{mean}} + \boldsymbol{y}_i^{\mathrm{noise}})$
for $i = 1, \dots, n_{\mathrm{out}}$.
*Activation f:
linear ("t2m") or
softplus ("ws")
[D, K]
2) $[\boldsymbol{s}^{(1)}, \dots, \boldsymbol{s}^{(D)}]^{\mathrm{T}}$
$h^{\boldsymbol{\delta}}$:
Multiple dense layers
$[D^{\mathrm{latent}}, 1]$
$\boldsymbol{\delta}$
$[D^{\mathrm{latent}}, 1]$
$\boldsymbol{\delta} \odot \boldsymbol{z}_i$
Latent variables:
$[D^{\mathrm{latent}}, 1]$
$\boldsymbol{z}_i \sim \mathcal{N}^{D_{\mathrm{latent}}}(\boldsymbol{0}, \boldsymbol{I})$
for $i = 1, \dots, n_{\mathrm{out}}$.
h^{noise}:
Multiple
dense layers
[D, 1]
$\boldsymbol{y}_i^{\mathrm{noise}}$
[D, (2K + 5)]
3) $\left[[\bar{\boldsymbol{X}}^{(1)}, \dots, \bar{\boldsymbol{X}}^{(D)}]^{\mathrm{T}}, [\boldsymbol{s}^{(1)}, \dots, \boldsymbol{s}^{(D)}]^{\mathrm{T}}, [\mathbf{loc}^{(1)}, \dots, \mathbf{loc}^{(D)}]^{\mathrm{T}}, \mathrm{doy}\right]$

Figure 2.2: Schematic illustration of the conditional generative model. The dimensions of the tensors at each step are indicated in the small box.

and we will highlight relevant differences in the following. The final output of the model is given by D-dimensional multivariate samples drawn from the post-processed joint distribution, $\{\hat{\boldsymbol{y}}_i, i = 1, \dots, n_{\mathrm{out}}\}$, which are composed of a mean component $\boldsymbol{y}^{\mathrm{mean}} \in \mathbb{R}^D$, and a noise component $\boldsymbol{y}_i^{\mathrm{noise}} \in \mathbb{R}^D$ that depends on the sample $\boldsymbol{z}_i \in \mathbb{R}^{D_{\mathrm{latent}}}$ from the latent noise distribution for $i = 1, \dots, n_{\mathrm{out}}$.

Our CGM architecture allows for incorporating arbitrary input predictors and for specifically tailoring the model structure to incorporate relevant exogenous information in the different components of the target distribution by separating the mean and noise component. To efficiently propagate the uncertainty inherent to the NWP forecasts, we dynamically reparametrize the latent distributions of the generative model conditionally on the standard deviations of the NWP ensemble forecasts. The overall model consists of three modules with different sets of inputs, which will be described in the following.

The first part of the model aims to learn the multivariate mean component $\boldsymbol{y}^{\mathrm{mean}}$ of the forecast distribution and can be considered as a multivariate deterministic bias-correction step of the mean ensemble forecasts, conditional on the available additional predictors. This mean module thus maps mean ensemble forecasts of all weather input variables to

$\boldsymbol{y}^{\text{mean}}$, i.e.,

$$\boldsymbol{y}^{\text{mean}} = h^{\text{mean}}\left(\bar{\boldsymbol{X}}^{(1)}, ..., \bar{\boldsymbol{X}}^{(D)}\right).$$

We utilize a linear model without hidden layers for h^{mean}, i.e., we have for each dimension $d = 1, ..., D$

$$y^{\text{mean}\,(d)} = w_{0,d} + \sum_{k=1}^{K} w_{k,d} \cdot \mu\left(\boldsymbol{X}_k^{(d)}\right).$$

The remaining two parts of the model aim to learn the noise component $\boldsymbol{y}^{\text{noise}}$ of the multivariate forecast distribution conditional on the available predictor information. The generative structure of our CGM approach becomes apparent in the second part of the model, where n_{out} random samples of D_{latent}-dimensional latent variables are drawn independently from a standard multivariate normal distribution, i.e.,

$$\boldsymbol{z}_1, ..., \boldsymbol{z}_{n_{\text{out}}} \sim \mathcal{N}^{D_{\text{latent}}}(\boldsymbol{0}, \boldsymbol{I}),$$

where $\boldsymbol{z}_i \in \mathbb{R}^{D^{\text{latent}}}$ for $i = 1, ..., n_{\text{out}}$. The number of samples n_{out} generated in this step directly controls the final number of output samples obtained from the multivariate CGM forecast distribution and thus allows for generating an arbitrary number of post-processed samples.[7] In a next step, the noise encoder module aims to encode the uncertainty from the ensemble weather forecasts into the latent distribution by adjusting the scale of the latent variables via a linear mapping. To that end, we first model the conditional variance of the latent distribution via a fully connected NN

$$\boldsymbol{\delta} = h^{\boldsymbol{\delta}}\left(\boldsymbol{s}^{(1)}, ..., \boldsymbol{s}^{(D)}\right),$$

where an exponential activation function is applied in the output layer to ensure positivity of the variances $\boldsymbol{\delta} \in \mathbb{R}^{D_{\text{latent}}}$. The scaling coefficient vector $\boldsymbol{\delta} = [\delta_1, ..., \delta_{D_{\text{latent}}}]^T$ is then used to adjust the variance of the latent noise variables, i.e.,

$$\tilde{\boldsymbol{z}}_i = \boldsymbol{\delta} \odot \boldsymbol{z}_i$$

[7] Note that the description in the following will focus on a single sample $\boldsymbol{z}_i$. During the model training process described in more detail below, we generate n_{out} independent CGM predictions.

for $i = 1, ..., n_{\text{out}}$.[8]

In the final part of the model, the scale-adjusted latent variables from the noise encoder module are now combined with additional predictors to yield the final conditional noise component $\boldsymbol{y}_i^{\text{noise}}$. This noise decoder module is a fully connected NN

$$\boldsymbol{y}_i^{\text{noise}} = h^{\text{noise}} \left(\bar{\boldsymbol{X}}^{(1)}, ..., \bar{\boldsymbol{X}}^{(D)}, \boldsymbol{s}^{(1)}, ..., \boldsymbol{s}^{(D)}, \mathbf{loc}^{(1)}, ..., \mathbf{loc}^{(D)}, \text{doy}, \tilde{\boldsymbol{z}}_i \right).$$

The final output of the model is given by the collection of realizations from the multivariate forecast distribution

$$\hat{\boldsymbol{y}}_i = \boldsymbol{y}^{\text{mean}} + \boldsymbol{y}_i^{\text{noise}},$$

for $i = 1, ..., n_{\text{out}}$ for temperature. For wind speed, we additionally apply a softplus activation function

$$\hat{\boldsymbol{y}}_i = \log\left(1 + \exp(\boldsymbol{y}^{\text{mean}} + \boldsymbol{y}_i^{\text{noise}})\right) \tag{2.5}$$

to ensure non-negativity of the obtained wind speed forecasts.

During training, for each training example $(\boldsymbol{X}_n, \boldsymbol{y}_n)$, where $\boldsymbol{X}_n$ and $\boldsymbol{y}_n$ represent the inputs and true observations at forecast case $n = 1, ..., N$, respectively, we generate n_{out} independent predictions $\hat{\boldsymbol{y}}_1, \ldots, \hat{\boldsymbol{y}}_{n_{\text{out}}}$ by querying the model n_{out} times. Each time the model generates predictions from a different sampled noise vector $\boldsymbol{z}_i, i = 1, ..., n_{\text{out}}$, but uses the same inputs $\boldsymbol{X}_n$. The training procedure is formalized in Algorithm 1.[9]

The CGM parameters, i.e., the weights and biases of h^{mean}, $h^{\boldsymbol{\delta}}$ and h^{noise}, are estimated by optimizing the energy score (see Section 2.5.1) as a loss function tailored to the specific situation of multivariate probabilistic forecasting based on an implicit representation of the forecast distribution in the form of a sample of size $n_{\text{train}} = 50$. We follow common practice in the machine learning literature and generate an ensemble of CGMs by repeating the estimation process multiple times from different random initializations

[8] Note that we now have $\tilde{\boldsymbol{z}}_i \sim N(\mathbf{0}, diag(\boldsymbol{\delta}))$, i.e., it is an isotropic Gaussian with a variance conditional on the ensemble standard deviations. Scale adjustment approaches that have been applied in the verification of gridded forecasts in the meteorological literature (Bouallegue et al., 2020) can be viewed as conceptually related.

[9] Note that this procedure might be slow on a CPU but the innermost loop in Algorithm 1 is fully parallelizable and can thus be efficiently implemented on GPUs.

Algorithm 1: CGM training algorithm

```
Input  : data $\{(\boldsymbol{X},\boldsymbol{y})_n\}_{n=1}^{N}$, initial model parameters $\boldsymbol{\theta}_0$, number of samples $n_{\text{train}}$,
         number of batches $B$, learning rate $\eta$
Output : Model parameters $\boldsymbol{\theta}^*$
for $N_{epochs}$ do
    Get mini batch $\{(\boldsymbol{X},\boldsymbol{y})_b\}_{b=1}^{B}$;
    For each sample $b$ generate a set of $n_{\text{train}}$ random noise samples $\{[\boldsymbol{z}_b^1,\dots,\boldsymbol{z}_b^{n_{\text{train}}}]\}_{b=1}^{B}$
    for $b = 1,\dots,B$ do
        for $s = 1,\dots,n_{train}$ do
            Compute forward pass $\hat{\boldsymbol{y}}_b^s \leftarrow g_{\boldsymbol{\theta}}(\boldsymbol{X}_b,\boldsymbol{z}_b^s)$
        end
    end
    Compute loss over batch $L \leftarrow \frac{1}{B}\sum_b \text{ES}\big(\boldsymbol{y}_b,[\hat{\boldsymbol{y}}_b^1,\dots,\hat{\boldsymbol{y}}_b^{n_{\text{train}}}]\big)$;
    Compute gradient $\nabla_{\boldsymbol{\theta}} L$;
    Update learning rate $\eta$ using ADAM;
    Update model parameters $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta\nabla_{\boldsymbol{\theta}} L$
end
```

to account for the randomness of the training process based on stochastic gradient descent methods (Lakshminarayanan et al., 2017; Schulz and Lerch, 2022a). Unless indicated otherwise, we will generate a set of $n_{\text{out}} = 50$ multivariate samples to ensure comparability with the benchmark methods when making predictions on the test set, and do so by repeating the model estimation 10 times and generating 5 samples each.

2.4.4 Implementation details and hyperparameter choices

Multiple hyperparameters need to be determined for the CGM implementation. For the specific setup of the individual modules of the model (h^{mean}, $h^{\boldsymbol{\delta}}$ and h^{noise}), we use two hidden layers with 100 nodes each in the noise decoder module h^{noise} for both target variables. As for the noise encoder module $h^{\boldsymbol{\delta}}$, we use two hidden layers with 100 nodes each for wind speed but one linear dense layer for temperature. Initial experiments indicated no substantial changes in the resulting model performance, we thus do not further optimize this component of the model architecture. Note that we also tested a nonlinear model based on a fully connected NN for $\boldsymbol{y}^{\text{mean}}$ which did not lead to substantial improvements. Details are provided in the Supplemental Material. The exponential linear unit (ELU; Clevert, 2015) activation function is applied in all hidden layers.

The number of latent variables, D^{latent}, the learning rate and the batch size are initially determined using the distributed asynchronous hyperparameter optimization technique from Bergstra et al. (2013) implemented in the `hyperopt` package. Based on the automated hyperparameter optimization and selected additional experiments we set D^{latent} to 10, and use a learning rate of 0.001 and a batch size of 64 for both target variables. The results are relatively robust to changes in these hyperparameters. Exemplary results are illustrated in the form of ablation studies in the Supplemental Material. The model is trained using stochastic gradient descent optimization based on the Adam optimizer (Kingma, 2014), where we employ an early stopping criterion with a patience of 10 epochs to avoid overfitting. The maximum number of epochs is set to 300 but generally, the CGM training and validation losses converge after a few epochs.

All available predictors (listed in Table 2.1) except for the target variable and date information are normalized by removing the mean and scaling to unit variance using the standard scaler from the `scikit-learn` package (Pedregosa et al., 2011).

2.5 Results

We here compare the CGM predictions with the various benchmark methods based on two-step procedures of separately modeling marginal distributions and multivariate dependencies introduced in Section 2.3.2, i.e., EMOS+ECC, EMOS+GCA, DRN+ECC and DRN+GCA. To do so, we first provide some background information on forecast evaluation methods (Section 2.5.1), and present the general setup of our experiments (Section 2.5.2) and univariate results (Section 2.5.3). The main focus is on the multivariate forecast performance presented in Sections 2.5.4–2.5.6.

2.5.1 Forecast evaluation methods

We briefly review key concepts relevant to the evaluation of probabilistic forecasts and refer to Rasp and Lerch (2018, Appendix A) and Lerch et al. (2020, Appendix B) for details. The general goal of probabilistic forecasting is to maximize the sharpness of a predictive distribution subject to calibration (Gneiting et al., 2007). In order to assess calibration and sharpness simultaneously, proper scoring rules are now widely used for comparative evaluation of probabilistic forecasts. A scoring rule $S(F, y)$ assigns a numerical score to a pair of a predictive distribution $F \in \mathcal{F}$ and a realizing observation $y \in \Omega$, where $\mathcal{F}$ is a class of probability distributions on Ω. It is called proper if the true distribution of the observation minimizes the expected score, i.e., if $\mathbb{E}S(G, Y) \leq \mathbb{E}S(F, Y)$ if $Y \sim G$ for all $F, G \in \mathcal{F}$ (Gneiting and Raftery, 2007). The continuous ranked probability score (CRPS; Matheson and Winkler, 1976) given by

$$\mathrm{CRPS}(F, y) = \int_{-\infty}^{\infty} \left(F(z) - \mathbb{1}\{z \geq y\}\right)^2 dz,$$

where $\mathbb{1}$ denotes the indicator function and F is assumed to have a finite first moment, is a popular proper scoring rule for univariate probabilistic forecasts (i.e., $\Omega \subset \mathbb{R}$). Closed-form analytical expressions are available for many parametric forecast distributions and probabilistic forecasts given in the form of a simulated sample (Jordan et al., 2019).

While the definition of proper scoring rules can in principle be straightforwardly extended towards multivariate settings with $\Omega \subset \mathbb{R}^D$, many practical questions remain open and a variety of multivariate proper scoring rules have been proposed over the past years (Petropoulos et al., 2022). Most of these multivariate proper scoring rules focus on multivariate probabilistic forecasts in the form of samples from the forecast distributions.

Using notation introduced in Section 2.3, the energy score (ES; Gneiting and Raftery, 2007),

$$\mathrm{ES}(F, \boldsymbol{y}) = \frac{1}{M} \sum_{i=1}^{M} \|\boldsymbol{X}_i - \boldsymbol{y}\| - \frac{1}{2M^2} \sum_{i=1}^{M} \sum_{j=1}^{M} \|\boldsymbol{X}_i - \boldsymbol{X}_j\|,$$

where $\| \cdot \|$ is the Euclidean norm on $\mathbb{R}^D$, and the variogram score of order p (VS^p; Scheuerer and Hamill, 2015),

$$\mathrm{VS}^p(F, \boldsymbol{y}) = \sum_{i=1}^{D} \sum_{j=1}^{D} w_{i,j} \left(\left| y^{(i)} - y^{(j)} \right|^p - \frac{1}{M} \sum_{k=1}^{M} \left| X_k^{(i)} - X_k^{(j)} \right|^p \right)^2 ,$$

are the most popular examples of multivariate proper scoring rules. In the definition of the VS, $w_{i,j} \geq 0$ is a non-negative weight for pairs of component combinations and p is the order of the VS. We use an unweighted (i.e. $w_{i,j} = 1$ for all i, j) version of the VS with order $p = 0.5$ throughout, following suggestions of Scheuerer and Hamill (2015) and utilizing implementations provided in Jordan et al. (2019). Other multivariate proper scoring rules have been proposed including copula scores focusing on the dependence structure (Ziel and Berk, 2019) and weighted versions of ES and VS (Allen et al., 2023), and we present additional results for some of these scores in the Supplemental Material.

To compare forecasting methods based on a proper scoring rule with respect to a benchmark, we will often calculate the associated skill score. With the mean score of the forecasting method of interest over a test dataset, $\bar{S}_{\text{f}}$, the corresponding mean scores of the benchmark, $\bar{S}_{\text{ref}}$, and the (hypothetical) optimal forecast, $\bar{S}_{\text{opt}}$, the skill score SS_{f} is calculated via

$$SS_{\text{f}} = \frac{\bar{S}_{\text{ref}} - \bar{S}_{\text{f}}}{\bar{S}_{\text{ref}} - \bar{S}_{\text{opt}}}.$$

For the scoring rules considered below, $S_{\text{opt}} = 0$. Skill scores are positively oriented with a maximum value of 1, values of 0 indicating no improvement over the benchmark and negative values indicating a worse predictive performance than the benchmark.

2.5.2 Setup of the multivariate post-processing experiments

To evaluate the multivariate forecast performance of different post-processing methods, we repeatedly sub-sample the station datasets described in Section 2.2. Focusing on spatial dependencies over geographically close stations in a setting that aims to mimic practical applications, we fix a number of dimensions $D \in \{5, 10, 20\}$, and proceed as follows.

We randomly pick a station and then select the $(D-1)$ stations which are geographically closest to obtain a set of D stations, based on which we implement the multivariate post-processing methods as described in Sections 2.3 and 2.4. Next, we apply the scoring rules introduced above to obtain corresponding mean scores over the test set, i.e., data from the calendar year 2016, and compute the corresponding skill scores.

To account for uncertainties, the above procedure is repeated 100 times for both temperature and wind speed. For skill score computations, EMOS+ECC is used as a reference method throughout and for all methods, we generate 50 multivariate samples

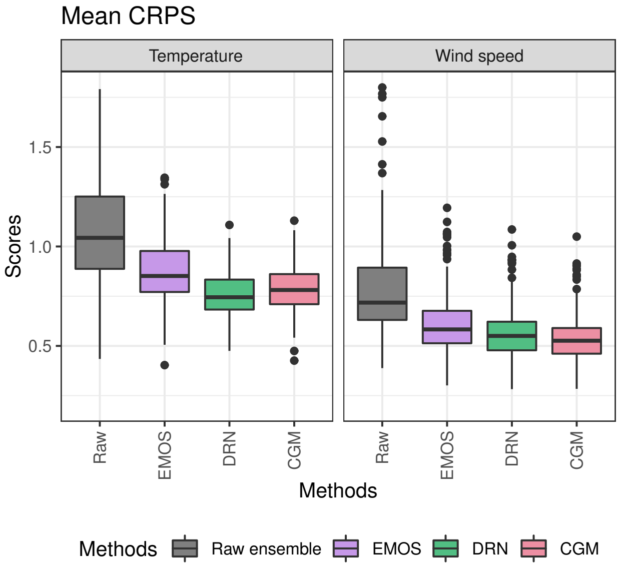


Figure 2.3: Boxplots of mean CRPS values of different multivariate post-processing methods with $D = 5$, including the scores of raw ensemble forecasts. The scores are based on 242 unique stations in case of temperature, and 178 unique stations in case of wind speed.

from all post-processed forecast distributions to ensure consistency and allow for a fair comparison.

2.5.3 Univariate results

While the focus of our study is on multivariate post-processing, the univariate predictive performance constitutes an important component of the overall forecast quality. Here, we therefore first focus on the univariate, marginal predictions of different post-processing methods to investigate the performance of our CGM approach in this setting.

To evaluate the univariate forecast performance in the experimental setup described above, we restrict our attention to the experiments for $D = 5$ and compute the mean CRPS over unique sets of stations present in the 100 repetitions of the sub-sampled station datasets. In case a station occurs multiple times in the randomly selected sets of stations, we only use data from the first occurrence within the 100 repetitions. We compare the univariate performance of our CGM to the raw ensemble forecasts and two univariate post-processing approaches discussed earlier, i.e., EMOS and DRN. Figure 2.3 shows boxplots of the mean CRPS values over the corresponding unique sets of stations for temperature and wind speed. As expected, all univariate post-processing methods notably improve the forecast performance over the raw ensemble predictions, and the variability among different stations is reduced. For temperature, the CGM forecasts generally outperform the EMOS predictions and are slightly worse than the DRN post-processed forecasts, where the mean CRPS over all stations is improved from 0.89 for EMOS to 0.76 for DRN and 0.79 for CGM. For wind speed, the CGM provides the overall best forecasts and clearly outperforms the DRN approach, with a mean CRPS improved from 0.62 for EMOS and 0.58 for DRN to 0.54 for CGM. The differences in performance can potentially be explained by the better fit of the chosen Gaussian parametric distribution for temperature which favors DRN, whereas the choice is less clear for wind speed, favoring the nonparametric CGM approach. Additional results are provided in the Supplemental Material, including an assessment of calibration which indicates that all post-processing methods generally provide relatively well-calibrated forecasts.

2.5.4 Multivariate results

We now turn to the key part of our results and compare the multivariate performance of our CGM approach to the two-step post-processing approaches used as benchmark methods. Table 2.2 summarizes the mean scores of different multivariate post-processing models for temperature and wind speed. For both target variables, all post-processing methods clearly improve the raw ensemble predictions. In comparison to the EMOS-based models, the DRN-based models show clear improvements in the multivariate performance. Regarding the choice of the reordering method, ECC and GCA lead to similar results in terms of the ES, but the GCA-based forecasts lead to better performance in terms of the VS. The CGM consistently provides the best multivariate forecasts and outperforms the state-of-the-art approaches across the variables, dimensions and evaluation metrics. The only exception to this observation are temperature forecasts evaluated with the ES, where the DRN+ECC and DRN+GCA models provide slightly better forecasts for $D = 10$ and $D = 20$. The values of all considered multivariate scoring rules increase with

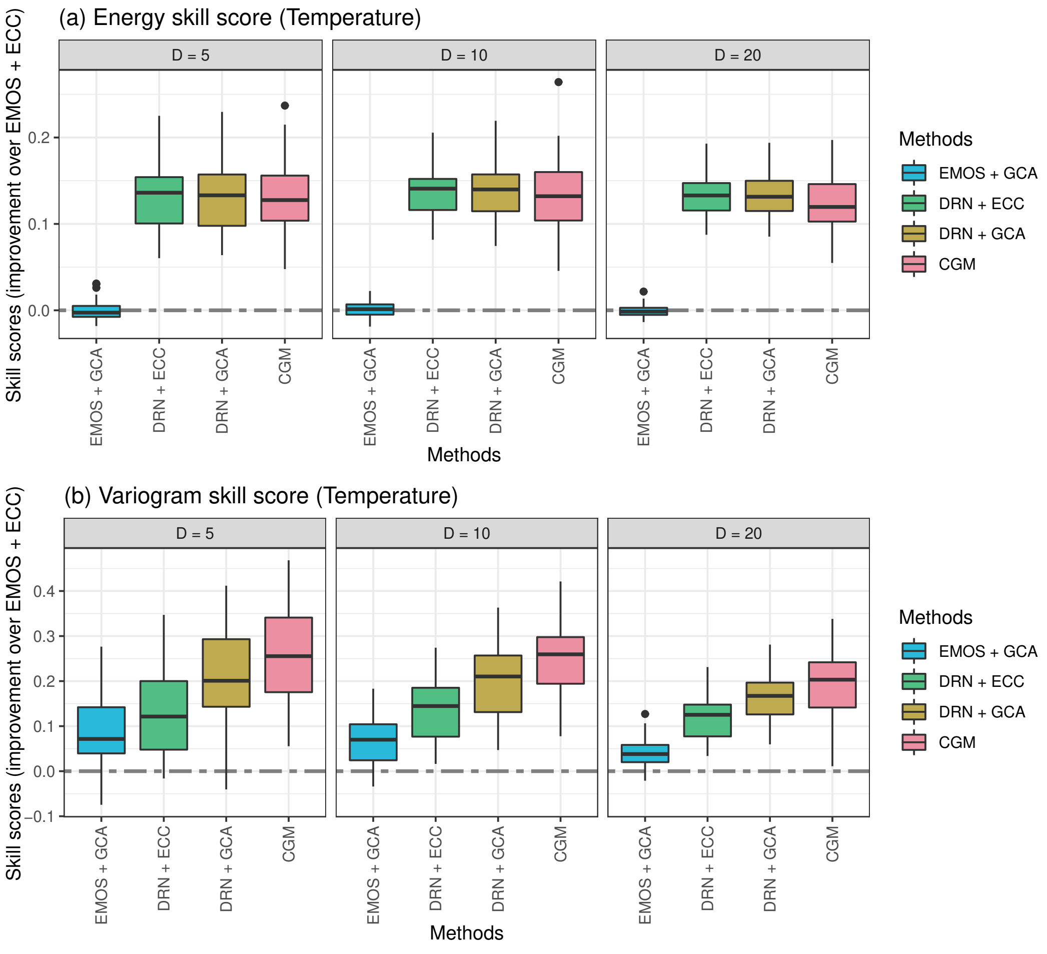


Figure 2.4: Boxplots of (a) energy skill scores and (b) variogram skill scores of different multivariate post-processing methods for temperature across the 100 repetitions of the experiment with different sets of stations. EMOS+ECC is used as reference forecast in both cases.

Table 2.2: Mean multivariate scores of different multivariate post-processing methods for temperature and wind speed, averaged over the 100 repetitions of the simulation experiment.

			Raw	EMOS+	EMOS+	DRN+	DRN+	
Variable	Score	D	ens.	ECC	GCA	ECC	GCA	CGM
Temperature	ES	5	2.81	2.27	2.27	**1.97**	**1.97**	**1.97**
		10	4.22	3.37	3.37	2.91	**2.90**	2.91
		20	6.09	4.87	4.87	**4.21**	4.22	4.26
	VS	5	8.22	4.81	4.36	4.12	3.74	**3.50**
		10	39.0	22.6	21.0	19.5	18.0	**16.9**
		20	153	96.7	92.8	85.0	80.7	**77.8**
Wind speed	ES	5	2.44	1.69	1.68	1.56	1.55	**1.44**
		10	3.67	2.55	2.53	2.31	2.30	**2.16**
		20	5.04	3.52	3.51	3.23	3.22	**3.04**
	VS	5	9.49	4.37	4.00	4.01	3.66	**3.31**
		10	39.7	20.2	19.0	18.0	16.9	**15.4**
		20	153	82.5	78.9	75.6	72.3	**67.0**

the spatial dimension D, which is to be expected from the definition of the scoring rules and consistent with findings in the extant literature.

To investigate the variability across the selected sets of stations, Figures 2.4 and 2.5 show boxplots of the multivariate skill scores for temperature and wind speed, respectively, using EMOS+ECC as reference method. For the temperature forecasts (Figure 2.4), the DRN-based two-step methods and the CGM provide consistent and comparable improvements over the reference in terms of the ES. The relative improvements in terms of the VS show a larger variability across the sets of stations, and indicate a superior performance of the CGM forecasts. Among the considered two-step approaches, applying GCA leads to improvements over ECC in terms of the VS, but similar results in terms of the ES. The above observations apply to all considered spatial dimensions and we do not observe any obvious trends in terms of D, indicating that consistent improvements can be observed also in the higher-dimensional settings in the experiments.

Qualitatively similar results can be observed for the multivariate wind speed forecasts shown in Figure 2.5. The main difference to the results for temperature are the notably larger improvements of the CGM forecasts in comparison to the DRN-based approaches, particularly in terms of the ES. Interestingly, in terms of the VS at $D = 5$, the DRN+ECC models here fail to outperform the EMOS+GCA forecasts despite the incorporation of additional predictor variables in the marginal distributions. A potential explanation for this observation is that the disadvantages due to the misspecifcations in the multivariate dependence structure of the raw ensemble forecasts which serve as a dependence template for ECC outweigh the benefits of incorporating additional predictors in the marginal distributions. Similar to the temperature forecasts, the DRN+GCA model results in better forecasts than the DRN+ECC approach, but performs notably worse than the CGM. Additional verification results including assessments of multivariate calibration, significance tests on score differences, and results for other multivariate proper scoring rules are available in the Supplemental Material.

2.5.5 The role of additional inputs for the CGM predictive performance

To assess the importance of incorporating additional input features for the CGM performance, Figure 2.6 includes a CGM variant which only uses ensemble forecasts of the target variable as input. While this CGM variant has access to the same information as the EMOS+ECC and EMOS+GCA models, it generally shows superior predictive performance, indicating improvements of the CGM approach beyond utilizing additional input features only. While the CGM variant without additional inputs typically fails to achieve forecast performance comparable to the DRN-based models, it on average outperforms the DRN+ECC model for wind speed in terms of the variogram score.

2.5.6 CGM sample size

In addition to incorporating arbitrary predictor variables, a particular advantage of the proposed CGM approach over ECC is that the generative procedure allows for generating an arbitrary number of samples from the predictive distribution instead of being limited to the number of ensemble members. To investigate the effect of the size n_{out} of the generated CGM ensemble, Figure 2.7 shows boxplots of the multivariate skill scores as functions of the ensemble size, based on the 100 repetitions of the experiment for $D = 10$.

As before, we repeat the model estimation procedure of the CGM approach 10 times and generate $\frac{n_{\text{out}}}{10}$ samples each time to obtain a final post-processed ensemble of size n_{out}.

Compared to the reference setting of 50 CGM ensemble members, generating a larger sample from the post-processed distributions generally improves the predictive performance, with median improvements in terms of the energy and variogram score of up to around 1.5%. The median skill score values increase notably up to an ensemble size of 200, after which some minor improvements can be observed. Additional results on other values of D are provided in the Supplemental Material.

Given a fixed CGM ensemble size n_{out}, various ensembling strategies for obtaining these n_{out} forecasts could be devised. For example, to obtain 50 CGM members, one could repeat the CGM model estimation 50 (or 25, 10, 5, 2, 1) time(s) and generate 1 (or 2, 5, 10, 25, 50) sample(s) each, respectively. While we found that in general, increasing the number of model runs leads to larger improvements in predictive performance compared to increasing the number of generated samples, this needs to be balanced against the added computational costs for repeating the CGM estimation. More details on the effects of different ensembling strategies are provided in the Supplemental Material.

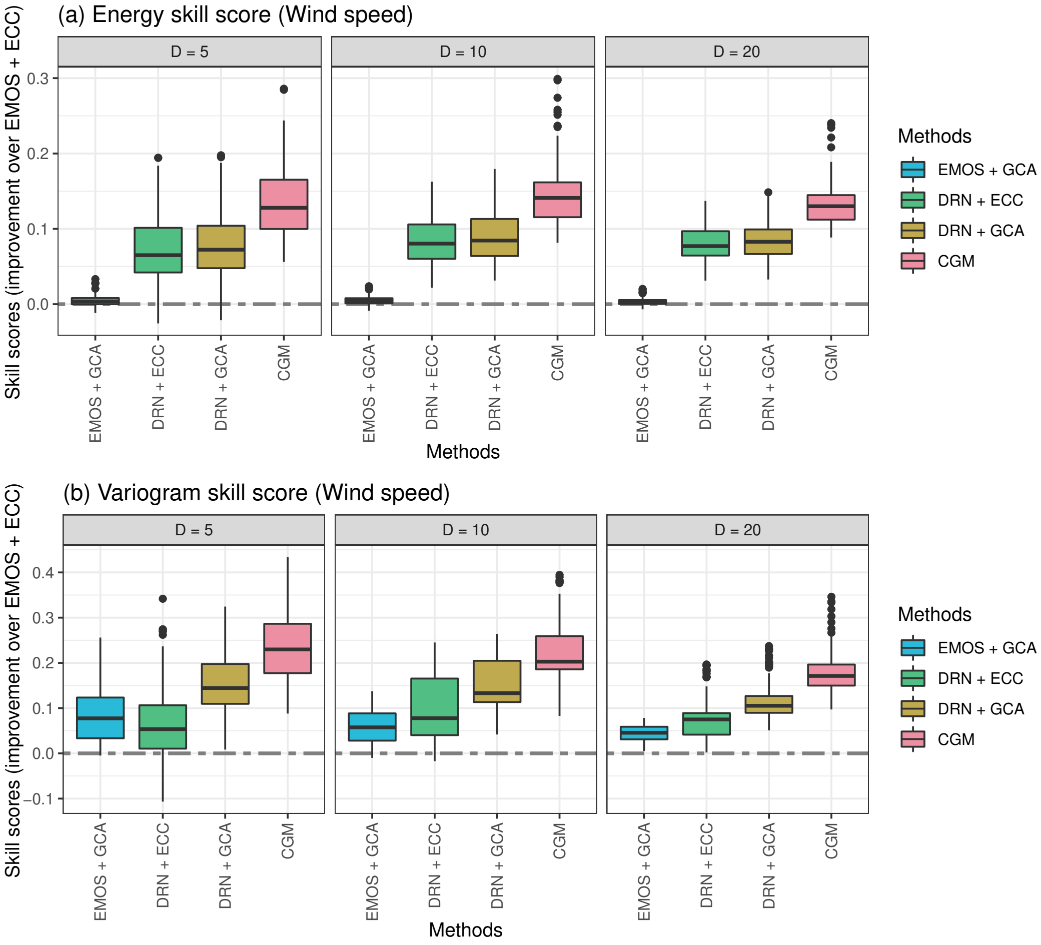


Figure 2.5: As Figure 2.4, but for wind speed.

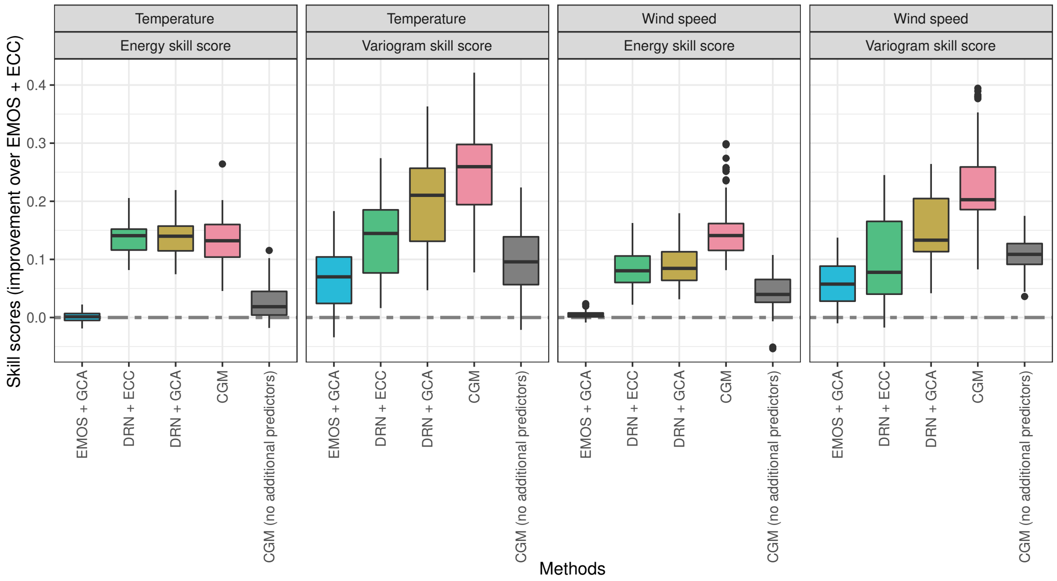


Figure 2.6: Box plots of energy skill scores and variogram skill scores of different multivariate post-processing methods analogous to Figures 2.4 and 2.5 for $D = 10$, but including a CGM variant without additional inputs. EMOS+ECC is used as reference forecast throughout.

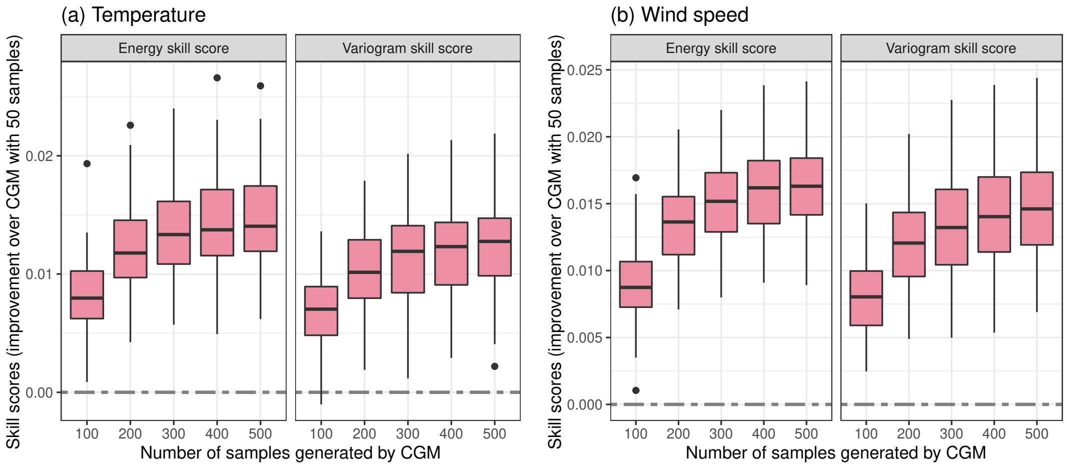


Figure 2.7: Boxplots of energy skill scores and variogram skill scores of CGM forecasts with different numbers of samples generated from the multivariate post-processed forecast distribution for (a) temperature and (b) wind speed over the 100 repetitions of the experiment with different sets of stations. The CGM approach with $n_{\text{out}} = 50$ is used as reference forecast and we only consider the case $D = 10$ here.

2.6 Discussion and conclusions

We propose a nonparametric multivariate post-processing method based on a conditional generative machine learning model which is trained by optimizing a suitable multivariate proper scoring rule. In our CGM approach, an arbitrary number of samples from the multivariate forecast distribution is directly obtained as output of a generative deep neural network which allows for incorporating arbitrary input predictors beyond ensemble predictions of the target variables only. By circumventing the two-step structure of the state-of-the-art multivariate post-processing approaches, the generative model aims to simultaneously correct systematic errors in the marginal distributions and the multivariate dependence structure. By contrast to the standard two-step methods, our CGM approach does not require the choice of parametric models. Furthermore, our CGM architecture can be specifically tailored to incorporate relevant exogenous information and domain knowledge in the different components of the target distribution. For example, our noise encoder module allows for dynamically reparametrizing the latent distributions of the generative model conditional on the standard deviations of the NWP ensemble forecasts to efficiently propagate uncertainty information.

In two case studies on spatial dependencies of temperature and wind speed forecasts at weather stations over Germany, our generative model outperforms state-of-the-art two-step methods for multivariate post-processing where univariate post-processing via DRN models is combined with ECC and GCA. Our CGM approach provides improvements in terms of the univariate forecast performance at individual stations, and produces the best overall multivariate forecasts in terms of the energy score and the variogram score. The observed score differences are statistically significant for a large fraction of the random repetitions of the experiments, even when compared to the best-performing benchmark methods, with details provided in the Supplemental Material. Overall, there are no clear differences in the performance across the considered spatial dimensions of 5, 10 and 20 stations, indicating that the CGM approach works well also in higher-dimensional settings. Regarding the two target variables, we observed more pronounced improvements over the state-of-the-art two-step methods for wind speed, potentially mainly due to larger improvements in the univariate forecast performance. In terms of the two considered multivariate proper scoring rules, the relative improvements are generally larger in terms of the variogram score, indicating that our CGM approach particularly succeeds in better modeling the multivariate dependence structure. The only case where we did not observe notable differences to the performance of the benchmark methods were the results for temperature in terms of the energy score.

The clear improvements in terms of the predictive performance are likely due to key conceptual advantages of our conditional generative models over the two-step approaches, in particular their ability to incorporate arbitrary predictor variables in both the modeling of the marginal distributions and the multivariate dependencies. CGM architectures without additional predictors beyond ensemble forecasts of the target variables reached a better multivariate predictive performance than EMOS+ECC and EMOS+GCA models, but failed to outperform the DRN-based models.

Two minor disadvantages of the CGM approach are on the one hand given by the slightly increased variability across the random repetitions of the experiments due to the generative procedure, which can lead to single outliers with a worse predictive performance. Generating ensembles of CGM predictions can help to alleviate this, in particular with an increased number of sub-ensembles and sample size (Schulz and Lerch, 2022a). Details on different strategies for generating CGM ensembles are discussed in the Supplemental Material. On the other hand, the CGM approach is conceptually somewhat simpler than the two-step methods in that it does not require any parametric assumptions on univariate forecast distributions or multivariate dependencies and produces forecasts in a single step only, however, the computational costs of model training are larger. That said, the computational costs of CGM for multivariate post-processing are still negligible compared to the computational costs of generating the raw ensemble forecasts and will not be a limiting factor in research or operations. For example, for a fixed set of $D = 20$ stations, estimating an ensemble of 10 CGMs and generating 5 samples each takes around 2 minutes on a Nvidia RTX A5000 GPU.

Over the past years, many techniques have been developed in order to better interpret and understand what machine learning methods have learned, in particular for NNs (see McGovern et al., 2019, for an overview from a meteorological perspective). Methods from interpretable machine learning have been applied in the literature on univariate post-processing (Taillardat et al., 2016; Rasp and Lerch, 2018; Schulz and Lerch, 2022b), but the problem is more involved for multivariate post-processing, in particular for the generative models proposed here. While there has been some progress for GANs (Chen et al., 2016; Adel et al., 2018), interpretation is challenging for generative models (Zhou, 2022) and the application of standard methods such as permutation feature importance is not straightforward.

Our results provide several avenues for further generalization and analysis. While we have focused on spatial dependencies across observation stations, it would be interesting to investigate the performance of the CGM approach on a gridded dataset. Motivated by the potential of score-based generative models to achieve comparable performance to GANs in image generation tasks (Song and Ermon, 2020), applications to multivariate post-processing similar to the GAN models proposed in Dai and Hemri (2021) constitute a natural starting point for future work. The CGM approach could further be applied to model temporal or inter-variable dependencies. In addition, while the focus of our case studies was on the multivariate forecast performance, the results presented in Section 2.5.3 indicate that the univariate CGM forecasts show competitive performance even with state-of-the-art NN-based post-processing models applied for the marginal distributions, despite being trained in a multivariate setting. Therefore, it would also be interesting to investigate the potential of the generative models for univariate probabilistic forecasting in more detail, ideally in conjunction with considering theoretical aspects such as the effects of choosing different proper scoring rules for optimization (Pacchiardi et al., 2024). Furthermore, the CGM architecture could be combined with additional predictors that aim to incorporate information from flow-dependent large-scale spatial structures in the raw forecast fields. For example, Lerch and Polsterer (2022) propose convolutional autoencoder NNs to learn low-dimensional representations of spatial forecast fields which could be used as additional CGM inputs to achieve a spatially-informed modeling of multivariate dependencies. Finally, as an alternative two-step strategy for multivariate post-processing, it is also possible to employ generative ML methods to learn conditional copula functions (Janke et al., 2021) in the second step which allow for incorporating arbitrary additional predictors.

The evaluation of multivariate probabilistic forecasts continues to represent an important methodological challenge, despite relevant recent work on multivariate proper scoring rules (Ziel and Berk, 2019; Alexander et al., 2022; Allen et al., 2023). Regarding the evaluation of the CGM forecasts, the question on how to best differentiate between the contributions of improvements in univariate and multivariate components of the overall forecast performance measured via multivariate proper scoring rules is a particular challenge. Another important aspect is the evaluation of multivariate extreme events (Lerch et al., 2017), where recent work from Allen et al. (2023) could serve as a starting point for systematically investigating the effect of the sample size of our CGM and alternative approaches on the ability of the post-processing models to provide reliable and accurate multivariate predictions of extreme events.

2.7 Supplemental Material

2.7.1 Strategies for generating CGM ensembles

While generating ensembles of CGM forecasts improves predictive performance (see Section 5.6 of the main paper), there exist a variety of strategies and configurations to obtain the final n_{out} samples. In the CGM implementation utilized in the main text, we repeated the model estimation 10 times and generated 5 samples each. Here, we investigate the effects of alternative strategies to obtain $n_{\text{out}} = 50$ samples by estimating 50 (or 25, 5, 2, 1) CGM model(s) and generating 1 (or 2, 10, 25, 50) sample(s) each. To that end, we repeat the CGM estimation 50 times and generate 100 samples each. From this large set of CGM predictions, we then randomly select a set of $R \in \{1, 2, 5, 10, 25, 50\}$ models runs and $S \in \{50, 25, 10, 5, 2, 1\}$ corresponding samples such that $R \cdot S = n_{\text{out}} = 50$. For each of these combinations, we compute the mean energy score over the 100 repetitions

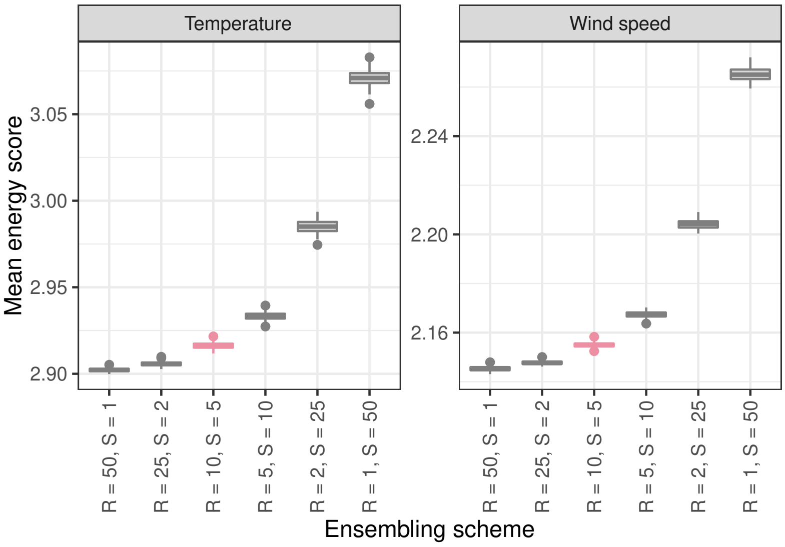


Figure 2.8: Box plots of mean energy scores of different CGM ensembling schemes comprised of R model runs producing S samples each. The mean scores are computed across the 100 sub-sampled station datasets, and the box plots summarize the variability across 100 repetitions of the ensembling procedure by randomly selecting R models runs and S generated samples. Our ensembling strategy from the main text is marked in red.

of the experiment (i.e., the 100 sets of sub-sampled stations) described in the main paper. This procedure is repeated 100 times.

Figure 2.8 shows box plots of the corresponding mean scores and clearly indicates that increasing the number of runs R substantially improves the predictive performance, with the best results obtained for $R = 50$ runs producing a single sample each. These improvements in predictive performance need to be balanced with against the increased computational costs. For example, increasing R from 10 to 50 roughly leads to a 5-fold increase in computational costs, since the computational costs of generating additional samples from an estimated CGM is negligible. However, the mean energy scores typically improve by less than 1%. The configuration we selected ($R = 10, S = 5$) represents a reasonable compromise between computational costs and forecast performance.

Note that multiple sources of randomness contribute to the variability of the performance of the CGM ensembles. Additional tests (not shown) indicate that the variability is dominated by the randomness introduced via repeated model runs via the stochastic training procedure, in comparison to which the randomness due to the sample generation process is negligible.

2.7.2 Ablation studies

We here present several ablation studies to illustrate the effects of different choices regarding the architecture and hyperparameters of our conditional generative model (CGM).

Model hyperparameters

Based on the hyperparameter optimization approach described in the main paper, our CGM implementation utilizes a multivariate normal latent distribution to generate samples of $D_{\text{latent}} = 10$ dimensional samples. The model estimation is based on a batch size of 64, a learning rate of 0.001, and utilizes early stopping with a maximum of 300 epochs and a patience of 10.

In the following, we present several ablation experiments to investigate different choices of hyperparameters. Thereby, we restrict our attention to $D = 10$ and consider a single hyperparameter only, while the rest is kept fixed at the default choice from the main paper.

All boxplots shown in the following summarize mean scores over the 100 repetitions of the sampling procedure over the test set (calendar year 2016), and boxplots in red color indicate the results from our CGM implementation.

Latent distribution Figure 2.9 illustrates the differences between a normal and uniform latent distribution. For both target variables and the two multivariate scores, the differences are very minor.

Latent dimensions The latent dimension, i.e., the number of latent variables has a considerable effect on the computational costs. However, as illustrated in Figure 2.10, different latent dimensions lead to very similar results in terms of the predictive performance of the corresponding generative models.

Learning rate Different choices of the learning rate lead to opposite effects for temperature and wind speed. Figure 2.11 illustrates that a larger learning rate results in slightly better temperature forecasts in terms of both multivariate scores, but a worse performance for wind speed forecasts. During our experiments, we further noted that a larger learning rate can sometimes lead to unstable model training for wind speed post-processing, likely due to the non-negativity constraint.

The effects of normalized input target variable Normalizing inputs to neural network (NN)-based models is generally recommended as a standard pre-processing step to stabilize and accelerate the training process. As described in the main paper, we normalize all meteorological weather variables except for the target variable in the first two CGM modules, according to our initial experiments. Figure 2.12 formally evaluates this choice based on the test data and confirms that normalizing the target variable forecasts has a negative impact on the forecast performance, in particular for wind speed.

The effects of early stopping during training We employ early stopping to avoid overfitting during training. An alternative strategy would be to train for a fixed number of epochs without early stopping. Different choices are compared in Figure 2.13. For both temperature and wind speed, the use of early stopping results in comparable or

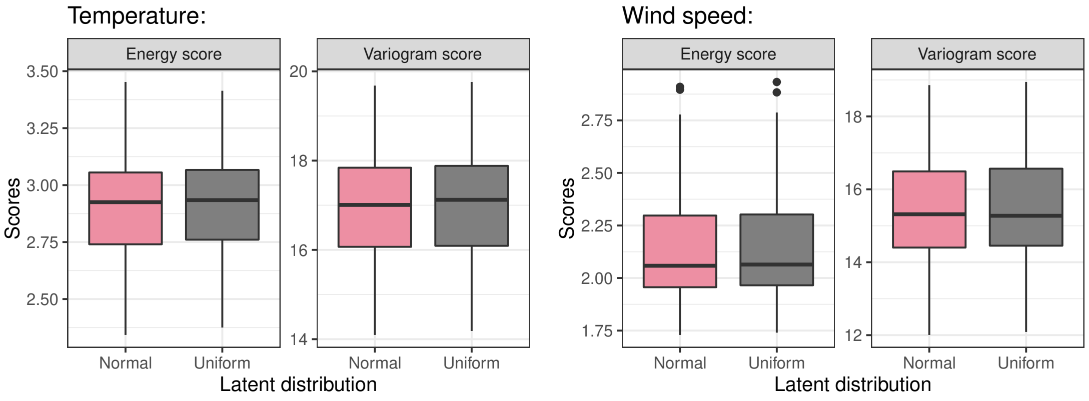


Figure 2.9: Boxplots of mean scores of different choices for the latent distribution.

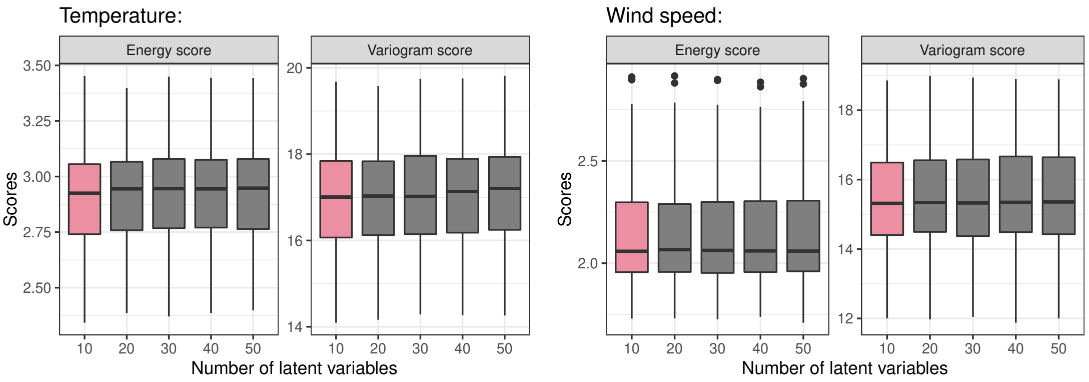


Figure 2.10: Boxplots of mean scores for different numbers of latent variables.

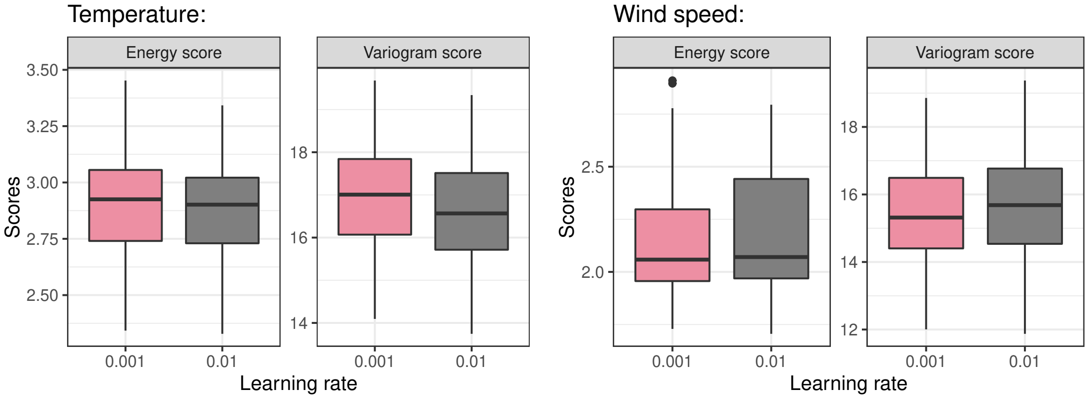


Figure 2.11: Boxplots of mean scores of different learning rates during model training.

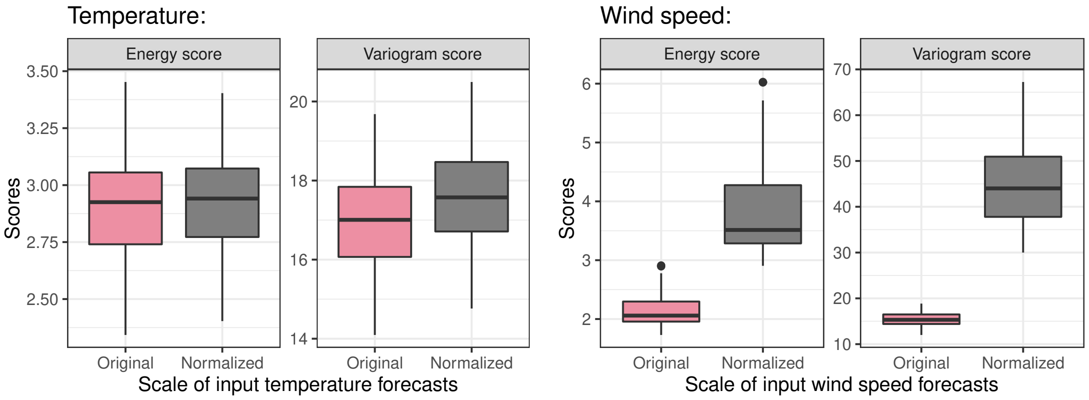


Figure 2.12: Boxplots of mean scores comparing normalized and non-normalized inputs of target variable forecasts.

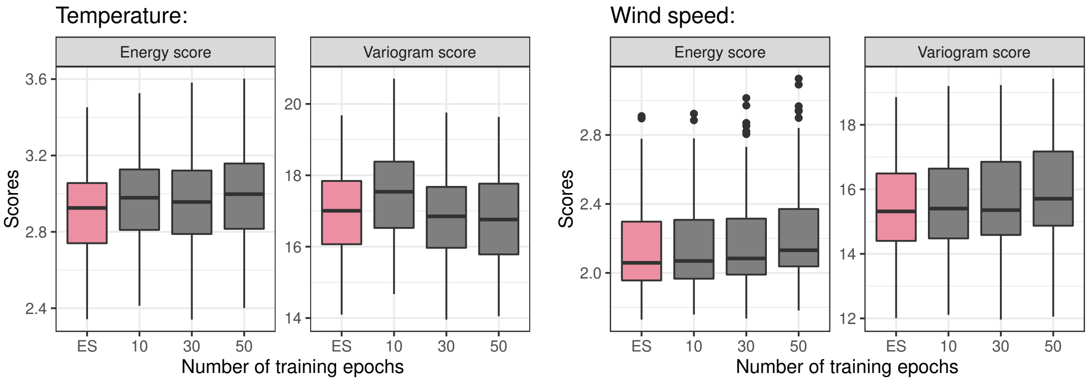


Figure 2.13: Boxplots of mean scores comparing the use of early stopping (‘ES’) or training for a fixed number of epochs.

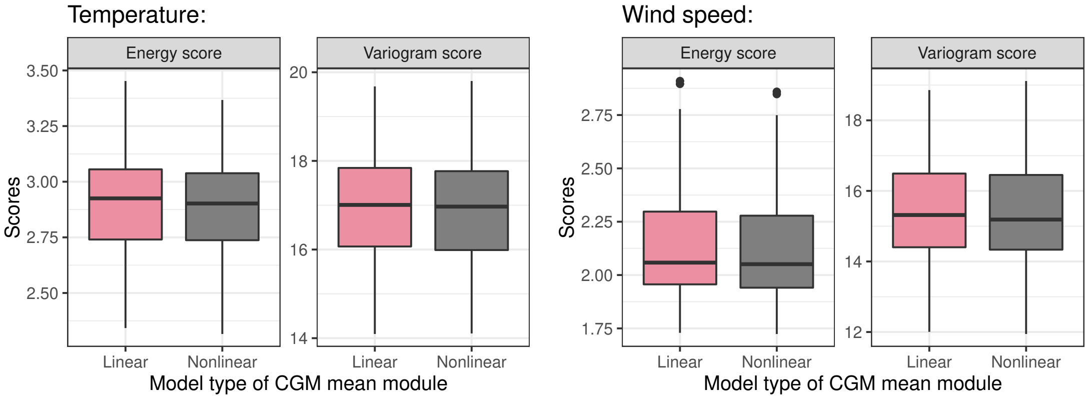


Figure 2.14: Boxplots of mean scores for different variants of the CGM mean module.

better forecasts with the ones trained for a fixed number of epochs. In practice, our CGM model generally stops training after around 20 epochs when applying early stopping.

CGM architecture choices

The effects of a nonlinear model for the mean module As illustrated in the schematic illustration in the Figure 2 of the main paper, we employ a linear model for the mean module of CGM, which implies that the mean component of the output samples is linearly dependent on the means of each meteorological input variable, where the relations (weights) are independent across dimensions. An alternative choice is given by a nonlinear model that utilizes fully connected dense layers for the mean module, similar to the noise decoder module of CGM. The nonlinear model enables the mean component of the output samples at each dimension to depend on the mean values of each meteorological variable in all dimensions. Figure 2.14 compares these two choices, where we utilize one dense layer with a linear activation function for temperature forecasts, and three dense layers (including two hidden layers consisting of 100 nodes) with an elu activation for wind speed. While results are comparable overall, the nonlinear models lead to slightly better multivariate forecasts.

2.7.3 Additional results

Here, we present additional results, e.g., results for other values of D not shown in the main paper, as well as alternative multivariate evaluation metrics. The general experimental setup follows our description in the main paper throughout.

Univariate results

Figure 2.15 presents univariate results analogous to those discussed in Section 5.3 of the main paper, but for $D = 10$ and $D = 20$. While there are minor differences due to random effects in the choices of stations, the results are overall similar, indicating that our CGM approach also provides skillful univariate forecasts, even when trained for higher-dimensional multivariate settings.

Verification rank and probability integral transform histograms are widely used tools to assess the calibration of univariate forecasts, see, e.g., Thorarinsdottir et al. (2016) for

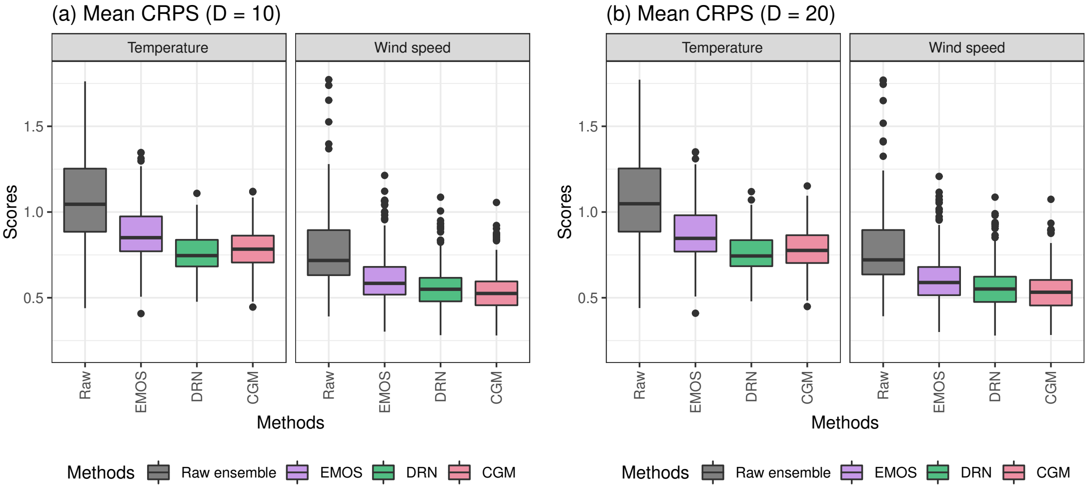


Figure 2.15: Boxplots of mean CRPS values of different multivariate post-processing methods with $D = 10$ and $D = 20$, including the scores of raw ensemble forecasts, analogous to Figure 3 in the main paper.

details. Verification rank histograms for the raw and post-processed forecasts are shown in Figures 2.16 and 2.17. In general, all post-processing methods notably improve the calibration of the under-dispersed ensemble forecasts. Nevertheless, we observe some minor deviations from uniformity in the rank histograms of the post-processed forecasts, most notably in the case of wind speed where the CGM forecasts show clear improvements over EMOS and DRN.

Multivariate results

Energy score and variogram score Figure 2.18 provides additional results on the effect of the CGM sample size for $D = 5$ and $D = 20$. The results are generally very similar to the case of $D = 10$ covered in the main paper, and we do not observe any structural differences.

Diebold-Mariano tests of equal predictive performance We further apply Diebold-Mariano (DM) tests of equal predictive performance (Diebold and Mariano, 1995) to assess the statistical significance of score differences between multivariate post-processing methods. To compare two forecasting methods F and G with based on a scoring

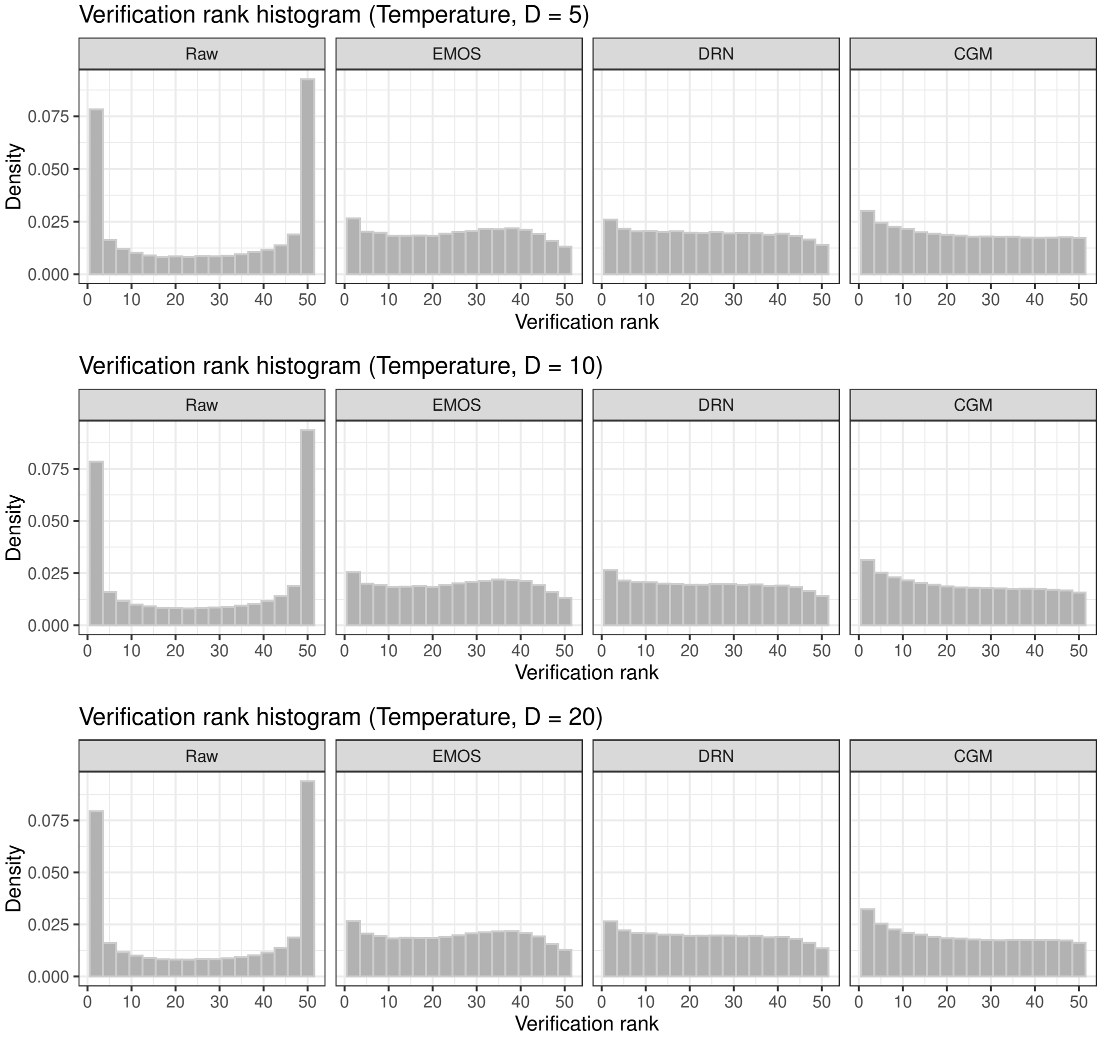


Figure 2.16: Verification rank histograms of different univariate post-processing methods, CGM, and the raw ensemble forecasts.

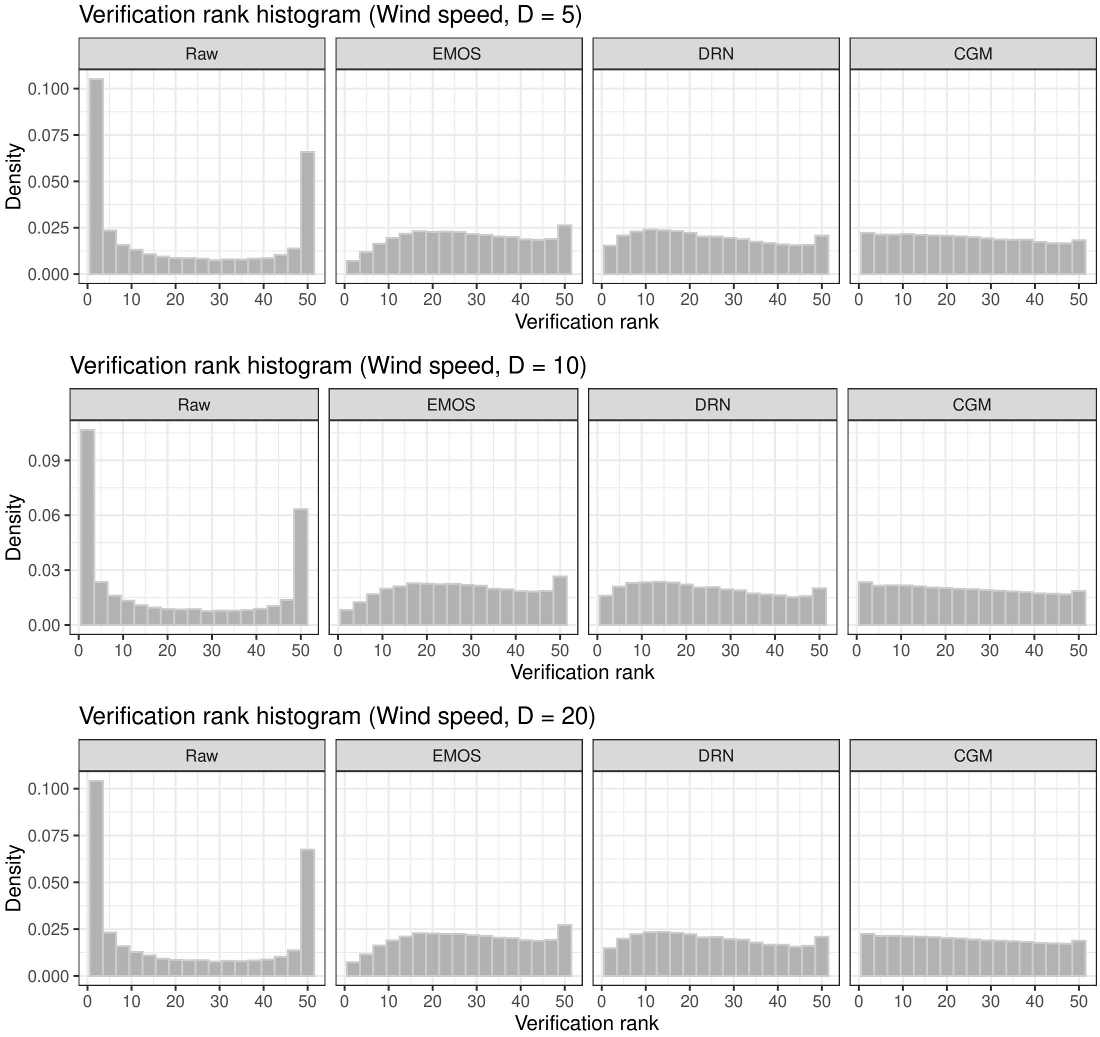


Figure 2.17: As Figure 2.16, but for wind speed.

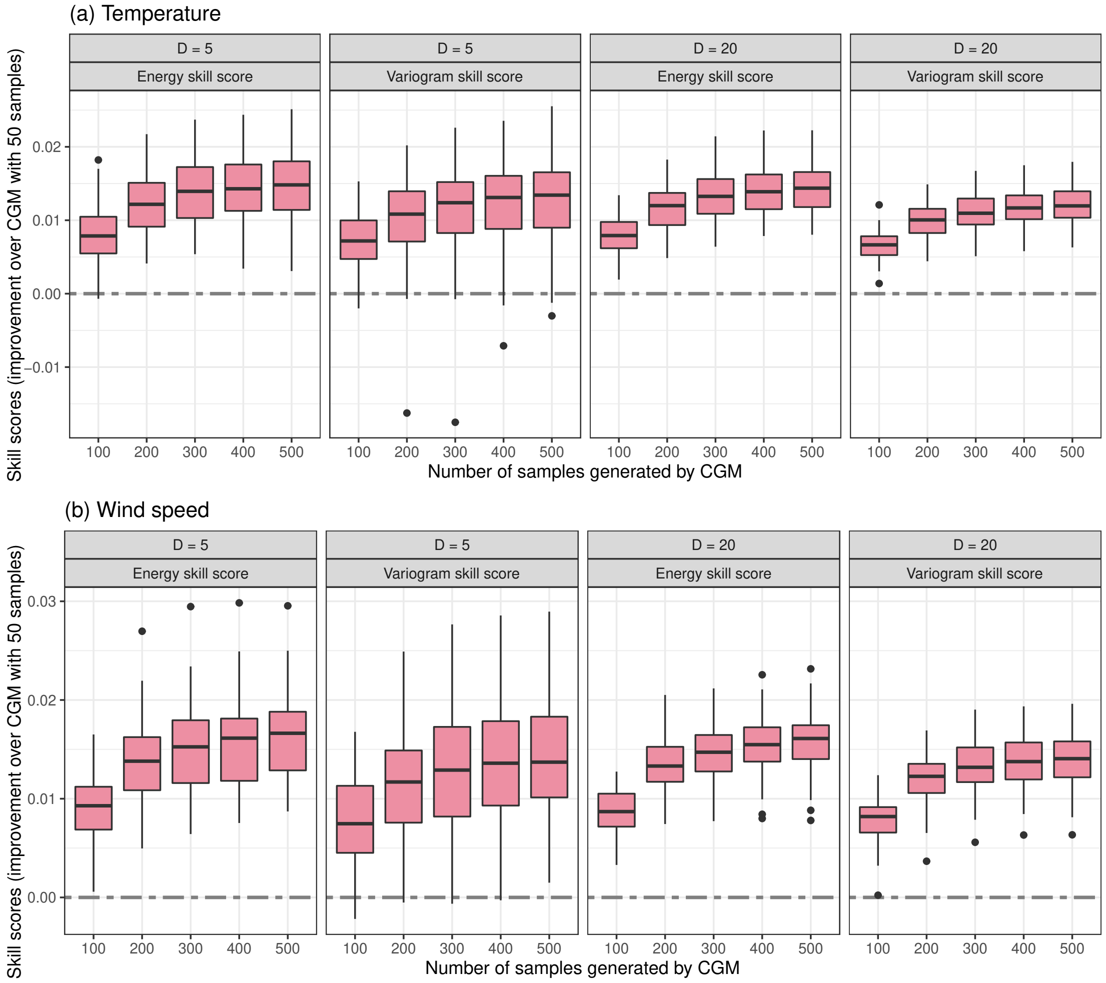


Figure 2.18: Boxplots of the energy skill scores and variogram skill scores of CGM forecasts with different numbers of samples generated from the multivariate post-processed forecast distribution, for (a) temperature and (b) wind speed, over the 100 repetitions of the experiment with different sets of stations. The CGM approach with $n_{\text{out}} = 50$ is used as reference forecast and we consider the cases $D = 5$ and $D = 20$.

rule S and corresponding mean scores $\bar{S}_n^F = \frac{1}{n}\sum_{i=1}^n S(F_i, \boldsymbol{y}_i)$ and $\bar{S}_n^G = \frac{1}{n}\sum_{i=1}^n S(G_i, \boldsymbol{y}_i)$ over n forecast cases, we employ two-sided tests based on the DM test statistic

$$t_n = \sqrt{n}\frac{\bar{S}_n^F - \bar{S}_n^G}{\hat{\sigma}_n}, \quad \text{where} \quad \hat{\sigma}_n^2 = \frac{1}{n}\sum_{i=1}^n \left(S(F_i, \boldsymbol{y}_i) - S(G_i, \boldsymbol{y}_i)\right)^2.$$

Under standard regularity conditions, t_n is asymptotically standard normal under the null hypothesis of equal predictive performance. We use the DM tests with a nominal level of $\alpha = 0.05$ and apply a Benjamini–Hochberg procedure (Benjamini and Hochberg, 1995) to account for multiple testing, see Schulz and Lerch (2022b) for details.

Tables 2.3 and 2.4 show the rejection rates of DM tests of equal predictive performance and thus allow for quantifying the statistical significance of the observed score differences between the multivariate post-processing methods across the repetitions of the experiment. In line with the results from the boxplots of the energy skill scores and variogram skill scores (see Section 5.4 of the main paper), we find that the observed score differences are significant to a large extent. The CGM forecasts show significant improvements over the other methods with the exception of temperature forecasts evaluated with the ES. For example, the null hypothesis of equal predictive performance is rejected in at least 99% of all cases in favor of the CGM forecasts when compared to all two-step approaches for wind speed, where we further do not observe a single case where the CGM is outperformed significantly by any one of the other models. In particular in terms of the ES, the differences between the multivariate re-ordering approaches tend to not be significant, with the notable exception of the EMOS+GCA approach showing comparable performance to the DRN+ECC model in case of wind speed forecasts evaluated with the VS. Tables 2.5, 2.6, 2.7 and 2.8 show the rejection rates of DM tests of equal predictive performance for $D = 5$ and $D = 20$. As before, we do not observe any substantial differences across the choices of D.

In addition to the statistical significance of the observed score differences between different multivariate post-processing methods, we perform similar tests to compare the CGM forecasts based on different number of samples generated from the post-processed multivariate distribution. The corresponding rejection rates of DM test of equal predictive performance are shown in Tables 2.9, 2.10, 2.11, 2.12, 2.13 and 2.14 for all choices of

Table 2.3: Proportion of pair-wise Diebold-Mariano tests for the temperature forecasts indicating statistically significant ES or VS differences after applying a Benjamini-Hochberg procedure to account for multiple testing for a nominal level of 0.05 of the corresponding one-sided tests. The (i, j)-entry in the i-th row and j-th column indicates the proportion of tests where the null hypothesis of equal predictive performance of the corresponding one-sided DM test is rejected in favor of the model in the i-th row when compared to the model in the j-th column. The remainder of the sum of (i, j)- and (j, i)-entry to 1 is the proportion of tests where the score differences are not significant. We consider the case $D = 10$ here.

Energy score

	EMOS+ ECC	EMOS+GCA	DRN+ECC	DRN+GCA	CGM
EMOS+ECC		0.04	0.00	0.00	0.00
EMOS+GCA	0.17		0.00	0.00	0.00
DRN+ECC	1.00	1.00		0.03	0.08
DRN+GCA	1.00	1.00	0.16		0.02
CGM	0.99	1.00	0.13	0.07	

Variogram score

	EMOS+ECC	EMOS+GCA	DRN+ECC	DRN+GCA	CGM
EMOS+ECC		0.00	0.00	0.00	0.00
EMOS+GCA	0.75		0.00	0.00	0.00
DRN+ECC	0.98	0.75		0.00	0.00
DRN+GCA	1.00	1.00	0.95		0.00
CGM	1.00	1.00	0.97	0.84	

Table 2.4: As Table 2.3, but for wind speed.

Energy score

	EMOS+ECC	EMOS+GCA	DRN+ECC	DRN+GCA	CGM
EMOS+ECC		0.00	0.00	0.00	0.00
EMOS+GCA	0.26		0.00	0.00	0.00
DRN+ECC	0.99	0.99		0.01	0.00
DRN+GCA	1.00	1.00	0.21		0.00
CGM	1.00	1.00	1.00	0.99	

Variogram score

	EMOS+ECC	EMOS+GCA	DRN+ECC	DRN+GCA	CGM
EMOS+ECC		0.00	0.00	0.00	0.00
EMOS+GCA	0.89		0.33	0.00	0.00
DRN+ECC	0.84	0.45		0.00	0.00
DRN+GCA	1.00	0.91	0.88		0.00
CGM	1.00	1.00	1.00	0.99	

Table 2.5: As Table 2.3, but for temperature in the case $D = 5$.

	EMOS+ECC	EMOS+GCA	DRN+ECC	DRN+GCA	CGM
Energy score					
EMOS+ECC		0.02	0.00	0.00	0.00
EMOS+GCA	0.08		0.00	0.00	0.00
DRN+ECC	1.00	1.00		0.00	0.06
DRN+GCA	1.00	1.00	0.00		0.00
CGM	1.00	1.00	0.06	0.08	
Variogram score					
EMOS+ECC		0.01	0.00	0.00	0.00
EMOS+GCA	0.75		0.07	0.00	0.00
DRN+ECC	0.86	0.43		0.00	0.00
DRN+GCA	0.98	0.97	0.86		0.00
CGM	0.99	0.99	0.97	0.73	

Table 2.6: As Table 2.3, but for wind speed in the case $D = 5$.

	EMOS+ECC	EMOS+GCA	DRN+ECC	DRN+GCA	CGM
Energy score					
EMOS+ECC		0.00	0.00	0.00	0.00
EMOS+GCA	0.18		0.00	0.00	0.00
DRN+ECC	0.87	0.84		0.00	0.00
DRN+GCA	0.87	0.85	0.27		0.00
CGM	1.00	1.00	0.97	0.96	
Variogram score					
EMOS+ECC		0.00	0.02	0.00	0.00
EMOS+GCA	0.89		0.48	0.00	0.00
DRN+ECC	0.66	0.28		0.00	0.00
DRN+GCA	0.96	0.69	0.87		0.00
CGM	1.00	0.98	1.00	0.92	

Table 2.7: As Table 2.3, but for temperature in the case $D = 20$.

Energy score					
	EMOS+ECC	EMOS+GCA	DRN+ECC	DRN+GCA	CGM
EMOS+ECC		0.00	0.00	0.00	0.00
EMOS+GCA	0.04		0.00	0.00	0.00
DRN+ECC	1.00	1.00		0.04	0.26
DRN+GCA	1.00	1.00	0.03		0.21
CGM	1.00	1.00	0.07	0.07	

Variogram score					
	EMOS+ECC	EMOS+GCA	DRN+ECC	DRN+GCA	CGM
EMOS+ECC		0.00	0.00	0.00	0.00
EMOS+GCA	0.75		0.00	0.00	0.00
DRN+ECC	1.00	0.97		0.00	0.00
DRN+GCA	1.00	1.00	0.98		0.01
CGM	0.99	0.99	0.84	0.67	

Table 2.8: As Table 2.3, but for wind speed in the case $D = 20$.

Energy score					
	EMOS+ECC	EMOS+GCA	DRN+ECC	DRN+GCA	CGM
EMOS+ECC		0.00	0.00	0.00	0.00
EMOS+GCA	0.15		0.00	0.00	0.00
DRN+ECC	1.00	1.00		0.00	0.00
DRN+GCA	1.00	1.00	0.18		0.00
CGM	1.00	1.00	1.00	1.00	

Variogram score					
	EMOS+ECC	EMOS+GCA	DRN+ECC	DRN+GCA	CGM
EMOS+ECC		0.00	0.00	0.00	0.00
EMOS+GCA	0.95		0.25	0.00	0.00
DRN+ECC	0.98	0.63		0.00	0.00
DRN+GCA	1.00	0.99	0.96		0.00
CGM	1.00	1.00	1.00	1.00	

Table 2.9: As Table 2.3, but comparing CGM forecasts with different numbers of samples generated from the multivariate post-processed forecast distribution. Here we consider the temperature forecasts in the case $D = 5$.

Energy score							Variogram score						
#	50	100	200	300	400	500	#	50	100	200	300	400	500
50		0.00	0.00	0.00	0.00	0.00	50		0.00	0.00	0.00	0.00	0.00
100	0.62		0.00	0.00	0.00	0.00	100	0.36		0.00	0.00	0.00	0.00
200	0.83	0.26		0.00	0.00	0.00	200	0.56	0.04		0.00	0.00	0.00
300	0.90	0.52	0.02		0.00	0.00	300	0.70	0.05	0.00		0.00	0.00
400	0.92	0.61	0.06	0.02		0.00	400	0.65	0.19	0.00	0.00		0.00
500	0.92	0.60	0.13	0.05	0.00		500	0.74	0.22	0.00	0.00	0.00	

Table 2.10: As Table 2.9, but for wind speed.

Energy score							Variogram score						
#	50	100	200	300	400	500	#	50	100	200	300	400	500
50		0.00	0.00	0.00	0.00	0.00	50		0.00	0.00	0.00	0.00	0.00
100	0.88		0.00	0.00	0.00	0.00	100	0.42		0.00	0.00	0.00	0.00
200	0.97	0.61		0.00	0.00	0.00	200	0.67	0.23		0.00	0.00	0.00
300	0.99	0.79	0.04		0.00	0.00	300	0.72	0.22	0.00		0.00	0.00
400	1.00	0.82	0.23	0.00		0.00	400	0.72	0.27	0.03	0.00		0.00
500	1.00	0.87	0.26	0.00	0.04		500	0.77	0.28	0.04	0.03	0.00	

Table 2.11: As Table 2.9, but for temperature in the case $D = 10$.

Energy score							Variogram score						
#	50	100	200	300	400	500	#	50	100	200	300	400	500
50		0.00	0.00	0.00	0.00	0.00	50		0.00	0.00	0.00	0.00	0.00
100	0.81		0.00	0.00	0.00	0.00	100	0.74		0.00	0.00	0.00	0.00
200	0.95	0.53		0.00	0.00	0.00	200	0.90	0.47		0.00	0.00	0.00
300	0.97	0.60	0.04		0.00	0.00	300	0.92	0.60	0.05		0.00	0.00
400	0.98	0.61	0.16	0.00		0.00	400	0.91	0.63	0.02	0.00		0.00
500	0.98	0.70	0.13	0.00	0.00		500	0.90	0.71	0.00	0.00	0.00	

Table 2.12: As Table 2.9, but for wind speed in the case $D = 10$.

Energy score							Variogram score						
#	50	100	200	300	400	500	#	50	100	200	300	400	500
50		0.00	0.00	0.00	0.00	0.00	50		0.00	0.00	0.00	0.00	0.00
100	0.98		0.00	0.00	0.00	0.00	100	0.90		0.00	0.00	0.00	0.00
200	1.00	0.83		0.00	0.00	0.00	200	0.94	0.66		0.00	0.00	0.00
300	1.00	0.94	0.32		0.00	0.00	300	0.98	0.77	0.14		0.00	0.00
400	1.00	0.95	0.59	0.08		0.00	400	0.99	0.83	0.31	0.05		0.00
500	1.00	0.96	0.60	0.17	0.01		500	0.99	0.87	0.38	0.08	0.00	

Table 2.13: As Table 2.9, but for temperature in the case $D = 20$.

Energy score							Variogram score						
#	50	100	200	300	400	500	#	50	100	200	300	400	500
50		0.00	0.00	0.00	0.00	0.00	50		0.00	0.00	0.00	0.00	0.00
100	0.87		0.00	0.00	0.00	0.00	100	0.98		0.00	0.00	0.00	0.00
200	0.99	0.64		0.00	0.00	0.00	200	1.00	0.70		0.00	0.00	0.00
300	1.00	0.79	0.12		0.00	0.00	300	1.00	0.88	0.15		0.00	0.00
400	1.00	0.86	0.23	0.01		0.00	400	1.00	0.89	0.40	0.00		0.00
500	1.00	0.92	0.39	0.04	0.00		500	1.00	0.90	0.54	0.08	0.01	

Table 2.14: As Table 2.9, but for wind speed in the case $D = 20$.

Energy score							Variogram score						
#	50	100	200	300	400	500	#	50	100	200	300	400	500
50		0.00	0.00	0.00	0.00	0.00	50		0.00	0.00	0.00	0.00	0.00
100	1.00		0.00	0.00	0.00	0.00	100	0.97		0.00	0.00	0.00	0.00
200	1.00	0.94		0.00	0.00	0.00	200	0.99	0.92		0.00	0.00	0.00
300	1.00	0.97	0.33		0.00	0.00	300	1.00	0.93	0.36		0.00	0.00
400	1.00	0.97	0.59	0.09		0.00	400	1.00	0.96	0.57	0.07		0.00
500	1.00	0.98	0.69	0.22	0.03		500	1.00	0.99	0.69	0.32	0.02	

D, and for both target variables and multivariate scores. Generating a larger number of samples from the CGM post-processed distribution generally leads to significant improvements over the default setup with 50 samples, in particular in higher-dimensional settings. As expected, the significance of the score differences diminishes when comparing to a larger number of CGM samples, for example the score differences between 500 and 400 sample-based CGM variants rarely are significant in only a small proportion of the cases.

Copula scores As noted in the main paper, a major challenge in the evaluation of multivariate forecasts is the distinction between contributions of improvements in the marginal distributions and the multivariate dependencies to the overall forecast quality. The univariate results presented above and in the main paper indicate that the DRN-based multivariate post-processing methods perform notably better in the calibration of marginal distributions than the EMOS-based methods, which is reflected on the improvement of multivariate scores given that they employ the same copula reordering (ECC or GCA) approach. For CGM forecasts, it is not trivial to assess to what extent the dependence structure improves the scores when compared with other methods.

Ziel and Berk (2019) propose the copula energy score and the copula variogram score as a novel approach to address this challenge. The copula scores are a new class of multivariate proper scoring rules that focus on the dependency structure of the multivariate forecast distribution. For more exact definitions, as well as further mathematical details and illustrations we refer to Ziel and Berk (2019).

We apply the copula energy score and the copula variogram score to assess the forecast performance of different multivariate post-processing methods, and the corresponding skill scores taking EMOS+ECC as reference are shown in Figures 2.19 and 2.20. In general, the performance of copula-based methods depends on the corresponding copulas, with GCA notably outperforming ECC. The choice of the univariate post-processing method only has a minor effect on the performance of copula-based methods (as expected), with DRN perhaps surprisingly leading to slightly worse results than EMOS. CGM shows comparable or slightly better performance than the GCA-based approaches, which

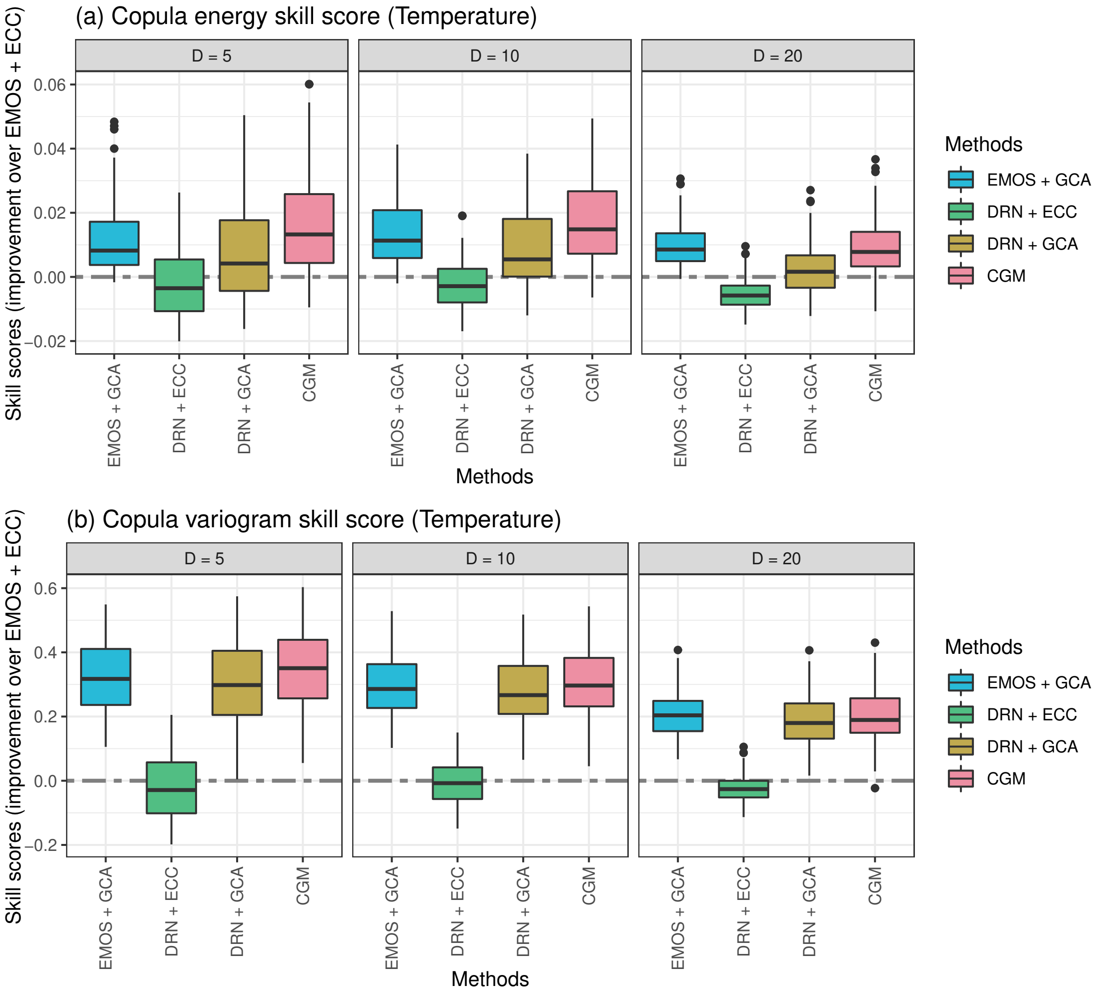


Figure 2.19: Boxplots of (a) copula energy skill scores and (b) copula variogram skill scores of different multivariate post-processing methods for temperature across the 100 repetitions of the experiment with different sets of stations. EMOS+ECC is used as reference forecast in both cases.

suggests that the dependence structures of the CGM forecasts better match those in the observations, when compared with the best-performing GCA-based approaches.

Weighted multivariate scores As discussed in the main paper, one benefit from the CGM approach is the option to efficiently generate a large number of samples from the post-processed multivariate distribution. This should in principle prove advantageous in correctly representing multivariate extreme events. To assess the multivariate performance for predicting extreme events we apply the weighted multivariate scoring rules proposed by Allen et al. (2023). Specifically, we consider the threshold-weighted energy score and variogram score in both their localizing ("twES-loc", "twVS-loc") and non-localizing ("twES", "twVS") variants, as well as the vertically re-scaled energy score and variogram score ("vrES", "vrVS"). For the exact definitions, and detailed mathematical illustration and simulation studies we refer to Allen et al. (2023). As a threshold, we here select the 95%-quantile of the observed values to determine extreme events. This corresponds to temperatures exceeding 17.5°C and wind speeds exceeding 8.0 m/s. Note that we here choose fixed thresholds to gain an overall perspective of the multivariate performance. In a more detailed future study, it might be interesting to account for seasonally adjusted and location-specific climatologies when determining the threshold values.

The mean scores over the 100 experiments are shown in Tables 2.15 and 2.16. We here additionally include the results for the CGM forecast with 500 samples for comparison. While the results show some variations across the two variables and the different weighted scores, the CGM variant with 500 samples generally provide the best forecasts. By contrast, the default CGM variant with 50 samples usually performs very similar to the best of the two-step methods (DRN+GCA) for temperature, and slightly better than DRN+GCA for wind speed.

Multivariate rank histograms Thorarinsdottir et al. (2016) propose several extensions of univariate rank histograms towards multivariate versions that allow for assessing multivariate calibration. We here consider the multivariate rank, average rank and band depth rank histograms, and refer to Thorarinsdottir et al. (2016) for the exact definitions and details of the interpretation of different histogram shapes.

Table 2.15: Mean weighted multivariate scores of different post-processing methods for temperature, averaged over the 100 repetitions of the simulation experiment. The weighted versions of the energy score (ES) and the variogram score (VS) are considered. We scale the scores by 10 where indicated by "×10" for better interpretation. The best scores are highlighted in bold.

Score	Raw ens.	EMOS+ ECC	EMOS+ GCA	DRN+ ECC	DRN+ GCA	CGM	CGM (500 samples)
				$D = 5$			
ES	2.81	2.27	2.27	1.97	1.97	1.97	**1.94**
twES	0.172	0.143	0.143	0.130	0.130	0.129	**0.127**
twES-loc (×10)	1.18	0.784	0.756	0.726	0.698	0.704	**0.690**
vrES	1.04	0.552	0.522	0.502	0.473	0.477	**0.468**
VS	8.22	4.81	4.36	4.12	3.74	3.50	**3.46**
twVS	0.876	0.699	0.644	0.590	0.566	0.561	**0.550**
vrVS	0.367	0.308	0.300	0.295	0.282	0.283	**0.277**
				$D = 10$			
ES	4.22	3.37	3.37	2.91	2.90	2.91	**2.87**
twES	0.254	0.210	0.208	0.187	0.187	0.185	**0.182**
twES-loc (×10)	1.44	0.715	0.671	0.654	**0.601**	0.626	0.617
vrES	1.25	0.434	0.398	0.400	**0.353**	0.366	0.360
VS	39.0	22.6	21.0	19.5	18.0	16.9	**16.7**
twVS	3.95	3.11	2.90	2.61	2.54	2.45	**2.41**
vrVS	1.36	0.865	0.820	0.815	**0.752**	0.767	0.756
				$D = 20$			
ES	6.09	4.87	4.87	4.21	4.22	4.26	**4.20**
twES	0.367	0.300	0.298	0.272	0.272	0.272	**0.267**
twES-loc (×10)	1.85	0.640	0.592	0.584	**0.527**	0.570	0.563
vrES	1.56	0.384	0.346	0.339	**0.292**	0.321	0.317
VS	153	96.7	92.8	85.0	80.7	77.8	**76.9**
twVS	15.8	12.6	12.1	10.8	10.7	10.5	**10.4**
vrVS	5.06	2.29	2.19	2.15	**1.97**	2.09	2.07

Table 2.16: As Table 2.15, but for wind speed.

Score	Raw ens.	EMOS+ ECC	EMOS+ GCA	DRN+ ECC	DRN+ GCA	CGM	CGM (500 samples)
$D = 5$							
ES	2.44	1.69	1.68	1.56	1.55	1.44	**1.42**
twES	0.230	0.168	0.166	0.149	0.149	0.133	**0.131**
twES-loc (×10)	0.460	0.337	0.337	0.330	0.309	0.304	**0.301**
vrES (×10)	1.40	0.955	0.966	0.941	0.883	0.860	**0.853**
VS	9.49	4.37	4.00	4.01	3.66	3.31	**3.26**
twVS	1.5	0.978	0.960	0.810	0.800	0.687	**0.675**
vrVS	0.221	0.178	0.179	0.167	0.160	0.160	**0.157**
$D = 10$							
ES	3.67	2.55	2.53	2.31	2.30	2.16	**2.12**
twES	0.504	0.369	0.365	0.323	0.323	0.288	**0.284**
twES-loc (×10)	0.541	0.325	0.334	0.311	**0.276**	0.314	0.313
vrES (×10)	1.54	0.759	0.775	0.771	**0.648**	0.754	0.757
VS	39.7	20.2	19.0	18.0	16.9	15.4	**15.2**
twVS	9.11	5.87	5.79	4.75	4.73	3.97	**3.92**
vrVS	0.866	0.506	0.518	0.491	**0.422**	0.514	0.517
$D = 20$							
ES	5.04	3.52	3.51	3.23	3.22	3.04	**2.99**
twES	0.587	0.435	0.428	0.386	0.387	0.352	**0.346**
twES-loc (×10)	0.266	0.118	0.121	0.124	0.110	0.101	**0.0959**
vrES (×10)	0.865	0.271	0.275	0.290	0.249	0.228	**0.215**
VS	153	82.5	78.9	75.6	72.3	67.0	**66.0**
twVS	28.3	18.8	18.6	15.5	15.4	13.3	**13.1**
vrVS	1.34	0.612	0.628	0.619	0.557	0.511	**0.480**

Figures 2.21, 2.22, 2.23, 2.24, 2.25 and 2.26 show the corresponding histograms. Overall, the CGM forecasts and the GCA-based forecasts show superior multivariate calibration performance to ECC-based forecasts, but neither of them consistently achieves uniformly distributed histogram across all types of ranks. In terms of the band depth rank, the ECC-based approaches lead to an over-estimation of the correlation among the ensemble members for both temperature and wind speed forecasts, which can also be observed in the average rank histograms. In general, there is no multivariate post-processing method that performs well consistently over all types of ranks, indicating that rankings of the different models will strongly depend on the employed notion of multivariate calibration.

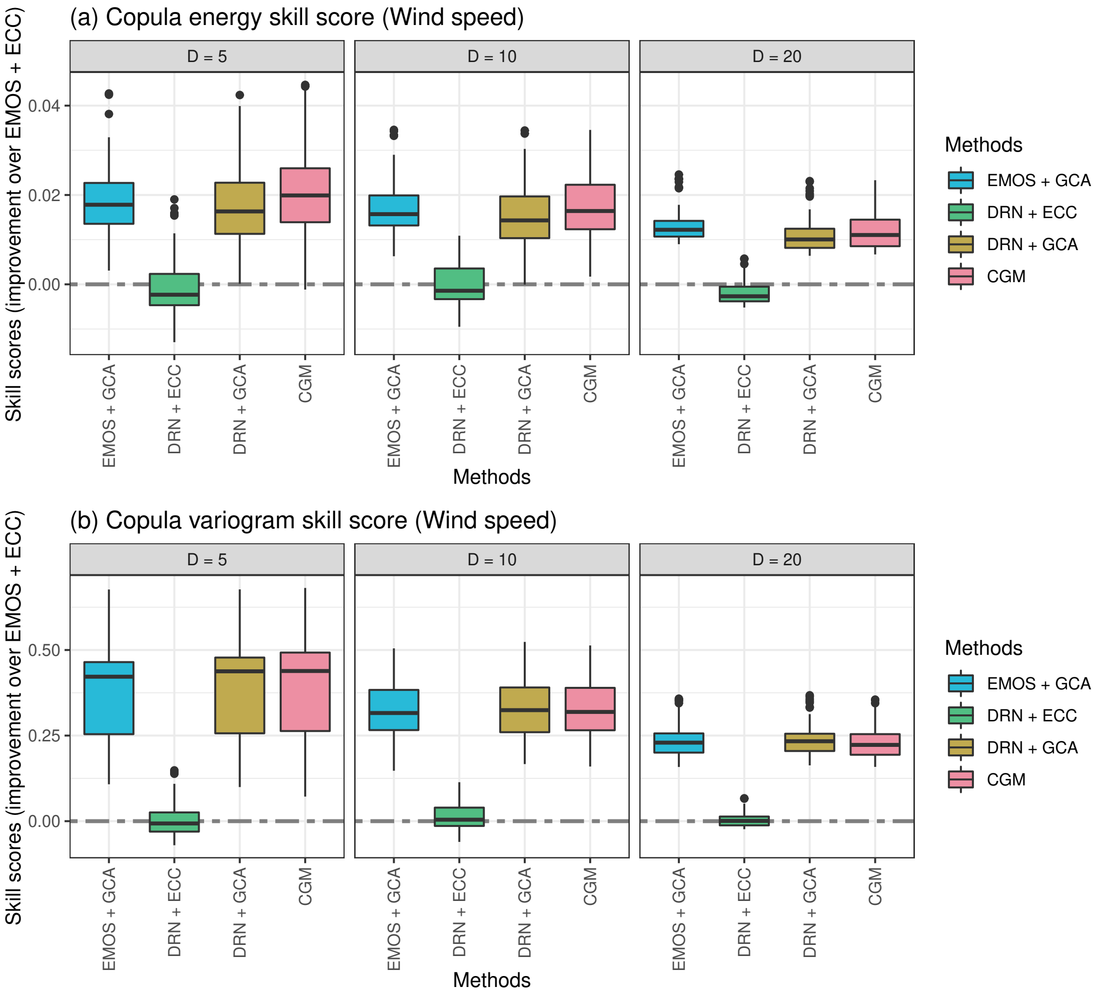


Figure 2.20: As Figure 2.19, but for wind speed.

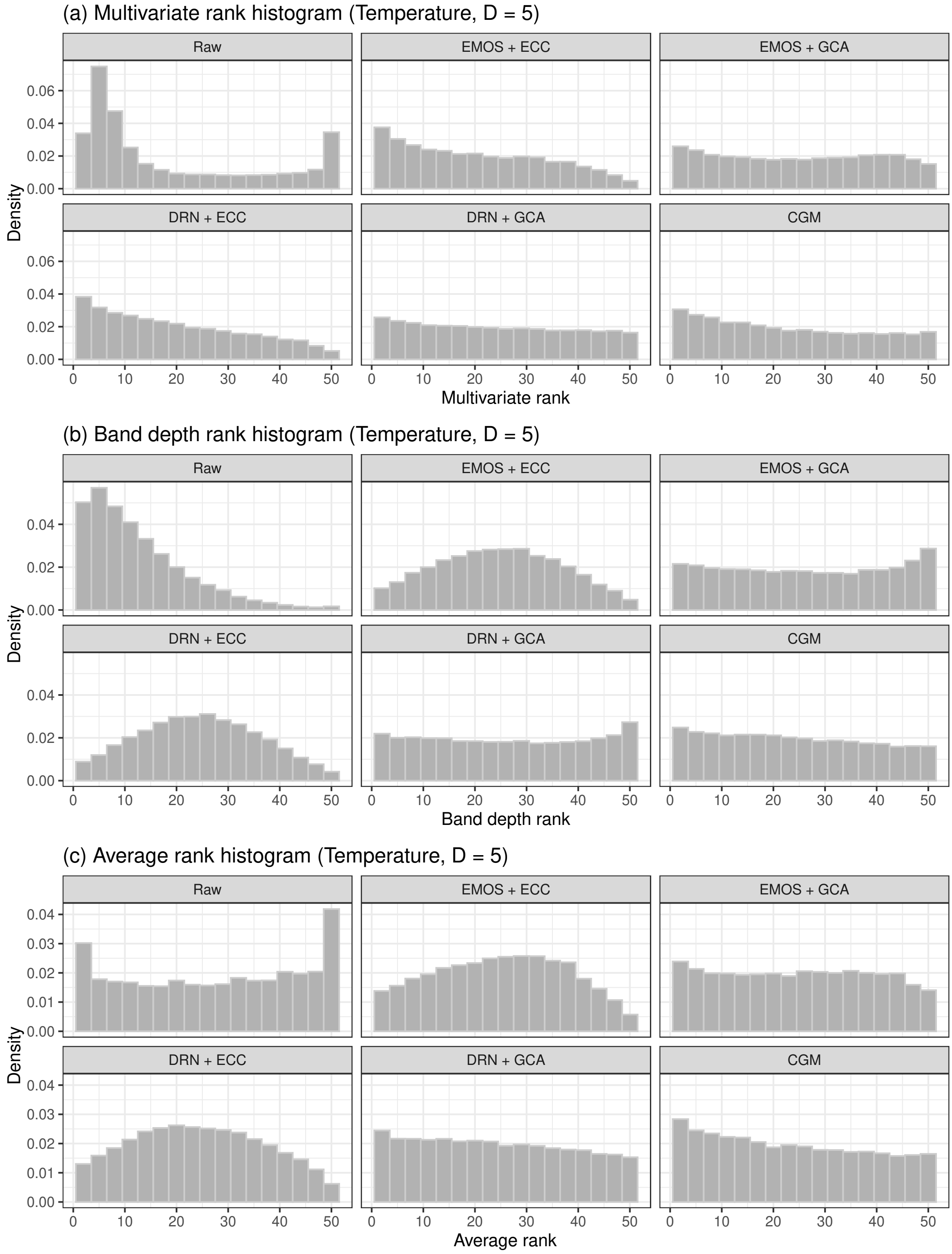


Figure 2.21: Histograms of (a) the multivariate rank, (b) the band depth rank, and (c) the average rank of different multivariate post-processing methods and the raw ensemble forecasts for temperature, across the 100 repetitions of the simulation experiment and the 366 days in the test set (calendar year 2016). We consider the case $D = 5$ here.

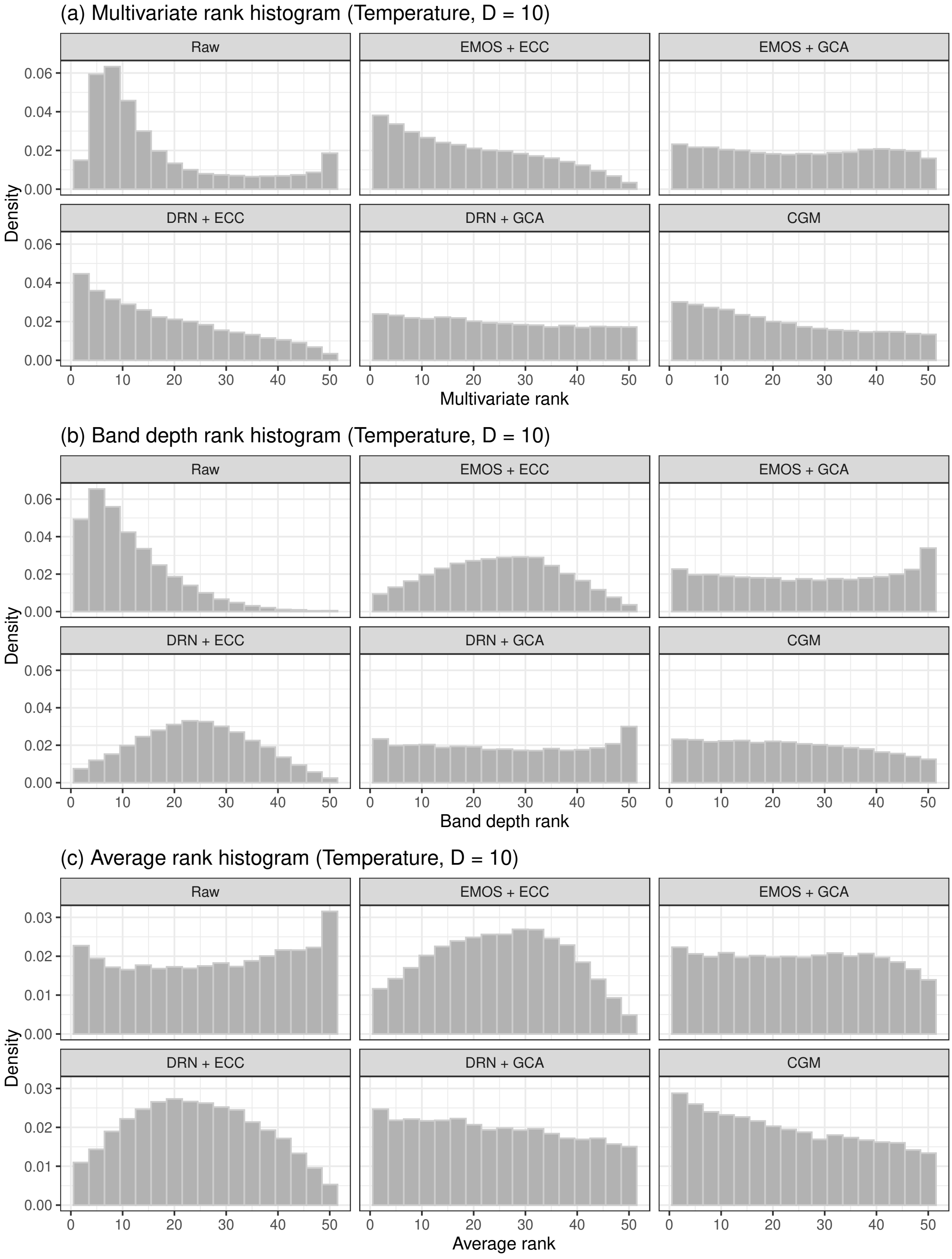


Figure 2.22: As Figure 2.21, but for temperature and $D = 10$.

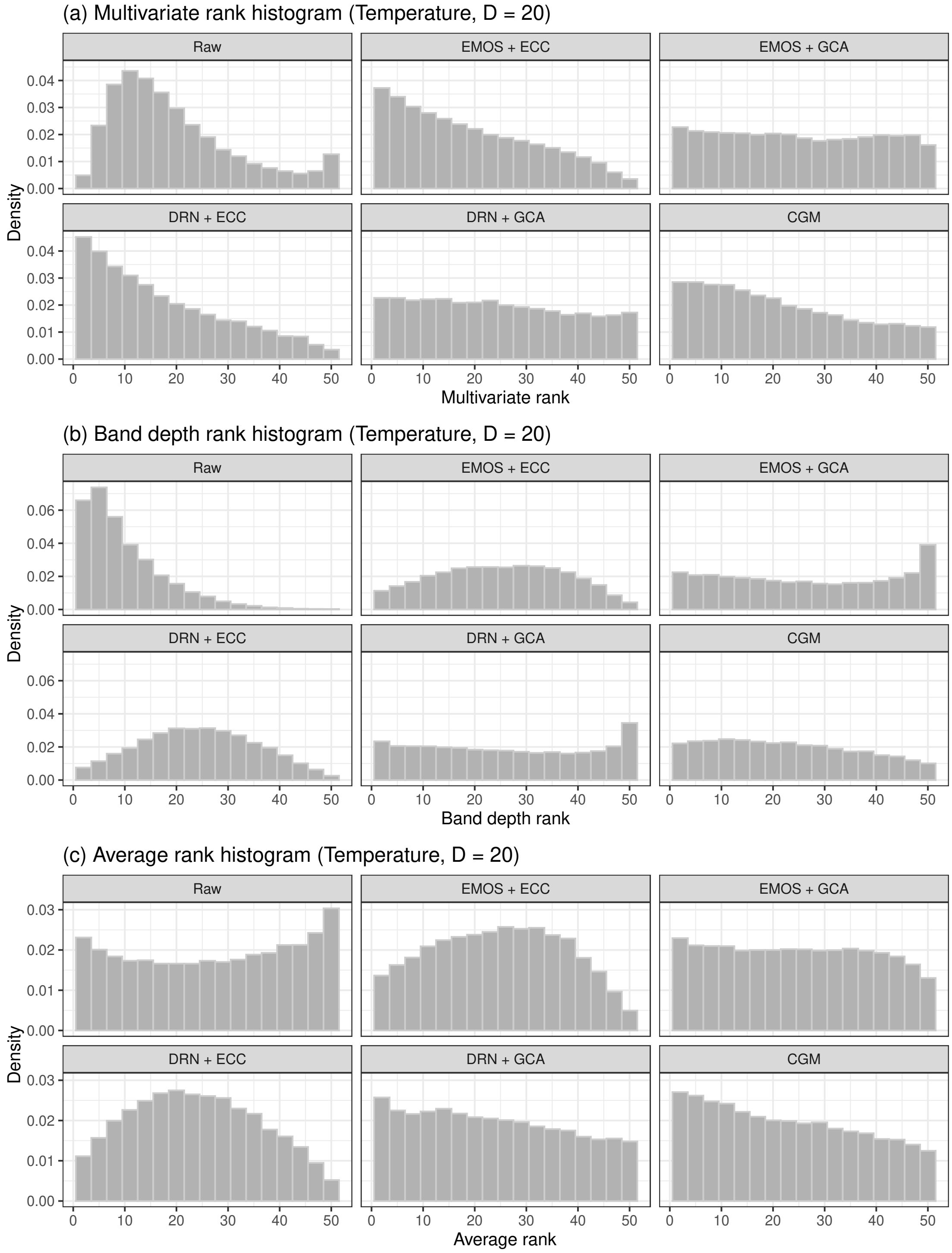
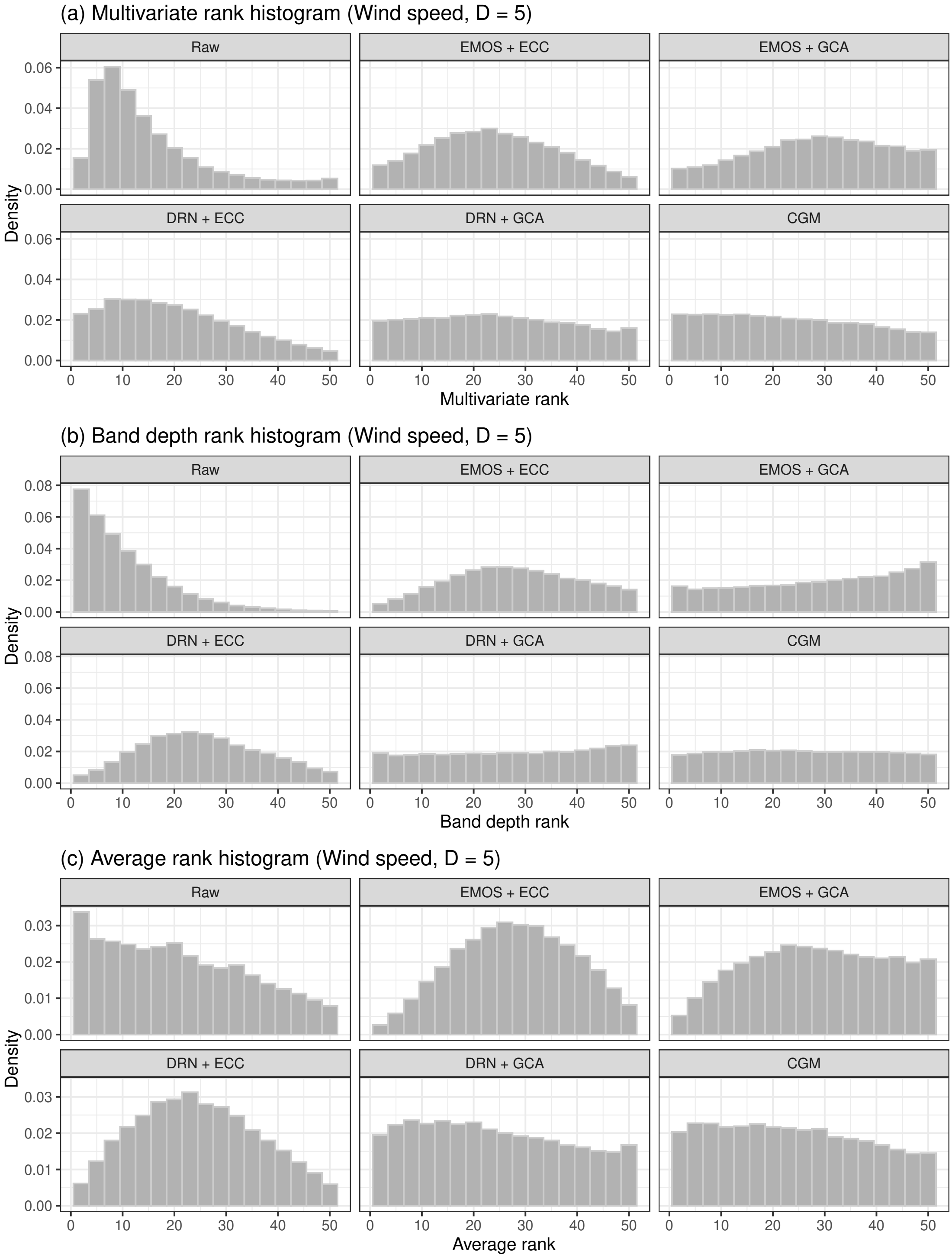


Figure 2.23: As Figure 2.21, but for temperature and $D = 20$.



(a) Multivariate rank histogram (Wind speed, D = 5)

(b) Band depth rank histogram (Wind speed, D = 5)

(c) Average rank histogram (Wind speed, D = 5)

Figure 2.24: As Figure 2.21, but for wind speed.

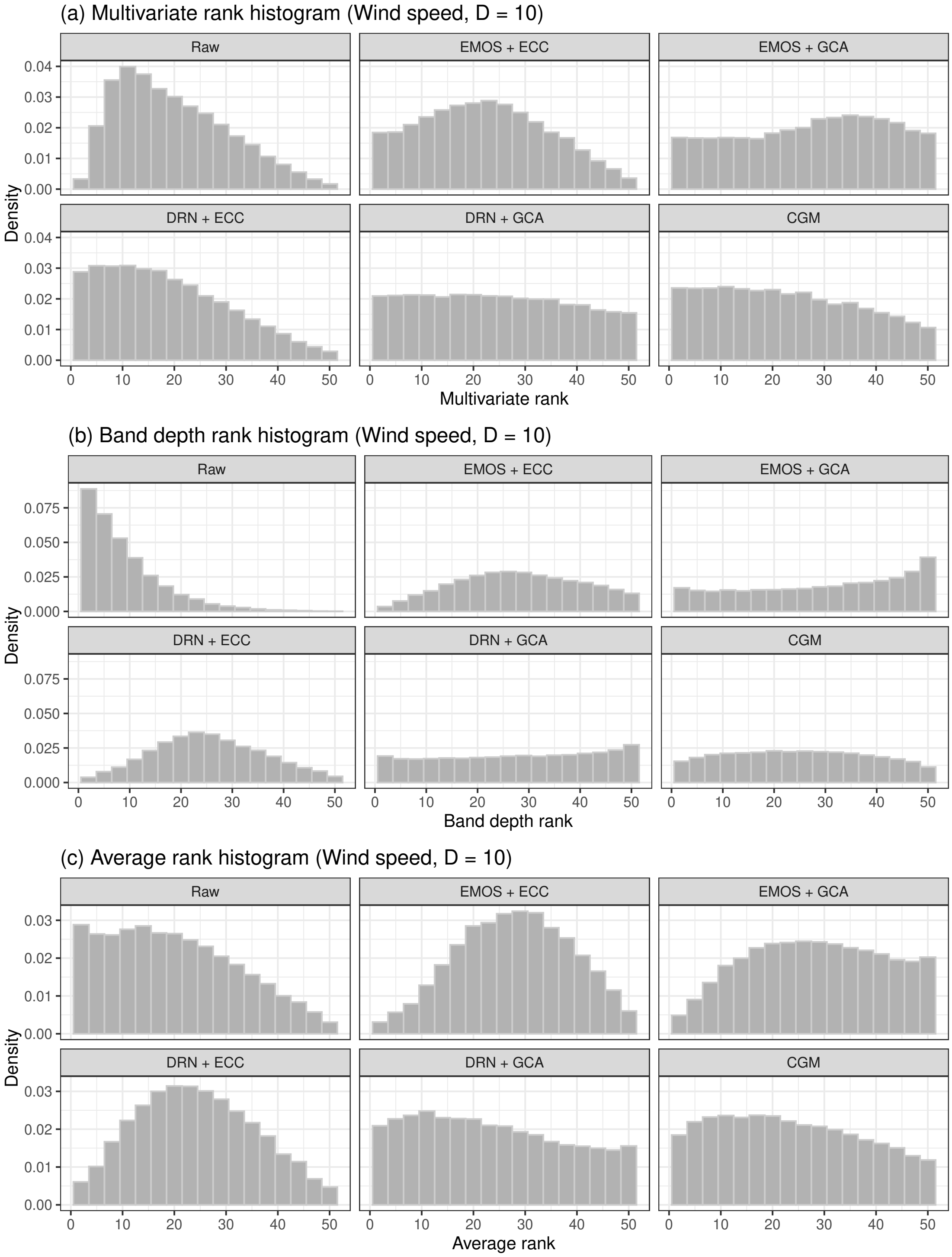


Figure 2.25: As Figure 2.21, but for wind speed and $D = 10$.

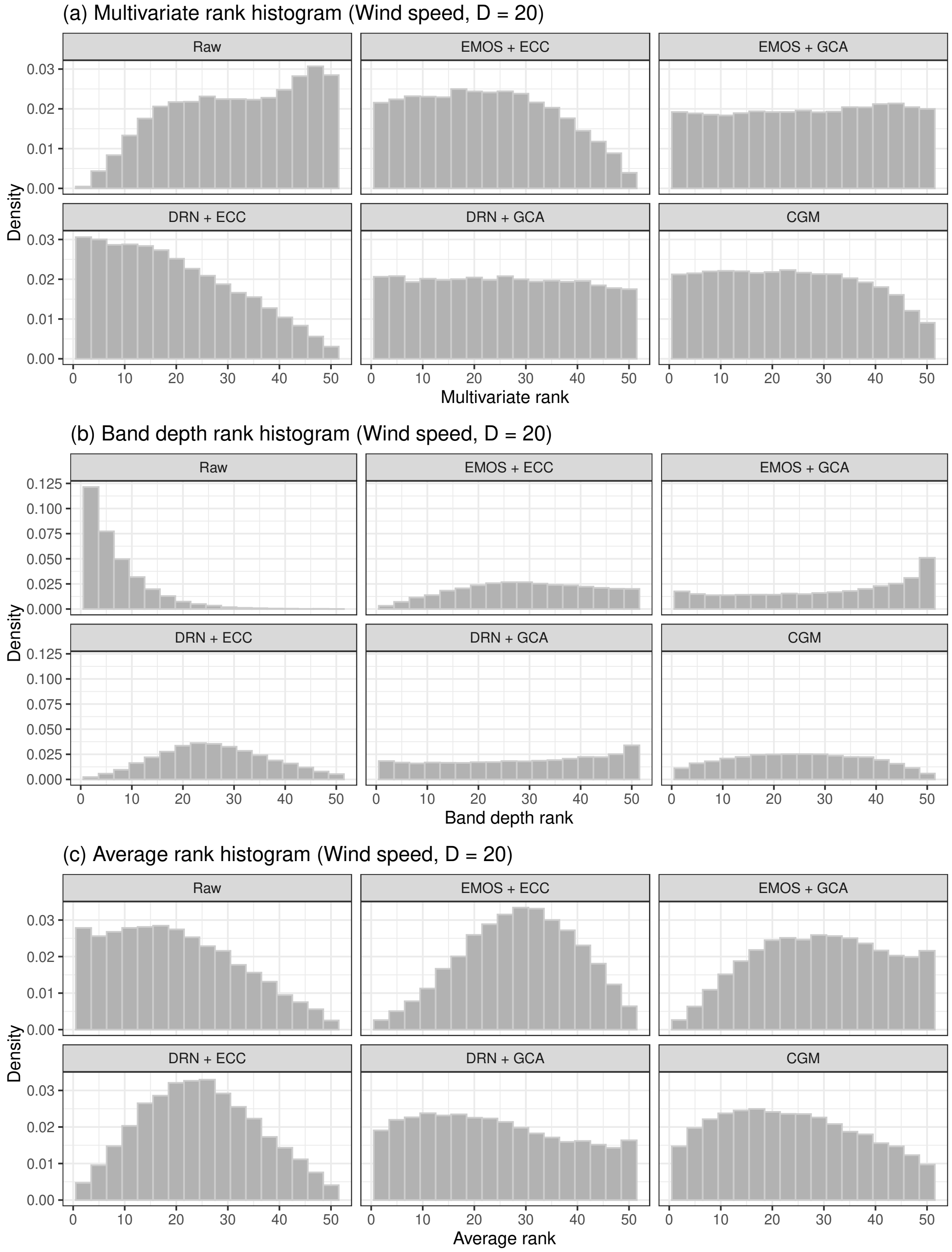


Figure 2.26: As Figure 2.21, but for wind speed and $D = 20$.

3 Learning low-dimensional representations of ensemble forecast fields using autoencoder-based methods

3.1 Introduction

Large-scale physics-based models are used across environmental sciences for prediction and modeling. A particularly important example is numerical weather prediction (NWP) models, where atmospheric processes are represented via partial differential equations. The forecast quality of NWP models has improved tremendously in recent decades due to continued scientific and technological advances (Bauer et al., 2015). Nowadays, NWP models are often run in an ensemble mode to quantify forecast uncertainty. Thereby, a collection of predictions of future weather states is obtained by running the model several times with varying initial conditions and perturbed model physics. Conceptually, these ensemble members can be considered equally likely realizations of an unknown probability distribution.

Due to their continuously increasing spatial and temporal resolution, ensemble weather forecasting models produce large amounts of data. However, such high-dimensional and complex data can be challenging to process in applications relying on weather predictions as inputs. Examples include weather forecasting applications such as post-processing and analog forecasting, and downstream applications such as hydrological and energy forecasting models. Therefore, summarizing relevant information from meteorological input data across space and time via learning low-dimensional representations is of interest beyond just reducing the amount of data that needs to be stored.

One example is ensemble post-processing, which aims at correcting systematic errors of NWP ensemble predictions via statistical or machine learning (ML) models (Vannitsem et al., 2021). Post-processing models use ensemble predictions of relevant meteorological variables as inputs, and produce corrected probabilistic forecasts in the form of probability distributions as their output. While recent ML-based approaches have enabled the incorporation of many predictor variables (Rasp and Lerch, 2018; Schulz and Lerch, 2022b; Chen et al., 2024) and there exist first spatial post-processing approaches (Scheuerer et al., 2020; Grönquist et al., 2021; Veldkamp et al., 2021; Chapman et al., 2022; Horat and Lerch, 2024), most post-processing models still tend to operate on localized predictions at individual stations or grid point locations. However, the restriction to localized predictions prevents the incorporation of predictability information from large-scale spatial structures, including flow-dependent error characteristics and weather regimes, which are inherent in physically consistent forecast fields (Rodwell et al., 2018; Allen et al., 2021). To address this limitation, Lerch and Polsterer (2022) propose the use of convolutional autoencoders to learn low-dimensional latent representations of the spatial forecast fields and demonstrate that using the learned representations as additional predictors to augment a NN-based post-processing model with information about the spatial structure of relevant forecast fields helps to improve predictive performance. However, Lerch and Polsterer (2022) only utilize learned representation of the mean ensemble field, where all ensemble member forecasts are averaged at every grid point. One potential drawback is that the mean field will be notably smoother than forecast fields from individual members. More importantly, however, such approaches ignore the underlying probabilistic information available in the ensemble simulations, which can be seen as samples from a multivariate probability distribution.

Our overarching aim is to propose dimensionality reduction methods to learn low-dimensional representations of ensemble forecast fields, which respect the inherently probabilistic nature of the input data. A variety of dimensionality reduction methods are available, ranging from classical principal component analysis (PCA; Pearson, 1901; Jolliffe and Cadima, 2016) to neural network (NN)-based autoencoder (AE) methods (Bourlard and Kamp, 1988; Kramer, 1991; Hinton and Zemel, 1993; Hinton and Salakhutdinov, 2006). However, the application of existing dimensionality reduction methods to ensemble forecast fields is not straightforward, since they tend to be tailored to deterministic input data. To the best of our knowledge, the problem of learning representations of ensemble simulation data has not been considered thus far, potentially since this type of data is somewhat specific to environmental modeling. The key challenge thus is to develop dimensionality reduction approaches that learn distributional representations in the latent space for an ensemble of forecast fields, allowing for random samples drawn from the latent distribution to be mapped back to a reconstructed forecast field, which ideally would be indistinguishable from a random member from the input ensemble.

To achieve this, we propose two approaches, one based on existing dimensionality reduction methods and one utilizing variational autoencoder (VAE; Kingma, 2013) architectures. The former is an extension of existing dimensionality reduction models with deterministic latent code and can be summarized as a two-step framework. In the first step, a dimensionality reduction model (e.g., PCA or an AE model) is employed to learn low-dimensional representations for each member of the ensemble forecast fields. In the second step, a multivariate Gaussian distribution in the latent space is fitted to the learned representations of all ensemble members. This distribution serves as a learned probabilistic representation of the entire ensemble and can be used to reconstruct ensemble members that are statistically indistinguishable from the original members. This is achieved by reverting the encoding process, i.e., drawing independent samples from the fitted distribution and applying the reverse step of the dimensionality reduction model (e.g., inverse PCA transform or the decoder of AE). In the latter VAE-based framework, on the other hand, we utilize a tailored VAE model that jointly considers all ensemble members as input and provides a distributional ensemble representation as the encoder posterior distribution defined on the VAE's latent space. A key design consideration is that the proposed VAE model should respect the interpretation of ensemble members as interchangeable samples from an unknown, multivariate probability distribution. Notably, the obtained distributional representation should be independent of any (arbitrary) ordering in which the ensemble members are sampled, held in memory, and supplied to the VAE. To this end, we use an invariant VAE (iVAE) architecture designed to be invariant to the reordering of ensemble members.

We systematically compare the two approaches in two case studies on ensemble forecast fields covering a region that roughly corresponds to Europe. We focus on 2-day ahead forecasts of temperature and wind speed, utilizing 10 years of daily forecasts from the European Centre for Medium-Range Weather Forecasts (ECMWF). To that end, we discuss appropriate evaluation approaches for the problem at hand, and consider an exemplary analysis of the learned representations.

The remainder of the paper is structured as follows. Section 3.2 provides an overview of the dataset, and Section 3.3 introduces the proposed two-step and iVAE approaches to learn distributional representations of ensemble forecast fields. The evaluation methods and main results are presented in Section 3.4, followed by conclusions and discussions in Section 3.5. Python code with implementations of all approaches is available online (`https://github.com/jieyu97/invariantVAE`).

3.2 Data

We focus on daily ensemble forecasts from the ECMWF's 50-member ensemble on a spatial domain roughly covering the European continent (-10E to 30E and 30N to 70N). The forcasts are available as gridded fields with regular $0.5^\circ \times 0.5^\circ$ resolution in latitude and longitude. This results in $81 \times 81 (= 6561)$ grid points over Europe. The forecasts are initialized daily at 00 UTC with a forecast lead time of 48 hours. We retrieve forecast data for all days in the time period from January 3, 2007 to January 2, 2017, and split the data into non-overlapping parts for training (03.01.2007–31.12.2014), validation (01.01.2015–31.12.2015), and testing (remainder).

For brevity, we select four exemplary meteorological variables as the basis of our evaluation[1]: 2-m temperature (`t2m`) in Kelvin, U component of wind at 10 meters (`u10`) in meter per second, V component of wind at 10 meters (`v10`), and geopotential height

[1] The datasets for the four variables considered are based on the underlying data from Rasp and Lerch (2018), and are available at `https://doi.org/10.6084/m9.figshare.28151213.v2` (`t2m`), `https://doi.org/10.6084/m9.figshare.28151372.v1` (u10), `https://doi.org/10.6084/m9.figshare.28151411.v1` (v10), `https://doi.org/10.6084/m9.figshare.28151444.v1` (z500).

at 500 hPa (`z500`). In the main paper, we present results exclusively for t2m and u10. Results for the remaining variables are available in the Supplemental Material.

For each weather variable, we apply standard normalization to the raw ensemble forecast data for more stable training of neural network models. The data is standardized by subtracting a global mean and dividing by a global standard deviation. The parameters are computed separately for each weather variable using the data from all grid points in the domain and all samples in the training dataset.

3.3 Learning distributional representations of ensemble forecast fields

This section first introduces required mathematical notation and outlines the problem to be addressed, and then presents two different frameworks for learning distributional representations of ensembles of spatial fields.

3.3.1 Mathematical notations and problem formulation

Throughout this paper, we aim to find lower-dimensional representations for ensembles of spatial forecast fields that summarize the ensemble's information and reduce the data complexity to simplify subsequent forecasting tasks. The required representations are learned in a data-driven way using suitable statistical models for encoding and decoding the inputs. Due to the stochastic characteristic of ensembles, we focus on distributional representations, which express the ensemble information through a suitably parameterized probability distribution defined on a low-dimensional latent space. The problem can thus be considered a dimensionality reduction task with low-dimensional distribution-based embeddings. The intended distributional representations encapsulate both the general spatial structure of the forecast fields and the variability among ensemble members. The field information of the original or new ensemble members can thus be reconstructed by sampling from the parametric distribution and decoding the samples.

For a specific weather variable (e.g., 2-m temperature, `t2m`) and time t, we denote the 50-member ensemble forecast by $\boldsymbol{X}^{\texttt{t2m},t} = \{\boldsymbol{X}_m^{\texttt{t2m},t}\}_{m=1}^{50}$, wherein $\boldsymbol{X}_m^{\texttt{t2m},t} \in \mathbb{R}^{d_{\text{data}}}$ represents the m-th member forecast field. The subscripts `t2m` and t will typically be omitted for brevity. We then write $\boldsymbol{X} = \{\boldsymbol{X}_m\}_{m=1}^{50}$ to denote the ensemble forecast for a given variable and a given time, e.g., to refer to a data point as one training example. The proposed dimensionality reduction methods process one meteorological variable at a time. Therefore, each forecast field $\boldsymbol{X}_m$ comprises scalar-valued forecast data for 81×81 grid locations, resulting in $d_{\text{data}} = 6561$. The 50 ensemble members are interpreted as independent samples from an unknown but identical multivariate probability distribution $\mathcal{P}$, which captures the uncertainty about the predicted weather state:

$$\boldsymbol{X}_m \sim \mathcal{P} \text{ for } m \in \{1, \ldots, 50\}.$$

Each dimensionality reduction consists of an encoding part $\texttt{E}$, a decoding part $\texttt{D}$, and a latent space $\mathbb{R}^{d_{\text{latent}}}$, which hosts the learned representations. The encoding part learns to translate the input ensemble into a representative distribution $\mathcal{D}$ in the latent space, i.e.,

$$\mathcal{D} = \texttt{E}(\boldsymbol{X}),$$

and the decoding part is trained to reconstruct an ensemble of forecast fields $\tilde{\boldsymbol{X}} = \{\tilde{\boldsymbol{X}}_n\}_{n=1}^{N}$ based on an ensemble of samples $\boldsymbol{z} = \{\boldsymbol{z}_n\}_{n=1}^{N}$, drawn from $\mathcal{D}$, i.e.,

$$\tilde{\boldsymbol{X}} = \texttt{D}(\boldsymbol{z}),$$

wherein $\boldsymbol{z}_n \sim \mathcal{D}$ for $n \in \{1, ..., N\}$. While N – the size of the reconstructed ensemble – can be arbitrary, in general, we mainly consider the case $N = 50$, which matches the number of members in the original ensemble forecast. We note, however, that even in this case $\tilde{\boldsymbol{X}}_n$ and $\boldsymbol{X}_m$ usually do not correspond, even if they have the same subscript value, since $\tilde{\boldsymbol{X}}_n$ is decoded from a random sample $\boldsymbol{z}_n$ which is not necessarily the adequate latent representation of $\boldsymbol{X}_m$ with $m = n$. The reconstruction of the original ensemble will be denoted as $\hat{\boldsymbol{X}} = \{\hat{\boldsymbol{X}}_m\}_{m=1}^{50}$.

Additionally, we usually have that the latent dimension $d_{\text{latent}} \ll d_{\text{data}}$. Therefore, d_{latent} controls the compression level of the methods, i.e., how much information from each ensemble field remains in the latent representations. As a hyperparameter of the proposed methods, the latent dimension can be adapted to the needs of downstream tasks.

In the presented case studies, we restrict our focus to relatively low latent dimensions ranging from 2 to 32. The information content of the representation is furthermore affected by the parametric form of the latent distribution $\mathcal{D}$, which is another design choice within the proposed method. We will assume $\mathcal{D}$ to be Gaussian, i.e., $\mathcal{D} = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with parameters $\boldsymbol{\mu} \in \mathbb{R}^{d_{\text{latent}}}$ and $\boldsymbol{\Sigma} \in \mathbb{R}^{d_{\text{latent}} \times d_{\text{latent}}}$ representing the distribution mean and covariance matrix, respectively.

The ultimate goal of the proposed learning framework is finding suitable mappings $\mathtt{E}$ and $\mathtt{D}$ such that for multi-samples $\boldsymbol{z}$ from $\mathcal{D}$, the reconstructed field ensemble $\mathtt{D}(\boldsymbol{z})$ becomes statistically indistinguishable from ensemble members sampled directly from the forecast distribution $\mathcal{P}$. The key challenge therein lies in performing dimensionality reduction on the space of probability distributions, which are represented through stochastically sampled ensembles of forecast fields. In this setting, the representations of each data point (i.e., ensemble $\boldsymbol{X}$) only convey incomplete and stochastic information about the underlying data (i.e., forecast distribution $\mathcal{P}$). This is in stark contrast to the assumption of standard dimensionality reduction problems, which presuppose complete and deterministic data representations. We propose two different approaches to address this problem, leveraging statistical and machine learning methods, which will be introduced in the following sections.

3.3.2 Two-step dimensionality reduction approaches

Our input data are 50-member ensembles of spatial forecast fields $\boldsymbol{X} = \{\boldsymbol{X}_m\}_{m=1}^{50}$. The most straightforward approach is to treat all members collectively as one input $\boldsymbol{X}$ and utilize existing dimensionality reduction methods. However, treating the ensemble members jointly ignores their nature as samples drawn from an identical distribution, leading to a deterministic latent representation for the entire ensemble. This conflicts with our goal of learning a representative low-dimensional distribution for the ensemble of forecast fields. Furthermore, the variabilities among different ensemble members are often less distinct than those among different grid locations in the spatial forecast fields. Consequently, the learned deterministic representation primarily captures the spatial structure in the data. Initial experiments indicated that the resulting reconstructed ensemble forecast fields approximate only the mean field, and fail to reproduce any variability between ensemble members.

To address the probabilistic nature of the ensemble forecast fields and preserve uncertainty information, we propose a two-step framework to identify a latent distribution capturing both general spatial structure and variability within the ensemble. This framework builds on existing methods, which are used to reduce the dimension of each ensemble member separately before merging the per-member representations into a distributional form. Specifically, we assume that a given standard dimensionality reduction approach provides a projection f, which maps a data item to its reduced representation, and a reconstruction function g, which restores a data item based on its latent code.

To encode an ensemble, we proceed by projecting each forecast field separately to obtain its low-dimensional representations as

$$\hat{\boldsymbol{z}}_m = f(\boldsymbol{X}_m) \quad \text{for } m \in \{1, \ldots, 50\}.$$

This yields an ensemble of latent representations to which we can fit a d_{latent}-dimensional Gaussian distribution:

$$\mathcal{D} = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \quad \text{with } \boldsymbol{\mu} = \frac{1}{50} \sum_{m=1}^{50} \hat{\boldsymbol{z}}_m, \text{ and } \boldsymbol{\Sigma} = \operatorname{Var}(\hat{\boldsymbol{z}}_1, \ldots, \hat{\boldsymbol{z}}_{50}).$$

Therein, $\boldsymbol{\mu}$ is the estimated mean vector and $\boldsymbol{\Sigma}$ is the estimated covariance matrix. This corresponds to the intended low-dimensional representation.

To reconstruct ensemble members, any number N of samples can be drawn from $\mathcal{D}$ and utilized to generate new forecast fields. A reconstructed ensemble is obtained as

$$\tilde{\boldsymbol{X}} = \{g(\boldsymbol{z}_n)\}_{n=1}^{N}, \quad \text{with } \boldsymbol{z}_n \sim \mathcal{D}, \quad \text{for } n \in \{1, \ldots, N\}.$$

These newly generated forecast fields can be considered to follow the same distribution as the reconstructions of the input ensemble members, $\hat{\boldsymbol{X}}_m = g(\hat{\boldsymbol{z}}_m)$, for $m \in \{1, ..., 50\}$.

We focus on two practical implementations of this approach using principal component analysis (PCA) and autoencoder (AE) neural networks as the underlying algorithms. A

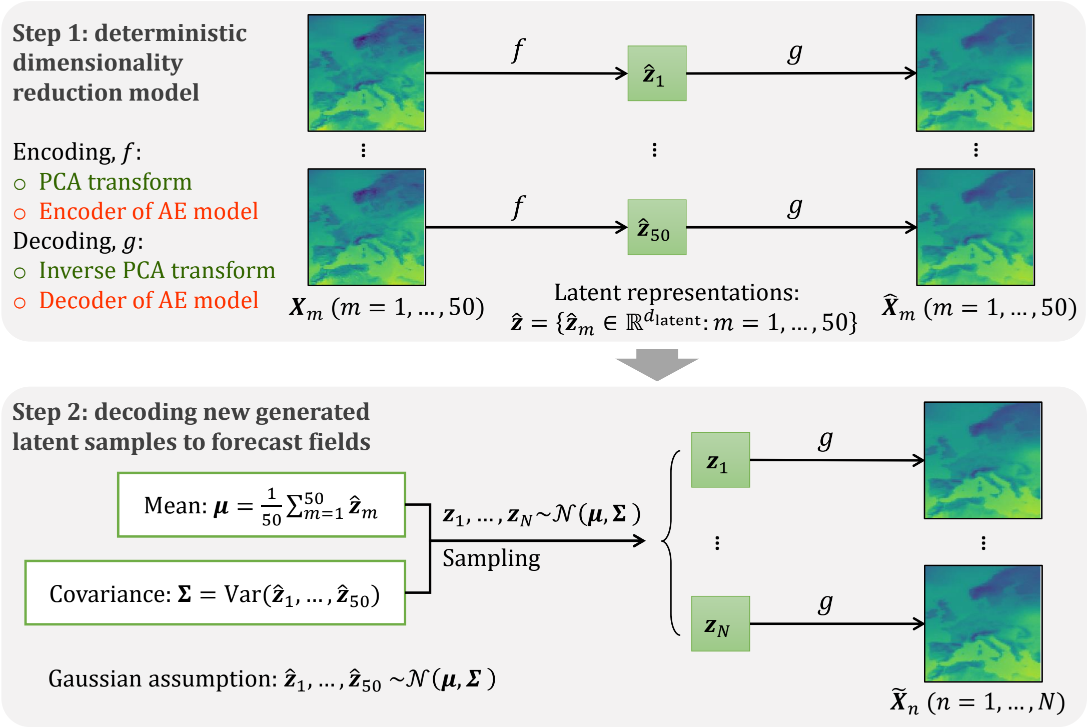

Step 1: deterministic dimensionality reduction model
Encoding, f:
○ PCA transform
○ Encoder of AE model
Decoding, g:
○ Inverse PCA transform
○ Decoder of AE model
f
g
$\hat{\boldsymbol{z}}_1$
$\hat{\boldsymbol{z}}_{50}$
$\boldsymbol{X}_m$ $(m = 1, \ldots, 50)$
Latent representations:
$\hat{\boldsymbol{z}} = \{\hat{\boldsymbol{z}}_m \in \mathbb{R}^{d_{\text{latent}}}: m = 1, \ldots, 50\}$
$\hat{\boldsymbol{X}}_m$ $(m = 1, \ldots, 50)$
Step 2: decoding new generated latent samples to forecast fields
Mean: $\boldsymbol{\mu} = \frac{1}{50}\sum_{m=1}^{50} \hat{\boldsymbol{z}}_m$
Covariance: $\boldsymbol{\Sigma} = \text{Var}(\hat{\boldsymbol{z}}_1, \ldots, \hat{\boldsymbol{z}}_{50})$
$\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$
Sampling
$\boldsymbol{z}_1$
$\boldsymbol{z}_N$
g
Gaussian assumption: $\hat{\boldsymbol{z}}_1, \ldots, \hat{\boldsymbol{z}}_{50} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$
$\tilde{\boldsymbol{X}}_n$ $(n = 1, \ldots, N)$

Figure 3.1: Schematic overview of the two-step dimensionality reduction methods based on PCA and AE models.

schematic illustration of the autoencoder approach is provided in Figure 3.1. PCA and AE are introduced in the following sections.

Principal Component Analysis approach

Principal Component Analysis (PCA) is a linear method that finds the projections of data onto the principal components, which capture the largest variation in the data. The basic idea is to project all samples into a new coordinate system, where the axes (principal components) are determined by the direction along which projections have the largest variance in descending order. For a dataset of d_{data} dimensions, the number of principal components is also d_{data}, and the dimensionality reduction is conducted by taking the projections onto only the first d_{latent} principal components. The reversibility of PCA transform allows data to be reconstructed from the low-dimensional representations.

To provide a benchmark for comparison, we employ PCA in the two-step framework, and refer to it as the PCA-based approach.

In many real-world datasets, linear transformations are not adequate to compress key information, and a variety of nonlinear dimensionality reduction techniques have been proposed. Examples include kernel PCA (Schölkopf et al., 1997), Isomap (Tenenbaum et al., 2000), and Locally Linear Embedding (LLE; Roweis and Saul, 2000). While these methods provide effective low-dimensional representations, the process of converting them back to the original data space introduces additional challenges. The inverse transformations for those nonlinear techniques often require additional training procedures, as exemplified by the pre-image problem for kernel PCA (Mika et al., 1998; Kwok and Tsang, 2004). Autoencoder neural network models, on the other hand, have emerged as a more flexible and now widely used alternative in reconstructing data from latent features. Since PCA can be considered as a linear case of a simple neural network, it is a natural reference method.

Autoencoder neural network approach

Autoencoders are neural network models for unsupervised learning that aim to replicate the input as their output. A typical autoencoder features an internal bottleneck layer with fewer nodes than the input and output layers, dividing the network into two distinct components, the encoder and the decoder, and induces an information bottleneck that prohibits the autoencoder network from memorizing every detail of the input. The encoder f and decoder g can be formulated as two mappings, where, following the notations in Section 3.3.1,

$$f(\boldsymbol{X}_m) = \hat{\boldsymbol{z}}_m, \quad g(\hat{\boldsymbol{z}}_m) = \hat{\boldsymbol{X}}_m, \quad \text{for } \boldsymbol{X}_m, \hat{\boldsymbol{X}}_m \in \mathbb{R}^{d_{\text{data}}},\ \hat{\boldsymbol{z}}_m \in \mathbb{R}^{d_{\text{latent}}} \text{ and } m \in \{1, \ldots, 50\}.$$

The encoder network f maps one input forecast field $\boldsymbol{X}_m$ to its latent representation $\hat{\boldsymbol{z}}_m$ from the bottleneck layer, with d_{latent} typically much smaller than d_{data}. The decoder network g maps one latent representation $\hat{\boldsymbol{z}}_m$ back to the corresponding reconstruction $\hat{\boldsymbol{X}}_m$, the output of the autoencoder, which aims to reproduce the input $\boldsymbol{X}_m$. Training autoencoder neural networks involves minimizing differences between input and output, often using mean square error as a loss function.

The latent representation $\hat{z}_m$ obtained from the encoder naturally functions as a compact representation of the input, capturing essential features needed for the decoder to reconstruct the original data. In addition to dimensionality reduction applications (Hinton and Salakhutdinov, 2006; Wang et al., 2014, 2016), autoencoder models have also found applications in other domains, such as anomaly detection (Sakurada and Yairi, 2014; Zhou and Paffenroth, 2017) and image denoising (Gondara, 2016). Autoencoders exist in many variants, developed for different applications, including, e.g., sparse autoencoders for classification tasks (Baccouche et al., 2012).

Our autoencoder model for the AE-based dimensionality reduction approach is a shallow neural network utilizing fully connected dense layers in both the encoder and decoder. The model is trained to minimize the mean absolute error (MAE) between the input forecast field and the reconstructed field obtained as output. Hyperparameter tuning is performed using the Bayesian optimization algorithm HyperBand (Li et al., 2018) implemented in the `Ray Tune` Python library (Liaw et al., 2018). The final AE model configuration includes layers of sizes "6561 - 4096 - d_{latent}" in the encoder and "d_{latent} - 4096 - 6561" in the decoder. The LeakyReLU activation function is applied in the hidden layer in both the encoder and the decoder. We utilize the AdamW optimizer (Loshchilov and Hutter, 2019) with a learning rate decay scheduler starting from $1e^{-4}$ to stabilize the training process. Mini-batch training is employed with a batch size of 1024 to enhance training efficiency, and samples in all batches are randomly shuffled in each training epoch. To prevent overfitting, an early stopping criterion with a patience of 20 epochs on the validation loss is applied. We also investigated more sophisticated frameworks for the encoder and decoder during initial experiments, including convolutional layers with residual blocks (He et al., 2016), and a vision transformer (ViT; Dosovitskiy et al., 2021) based architecture. However, these more complex frameworks did not yield significant improvements and incurred higher computational costs. Therefore, we prioritize a simpler framework with only dense layers for our neural network models, allowing efficient computation without requiring a GPU.

3.3.3 Invariant variational autoencoder approach

The two-step framework developed for ensemble forecast fields can, in principle, be generalized to other dimensionality reduction techniques beyond PCA and AE. However, a conceptual disadvantage of the approach is the assumption of Gaussian-distributed representations in the latent space, which is somewhat decoupled from the training process.

To address this limitation, we propose an ensemble-invariant framework based on variational autoencoders (VAEs). VAEs (Kingma, 2013) are generative machine learning methods that leverage variational inference to learn a probabilistic representation in latent space. In contrast to standard autoencoders, VAEs connect an encoder network to its decoder through a probabilistic latent space, which corresponds to the parameters of a pre-specified probability distribution. Thereby, the encoder network maps input samples to parameters of the latent space distribution, and the decoder network maps samples drawn from the distribution in the latent space back to the data space by generating new data points decoded from the samples. The reparameterization trick (Kingma and Welling, 2019) enables the simultaneous training of the encoder and decoder using backpropagation by transforming the sampling process to make it differentiable. VAEs have been widely applied in various domains, including image generation, denoising, and inpainting (An and Cho, 2015; Pu et al., 2016). Moreover, the VAE framework has inspired the development of extensions targeting different aspects of feature representations and applications. Examples include Importance Weighted Autoencoders (Burda et al., 2016), the combination of a VAE with a Generative Adversarial Network (Larsen et al., 2016), Wasserstein Autoencoders (Tolstikhin et al., 2019) and Sinkhorn Autoencoders (Patrini et al., 2020).

The inherently probabilistic nature of VAEs makes them potentially effective for the problem at hand. The standard VAE model is trained to learn a latent distribution for a single instance from the input data, where samples drawn from the latent distribution are decoded to data points close to the corresponding instance. If we consider each ensemble member separately, the VAE model would thus learn different latent distributions for different members from the same forecast case. Therefore, we need to adapt the VAE framework to jointly learn one latent distribution for all ensemble members, and decode

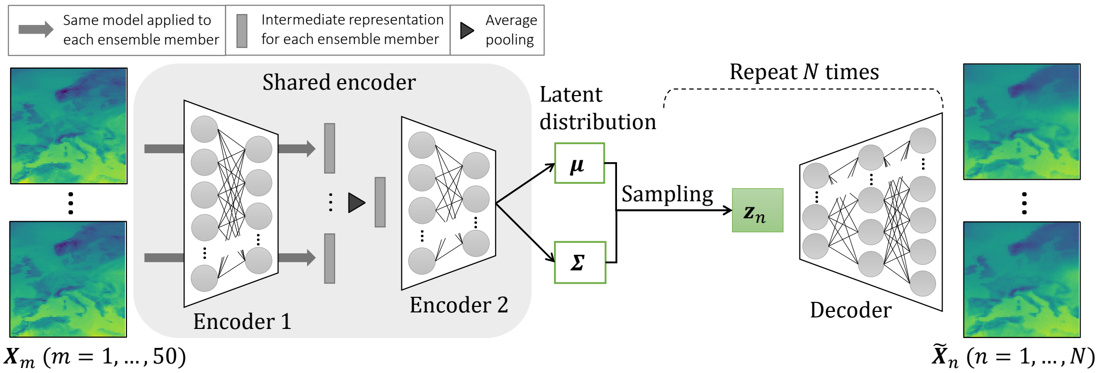


Figure 3.2: Schematic illustration of the invariant variational autoencoder (iVAE) model.

samples from the latent distribution to newly generated members that follow the same distribution as the inputs. To address this challenge, we propose an invariant VAE (iVAE) model, inspired by the permutation-invariant neural network framework in the Deep Sets architecture (Zaheer et al., 2017). The encoder of our iVAE model follows such a permutation-invariant framework and is invariant to any permutation on the order of ensemble members. A schematic overview of our iVAE model is available in Figure 3.2.

The main difference between our iVAE model and a standard VAE is that the encoder is shared for all 50 members of the ensemble forecast fields as input, and is invariant to their order. The shared encoder comprises two separate encoder parts, which we denote by e_1 and e_2, see Figure 3.2. For a given 50-member ensemble $\boldsymbol{X} = \{\boldsymbol{X}_m\}_{m=1}^{50}$, the first encoder is applied to each ensemble member forecast field $\boldsymbol{X}_m$ iteratively to obtain intermediate representations $\boldsymbol{y} = \{\boldsymbol{y}_m\}_{m=1}^{50}$, i.e.,

$$\boldsymbol{y}_m = e_1(\boldsymbol{X}_m), \quad \text{for } m \in \{1, \ldots, 50\}.$$

These intermediate representations are interchangeable since their original input fields are assumed to follow the same distribution. Next, we average the 50 intermediate representations to summarize key features learned from all ensemble members, i.e.,

$$\bar{\boldsymbol{y}} = \frac{1}{50} \sum_{m=1}^{50} \boldsymbol{y}_m,$$

which ensures that the shared encoder is permutation-invariant. Subsequently, the second encoder is applied after this average pooling step. Similar to standard VAEs, a probabilistic encoder e_2 with parameters ϕ is applied to approximate the posterior distribution $p(z|\boldsymbol{X})$ in the latent space using a parameterized distribution $q_\phi(\boldsymbol{z}|\boldsymbol{X}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}, \operatorname{diag}(\boldsymbol{\sigma}^2))$. After applying the reparametrization trick, we obtain

$$\begin{aligned} \boldsymbol{\epsilon}_n &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \\ (\boldsymbol{\mu}, \log \boldsymbol{\sigma}) &= e_2\left(\frac{1}{50}\sum_{m=1}^{50} e_1(\boldsymbol{X}_m)\right), \\ \boldsymbol{z}_n &= \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}_n. \end{aligned}$$

The latent distribution $\mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}, \operatorname{diag}(\boldsymbol{\sigma}^2))$ is thus the low-dimensional probabilistic representation of the ensemble forecast fields that we aimed for. The decoder d with parameters θ is then applied to the sample $\boldsymbol{z}_n$ from the latent distribution to generate reconstructed forecast field $\tilde{\boldsymbol{X}}_n$, parameterizing the likelihood $p_\theta(\boldsymbol{X}|\boldsymbol{z})$ in the data space. In contrast to standard VAEs, we decode an arbitrary number N of samples $\boldsymbol{z} = \{\boldsymbol{z}_n\}_{n=1}^N$ for each data point, producing an ensemble of reconstructed forecast fields as output.

The architecture of our iVAE is built on the AE model discussed above and employs fully-connected dense layers. The shared encoder adds an average pooling layer to the encoder of the AE model, consisting of e_1 with one layer of size "6561 - 4096" and e_2 with two layers of sizes "4096 - 4096 - d_{latent}", while the decoder follows the same structure as the decoder of the AE with layers of sizes "d_{latent} - 4096 - 6561". The LeakyReLU activation function is again applied in the hidden layers, and the same AdamW optimizer and early stopping criterion are applied. Due to the significantly increased memory requirements for training the iVAE model, we employ mini-batch training with a batch size of 64.

The training objective of a standard VAE is to maximize the evidence lower bound on the marginal likelihood of the data,

$$\mathcal{L}(\theta, \phi) = \log p_\theta(\boldsymbol{X}|z) - D_{\text{KL}}(q_\phi(\boldsymbol{z}|\boldsymbol{X}) \| p_\theta(\boldsymbol{z})),$$

which consists of a negative reconstruction error and a regularization term. The reconstruction error $-\log p_\theta(\boldsymbol{X}|\boldsymbol{z})$ measures how well the model reconstructs the input data, which is proportional to the mean square error (MSE) with the Gaussian assumption on the data distribution for deterministic input and output. The regularization term is the Kullback-Leibler divergence between the approximate posterior $q_\phi(\boldsymbol{z}|\boldsymbol{X})$ from the encoder and the prior $p_\theta(\boldsymbol{z})$ of the latent code $\boldsymbol{z}$, where standard multivariate Gaussian distributions are often used as the prior.

In our case, however, the probabilistic nature of both input and output of the iVAE necessitates a different notion of the reconstruction error. Our iVAE model takes an ensemble of forecast fields $\boldsymbol{X} = \{\boldsymbol{X}_m\}_{m=1}^{50}$ as input and generates an ensemble of reconstructed fields $\tilde{\boldsymbol{X}} = \{\tilde{\boldsymbol{X}}_n\}_{n=1}^{N}$ with size N as output. The number N is not necessarily equal to 50, and each input member $\boldsymbol{X}_m$ does not match the corresponding output member $\tilde{\boldsymbol{X}}_n$ for $m = n$ due to the random sampling of latent distribution. Therefore, the MSE between $\boldsymbol{X}_m$ and $\tilde{\boldsymbol{X}}_n$ is not a suitable choice for estimating the reconstruction error. Given that both the input and output ensembles of the iVAE can be considered to be multivariate empirical probability distributions, notions of the distance between the two distributions yield a more appropriate choice. We thus incorporate two such metrics into the iVAE training objective. Specifically, we use the energy distance and the Sinkhorn distance, which will be introduced in Section 3.4.1, for measuring different aspects of distances between multivariate probability distributions. The reconstruction error of our iVAE is defined as the weighted sum of the energy distance and the Sinkhorn distance, complemented by the Kullback-Leibler divergence as a regularization term. The three loss components exhibit significantly different scales, necessitating rescaling to ensure that their value ranges are comparable. By comparing the mean values of different loss components over the first 20 epochs of training, we applied the following adjustments: the KL divergence was divided by 10, the energy distance was multiplied by 2 for both weather variables, and the Sinkhorn distance was divided by 50 for temperature and by 500 for wind speed. The loss function of our iVAE model for temperature data thus is

$$\ell(\boldsymbol{X}, \tilde{\boldsymbol{X}}) = \omega_1 \cdot 2D(\boldsymbol{X}, \tilde{\boldsymbol{X}}) + \omega_2 \cdot \frac{1}{50}\mathrm{SD}(\boldsymbol{X}, \tilde{\boldsymbol{X}}) + \omega_3 \cdot \frac{1}{10} D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}|\boldsymbol{X}) \| p_\theta(\boldsymbol{z})), \tag{3.1}$$

where $D(\cdot)$ represents the energy distance and $\mathrm{SD}(\cdot)$ denotes the Sinkhorn distance. In our preliminary experiments, we observed that assigning a high weight to the KL divergence component in the loss function restricts the information flow through the bottleneck of the network, which results in outputs that fail to preserve the general spatial patterns of the input forecast fields. To mitigate this issue and alleviate posterior collapse, we, like many other studies, heuristically selected a small weight $\omega_3 = 0.01$ for the KL divergence component. For a more comprehensive understanding of the posterior collapse problem, we refer to Lucas et al. (2019). Regarding the two components used to measure reconstruction error, we assign equal weights of $\omega_1 = \omega_2 = 0.5$. Further discussions on the impact of different weighting schemes on the evaluation results are provided in the ablation studies in Section 3.4.3.

3.4 Results

In the following, we first briefly introduce the evaluation methods tailored for our specific problem, and then present and discuss the corresponding results for the dimensionality reduction methods introduced above. Finally, we analyze the learned low-dimensional representations from different methods in an exemplary use case.

3.4.1 Evaluation methods

Choosing appropriate evaluation methods for our specific setting presents a challenge. Two primary perspectives guide our evaluation: assessing the accuracy of the reconstructed ensemble forecast fields in comparison to the original ensemble fields, and analyzing the information content of the learned low-dimensional representations. Evaluating the latter is particularly challenging as there naturally is no ground truth information for the representations, and the suitability will strongly depend on the application use case. Therefore, our main focus is on discrepancy measures between the reconstructed output and the original input ensemble fields. The discrepancy could be evaluated in terms of independent pixel-wise errors at each grid point, and joint, whole-image evaluation, where the entire forecast field is considered at once. Several evaluation metrics are available for both settings and will be introduced in the following.

All three dimensionality reduction approaches learn a low-dimensional Gaussian distribution for representing an ensemble of forecast fields. We draw 50 samples from the learned distribution and decode them into reconstructed forecast fields to enable a fair comparison when assessing the discrepancy with the raw 50-member ensemble. As discussed above, the reconstructed ensemble members do not necessarily match the individual raw ensemble members, which prohibits measuring the pair-wise differences directly. As an alternative, we compare the mean and standard deviation of all ensemble members between the raw and reconstructed fields at each grid point, providing insight into how well the model captures general characteristics of the input ensemble. Given an ensemble of spatial forecast fields $\boldsymbol{X} = \{\boldsymbol{X}_m\}_{m=1}^{50}$ and an ensemble of reconstructed fields $\tilde{\boldsymbol{X}} = \{\tilde{\boldsymbol{X}}_n\}_{n=1}^{N}$ with $N = 50$, we compute the absolute difference of ensemble means and the standard deviation difference at each grid point (i, j),

$$e_{(i,j)} = \left| \frac{1}{50} \sum_{m=1}^{50} \boldsymbol{X}_{m,(i,j)} - \frac{1}{N} \sum_{n=1}^{N} \tilde{\boldsymbol{X}}_{n,(i,j)} \right|, \quad \Delta(\sigma)_{(i,j)} = \sigma(\boldsymbol{X}_{(i,j)}) - \sigma(\tilde{\boldsymbol{X}}_{(i,j)}),$$

where $i, j \in \{1, \dots, 81\}$ and $\sigma(\cdot)$ denotes the standard deviation of the respective ensemble.

We further consider probabilistic measures to quantify the discrepancy between two distributions of the ensembles used as inputs and obtained as outputs. The evaluation of ensemble forecast fields could be executed pixel-wise, considering the (one-dimensional) univariate distribution at each grid point, or for the whole image, treating all grid points together as a high-dimensional multivariate distribution. Measuring the divergence between two distributions for evaluating climate models in the univariate setting has been studied by Thorarinsdottir et al. (2013), and here we consider the distance measures for both univariate and multivariate settings, utilizing the energy distance and optimal transportation distances. The multivariate measures are also integrated into the reconstruction error component of the loss function for training our iVAE models, as discussed earlier in Section 3.3.3.

The energy distance introduced by Székely and Rizzo (2013) is a metric that measures the distance between two probability distributions. Following our notations in Section 3.3.1, consider an ensemble of forecast fields $\boldsymbol{X} = \{\boldsymbol{X}_m\}_{m=1}^{50}$ and reconstructed fields $\tilde{\boldsymbol{X}} = \{\tilde{\boldsymbol{X}}_n\}_{n=1}^{N}$ in $\mathbb{R}^{d_{\text{data}}}$ with $N = 50$, with the assumption that the ensemble members follow an idential distribution, i.e., $\boldsymbol{X}_m \sim \mathcal{P}$ for $m \in \{1, \dots, 50\}$ and $\tilde{\boldsymbol{X}}_n \sim \tilde{\mathcal{P}}$ for $n \in \{1, \dots, N\}$. The squared energy distance between $\mathcal{P}$ and $\tilde{\mathcal{P}}$ can be estimated in terms of expected pairwise distances between the two ensembles of samples,

$$D^2(\boldsymbol{X}, \tilde{\boldsymbol{X}}) = \frac{2}{50N} \sum_{m=1}^{50} \sum_{n=1}^{N} \|\boldsymbol{X}_m - \tilde{\boldsymbol{X}}_n\| - \frac{1}{50^2} \sum_{m_1=1}^{50} \sum_{m_2=1}^{50} \|\boldsymbol{X}_{m_1} - \boldsymbol{X}_{m_2}\| - \frac{1}{N^2} \sum_{n_1=1}^{N} \sum_{n_2=1}^{N} \|\tilde{\boldsymbol{X}}_{n_1} - \tilde{\boldsymbol{X}}_{n_2}\|,$$

where $\|\cdot\|$ is the Euclidean norm in $\mathbb{R}^{d_{\text{data}}}$. The energy distance $D(\boldsymbol{X}, \tilde{\boldsymbol{X}})$ is negatively oriented and is zero if and only if the two empirical distributions with samples $\boldsymbol{X}$ and $\tilde{\boldsymbol{X}}$ coincide. In the univariate setting of pixel-wise evaluation, the squared energy distance at each grid point (i, j) is thus

$$\begin{aligned} D^2_{(i,j)} = \frac{2}{50N} \sum_{m=1}^{50} \sum_{n=1}^{N} |\boldsymbol{X}_{m,(i,j)} - \tilde{\boldsymbol{X}}_{n,(i,j)}| &- \frac{1}{50^2} \sum_{m_1=1}^{50} \sum_{m_2=1}^{50} |\boldsymbol{X}_{m_1,(i,j)} - \boldsymbol{X}_{m_2,(i,j)}| \\ &- \frac{1}{N^2} \sum_{n_1=1}^{N} \sum_{n_2=1}^{N} |\tilde{\boldsymbol{X}}_{n_1,(i,j)} - \tilde{\boldsymbol{X}}_{n_2,(i,j)}|. \end{aligned}$$

The univarite squared energy distance is closely related to the Cramér distance (Rizzo and Székely, 2016), which is also known as the integrated quadratic distance (Thorarinsdottir et al., 2013) for evaluating probabilistic forecasts from climate models. The computation of univariate energy distance $D_{(i,j)}$ follows existing Python implementations from the `scikit-learn` library (Pedregosa et al., 2011), while the multivariate energy distance $D(\boldsymbol{X}, \tilde{\boldsymbol{X}})$ is implemented by custom code.

The optimal transportation distances, also known as the p-Wasserstein distances (Kantorovich, 1960), are another type of metric that measure distances between two probability distributions. The general optimal mass transport problem aims to find the optimal strategy to transport probability mass from one probability measure into another while minimizing the transportation cost, where the p-Wasserstein distance is the minimum total cost with the cost function $c(x, y) = |x - y|^p$. Optimal transportation distances have found broad applications in machine learning in recent years (Frogner et al., 2015; Courty et al., 2016; Arjovsky et al., 2017). For an overview of the theory and methodology of optimal transportation distances and their applications, we refer to Kolouri et al. (2017). In our univariate setting of pixel-wise evaluation, the empirical format of 1-Wasserstein distance is considered. The 1-Wasserstein distance between an input ensemble of forecast fields $\boldsymbol{X}$ and an output ensemble of reconstructed fields $\tilde{\boldsymbol{X}}$ at each grid point (i,j) is estimated based on order statistics,

$$W_{1,(i,j)} = \frac{1}{50}\sum_{k=1}^{50} \left|\boldsymbol{X}_{(k),(i,j)} - \tilde{\boldsymbol{X}}_{(k),(i,j)}\right|,$$

where the subscript $\cdot_{(k)}$ denotes the k-th order of values. We follow existing Python implementations from the `scikit-learn` library (Pedregosa et al., 2011) for the computation of univariate 1-Wasserstein distance $W_{1,(i,j)}$. In the multivariate setting of whole-image evaluation, estimating the Wasserstein distance between two high-dimensional distributions requires substantial computational cost and is thus not suitable for both evaluation tasks and particularly the integration into the iVAE training loss. The Sinkhorn distance proposed by Cuturi (2013) provides a computationally efficient approximation of the Wasserstein distance by leveraging entropic regularizations. We thus utilize the Sinkhorn distance as an alternative, and follow the Sinkhorn algorithm proposed in Eisenberger et al. (2022) for a memory-efficient estimation. The MSE is used as the cost function, i.e., $c(x,y) = \|x-y\|^p$ with $p=2$. Thereby, the estimated Sinkhorn distance $\mathrm{SD}(\boldsymbol{X}, \tilde{\boldsymbol{X}})$ approximates the 2-Wasserstein distance between $\boldsymbol{X}$ and $\tilde{\boldsymbol{X}}$,

$$W_2 = \inf_{\pi} \left(\sum_{k=1}^{50} \left\|\boldsymbol{X}_k - \tilde{\boldsymbol{X}}_{\pi(k)}\right\|^2\right)^{1/2},$$

where π denotes all possible permutations.

To compare different methods based on the same distance measure with respect to a benchmark, we further compute the associated skill score for analyzing the relative performances. Among our three dimensionality reduction approaches, the PCA-based approach is naturally taken as the reference baseline method. For each day in the test set, we first compute either the mean distance over all grid points for the pixel-wise metric, or directly take the distance measure for the entire grid as the score S_a of a certain approach. The skill score SS_a is then calculated via

$$SS_a = \frac{S_{\mathrm{ref}} - S_a}{S_{\mathrm{ref}} - S_{\mathrm{opt}}},$$

where S_{ref} is the corresponding score of the reference PCA-based approach, and $S_{\mathrm{opt}} = 0$ represents the score of an optimal method, i.e., an ideal model that replicates its input ensemble members perfectly. Skill scores are positively oriented and have an upper bound of 1, where a positive value indicates better performance than the benchmark, and a value of 0 indicates no improvement over the benchmark.

For brevity, we present only the results for the energy distances in the main text, whereas the results concerning the optimal transport distances are deferred to the Supplemental Material.

3.4.2 Reconstruction accuracy

We compare the accuracy of the reconstructed ensemble forecast fields obtained by the three different approaches (i.e., PCA-based and AE-based two-step methods and the iVAE approach). We here focus on 2-m temperature and the U component of 10-m wind speed, and present results for the corresponding V component and for the geopotential height at 500 hPa in the Supplemental Material.

Figure 3.3 shows examples of input forecast fields along with reconstructed ensemble members produced as output of the different approaches. As discussed above and in light of our aim of learning representations of the underlying probability distributions, the reconstructed ensemble members should not be expected to perfectly match the input even though the selected fields are from the same forecast day. For both target variables, all approaches are able to capture the general spatial structure in the raw fields despite reducing the dimensionality from 6561 to 32. However, neither of them is able to realistically replicate the localized fine patterns in the raw fields, with the AE and iVAE methods showing slightly better, specifically for the U component of wind speed based on visual inspection across ensemble members. The reconstruction quality and level of fine-scale details can be improved for higher-dimensional latent representation. Animated figures cycling through all ensemble members of reconstructed fields in comparison to the corresponding ensemble of raw fields for the same forecast day indicate that the iVAE method tends to reproduce larger but still realistic variability among different ensemble members compared to the other two methods. The animated figures are available in the Supplemental Material.

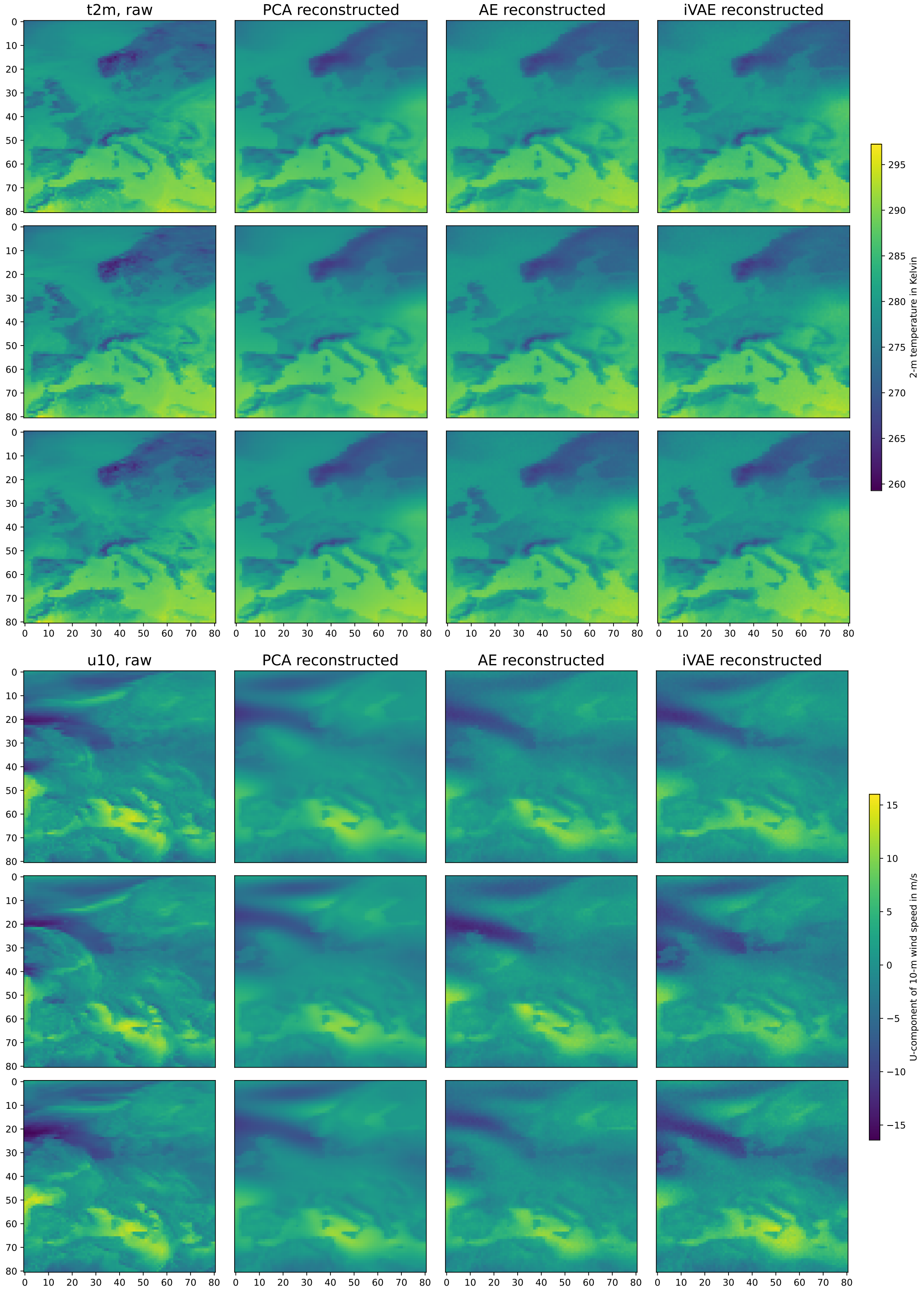


Figure 3.3: Exemplary raw forecast fields and reconstructed forecast fields of 2-m temperature (top) and the U component of 10-m wind speed (bottom) by different methods, with a latent dimension of 32. The rows correspond to different ensemble members for the same forecast day.

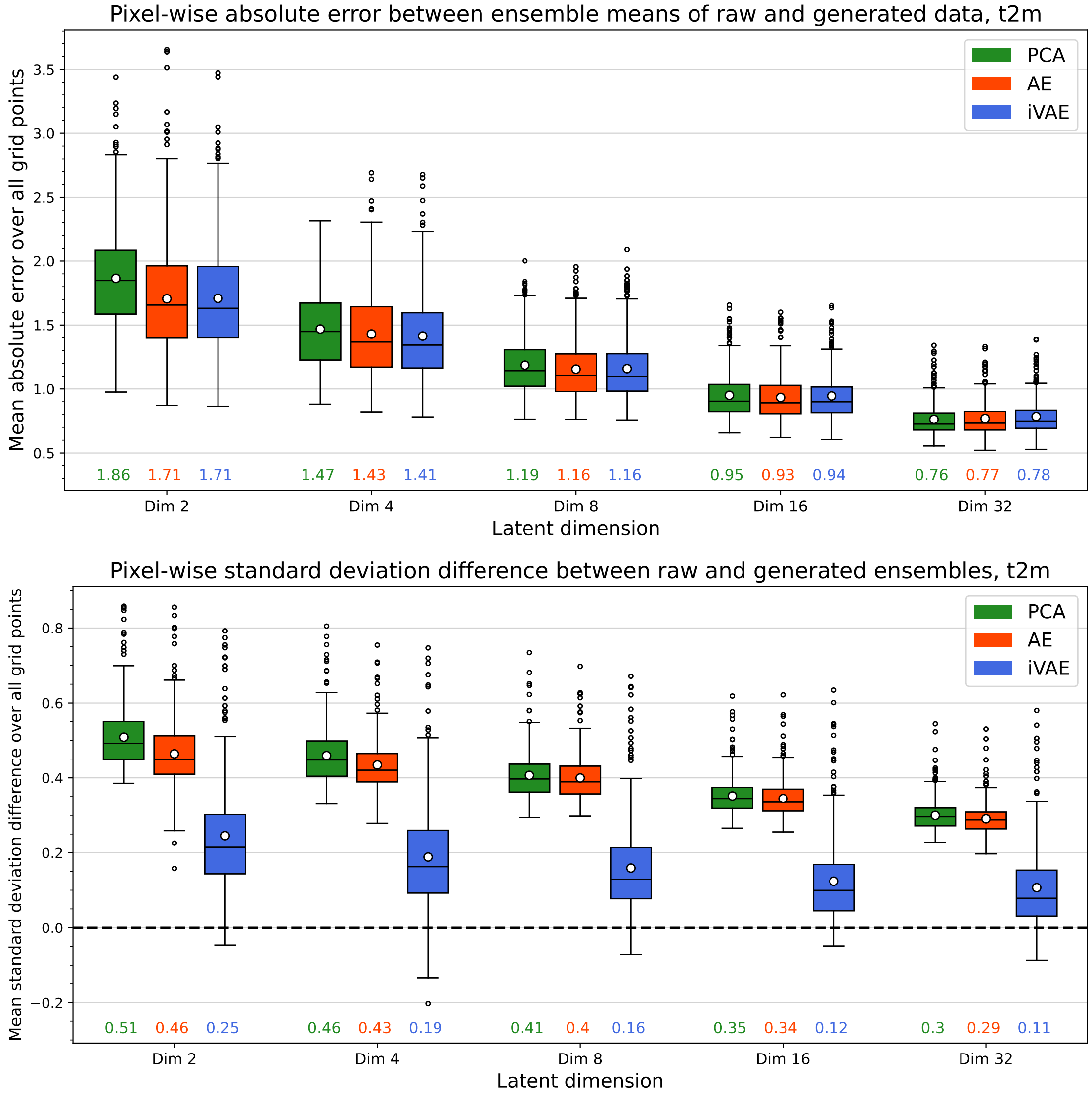


Figure 3.4: Boxplots of mean absolute differences between the mean values of input and reconstructed ensemble fields (top) and differences between the standard deviations of input and reconstructed ensemble fields (bottom) at each grid point. Boxes show performance variability over 366 days in the test set of different methods for 2-m temperature data, considering 5 different dimensionalities of the latent representation. The mean values of the (absolute) differences are indicated below each box. The differences between the standard deviations are computed such that negative values indicate a larger variability of the reconstructed ensemble compared to the input ensemble.

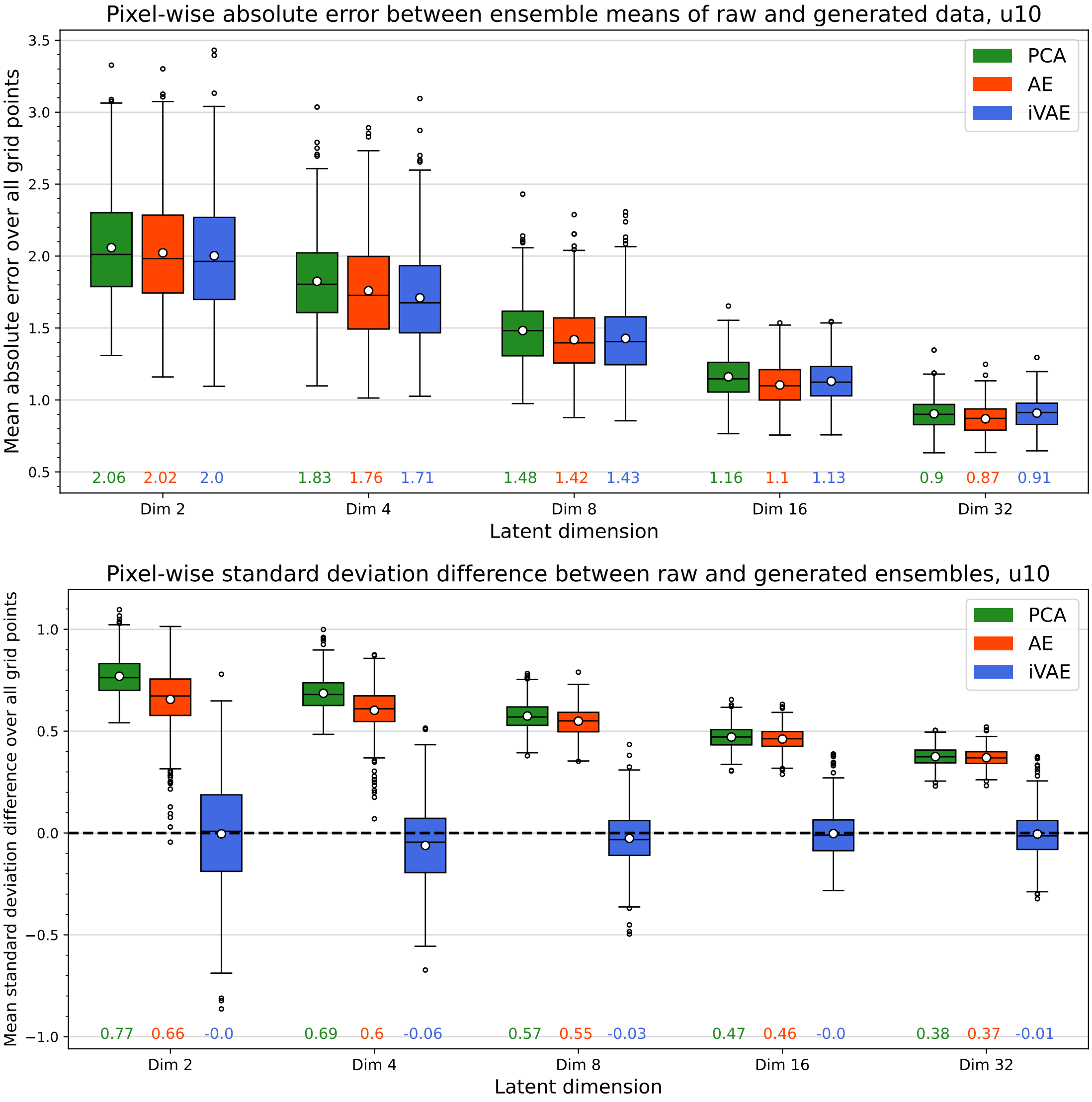


Figure 3.5: As Figure 3.4, but for the U component of 10-m wind speed.

To assess the reconstruction quality in terms of general summary statistics of the input and output ensemble fields, Figure 3.4 shows pixel-wise absolute differences of the corresponding ensemble mean and standard deviation values for temperature data. For all methods, the differences between the summary statistics of the input and reconstructed ensemble fields decrease with increasing dimensionality of the latent representations. In terms of the deviations of the ensemble mean, the two neural network-based methods show slightly better performance in lower-dimensional settings, while PCA shows minimally better performance for the largest dimensionality considered here. However, these differences are very minor, in particular when compared to the variability within the boxplots. More substantial differences can be observed for the standard deviation among the reconstructed ensemble members, where only the iVAE approach is able to produce variability among the members of a magnitude that matches the raw ensemble, whereas the reproduced ensembles from both two-step methods notably underestimate the variability of the input. Qualitatively similar results are obtained for the wind speed data, shown in Figure 3.5, where the better performance of the iVAE approach at correctly reproducing the variability across ensemble members is even more apparent.

For a more fine-grained assessment of the probabilistic reconstruction quality, Figure 3.6 shows skill scores based on energy distances between input and reconstructed temperature ensemble forecast fields for the AE-based and iVAE methods, with the PCA-based method as a baseline. Positive values indicate an improvement in terms of the energy distance over the reference method. For example, a skill score of 0.1 corresponds to a 10% lower energy distance compared to PCA. The iVAE method consistently outperforms the other two approaches across all latent dimensions and in both univariate and multivariate evaluation. These improvements are likely due to the variability of the reconstructed ensemble fields being close to the input ones as discussed in Figure 3.4, and potentially benefit from the inclusion of the multivariate energy distance as part of the loss function. The AE-based two-step approach generally outperforms the PCA-based approach, but the improvements are less pronounced than for the iVAE method, and poorer performance is observed for multivariate energy distances with a latent dimension greater than 8. Both neural network-based approaches show less distinct improvements over the PCA-based method with increasing latent dimensions, possibly because PCA is sufficient to capture

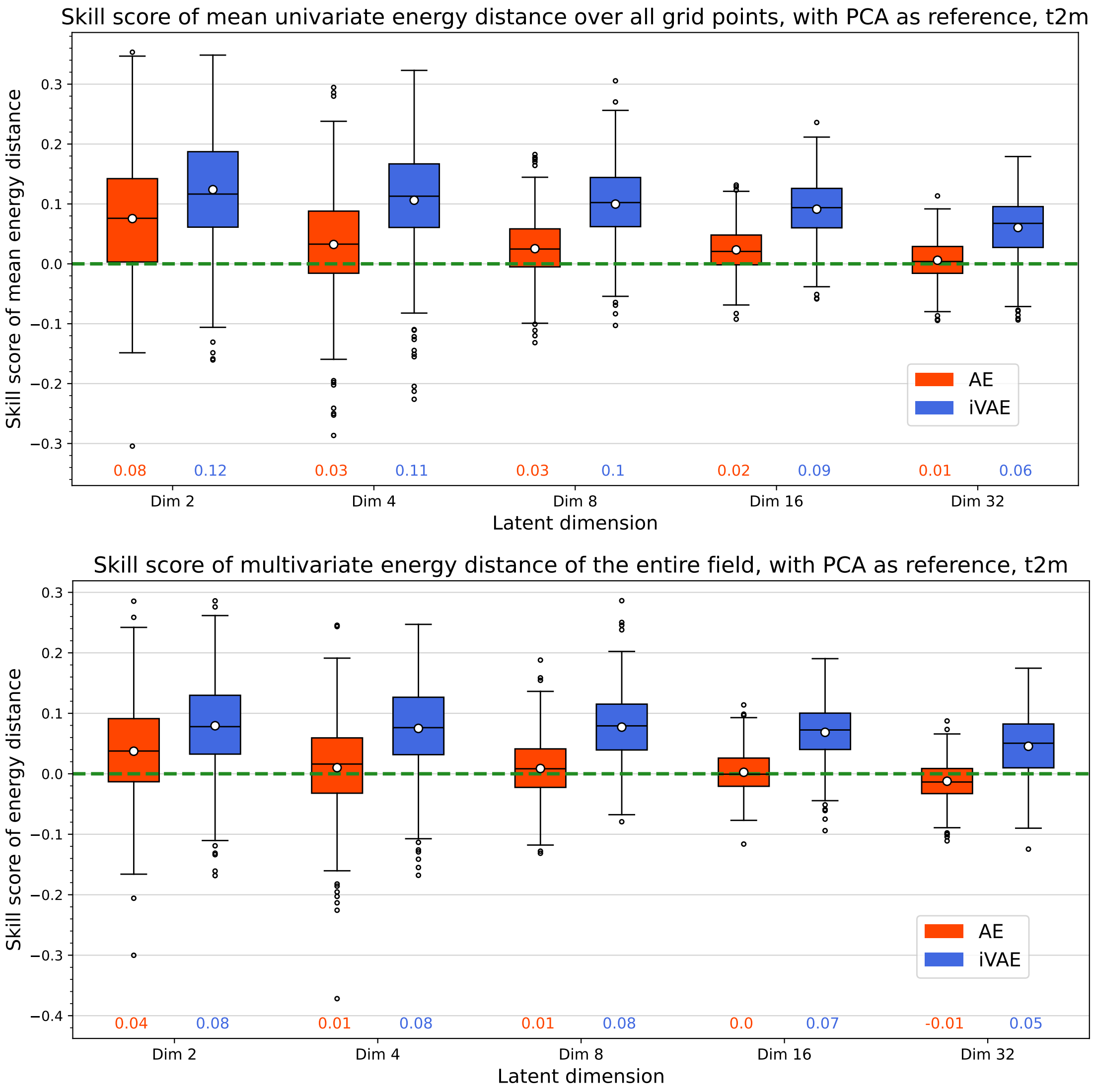


Figure 3.6: Boxplots of skill scores based on energy distances between the input and reconstructed ensemble fields over the 366 days in the test set for temperature data. The panels show mean univariate energy distances over all grid points (top) and multivariate energy distances computed for the entire fields (bottom). PCA-based approach shown in green dashed line is the reference method. The respective mean skill values are indicated below each box.

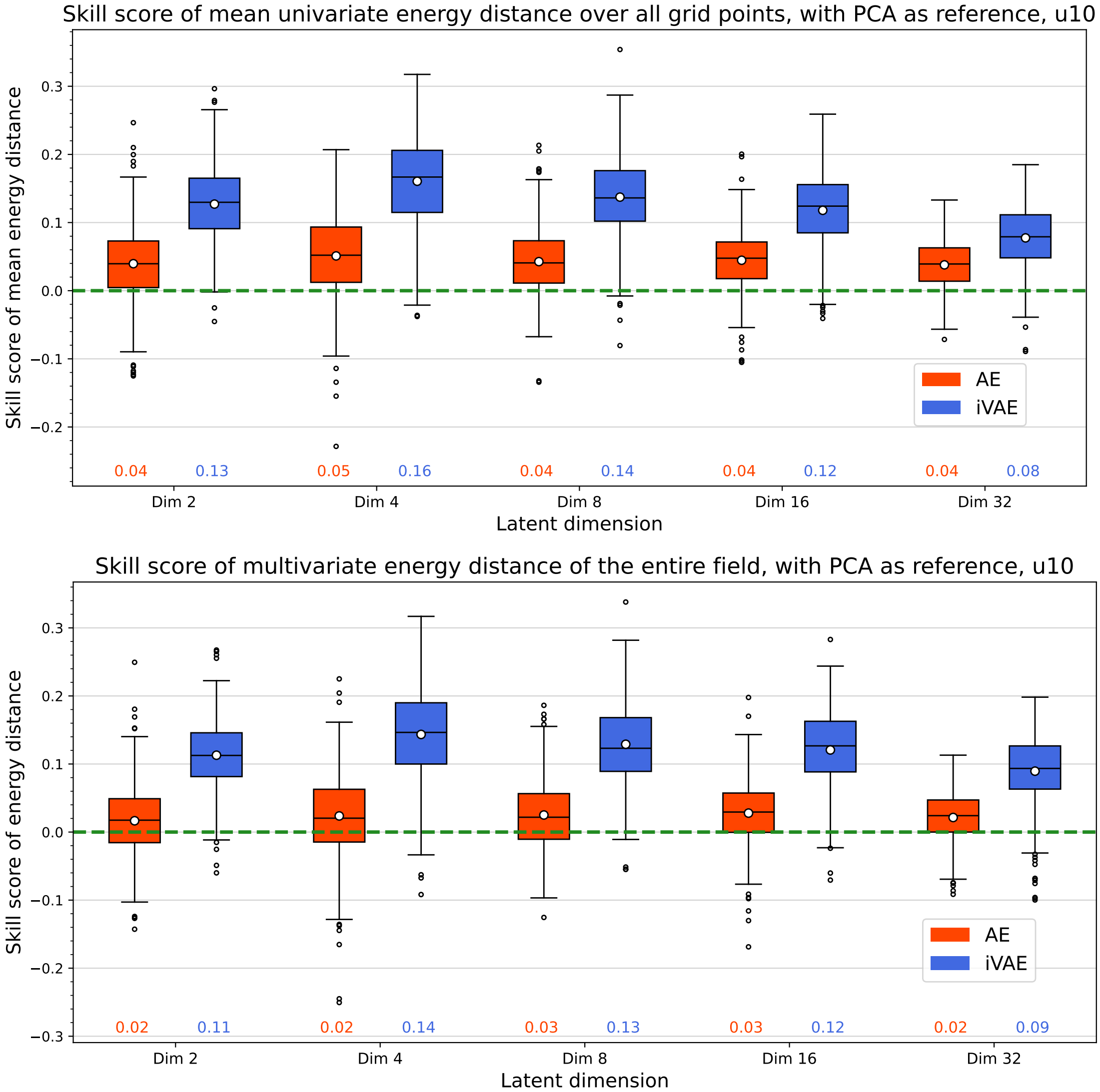


Figure 3.7: As Figure 3.6, but for the U component of 10-m wind speed.

most of the variability information in the raw ensemble forecast fields when the dimension of representations is suitably large. Similar conclusions can be drawn for the wind speed data, see Figure 3.7. The most notable difference to the results for temperature data is that the AE-based method consistently outperforms the PCA-based method here, and the largest improvements from the iVAE method occur at a latent dimension of 4.

3.4.3 Ablation studies on the weighted loss function of iVAE

As discussed in Section 3.3.3, the loss function of our iVAE model incorporates both the multivariate energy distance and the Sinkhorn distance to measure reconstruction error. Given that both distances are used to evaluate the performance of reconstructed ensemble forecast fields, we aim to investigate the impact of different weighting schemes on the evaluation results.

To this end, we conduct ablation studies on several weighting scheme choices for temperature data. To ensure comparable scales of the weighted loss, we maintain $\omega_1 + \omega_2 = 1$ in the loss function (3.1), and vary ω_2 between 0 and 1. The averages of the corresponding evaluation distances in the test set are presented in Figure 3.8.

As expected, our iVAE model demonstrates better performance in terms of Sinkhorn distance when a larger weight is assigned to the Sinkhorn distance component in the loss function during training, and vice versa. This results in a trade-off between improved energy distance performance and improved Sinkhorn distance performance. Our initial choice of an equal weighting scheme appears to be a suitable compromise.

3.4.4 Exploratory analysis of the learned representations

Here, we focus on another interesting question, which is whether the learned representations in the low-dimensional latent space carry any relevant meteorological information or can offer additional insights about the data at hand. To that end, we focus on the temperature forecast data and try to detect seasonal patterns in the learned representations. We restrict our attention to latent representations of dimension 2 to enable graphical visualization. Figure 3.9 shows scatterplots of the components of the mean vector of the learned latent distributions for the different approaches.

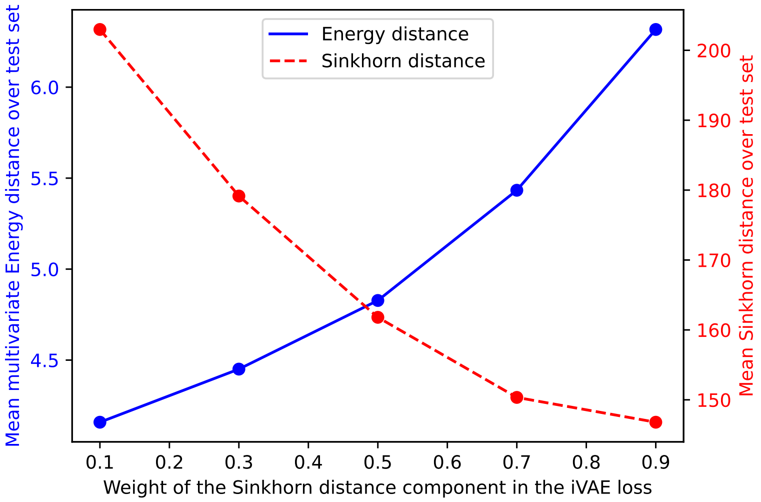


Figure 3.8: Mean multivariate energy distance (blue) and Sinkhorn distance (red) over the test set for reconstructed ensemble forecast fields of 2-m temperature generated by the iVAE method, shown as a function of the weight of the Sinkhorn distance component in the loss function.

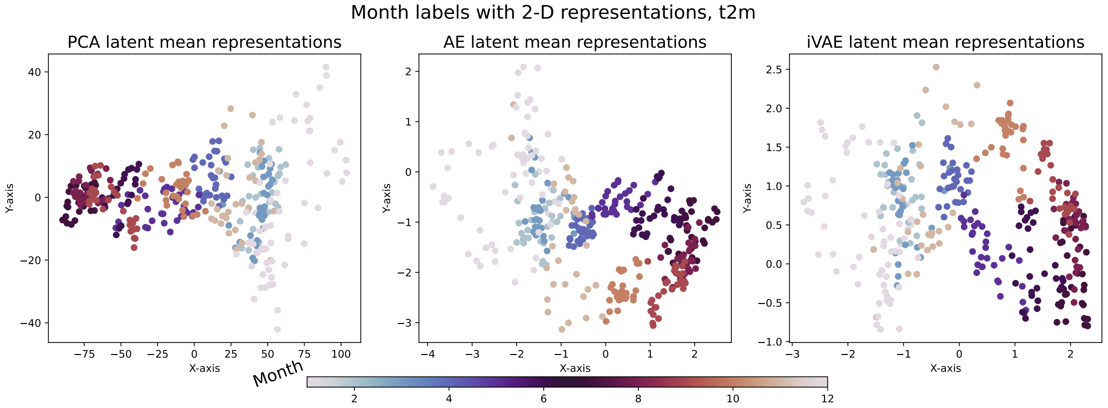


Figure 3.9: Scatterplots of the components of the mean vector of learned two-dimensional representations of temperature for the three different dimensionality reduction methods. The points are colored according to the month of the corresponding forecast date.

The mean vectors of the latent representations from all three methods show clear patters corresponding to the seasonality of the 2-m temperature, with data points representing warmer and colder weather clustered together, respectively. Notably, the two neural network-based approaches allow for a clearer distinction between different months, while the PCA-based representations tend to partially overlap for several months.

Similar scatterplots for the two components of 10-m wind speed are available in the Supplemental Material, however, there is a less clear seasonal pattern and a larger variability within individual months. We further explored the connection of the learned representations of ensemble forecast fields of geopotential height at 500 hPa to quasi-stationary, recurrent, and persistent large-scale atmospheric circulation patterns, so-called weather regimes (Grams et al., 2017). While there are noticeable patterns and clusters which could be incorporated into post-processing models (Mockert et al., 2024), the analysis is more involved and a more detailed investigation is left for future work. Some first results are available in the Supplemental Material.

3.5 Discussion and conclusions

We propose two types of approaches to learning probabilistic low-dimensional representations of ensemble forecast fields which aim to treat them as interchangeable samples from an underlying high-dimensional probability distribution. Two-step methods based on PCA or AE models first learn to assign a deterministic latent representation to each ensemble member, and proceed by fitting a multivariate probability distribution to the learned representations in the latent space. By contrast, the iVAE approach directly learns a probability distribution as latent representation and treats the ensemble members as invariant inputs. Both approaches allow for efficiently reducing the dimensionality of ensemble forecast fields within a probabilistic setting, where the learned distributions in the latent space enable the generation of arbitrarily many reconstructed forecast fields. Systematic comparisons of PCA- and AE-based approaches and the iVAE in case studies on temperature and wind speed forecasts over Europe indicate that the two NN-based approaches show promising results both in terms of the quality of reconstructed forecast fields and the informativeness of learned latent representations. While the results vary across all considered evaluation metrics, the iVAE model specifically performs best at preserving the variability information of the input ensemble forecasts. When compared to PCA, the NN-based methods generally show better performance for lower latent dimensions, whereas PCA yields equally good or even better reconstructions when the latent dimension is large. Overall, the computational costs of the iVAE are substantially higher than those of the AE-based and PCA-based approaches, although they remain manageable since the architectures are only comprised of dense layers. On a multiple-node CPU cluster, it takes less than one hour to train the iVAE model, while it takes only about 10 minutes to train the AE- or PCA-based model.

Despite the promising results, there are limitations to both types of approaches. All three methods assume a multivariate Gaussian distribution in the latent space, which might limit their applicability across different weather variables, specifically for variables such as precipitation, where the distribution should account for potential point masses at zero. A particular challenge in the specific setting of our dimensionality reduction problem is the evaluation of different approaches. In our experiments, we considered both deterministic and probabilistic metrics to assess the quality of reconstructed forecast fields. However, there is a general lack of suitable probabilistic evaluation tools that take into account structural aspects of the (ensemble of) forecast fields, compared to perception-based metrics proposed in the computer vision literature such as the widely used structural similarity index (Wang et al., 2004). Spatial evaluation approaches proposed in the meteorological literature might offer useful starting points, but are often heuristics-based, tailored to specific variables such as precipitation, and not straightforward to extend towards probabilistic settings, see, for example, Gilleland et al. (2010) and Dorninger et al. (2018) for overviews.

The proposed approaches provide several avenues for further generalization and analysis. Evidently, it would be of interest to investigate the scalability of the proposed methods towards larger grids and higher resolution forecast fields, as well as other variables. In particular, spatial forecasts of precipitation have been a focal point of research interest, for example in spatial verification (Roberts and Lean, 2008). Further, while we applied the dimensionality reduction methods separately to ensemble forecast fields of different variables, it would be interesting to apply them jointly to forecasts of multiple variables, and potentially over multiple time steps, at the same time. In addition, more advanced NN architectures such as transformers could be used as components of the AE-based or iVAE approaches. While we did not observe any benefits from using vision transformer architectures in initial experiments in the context of our case study, the results may vary if more training data was available. Given the recent developments in modern AI-based weather forecasting, including the generation of AI-based ensemble forecasts (Kochkov et al., 2024; Price et al., 2023; Bülte et al., 2024; Mahesh et al., 2024; Zhong et al., 2024), generating the corresponding ensemble forecast fields may become feasible and relevant for applications.

Finally, an important next step is to make progress towards integrating the proposed dimensionality reduction methods into downstream tasks. As discussed in the introduction, a key motivation for learning low-dimensional latent representations was to use those representations either as input data in, for example, hydrological or energy forecasting models, or to augment the input for neural network-based post-processing models (Lerch and Polsterer, 2022). Further examples of potential applications include (sub)seasonal weather prediction, where PCA-based representations of ensemble forecast fields have been used as inputs to machine learning models (Kiefer et al., 2023, 2024; Scheuerer et al., 2024), as well as analog forecasting, where compressed information from raw forecasts could be utilized for a more efficient generation of analogs (Grooms, 2021; Yang and Grooms, 2021). It is important to note that the quality requirements relevant to such applications differ from those in data reduction applications with a focus on data storage and management (see, e.g., Düben et al., 2019; Höhlein et al., 2022). While such approaches pursue accurate reconstruction as the central quality criterion, the achievable reconstruction quality may be decoupled from the information value of compressed representations in integrated modeling workflows. For example, in the specific context of incorporating learned representations of spatial input fields into NN-based post-processing models, one has to carefully balance the added value of the spatial information and the increased number of input predictors when the availability of training data is limited. In the setting of Lerch and Polsterer (2022), we did not find any improvements when using learned representations of the full ensemble of forecast fields instead of the mean forecast only. One explanation in line with other findings from related work (Höhlein et al., 2024;

Feik et al., 2024) might be that for post-processing, there seems to often be little value of including full information from an ensemble beyond simple summary statistics.

3.6 Supplemental Material

This Supplemental Material provides additional evaluation results for the three dimensionality reduction approaches presented in the main paper.

Figure 3.10 illustrates the proportion of variance explained by PCA with different choices of retained principal components. The first 5 principal components account for more than 90% of the variance in the original data, while the first 16 principal components account for more than 95%. This observation helps to explain the strong performance of the PCA-based approach when the latent dimension is large, as discussed in Section 4.2 of the main paper.

The evaluation results for optimal transport distances for temperature and the U component of wind speed data are presented in Figures 3.11 and 3.12. We observe slightly worse performance from the two neural network-based approaches for temperature data, where both methods fail to outperform the PCA-based method with a high latent dimension of 32. Notably, the iVAE method performs the worst in terms of Sinkhorn distance, despite incorporating the Sinkhorn distance component in its loss function. Conversely, both neural network-based approaches demonstrate significantly better

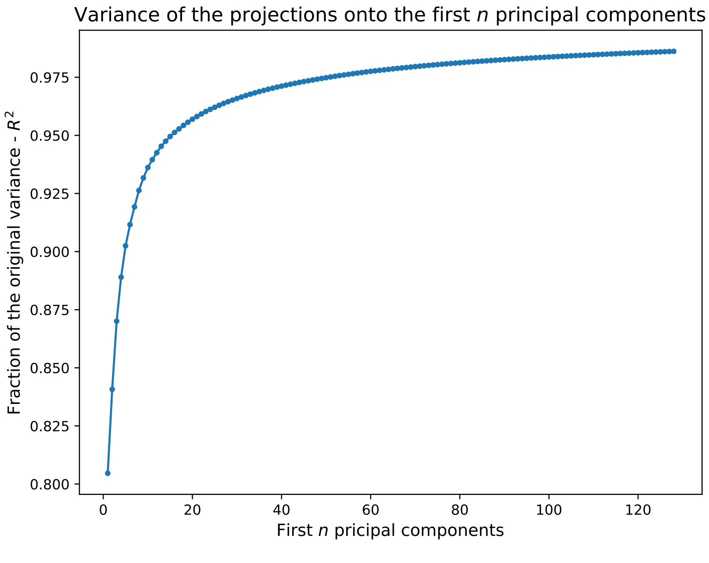


Figure 3.10: Proportion of variance explained depending on the number of retained principal components in PCA applied to temperature data.

performance for wind speed data, consistent with our findings on energy distances in the main paper.

Further evaluation results for two additional weather variables, the V component of 10-m wind speed and the geopotential height at 500 hPa, are presented in Figures 3.13–3.18.

The visualization of connections between the month information of the forecast date and two-dimensional mean representations learned by all three dimensionality reduction approaches, for geopotential height and the U and V components of wind speed, is provided in Figures 3.27–3.29. The visualization of connections between the specific weather regime at the forecast date and the learned two-dimensional mean representations is provided in Figures 3.30–3.33.

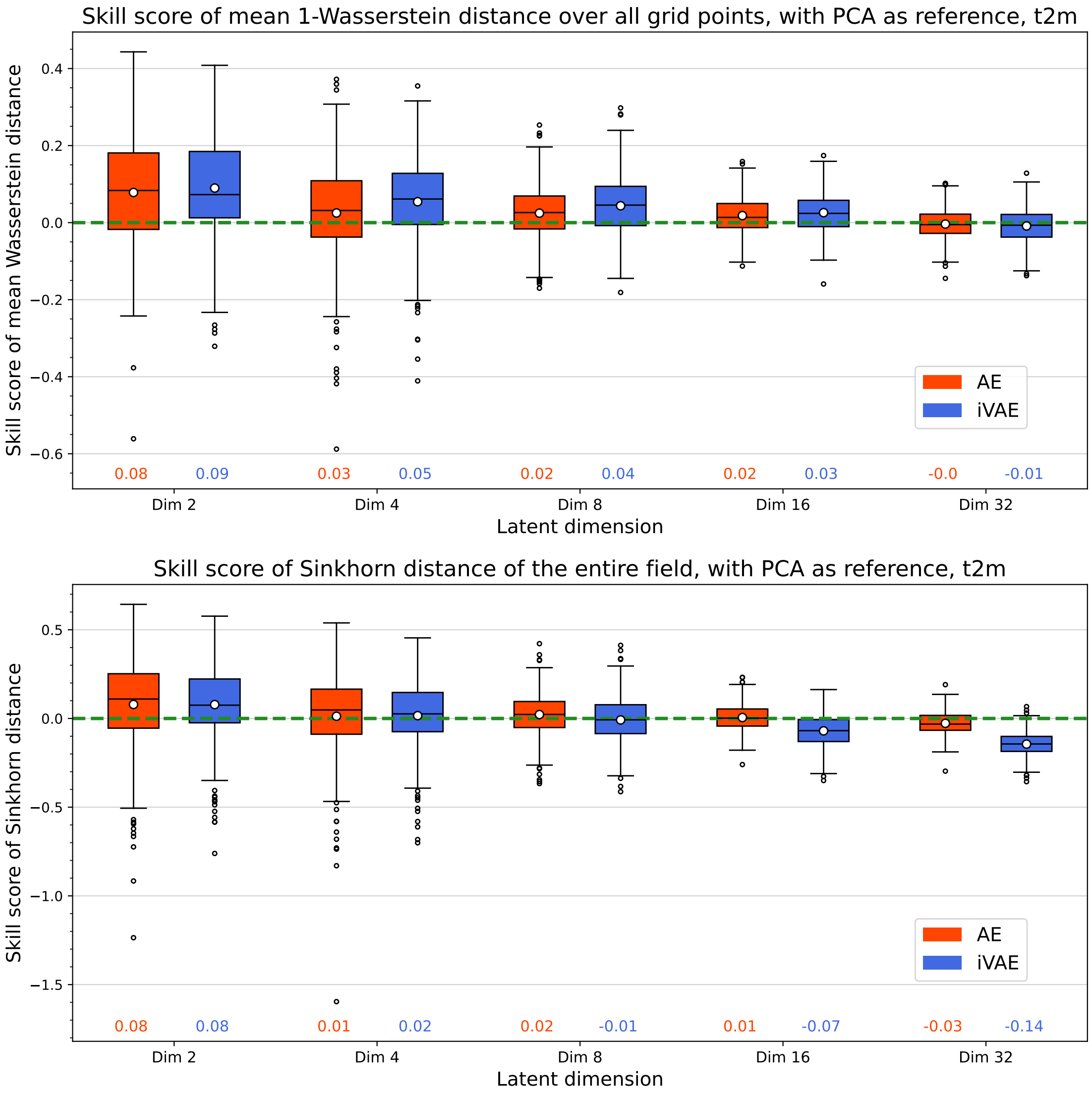
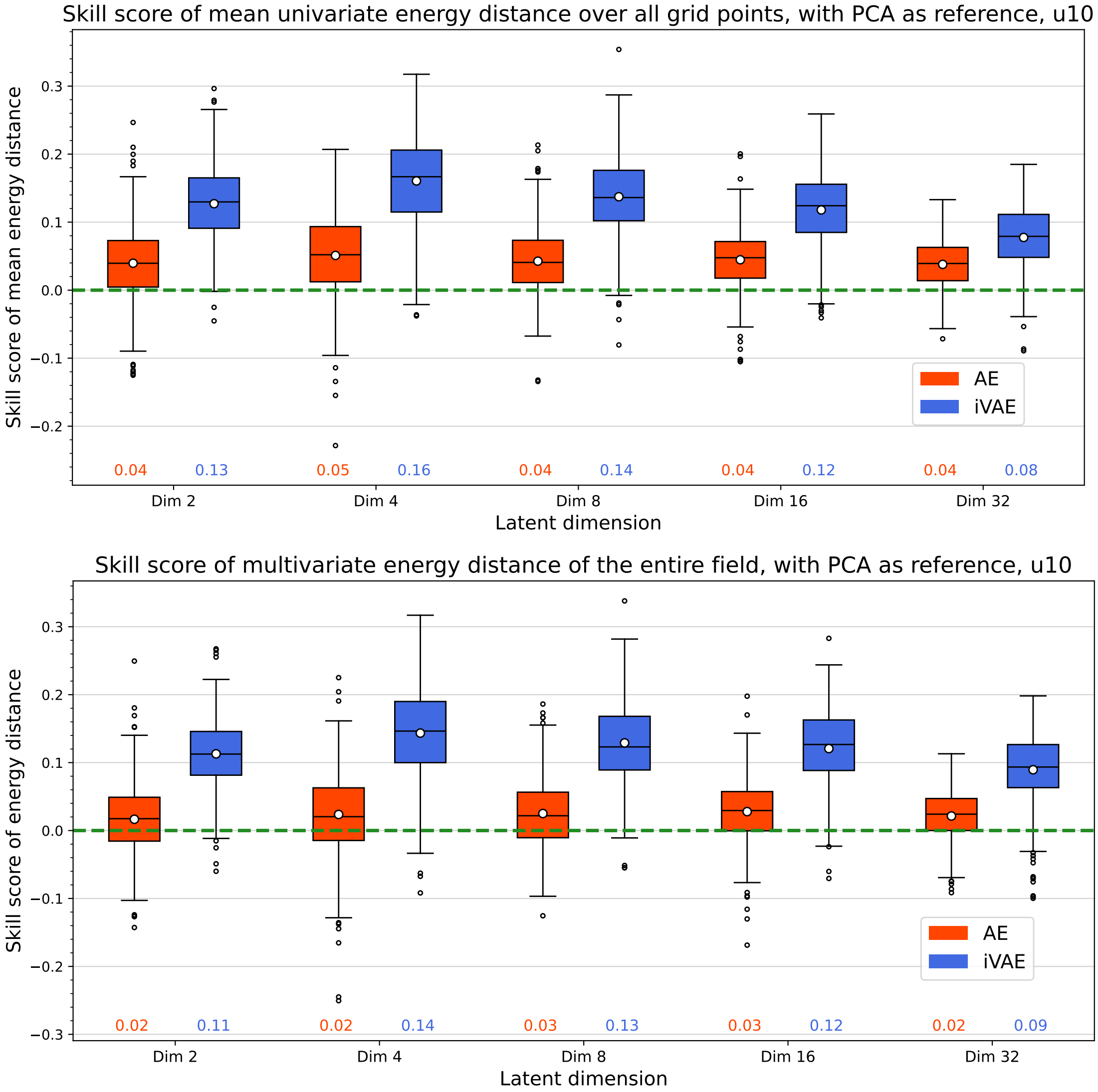


Figure 3.11: Boxplots of skill scores based on optimal transportation distances between the input and reconstructed ensemble fields over the 366 days in the test set for temperature data. The panels show mean univariate 1-Wasserstein distances over all grid points (top) and multivariate Sinkhorn distances computed for the entire fields (bottom). PCA-based approach shown in green dashed line is the reference method. The respective mean skill values are indicated below each box.



Skill score of mean univariate energy distance over all grid points, with PCA as reference, u10

Skill score of multivariate energy distance of the entire field, with PCA as reference, u10

Figure 3.12: As Figure 3.11, but for the U component of 10-m wind speed.

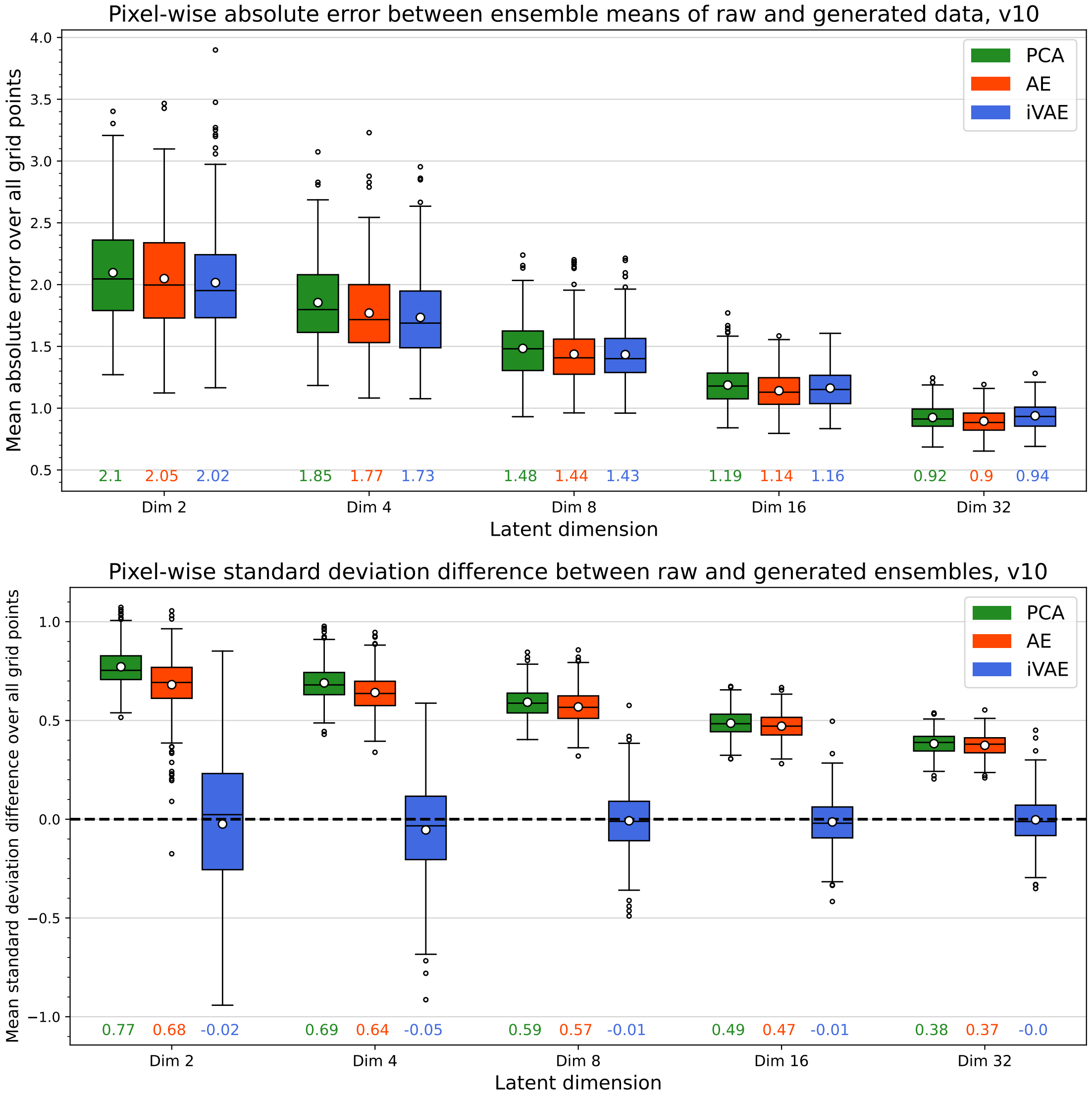


Figure 3.13: Boxplots of absolute differences between the mean values of input and reconstructed ensemble fields (top) and differences between the standard deviations of input and reconstructed ensemble fields (bottom) at each grid point. Boxes show the variability over 366 days in the test set of different methods for the V component of 10-m wind speed data, considering 5 different dimensionalities of the latent representation. The mean values of the (absolute) differences are indicated below each box. The differences between the standard deviations are computed such that negative values indicate a larger variability of the reconstructed ensemble compared to the input ensemble.

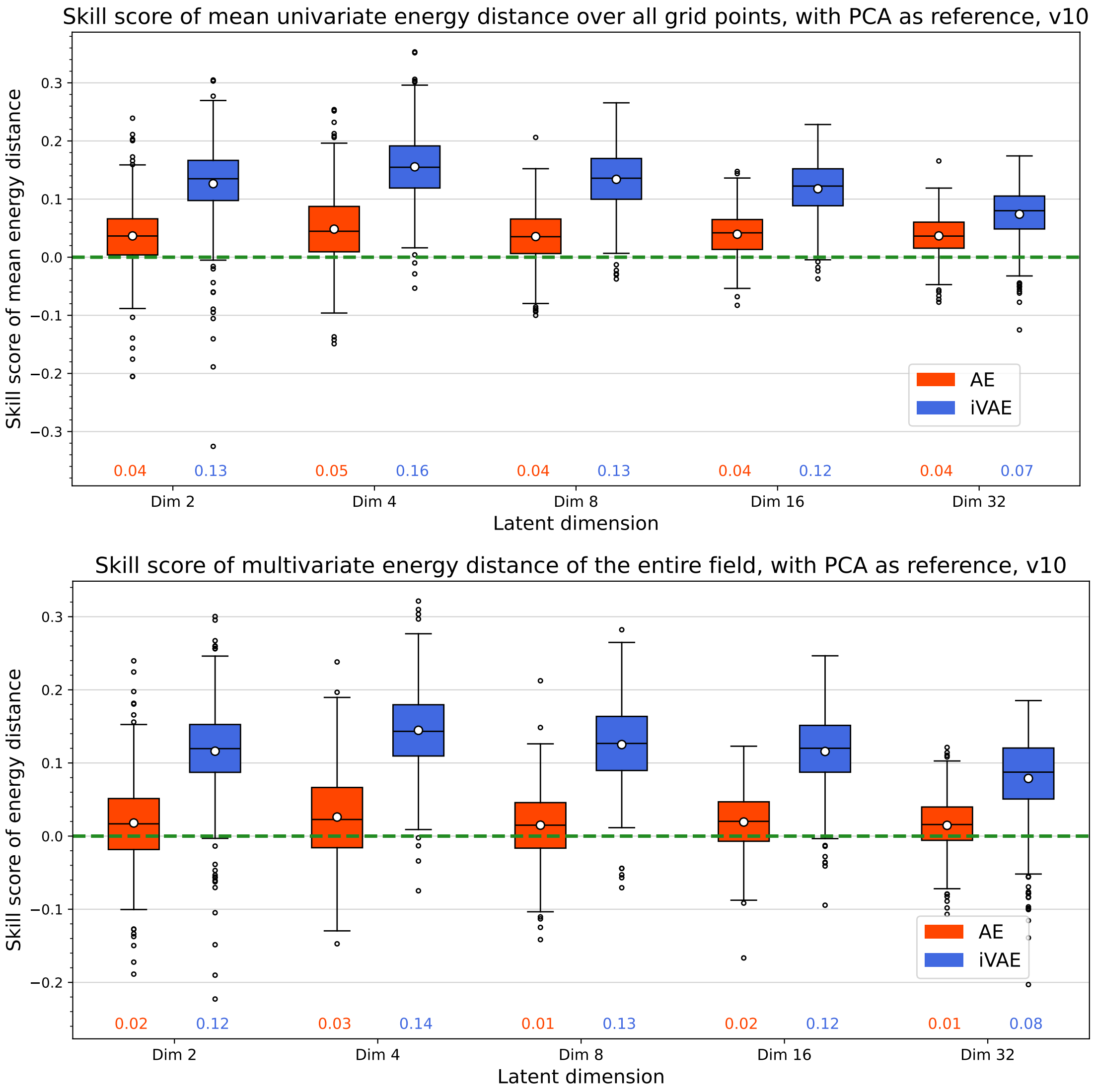


Figure 3.14: Boxplots of skill scores based on energy distances between the input and reconstructed ensemble fields over the 366 days in the test set for the V component of wind speed. The panels show mean univariate energy distances over all grid points (top) and multivariate energy distances computed for the entire fields (bottom). The PCA-based approach shown in green dashed line is the reference method. The respective mean skill values are indicated below each box.

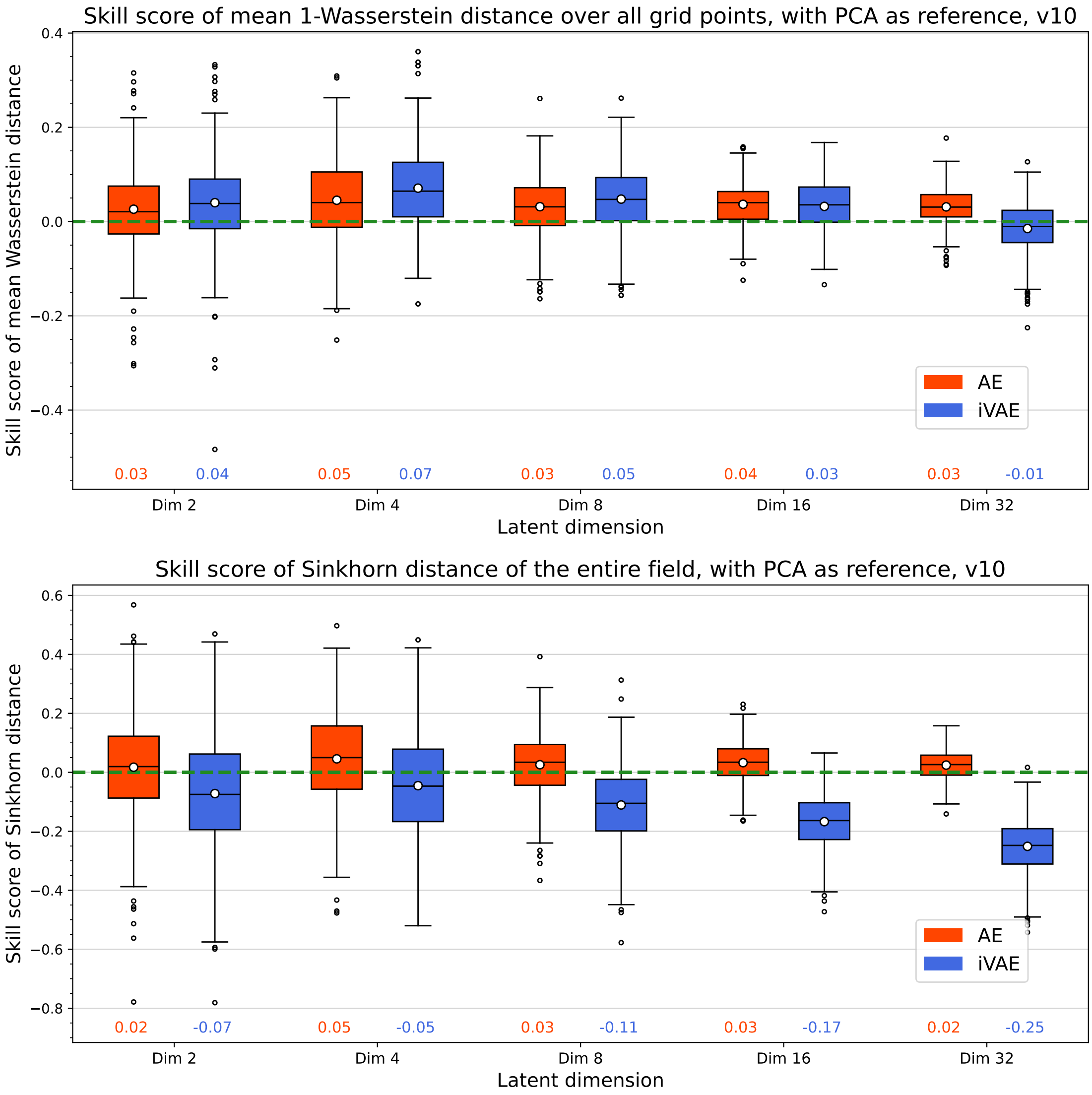


Figure 3.15: As Figure 3.11, but for the V component of 10-m wind speed.

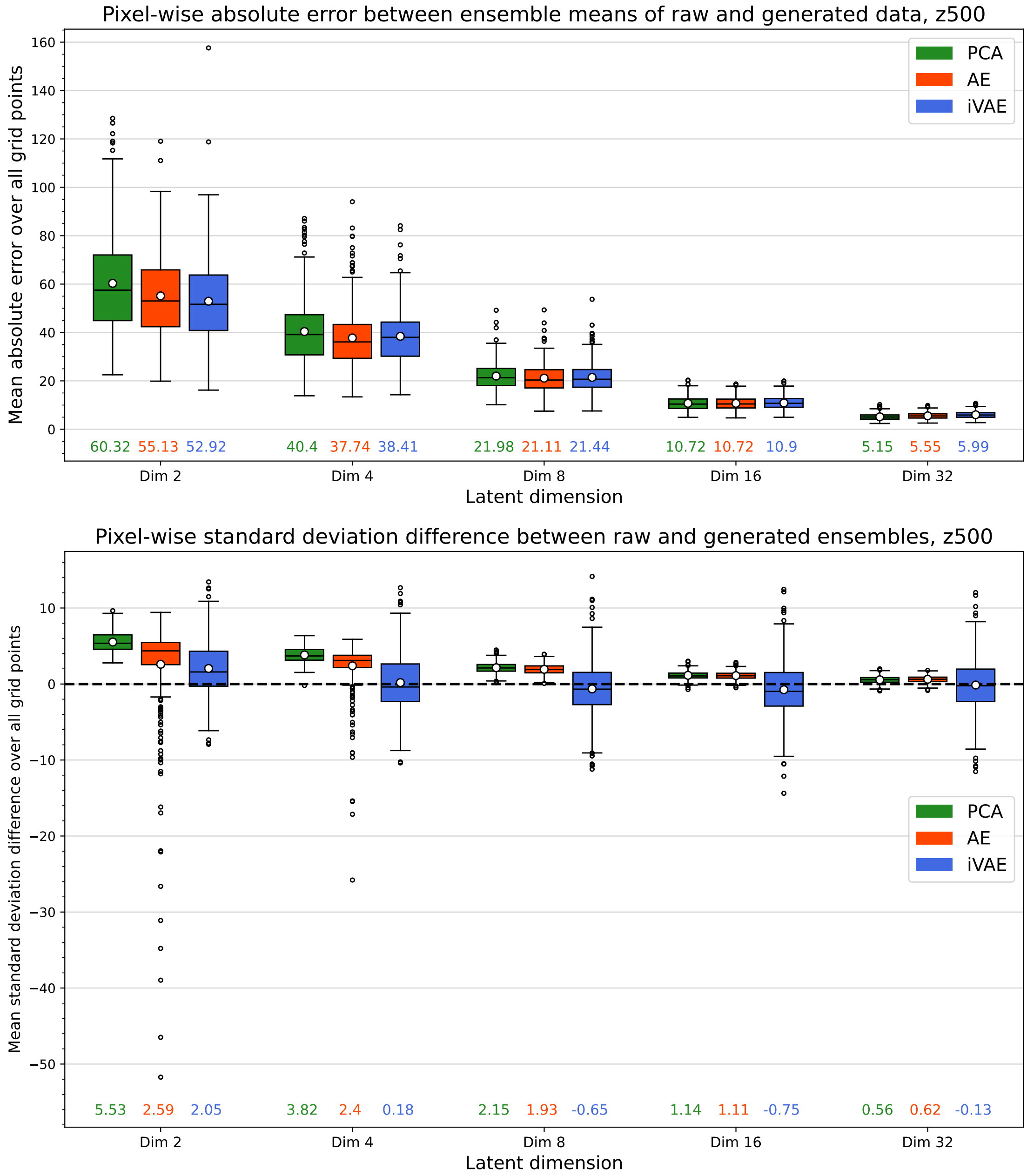


Figure 3.16: As Figure 3.13, but for geopotential height at 500 hPa.

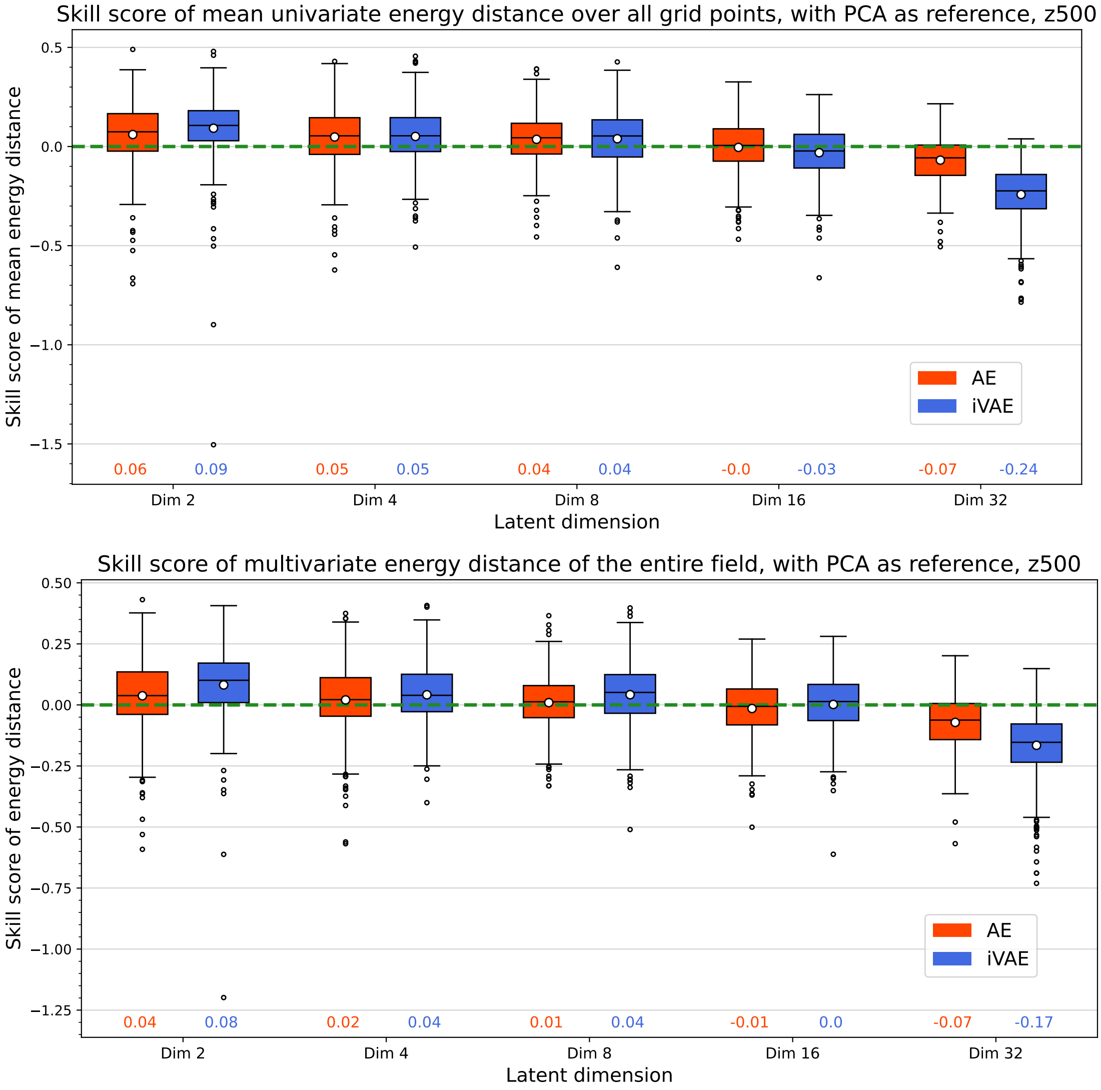


Figure 3.17: As Figure 3.14, but for geopotential height at 500 hPa.

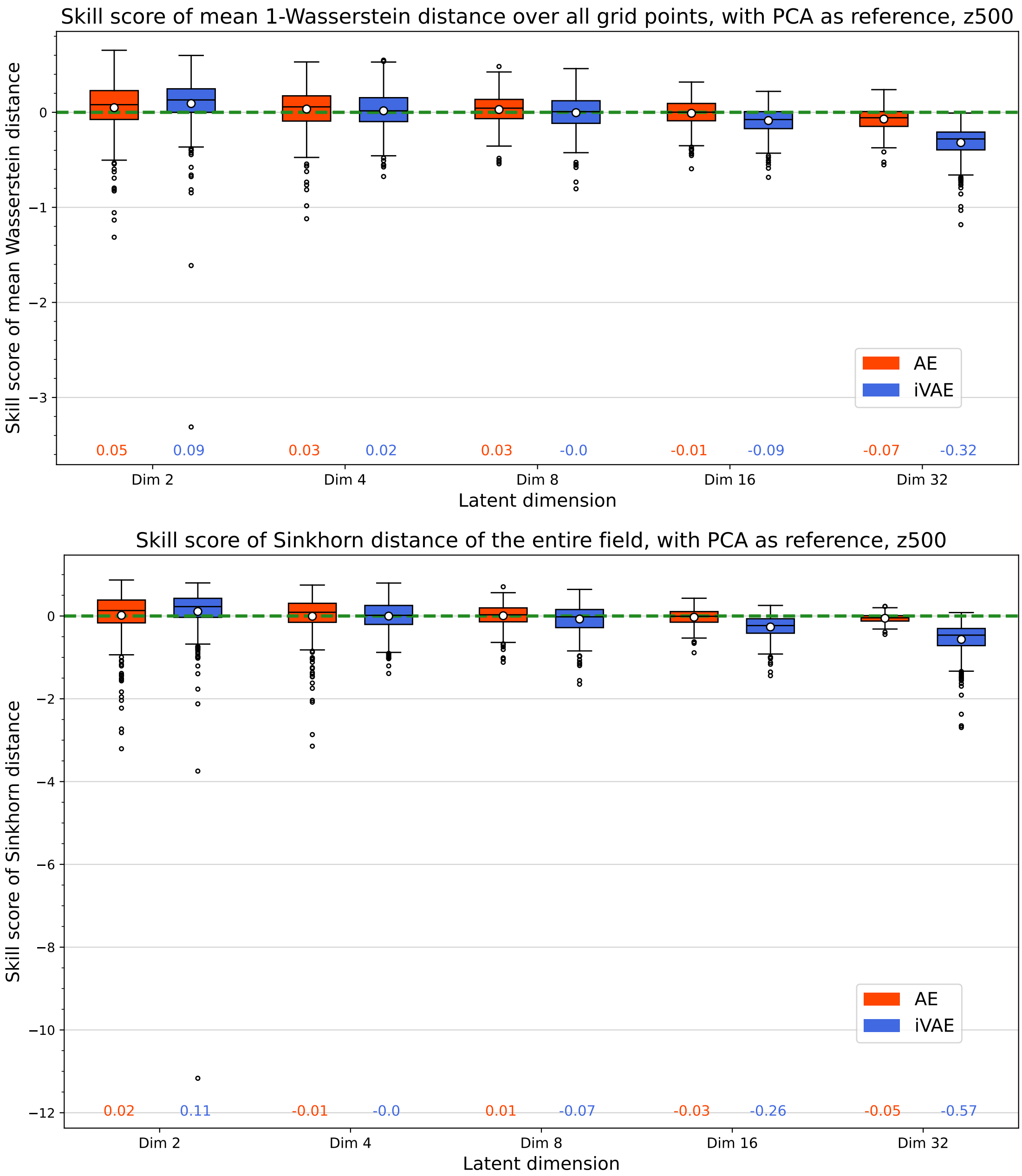


Figure 3.18: As Figure 3.11, but for geopotential height at 500 hPa.

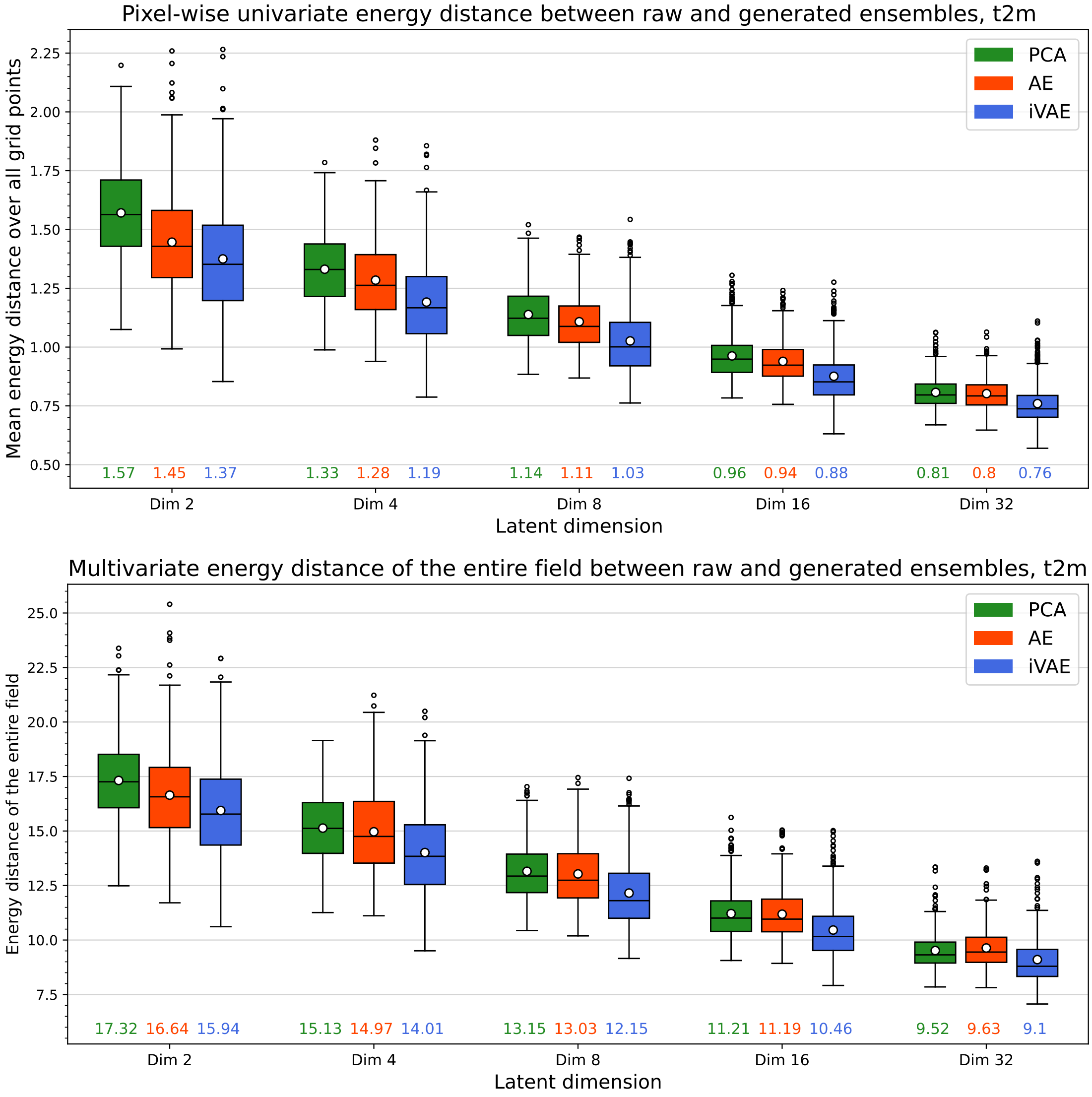


Figure 3.19: Boxplots of energy distances between the input and reconstructed ensemble fields over the 366 days in the test set for temperature data. The panels show mean univariate energy distances over all grid points (top) and multivariate energy distances computed for the entire fields (bottom). The respective mean values are indicated below each box.

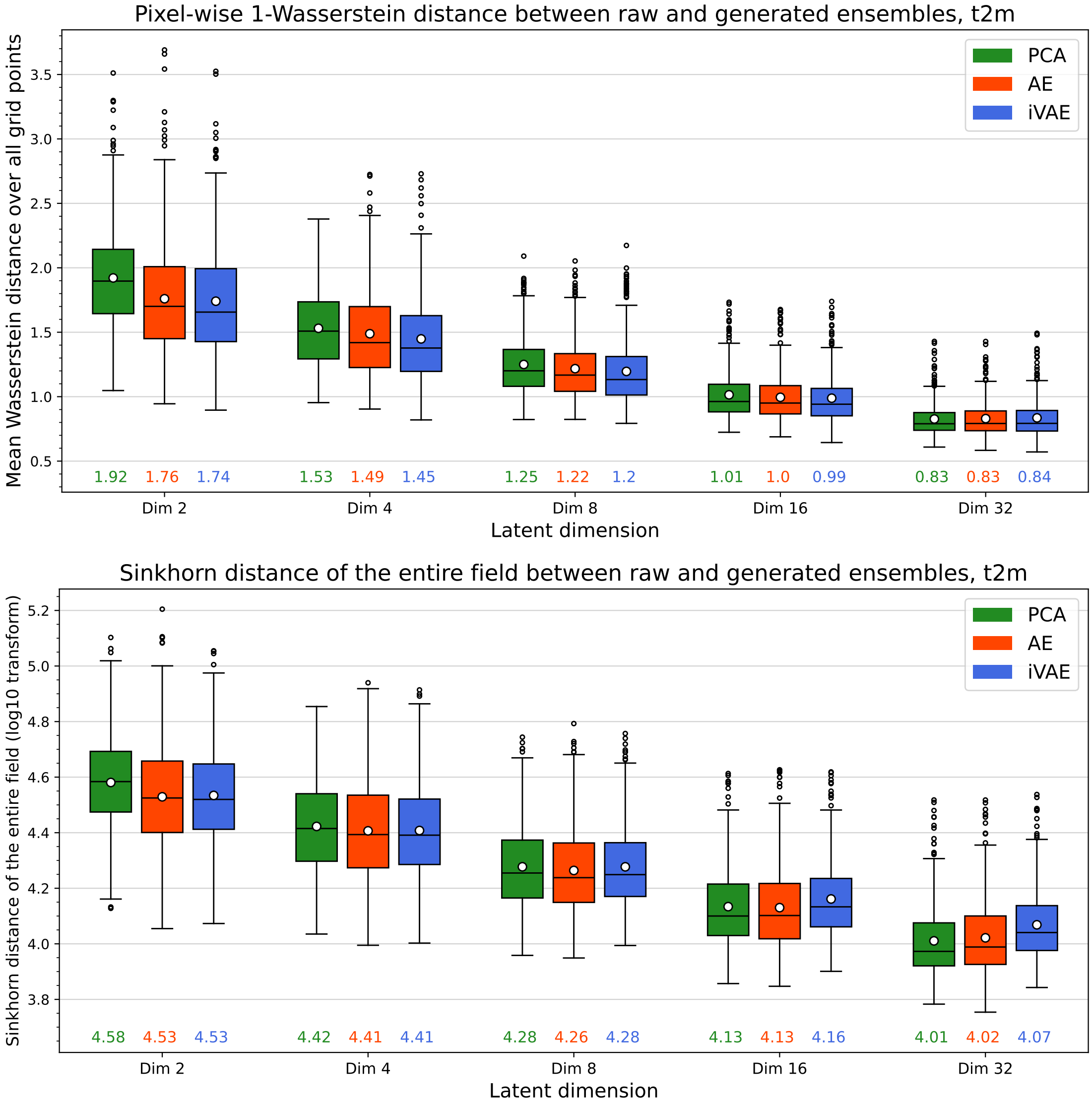


Figure 3.20: Boxplots of optimal transportation distances between the input and reconstructed ensemble fields over the 366 days in the test set for temperature data. The panels show mean univariate 1-Wasserstein distances over all grid points (top) and multivariate Sinkhorn distances computed for the entire fields (bottom). The respective mean values are indicated below each box.

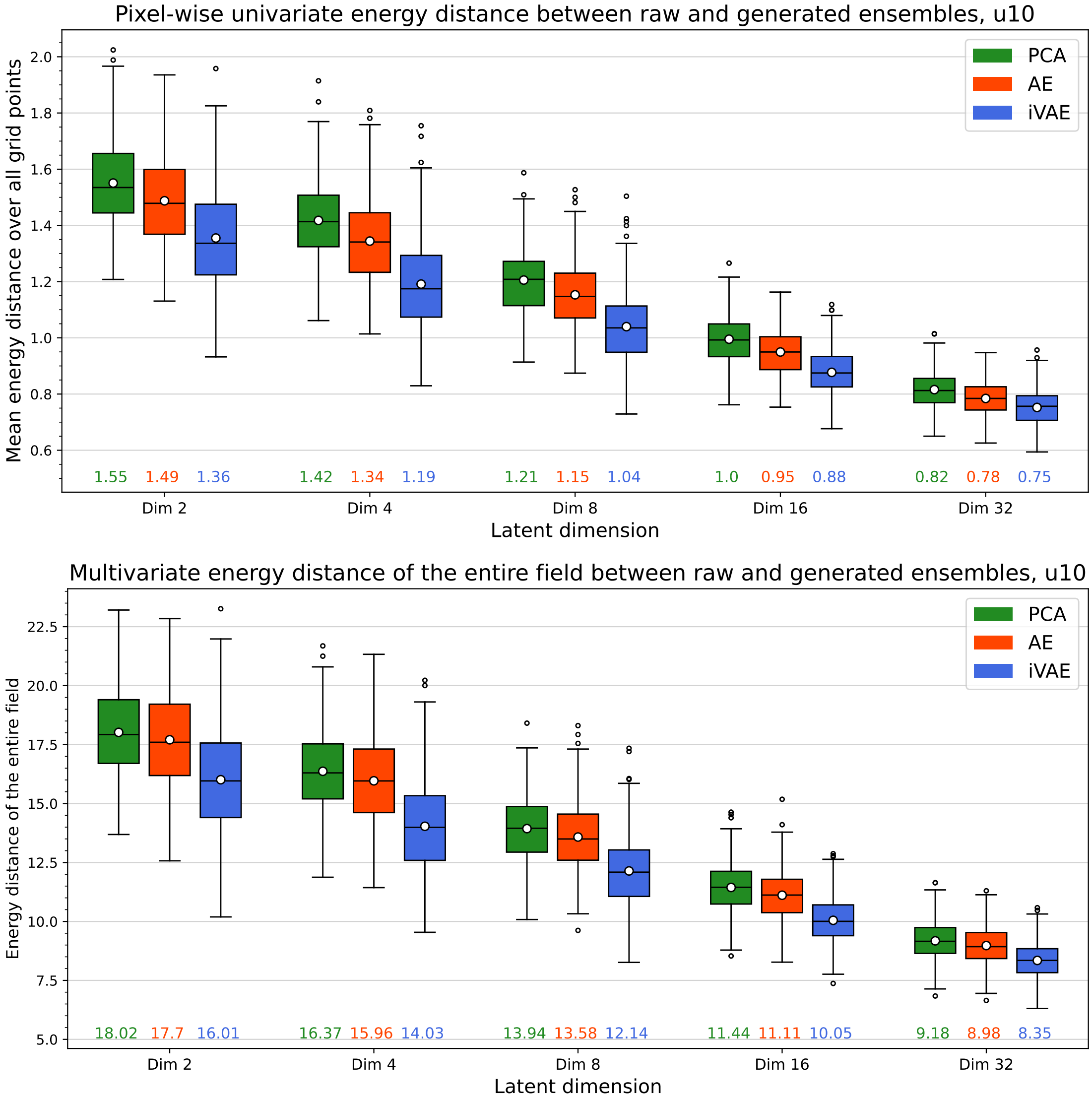


Figure 3.21: As Figure 3.19, but for the U component of wind speed.

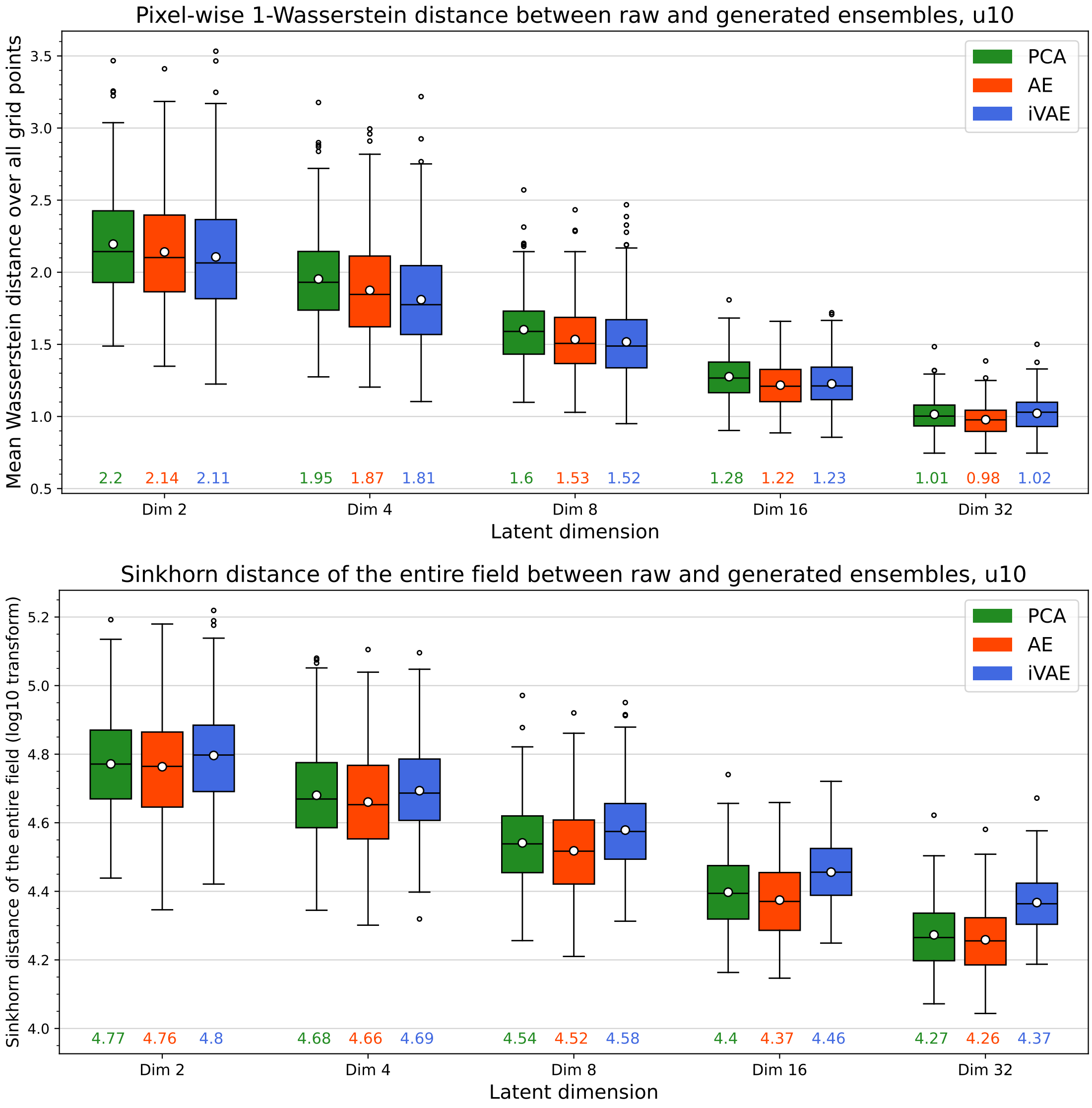


Figure 3.22: As Figure 3.20, but for the U component of wind speed.

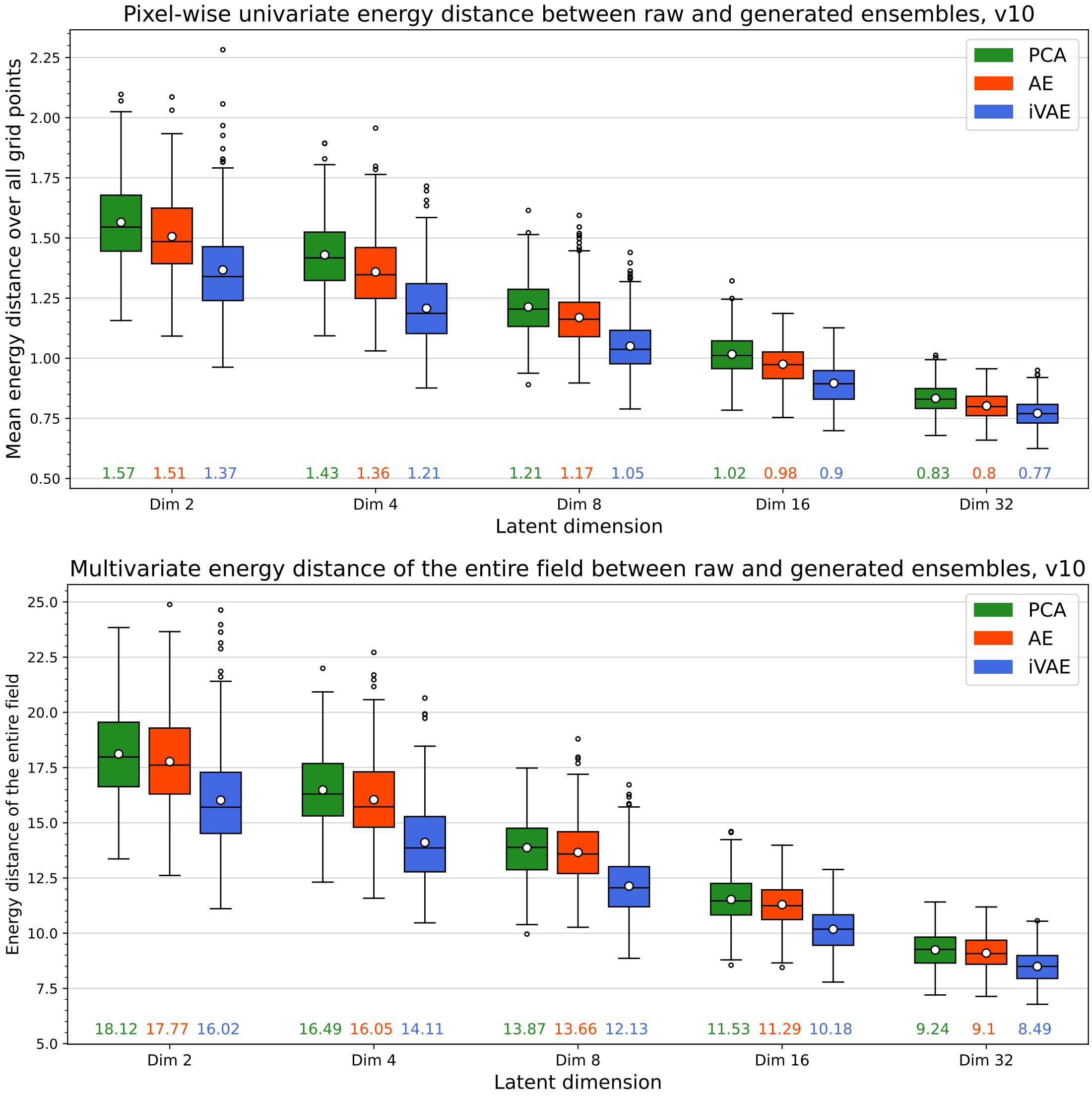


Figure 3.23: As Figure 3.19, but for the V component of wind speed.

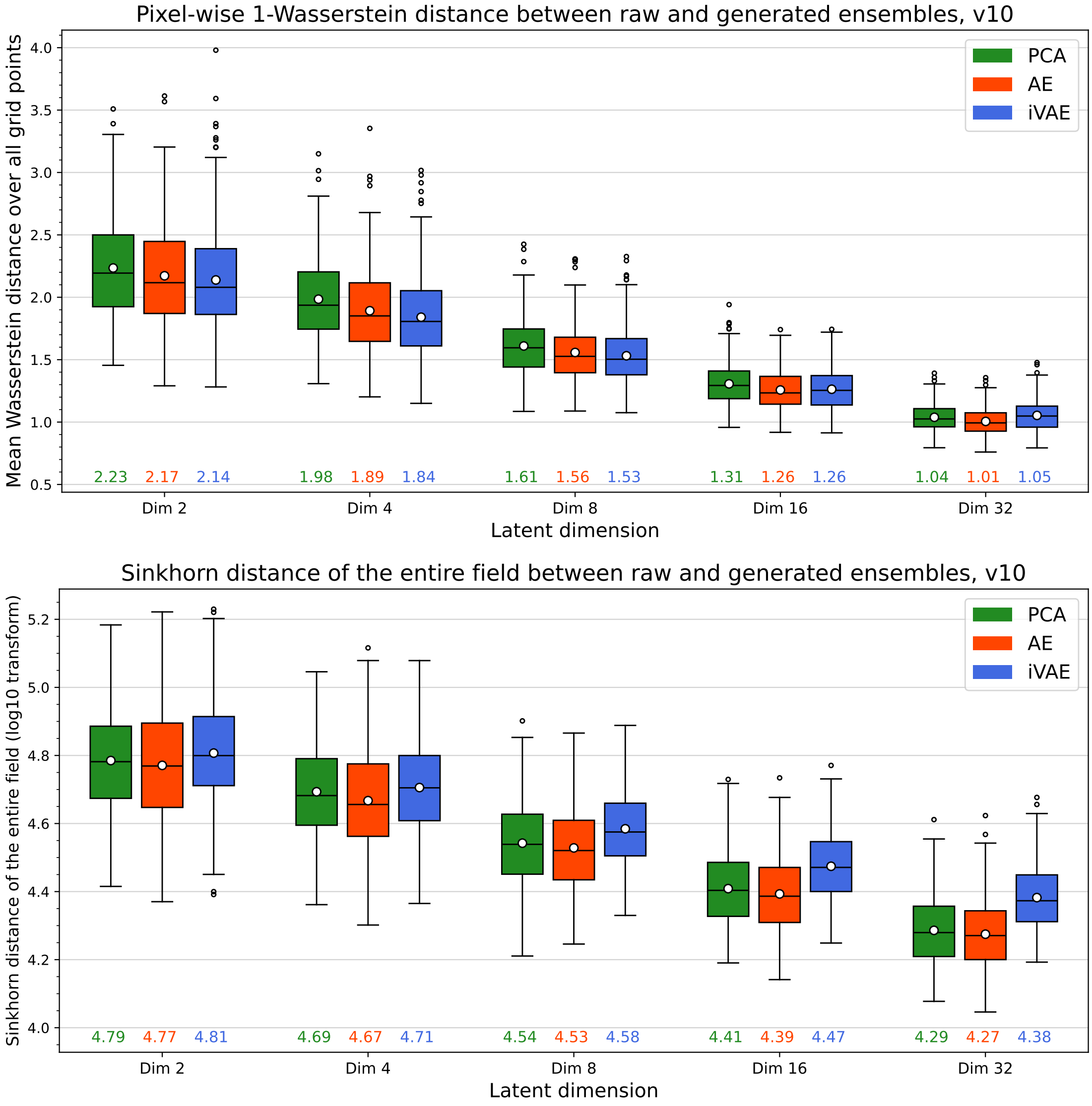


Figure 3.24: As Figure 3.20, but for the V component of wind speed.

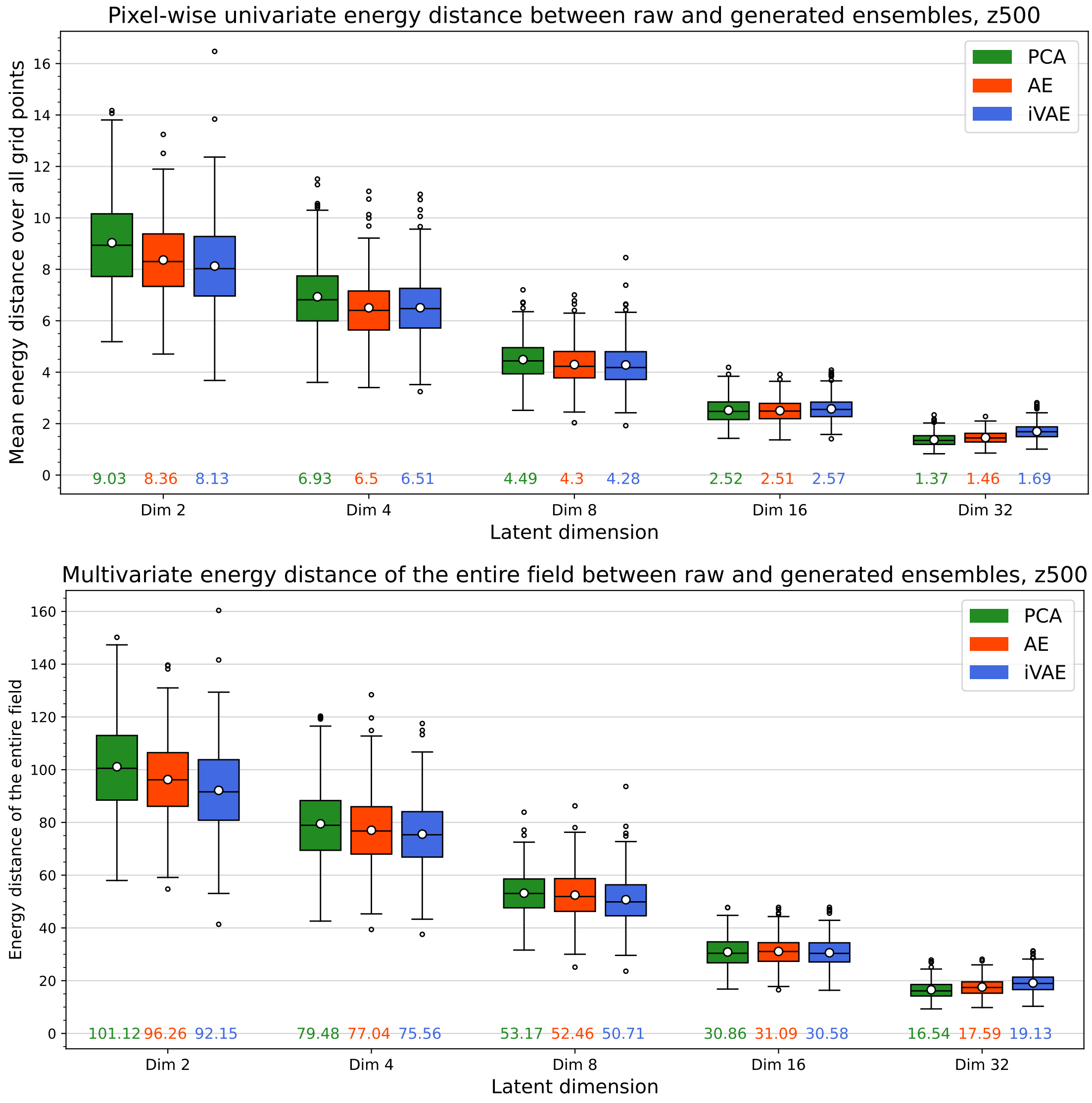


Figure 3.25: As Figure 3.19, but for the geopotential height.

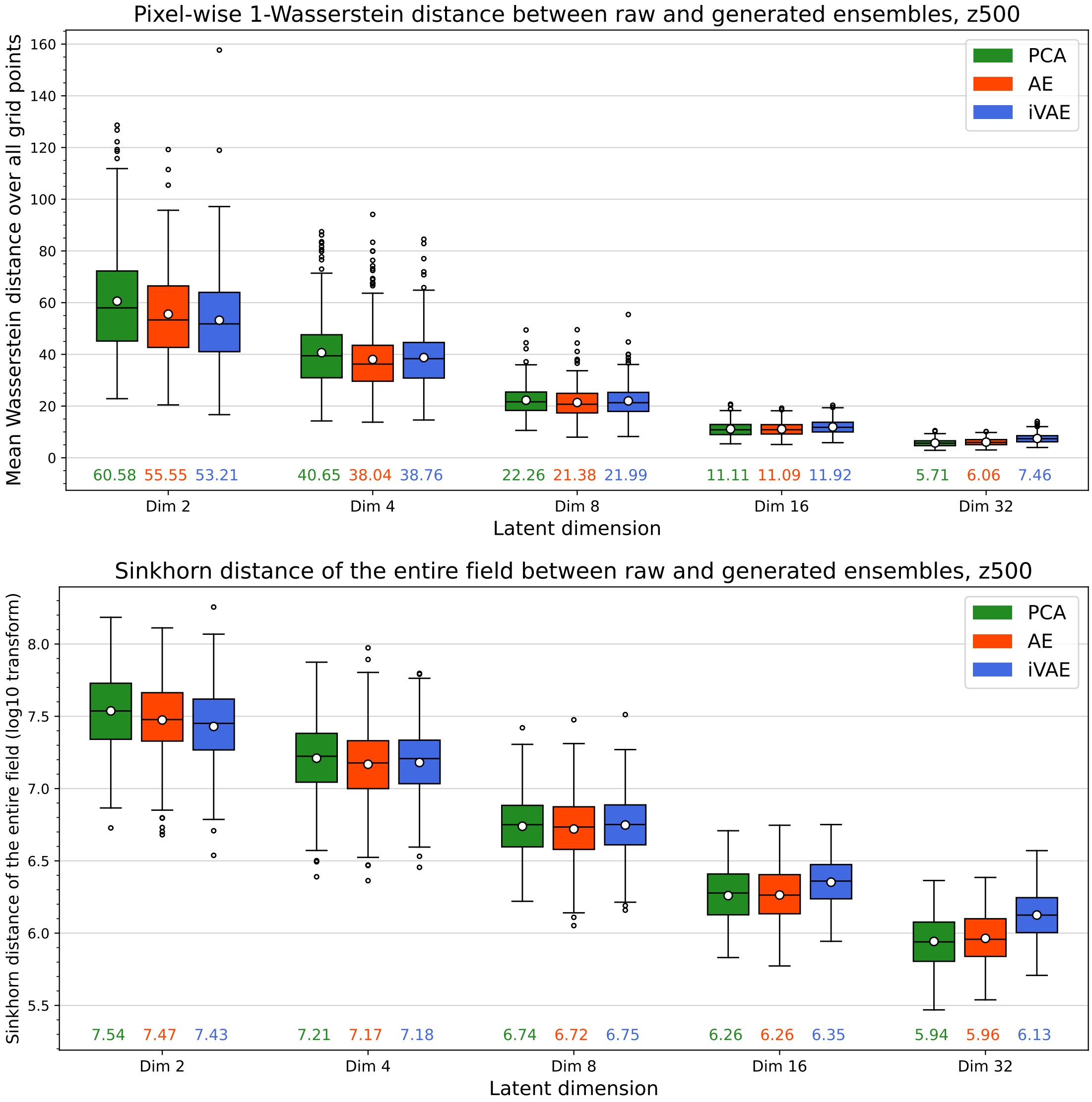


Figure 3.26: As Figure 3.20, but for the geopotential height.

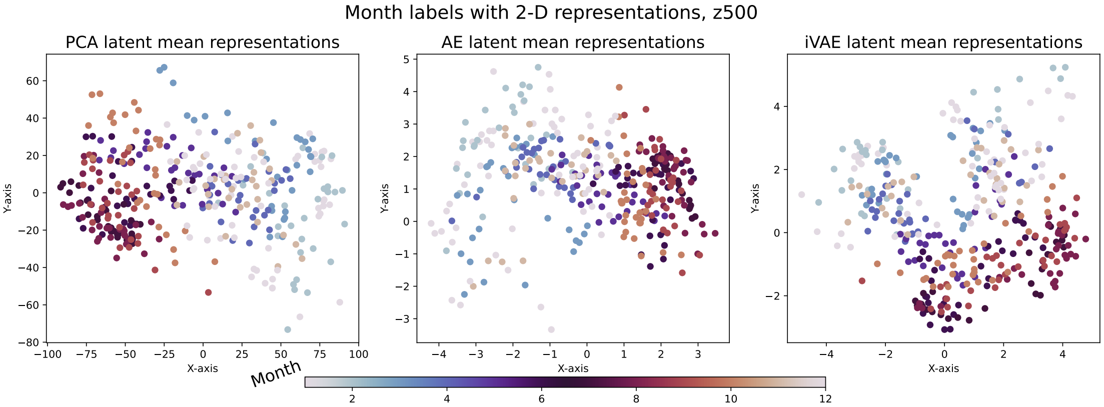


Figure 3.27: Seasonality visualization of learned 2-D representations from three different methods for geopotential height data, for each day the mean of the latent distribution is plotted, colors indicate month information of the forecast date.

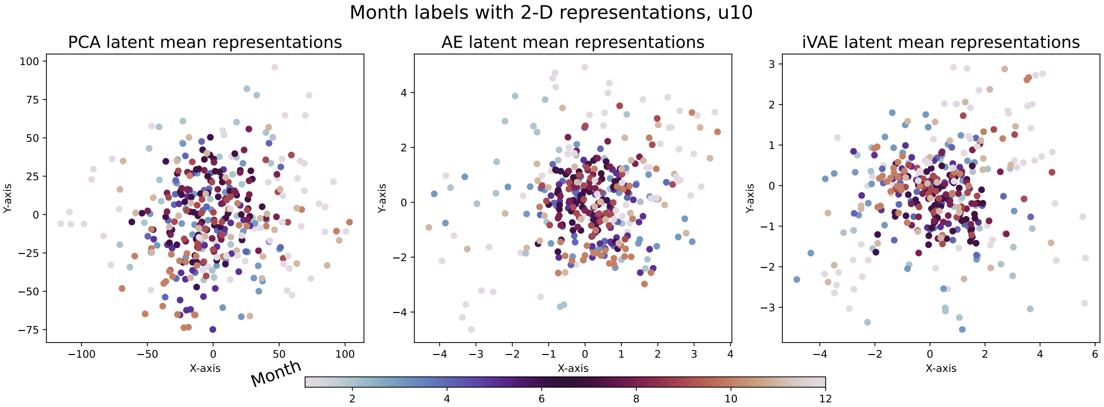


Figure 3.28: As Figure 3.27, but for the U component of wind speed.

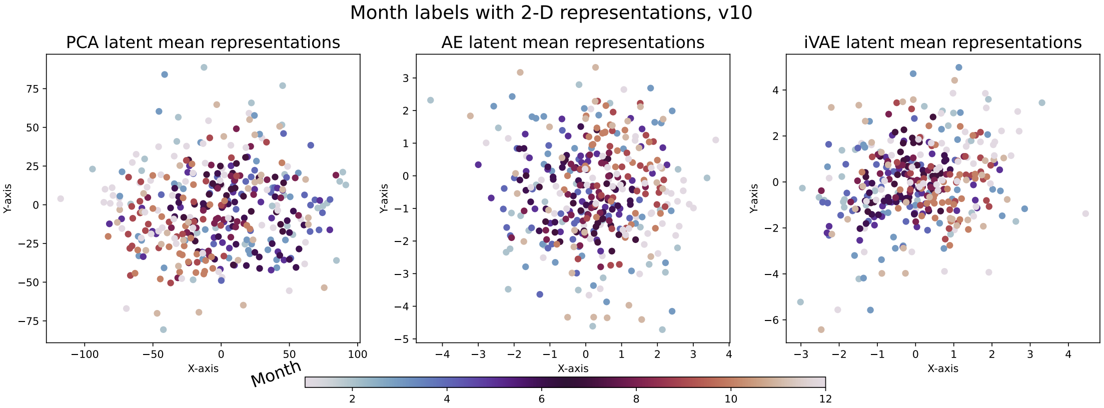


Figure 3.29: As Figure 3.27, but for the V component of wind speed.

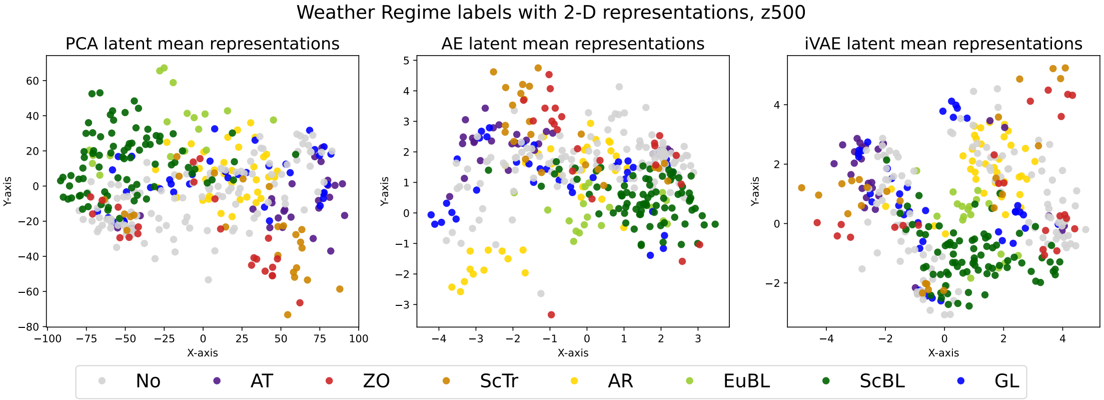


Figure 3.30: Visualization of learned 2-D representations from three different methods for geopotential height data, labeled and colored by the corresponding weather regime, for each day the mean of the latent distribution is plotted.

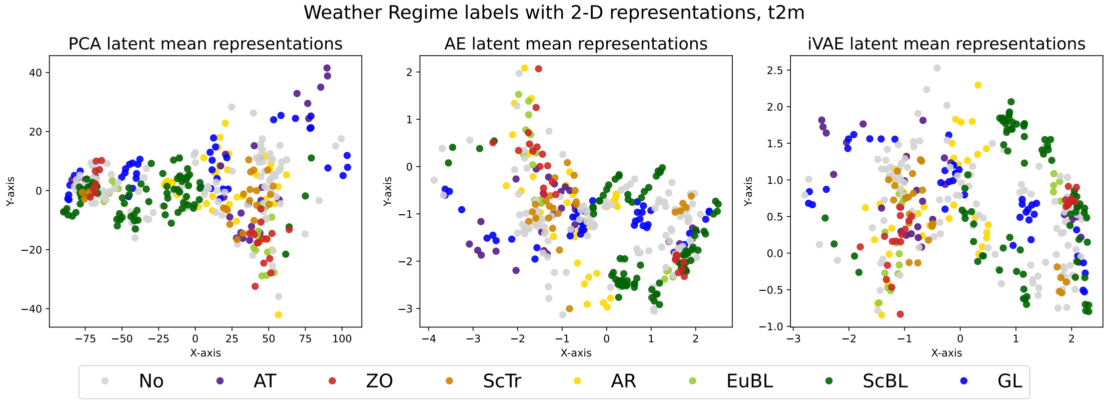


Figure 3.31: As Figure 3.30, but for temperature.

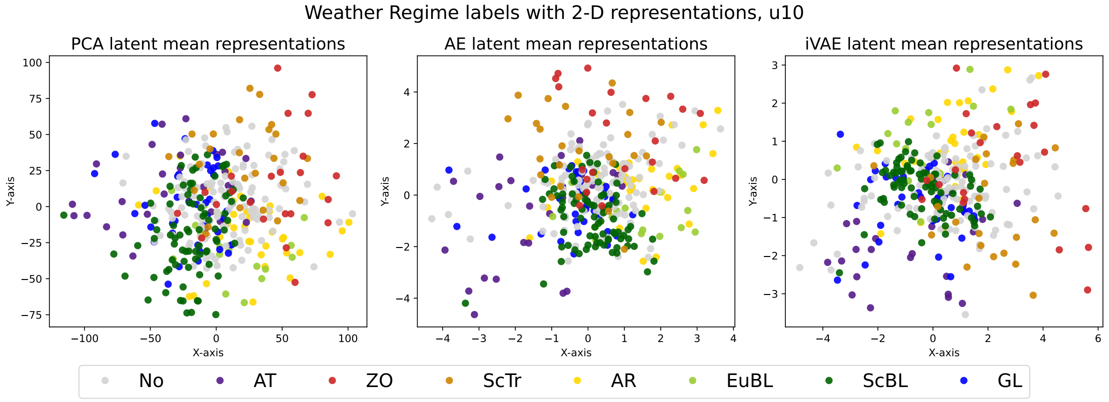


Figure 3.32: As Figure 3.30, but for the U component of wind speed.

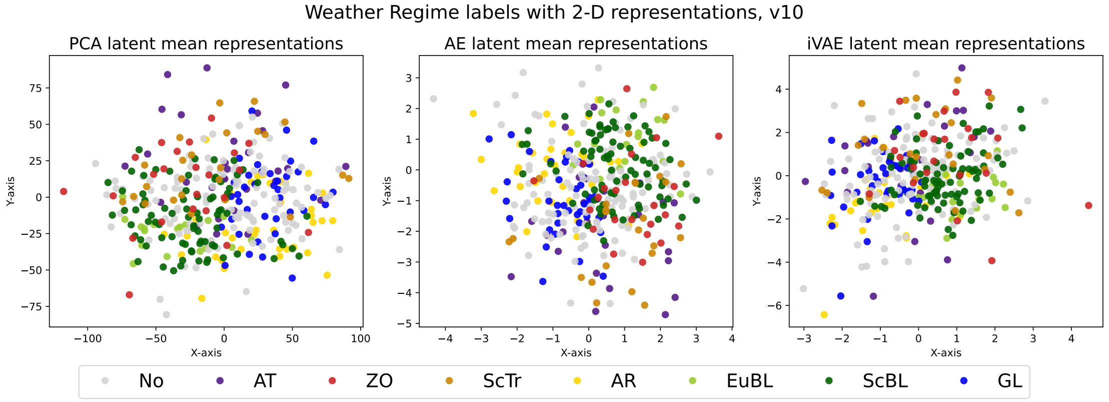


Figure 3.33: As Figure 3.30, but for the V component of wind speed.

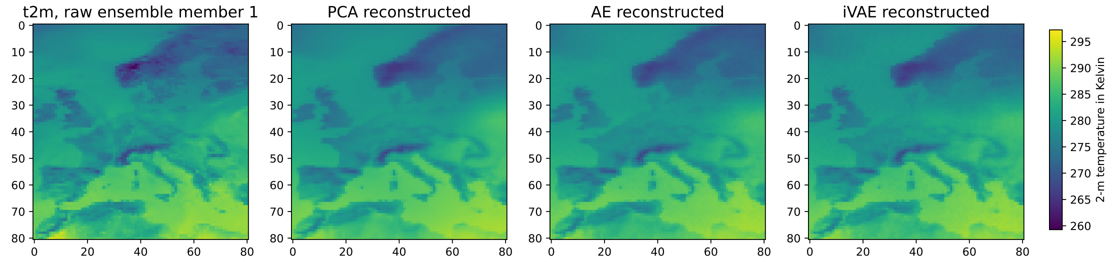


Figure 3.34: Animated figures of ensemble forecast fields from raw data and reconstructed data using different dimensionality reduction methods for temperature [click to make it move].

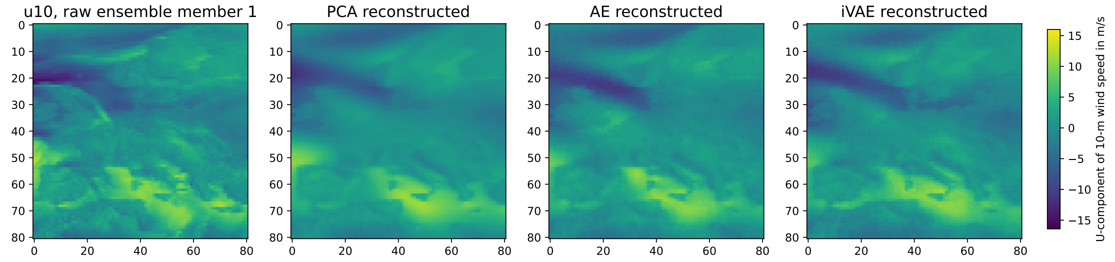


Figure 3.35: As Figure 3.34, but for the U component of wind speed [click to make it move].

4 Generative machine learning method for short-term path forecasts of intraday electricity prices

4.1 Introduction

Since the introduction of competitive electricity markets in the 1990s, the day-ahead (DA) auction has played a central role in power trading (Mayer and Trück, 2018; Weron, 2014). However, the increasing use of renewable energy sources (RES) is gradually shifting market activity toward intraday (ID) trading. Since 2015, trading volumes in the European ID markets operated by the European Power Exchange (EPEX) have increased by 300%, while DA volumes have risen by only 30% (EPEX, 2023).

This trend is making its way into the electricity price forecasting (EPF) literature, albeit with some delay. Of all Scopus-indexed publications from the years 2000-2009, only 5% focused on predicting ID (or real-time) prices.[1] The share increased to 11% in the next decade and then rapidly rose to 17% in just the last five years. One possible reason for the delay and lower interest is the variety of market designs (Glachant et al., 2021), making it difficult to compare the findings between studies. North American real-time markets follow a mandatory, security-constrained economic dispatch. In contrast, European markets often rely on voluntary ID auction and/or continuous-time trading,

[1] We used the Scopus query `TITLE((forecast* OR predict*) AND price*) AND TITLE-ABS-KEY("electric* market" OR "power market")` combined either with `AND TITLE-ABS-KEY("day-ahead" OR "spot" OR "next-day")` to identify DA-related or with `AND TITLE-ABS-KEY("intraday" OR "intra-day" OR "real-time")` to identify ID-related publications. Naturally, some of these papers concern both DA and ID price forecasting.

which precedes the final settlement in the balancing market (Backer et al., 2023; Cramton, 2017; Maciejowska et al., 2023).

The existing literature on price forecasting in intraday electricity markets considers different perspectives. Some authors predict ID prices for the next day to take advantage of arbitrage opportunities (Maciejowska et al., 2021), to optimize the scheduling of a behind-the-meter storage system (Chitsaz et al., 2018) or to manage the risk associated with trading (Klein et al., 2023; Janczura and Wójcik, 2022; Browell, 2018). Other studies focus on very short-term forecasting with lead times ranging from a few hours (Monteiro et al., 2016; Uniejewski et al., 2019; Narajewski and Ziel, 2020a) to an hour or less before delivery (Browell and Gilbert, 2022; Bunn et al., 2018). Many of these studies focus on probabilistic forecasts in the form of predictive distributions. Those quantify predictive uncertainty and thus offer essential information for market participants. In particular, multivariate probabilistic forecasts that account for temporal dependencies over the evolution of ID price trajectories across different times are of central interest for decision-making in the trading markets, as discussed in a recent study by Hirsch and Ziel (2024).

Regarding forecasting techniques, traditional econometric models are still in use (Janczura and Puć, 2023; Maciejowska, 2022; Russo et al., 2022), but they are increasingly being replaced by statistical learning methods (e.g., LASSO; Narajewski and Ziel, 2020a; Uniejewski et al., 2019) and deep neural networks (Oksuz and Ugurlu, 2019; Zhang and Wu, 2022; Klein et al., 2023; Cramer et al., 2023), which generally achieve higher accuracy. However, to the best of our knowledge, with the exception of Janke and Steinke (2019) and Hirsch and Ziel (2024), no neural network model has been proposed in the literature to predict marginal distributions or joint multivariate distributions with temporal dependence of intraday prices in a continuous-time electricity market.

Model features also differ, especially for studies of European continuous-time ID markets, where most rely on aggregate price indicators such as the ID3 index, i.e., the weighted average price of all transactions executed within the last 3 hours of a delivery contract (Maciejowska, 2022; Uniejewski et al., 2019; Narajewski and Ziel, 2020a; Russo et al., 2022; Cramer et al., 2023). Although it is convenient to summarize the evolution of a price trajectory with a single value, this approach neglects the potential trading opportunities resulting from the RES generation updates (Kuppelwieser and Wozabal, 2023). Furthermore, the timing of transactions has a significant impact on trading revenues (Serafin et al., 2022; Janke and Steinke, 2019). Therefore, the ability to simulate plausible intraday price paths is highly valuable. However, apart from Narajewski and Ziel (2020b), Serafin et al. (2022) and Hirsch and Ziel (2024), research on this topic remains limited, despite its high relevance for market participants.

To address this gap, we propose a generative neural network model designed to predict multivariate distributions of the ID price path, capturing temporal dependencies to generate realistic price path trajectories. Our method is a data-driven, nonparametric approach, where the neural network directly outputs ID price path trajectories, incorporating information from historical price data and relevant exogenous input variables. This approach builds on the conditional generative model (CGM) developed for multivariate probabilistic weather forecasting by Chen et al. (2024), which in turn extends earlier work of Janke and Steinke (2020) on multivariate prediction of DA prices. The CGM belongs to the class of scoring rule-based generative neural networks, where the model generates meaningful data from noise and is optimized using a loss function that measures the discrepancy between generated and real data. Training the CGM involves optimizing a suitable multivariate proper scoring rule, e.g., the energy score, that quantifies the discrepancy between multivariate forecast samples (i.e., the price path trajectories) and a realization vector representing the temporal path of observed ID prices. By conditioning on explanatory inputs, the model effectively captures nonlinear relationships for both marginal forecast distributions and temporal dependencies in the price paths, and integrates them into the output path trajectories. Our CGM approach is in contrast to the commonly followed two-step framework for multivariate probabilistic forecasting, which proceeds by separately modeling the marginal distributions and the multivariate dependencies. Such two-step framework has been adopted in many disciplines, including EPF (Ziel and Weron, 2018) and weather forecasting based on ensemble post-processing (e.g., Schefzik et al., 2013; Lerch et al., 2020; Lakatos et al., 2023).

The specific application in electricity markets highlights the need for evaluating the performance of probabilistic forecasts from both statistical and economic perspectives. Although many statistical measures have been proposed to assess the calibration and accuracy of univariate and multivariate probabilistic forecasts, these metrics typically do not directly correspond to the economic value obtained in real market scenarios. The utilization of probabilistic multivariate forecasts for making optimal trading decisions and the economic evaluation of specific trading behaviors thus is of particular importance in this context. Following previous research by Serafin et al. (2022), we consider a simple trading scenario and propose several strategies to make optimal trading decisions based on multivariate probabilistic forecasts of electricity prices, and evaluate their performance based on an economic assessment of profit gains.

The remainder of this paper is structured as follows. Section 4.2 provides a comprehensive description of the datasets used in this study. In Section 4.3, we present three approaches to probabilistic path forecasting of ID electricity prices, including the proposed CGM and two statistical benchmark methods. In Section 4.4, we describe the scoring rules utilized to evaluate the accuracy of path forecasts and introduce trading strategies applied for economic assessment in a case study. Section 4.5 presents the results of both the statistical and economic evaluations, and discusses the practicality and effectiveness of these methodologies. Finally, Section 4.6 concludes the key findings of this research. Python code for implementations of all forecasting methods is available online (`https://github.com/jieyu97/epf_cgm`).

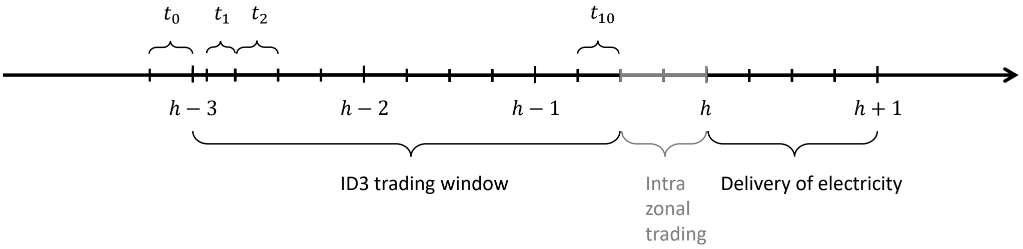


Figure 4.1: Timeline of the forecasting framework. Forecasts for ten 15-minute subperiods, denoted by $t_1, \ldots, t_{10}$, are generated three hours prior to delivery. The last 30 minutes before delivery, during which trading is restricted to within control zones, are excluded from the analysis. Note that the first subperiod, t_1, covers only 10 minutes, as the first five minutes are reserved for data collection and model execution.

4.2 Data

The German intraday market for electricity offers both auction-based and continuous-time trading for hourly, half-hourly, and quarter-hourly products. In this study, we focus exclusively on the continuous-time market for hourly delivery periods, which represents the most liquid segment (EPEX, 2023; Narajewski and Ziel, 2020a). Trading for these products begins at 16:00 on the day before delivery and ends 30 minutes prior to delivery, or 5 minutes prior within control zones. Unlike auctions, prices in the continuous-time market evolve in real time as transactions occur between participants, resembling the dynamics of a financial market with a limit order book (Kuppelwieser and Wozabal, 2021).

We consider ID price trajectories spanning the period from 15.06.2017 to 29.09.2019[2], before the start of the crisis periods with COVID-19 and the Russian attack on Ukraine. Like Serafin et al. (2022), we focus on the Volume Weighted Average Prices (VWAPs) of all transactions in the ten 15-minute subperiods $t_1, t_2, ..., t_{10}$ ranging from 3 hours to 30 minutes before delivery, see Figure 4.1. Note that the first subperiod, t_1, covers only 10 minutes, as the first five minutes are reserved for data collection and model execution. The last 30 minutes before delivery, during which trading is restricted to within control zones, are excluded from the analysis. The ID VWAPs, denoted as $\boldsymbol{X}_{d,h} = (X_{d,h,t_j})_{j=1}^{10}$, at the ten subperiods $\{t_j\}_{j=1}^{10}$ for a specific hourly market at day d and hour h, are the targets to predict. Our goal is to generate path trajectories that form probabilistic multivariate forecasts, incorporating temporal dependencies within the path forecasts.

In addition to the ID VWAPs for 15-minute subperiods, six explanatory variables are available to be used as predictors for making path forecasts, including

- the ID3 index $\texttt{ID3}_{d*,h*}$, which is defined as the VWAP of all transactions that took place in the last 3 hours before the delivery of a given hourly product, and corresponds to the volume weighted average of the VWAPs over the t_j's for $j = 1, \ldots, 12$;
- the day-ahead (DA) price $\texttt{DA}_{d*,h*}$, provided by the EPEX SPOT exchange[3];

[2] The same dataset as in Serafin et al. (2022).

[3] See `https://www.epexspot.com/en/indices`.

- the real values of total load L_{d^*,h^*} and its day-ahead forecasts $\hat{L}_{d^*,h^*}$, provided by the transmission system operator (TSO);
- the real values of wind generation W_{d^*,h^*} and its day-ahead forecasts $\hat{W}_{d^*,h^*}$, provided by the TSO.

The indices d^* and h^* represent the day and the hour, respectively. Multiple selected values (d^*, h^*) are utilized to make path forecasts for the target hourly market at day d and hour h. All data series except the ID3 index are freely available from the ENTSO-E platform[4]. We assume that the actual values of the load and wind generation are available with a delay of less than 3 hours in real-time operation.

Like in Serafin et al. (2022), the out-of-sample test period comprises the last 200 days (from 13.03.2019 to 29.09.2019). The preceding data is used for model training and generating path trajectories using different approaches. All predictors and target variables are normalized to ensure more stable and efficient training, where different standardization schemes are applied and will be introduced separately for each approach in the following.

4.3 Methods

This section introduces three approaches to multivariate probabilistic time series forecasting by generating ID price trajectories across multiple subperiods, including the proposed generative machine learning method using a conditional generative model (CGM), and two state-of-the-art statistical benchmark methods, which were originally proposed in Serafin et al. (2022). A schematic overview of the three approaches is provided in Figure 4.2. Many time series path forecasting approaches used in EPF literature (Ziel and Weron, 2018) follow a two-step framework that separately estimates the marginal forecasts and temporal dependencies, including the two statistical benchmark methods introduced later, while our CGM bypasses the two steps.

[4] See `https://transparency.entsoe.eu/`.

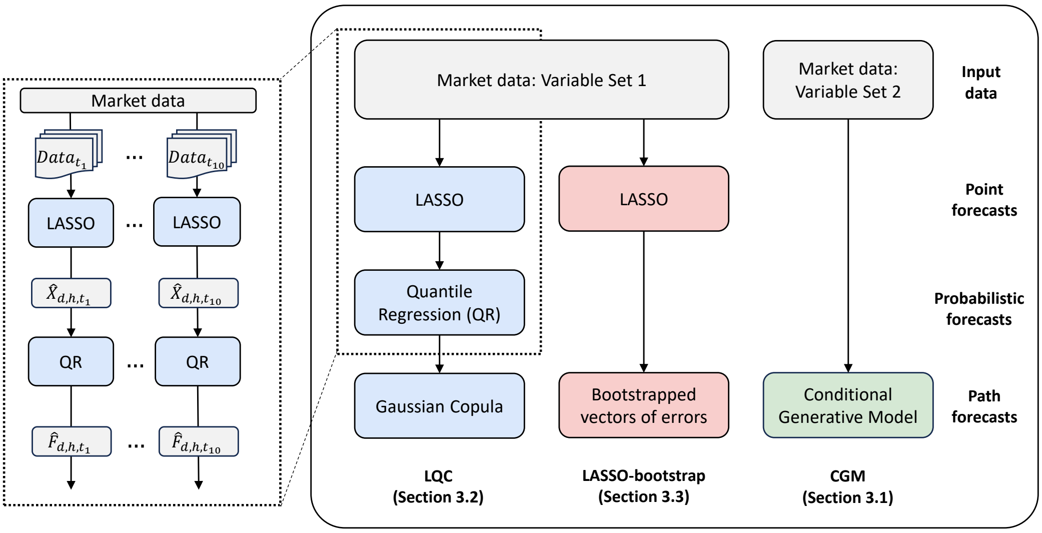


Figure 4.2: Schematic overview of the three approaches to multivariate probabilistic time series forecasting.

4.3.1 The proposed conditional generative model

We propose a novel approach to directly produce multivariate time series forecasts in the form of path trajectories using generative machine learning. This approach builds on the framework developed by Chen et al. (2024) in the context of multivariate post-processing of ensemble weather forecasts. Our conditional generative model (CGM) is a nonparametric approach which does not require parametric assumptions on the marginal distribution or the multivariate dependence structure. This is achieved by utilizing an implicit generative neural network that parametrizes the stochastic process of generating meaningful data from noise, and directly yields simulated time series path trajectories as output. Incorporating information from the available exogenous predictors as inputs enables the CGM to learn complex and nonlinear relationships within the data. The CGM is trained by minimizing the energy score, which will be introduced in Section 4.4.1, as a loss function that measures the discrepancy between the generated path trajectories and the multivariate vector of observed ID prices. For a more detailed description of the mathematical background of generative models and the CGM, we refer to Chen et al. (2024).

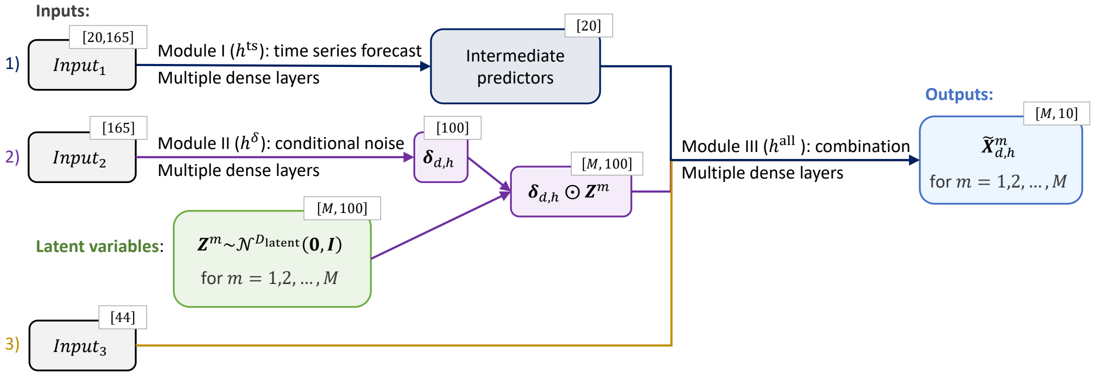

Inputs:
[20,165]
1)
$Input_1$
Module I (h^{ts}): time series forecast
Multiple dense layers
[20]
Intermediate predictors
[165]
2)
$Input_2$
Module II (h^{δ}): conditional noise
Multiple dense layers
[100]
$\boldsymbol{\delta}_{d,h}$
$[M, 100]$
$\boldsymbol{\delta}_{d,h} \odot \boldsymbol{Z}^m$
Latent variables:
$[M, 100]$
$\boldsymbol{Z}^m \sim \mathcal{N}^{D_{\text{latent}}}(\boldsymbol{0}, \boldsymbol{I})$
for $m = 1,2,\dots,M$
[44]
3)
$Input_3$
Module III (h^{all}): combination
Multiple dense layers
Outputs:
$[M, 10]$
$\tilde{\boldsymbol{X}}^m_{d,h}$
for $m = 1,2,\dots,M$

Figure 4.3: Schematic illustration of the conditional generative model (CGM) for generating M path trajectories of the multivariate ID price forecast at hourly market day d and hour h. The dimensions of the tensors at each module are indicated in the small box, with the batch size omitted.

From a conceptual point of view, the CGM comes with the advantage of simplifying the training procedure of common two-step frameworks in statistical path forecasting approaches by eliminating the need for separate training of different components. Compared to the statistical methods introduced above, the CGM is able to flexibly incorporate additional information from exogenous predictors when modeling temporal dependencies in the path forecasts while directly producing probabilistic, multivariate forecasts as output. A specific advantage is that this additional information can also be utilized in modeling the temporal dependence structure, whereas the LQC and LASSO approaches solely rely on observed time series of the target variable. Furthermore, the loss function of the CGM is not necessarily restricted to the energy score. While Pacchiardi et al. (2024) discuss the use of other multivariate proper scoring rules in a similar setting, we explore the use of a custom loss function tailored to the needs of the economic evaluation, see Section 4.4.2.

Model architecture

Figure 4.3 provides a schematic illustration of our CGM. The output of the CGM is a set of 10-dimensional vectors,

$$\tilde{\boldsymbol{X}}^m_{d,h} = (\tilde{X}^m_{d,h,t_1}, \dots, \tilde{X}^m_{d,h,t_{10}}),$$

representing sample paths of the ID prices over the 10 subperiods from the underlying multivariate forecast distribution for a target hourly market at day d and hour h. The model comprises three components to efficiently incorporate relevant exogenous predictors in different segments and to propagate relevant uncertainty information to the generated ID paths by transforming the input noise of the generative model. This design results in three separate input modules, with the corresponding parts represented in different colors in the schematic illustration in Figure 4.3.

The first module of the model, denoted by h^{ts}, aims at generating intermediate predictions as latent information for the subsequent parts. It is designed to mimic deterministic time series forecasting and utilizes a fully-connected feed-forward neural network. The input for this module, denoted by Input_1, consists of 20 predictor variables, including the 6 exogenous variables introduced in Section 4.2, the ID prices of 12 subperiods (from t_1 to t_{12}, spanning the 3-hour period before delivery) along with their standard deviations, and the VWAP of the last subperiod preceding t_1. This VWAP, which is denoted by X_{d^*,h^*,t_0}, corresponds to the period from 3 hours 15 minutes to 3 hours before the delivery of a target hourly market at day d^* and hour h^*. For all 20 input variables, we use a window of historical data ranging from 1 week to 4 hours before the delivery time. For the ID price-related predictors, the data corresponds to historical hourly markets. The full list of inputs for this first module thus is

$$\mathit{Input}_1 = \Big\{\texttt{ID3}_{d,h-i}, \texttt{DA}_{d,h-i}, L_{d,h-i}, \hat{L}_{d,h-i}, W_{d,h-i}, \hat{W}_{d,h-i}, \{X_{d,h-i,t_j}\}_{j=0}^{12}, \sigma\big(\{X_{d,h-i,t_j}\}_{j=1}^{12}\big)\Big\}_{i=4}^{168}.$$

The second module, denoted by h^{δ}, is the core of the generative model based on which it learns to produce meaningful noise estimates conditional on the available input data. We generate latent noise variables by sampling from a standard multivariate Gaussian distribution, which is a common choice in generative models. The dimensionality of the latent variables D_{latent} is a hyperparameter of the model that controls the complexity of randomness for each sample and needs to be determined through hyperparameter tuning. We use $\boldsymbol{Z}^m$ to denote a single sample of the noise vector from which we eventually obtain the corresponding output sample $\tilde{\boldsymbol{X}}_{d,h}^m$ as the final output of our generative model. By repeatedly generating samples from the noise distribution and propagating them through the generative model, we obtain a multivariate probabilistic forecast in the form of samples of output vectors. The number of noise samples we draw during training (and inference) determines the number of output path trajectories, and thus enables the generation of arbitrarily many sample trajectories.

The scale of the generated latent noise $\boldsymbol{Z}^m$ is adjusted by incorporating uncertainty information from the second part of the available inputs, denoted by Input_2. We refer to the output of this scale adjustment as conditional noise. Input_2 utilizes the standard deviation predictor from Input_1, i.e., $\sigma\big(\{X_{d,h-i,t_j}\}_{j=1}^{12}\big)$ for $i = 4, \ldots, 168$. We employ a fully-connected feed-forward neural network to learn the adjusted scales $\boldsymbol{\delta}_{d,h}$ for all latent variables, and obtain the conditional noise via

$$h^{\delta}(\mathit{Input}_2) \odot \boldsymbol{Z}_m,\ \boldsymbol{Z}_m \sim \mathcal{N}^{D_{\text{latent}}}(\boldsymbol{0}, \boldsymbol{I}); \quad \text{with } \mathit{Input}_2 = \Big\{\sigma\big(\{X_{d,h-i,t_j}\}_{j=1}^{12}\big)\Big\}_{i=4}^{168}.$$

The third and final module, denoted by h^{all}, further incorporates more recent historical information available within the 4 hours before the delivery, and integrates the intermediate predictions and conditional noise from the previous two modules to generate output sample trajectories. The inputs for this module, denoted by Input_3, contain the 20 variables from Input_1, but only for the specific values at (d^*, h^*) corresponding to 4 hours before the delivery of the target hourly market. Additionally, it incorporates four variables, the day-ahead price and the last VWAP, as well as the day-ahead forecasts of wind generation and load, available from 3 hours before up to the delivery time. While the complete observed ID price path of previous hourly markets within 3 hours before the target delivery is not available at the time of forecasting, partial paths are accessible and can provide valuable insights into the latest real-time ID prices. Therefore, we incorporate these available ID prices, specifically $\{X_{d,h-2,t_j}\}_{j=9}^{12}$ and $\{X_{d,h-3,t_j}\}_{j=5}^{12}$. We also incorporate time dummy variables to convey the time information of the target hourly market, including both sine and cosine transforms of "the day of the year” d, and the hour of the day h. The weekday information, ranging from 1 to 7, is treated as a separate input component. This information is integrated with the other inputs after being processed through an embedding layer that converts categorical integer values into 2-dimensional vectors, following related work in probabilistic weather forecasting Rasp and Lerch (2018), which in turn is based on widely used embedding techniques in natural language processing. The complete list of Input_3 thus is

$$\mathit{Input}_3 = \Big\{\mathit{Input}_1(i=4), \{\mathtt{DA}_{d,h-i}, \hat{L}_{d,h-i}, \hat{W}_{d,h-i}, X_{d,h-i,t_0}\}_{i=0}^{3}, \\ \{X_{d,h-2,t_j}\}_{j=9}^{12}, \{X_{d,h-3,t_j}\}_{j=5}^{12}, \text{time indicators}\Big\}.$$

As final output of the CGM, we thus obtain samples of path trajectories via

$$\tilde{\boldsymbol{X}}_{d,h}^{m} = h^{\mathrm{all}}\big(h^{\mathrm{ts}}(\mathit{Input}_1), h^{\delta}(\mathit{Input}_2) \odot \boldsymbol{z}_m, \mathit{Input}_3\big),\ m = 1, 2, \ldots$$

by repeatedly generating samples from the latent noise distribution.

Implementation details

The CGM is trained by minimizing the empirical energy score, see Section 4.4.1. To reduce the randomness inherent in neural network training, we generate an ensemble of 10 CGMs by training models with identical hyperparameters on the same data, but with different random seeds. This strategy has proven effective in improving robustness and overall forecast quality, and is competitive with other ensemble generation mechanisms for neural network-based forecasting models (Schulz and Lerch, 2022a). Each ensemble run generates 1 000 output samples, and the combined output of all ensemble runs yields a total of 10 000 forecast path trajectories as the final outcome of the model. For a detailed investigation of strategies to generate ensembles of CGMs, see Chen et al. (2024).

The hyperparameters determining the structure of the CGM, including the number of layers, nodes, activation functions in each layer, and the number of latent variables, need to be determined through hyperparameter tuning. The hyperparameters were determined based on a combination of exploratory experiments and an additional grid search. The three model components have different hyperparameter configurations, and the overall framework consists of 100 latent variables for the noise component, 10 dense layers, with the ELU activation function (Clevert, 2015) used for most layers. For more a complete list of all hyperparameter choices of the CGM, we refer to the Python code accompanying this work. The model is trained using stochastic gradient descent optimization with the

Adam optimizer (Kingma, 2014) with a learning rate of 1×10^{-4}, a batch size of 1 024, and an early stopping criterion with a patience of 10 epochs to avoid overfitting.

The CGM is trained over a fixed period of 630 days (22.06.2017–13.03.2019), with 20 percent of the data randomly selected as the validation set. The training period begins one week after the first date in the original dataset to ensure historical input variables are available. All input variables are normalized by subtracting the mean and dividing by the standard deviation of the data over the training period. Preliminary experiments indicated no improvements, and sometimes worse performance, when training the CGM with a sliding window, likely due to higher variability in the training data. Therefore, in contrast to the statistical benchmark methods, we do not employ sliding window training for the CGM, even though the comparatively low computational cost would have made this technically possible, as the training process only takes a few minutes on multiple CPUs. That said, rolling window training may still offer advantages for different datasets and contexts, and thus may be worth considering to enable the model to better adapt to structural changes in the data over time.

4.3.2 LQC benchmark

The LQC approach (Serafin et al., 2022) comprises three components: a deterministic point prediction model, a transformation of those point predictions to probabilistic forecasts, and a restoration of temporal dependencies. The specific methods applied in those components lend the LQC approach its name: The point predictions are obtained via (L)ASSO-estimated (auto)regression (also known as the LEAR model; Lago et al., 2021), and are converted to probabilistic predictions (as in the quantile regression averaging approach proposed by Nowotarski and Weron, 2015) via (Q)uantile regression. Finally, a Gaussian (C)opula is employed for modeling temporal dependencies (as suggested in Pinson et al., 2009).

In the first step, point predictions are made using the LEAR model, utilizing 102 inputs (or regressors) derived from the six explanatory variables introduced in Section 4.2:

- $\{\mathtt{ID3}_{d,h-i}\}_{i=4}^{24}$, i.e., the most recent 21 past ID3 index values available at the time of prediction;

- $\{\mathtt{DA}_{d,h-i}\}_{i=0}^{24}$, i.e., 25 day-ahead prices available within one day before the delivery;
- $\{\hat{W}_{d,h-i}, \hat{L}_{d,h-i}\}_{i=0}^{24}$, i.e., 25 hourly values of day-ahead wind generation and load forecasts available within one day before the delivery;
- $\{W_{d,h-4}, W_{d,h-24}, L_{d,h-4}, L_{d,h-24}\}$, i.e., the actual wind power production and observed load for the last observed hour (4 hours before delivery) and 24 hours ago;
- $\{X_{d,h,t_0}\}$, i.e., the last VWAP spanning the transaction period from 3 hours 15 minutes to 3 hours before the delivery;

where day d, hour h represent the delivery time of the target hourly market. Separate models are constructed for each of the 10 subperiods $j = 1, 2, \dots, 10$, and the least absolute shrinkage and selection operator (LASSO; Tibshirani, 1996) is used to remove redundant features.

Following Tschora et al. (2022) and Ziel and Weron (2018), we transform the inputs by applying the *area hyperbolic sine*[5]. As suggested by Uniejewski et al. (2018), each input series is first independently normalized by subtracting the in-sample median and dividing by the in-sample median absolute deviation, adjusted by the 75th percentile of the standard normal distribution. Once the point prediction $\hat{X}_{d,h,t_j}$ for each subperiod j is generated, the transformation and normalization are inverted.

Based on the point forecast processed separately for different subperiods, we use quantile regression (QR; Koenker, 2005) in the next step to compute empirical forecasts in the form of 99 percentiles of the predictive distribution $\hat{F}_{d,h,t_j}$ at each margin, i.e., for each subperiod t_j before the delivery at day d and hour h. The LASSO and QR steps result in probabilistic forecasts of the marginal distribution and thus constitute the first part of the two-step framework for multivariate forecasting. The 99 percentiles are linearly interpolated, with linear extrapolation applied to the minimum and maximum prices for the extreme values, to allow for drawing arbitrarily many quantiles in the subsequent step.

In the final step, multivariate path trajectories of ID prices across multiple subperiods are generated based on the predicted quantiles, with temporal dependencies between

[5] The area (inverse) hyperbolic sine can be computed by $\operatorname{arsinh}(x) = \ln\left(x + \sqrt{x^2+1}\right)$.

subperiods modeled using a Gaussian copula, as presented in Serafin et al. (2022). The probabilistic path forecasts consisting of M trajectories

$$\left\{\tilde{\boldsymbol{X}}_{d,h}^{m} = (\tilde{X}_{d,h,t_1}^{m}, \dots, \tilde{X}_{d,h,t_{10}}^{m})\right\}_{m=1}^{M}$$

of the target ID prices

$$\boldsymbol{X}_{d,h} = (X_{d,h,t_1}, \dots, X_{d,h,t_{10}})$$

are derived from M random samples $\{\boldsymbol{Z}_{d,h}^{m} = (Z_{d,h,t_1}^{m}, \dots, Z_{d,h,t_{10}}^{m})\}_{m=1}^{M} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{d,h})$ from a multivariate Gaussian distribution,

$$\tilde{X}_{d,h,t_j}^{m} = \hat{F}_{d,h,t_j}^{-1}\left(\Phi\left(Z_{d,h,t_j}^{m}\right)\right), \quad \text{for } j = 1, \dots, 10,$$

where $\hat{F}_{d,h,t_j}^{-1}$ denotes the inverse transformation of the marginal forecast CDF $\hat{F}_{d,h,t_j}$, and Φ represents the standard Gaussian CDF. The covariance matrix $\boldsymbol{\Sigma}_{d,h}$ is estimated based on the transformed historical ID prices from a preceding calibration window $\mathcal{C} = [d - 120, d)$,

$$\boldsymbol{\Sigma}_{d,h} = \operatorname{cov}\left(\{\hat{\boldsymbol{Z}}_{d^*,h} = (\hat{Z}_{d^*,h,t_1}, \dots, \hat{Z}_{d^*,h,t_{10}})\}_{d^* \in \mathcal{C}}\right), \quad \text{with } \hat{Z}_{d^*,h,t_j} = \Phi^{-1}(\hat{F}_{d^*,h,t_j}(X_{d^*,h,t_j})).$$

For all three steps of the LQC approach, a rolling window scheme is employed. Each day, the calibration windows are moved forward by one day to produce the next day's forecasts, with different window sizes used for each step. We first use LASSO-estimated regression fitted to data from a 396-day calibration window (sliding window initially starting from 16.06.2017) to compute point predictions, then apply quantile regression with parameters estimated using a 120-day calibration window (sliding window initially starting from 16.07.2018). Once computed, the predictive distributions are converted into path forecasts using a Gaussian copula fitted over a 120-day calibration window (sliding window initially starting from 13.11.2018).

4.3.3 LASSO bootstrap benchmark

The LASSO bootstrap approach uses the same point predictions from the LEAR model as the LQC approach. These point predictions serve as the basis for obtaining probabilistic price path forecasts without the need to compute predictive distributions, utilizing a bootstrapping method. Thereby, vectors of historical point forecast errors are sampled to incorporate temporal dependencies based on past obseervations.

To obtain a multivariate path trajectory $\tilde{\boldsymbol{X}}^m_{d,h}$ of ID price forecast for the delivery at day d and hour h, we first compute vectors of past point forecast errors from a preceding calibration window, i.e.,

$$\boldsymbol{\varepsilon}_{d^*,h} = \hat{\boldsymbol{X}}_{d^*,h} - \boldsymbol{X}_{d^*,h}, \quad \text{with } d^* \in [d-240, d),$$

and proceed by adding bootstrapped error vectors to the point predictions for the target path,

$$\tilde{\boldsymbol{X}}^m_{d,h} = \hat{\boldsymbol{X}}_{d,h} + \boldsymbol{\varepsilon}^m_{d^*,h}, \quad \text{with } \boldsymbol{\varepsilon}^m_{d^*,h} \in \{\boldsymbol{\varepsilon}_{d^*,h}\}_{d^* \in [d-240,d)}, \quad \text{for } m = 1, \dots, M,$$

where $\hat{\boldsymbol{X}}_{d,h} = (\hat{X}_{d,h,t_1}, \dots, \hat{X}_{d,h,t_{10}})$ are the point predictions for all subperiods.

The LASSO bootstrap approach also employs a rolling window scheme. The first step is the same as the LASSO step in the LQC approach, where we fit a LASSO-estimated regression model using data from a 396-day calibration window. In the next step, randomly sampled historical error vectors from a 240-day calibration window (sliding window initially starting from 16.07.2018) are added to the point predictions to obtain multivariate probabilistic forecasts of ID prices.

4.4 Statistical and economic evaluation methods

We here introduce various evaluation metrics that will be used in Section 4.5 to compare the CGM against the two statistical benchmarks. We present widely used statistical metrics for probabilistic forecasts that account for prediction uncertainty, and propose economic evaluation methods based on trading strategies on top. These are motivated from a practical perspective where a manager has to make a decision, and different evaluation metrics may point to different suggested actions (Kolassa, 2020). At the same time, the optimal choice will be affected by the decision maker's preferences, e.g., regarding profit maximization or risk reduction. Statistical evaluation alone thus does not provide the necessary information, as there is no clear and obvious relationship between scoring metrics and the expected outcome of economic decisions. This makes it unclear whether higher accuracy in terms of statistical evaluation metrics translates into better economic results in practice (Maciejowska et al., 2023; Yardley and Petropoulos, 2021). To address this, we consider a range of trading strategies, which will be introduced in Section 4.4.2 and utilize the generated ID price path forecasts and evaluate different methods in a case study involving a fixed-volume scenario.

4.4.1 Statistical evaluation

Since probabilistic forecasts capture prediction uncertainty, respective statistical evaluation metrics should also take uncertainty information into account. The widely accepted standard tools for probabilistic forecast evaluation are proper scoring rules (Gneiting and Raftery, 2007), which simultaneously assess calibration and sharpness of predictive distributions. In a nutshell, a scoring rule $S(F, x)$ assigns a numerical score to a pair of a forecast distribution F and a realizing observation x. It is called proper, if the true distribution of the observation achieves the best (i.e., lowest) possible score in expectation, i.e., $\mathbb{E}_{X\sim G}S(G, X) \leq \mathbb{E}_{X\sim G}S(F, X)$ for all pairs of forecast distributions F, G from a suitably chosen class of probability distributions. For details, we refer to Gneiting and Raftery (2007), available software implementations (e.g., Jordan et al., 2019), and the wide variety of research in statistics and application disciplines, including, e.g., Lauret et al. (2019) with a focus on energy forecasting.

The continuous ranked probability score (CRPS), proposed by Matheson and Winkler (1976), is a proper scoring rule widely used for evaluating univariate probabilistic forecasts. Given marginal forecast CDF $\hat{F}_{d,h,t_j}$ and the real price X_{d,h,t_j} at subperiod t_j for hourly market day d and hour h, the CRPS is defined as

$$\text{CRPS}_{d,h,t_j}(\hat{F}_{d,h,t_j}, X_{d,h,t_j}) = \int_{-\infty}^{\infty} \left(\hat{F}_{d,h,t_j}(z) - \mathbb{I}\{z \geq X_{d,h,t_j}\}\right)^2 dz,$$

where $\mathbb{I}$ denotes the indicator function. Based on samples $\{\tilde{X}^m_{d,h,t_j}\}^M_{m=1}$ from the predictive distribution, it can be formulated as

$$\mathrm{CRPS}_{d,h,t_j} = \frac{1}{M}\sum_{m=1}^{M} |\tilde{X}^m_{d,h,t_j} - X_{d,h,t_j}| - \frac{1}{2M^2}\sum_{m=1}^{M}\sum_{n=1}^{M} |\tilde{X}^m_{d,h,t_j} - \tilde{X}^n_{d,h,t_j}|.$$

The CRPS is negatively oriented and equals zero for a forecast that perfectly matches the observed distribution. In the special case where the forecast is a deterministic point prediction, the CRPS reduces to the mean absolute error.

The direct generalization of CRPS to multivariate forecasts is the energy score, which will be introduced below. In addition to the energy score, several proper scoring rules have been proposed for evaluating multivariate probabilistic forecasts. However, all of them come with certain shortcomings in terms of sensitivity to certain types of misspecifications of the multivariate forecast distribution (Scheuerer and Hamill, 2015; Alexander et al., 2022). A comprehensive understanding of contributions to various types of misspecifications to the behavior of multivariate proper scoring rules remains an open question and subject of current research, see the discussion in Chen et al. (2024) and references therein. We here use three popular multivariate proper scoring rules: the energy score (ES), the Dawid-Sebastiani score (DSS) and the variogram score (VS).

Energy score

The energy score (ES; Gneiting and Raftery, 2007) is given by

$$\mathrm{ES}_{d,h} = \frac{1}{M}\sum_{m=1}^{M} \left\|\tilde{\boldsymbol{X}}^m_{d,h} - \boldsymbol{X}_{d,h}\right\|_2 - \frac{1}{M(M-1)}\sum_{m=1}^{M-1}\sum_{n=m+1}^{M} \left\|\tilde{\boldsymbol{X}}^m_{d,h} - \tilde{\boldsymbol{X}}^n_{d,h}\right\|_2, \tag{4.1}$$

where $\tilde{\boldsymbol{X}}^m_{d,h} = (\tilde{X}^m_{d,h,t_1}, \ldots, \tilde{X}^m_{d,h,t_{10}})$ is the m-th multivariate realization of ID price path forecast for day d and hour h, $\boldsymbol{X}_{d,h}$ is the corresponding observed ID price path, and M is the number of generated path trajectories. A number of studies have noted that the ES lacks sensitivity to misspecifications of the dependence structure (e.g., Pinson and Girard, 2012; Alexander et al., 2022).

Dawid-Sebastiani score

The Dawid-Sebastiani score (DSS; Dawid and Sebastiani, 1999) is a multivariate proper scoring rule based on the mean vector and covariance matrix of the predictive distribution

$$\mathrm{DSS}_{d,h} = \log\left(\det\left(\mathbf{S}_{d,h}\right)\right) + \mathbf{K}^T \mathbf{S}_{d,h}^{-1} \mathbf{K}, \tag{4.2}$$

where in our case $\mathbf{K}_{d,h} = (K_{d,h,t_1}, \ldots, K_{d,h,t_{10}})$ is a vector of 10 differences,

$$K_{d,h,t_j} = X_{d,h,t_j} - \frac{1}{M} \sum_{m=1}^{M} \tilde{X}^m_{d,h,t_j}$$

and $\mathbf{S}_{d,h}$ is the covariance matrix estimated from the simulated scenarios. The DSS corresponds to the logarithmic score for multivariate Gaussian predictive distributions and is a proper scoring rule for a broad class of probability distributions.

In addition to shortcomings that have been noted in cases where forecast accuracy is moderate (Wilks, 2020), a major limitation of this score is the potential numerical issue when inverting the covariance matrix if the sample size is small relative to the number of ensemble members (Scheuerer and Hamill, 2015).

Variogram score

The variogram score (VS; Scheuerer and Hamill, 2015) has been proposed as an alternative multivariate proper scoring rule and is given by

$$\mathrm{VS}_{d,h} = \sum_{i,j=1}^{10} w_{i,j} \left(\left| X_{d,h,t_i} - X_{d,h,t_j} \right|^p - \frac{1}{M} \sum_{m=1}^{M} \left| \tilde{X}^m_{d,h,t_i} - \tilde{X}^m_{d,h,t_j} \right|^p \right)^2 , \tag{4.3}$$

where p is the order of the VS, and $w_{i,j}$ is an optional weight parameter. We here consider the unweighted version with $w_{i,j} = \frac{1}{100}$. It has been argued that the VS tends to be more discriminative than the ES and DSS when the correlation structure of ensemble forecasts is misspecified (Scheuerer and Hamill, 2015). The order p needs to be chosen by the user, with Alexander et al. (2022) noting that the VS with $p = 0.5$ has a superior discriminative ability when dealing with relatively accurate forecasts, whereas $p = 1$ should be used in cases with moderate prediction accuracy.

4.4.2 Economic evaluation

To evaluate the generated path forecasts from an economic perspective, we consider a range of trading strategies for the fixed-volume scenario introduced by Serafin et al. (2022) in the continuous-time ID market. The fixed-volume scenario assumes that an energy producer owning intermittent RES sells the surplus of 1 MWh of electricity in each hour of the day. A similar setup has been considered by Kath and Ziel (2018) and Janczura and Puć (2023), among others. We make the standard assumption that the impact of our trades on the ID prices is negligible and ignore the transaction costs. The decision problem can then be treated as finding the optimal time to enter the market for selling the fixed amount electricity for each individual hourly delivery period.

In the following, we present two classes of strategies that rely on multivariate path forecasts for the fixed-volume scenario, where one is based directly on the multivariate trajectories and the other utilizes prediction bands derived from the path forecasts. In addition, we describe the naive benchmark strategies and introduce a crystal ball (or orcale) benchmark to evaluate the realized trading potential when using multivariate price forecasts of the benchmark models and the proposed CGM.

Trading strategies based on probabilistic forecasts

Majority vote strategy

Given a single path forecast in the form of a trajectory of ID price across multiple subperiods, the most intuitive and simple approach to determining the optimal time for selling the fixed amount of electricity is to simply use the subperiod when the predicted path trajectory reaches its maximum price. Based on the collection of M generated path trajectories which are obtained as outputs of the different forecasting methods, we use a majority-vote strategy to identify the most frequent subperiod with the maximum price. The optimal time for entering the market using the majority-vote strategy for M path trajectories $\{\tilde{\boldsymbol{X}}^m_{d,h} = (\tilde{X}^m_{d,h,t_1}, \dots, \tilde{X}^m_{d,h,t_{10}})\}^M_{m=1}$ is then given by

$$J_{d,h} = \text{mode}\left(\left\{\underset{j\in\{1,\dots,10\}}{\arg\max}\ \tilde{X}^m_{d,h,t_j}\right\}^M_{m=1}\right), \tag{4.4}$$

where $J_{d,h}$ is the index of the optimal subperiod for selling the fixed amount of electricity.

Prediction band-based strategy

In addition to selecting the optimal time for entering the market directly from the simulated trajectories of future price paths, we further explore strategies based on prediction bands derived from the collection of path forecasts, which were first proposed in Serafin et al. (2022).

Prediction bands, unlike a set of prediction intervals, account for the temporal dependence in the evolution of predicted prices over time. Each prediction band (upper or lower) is defined by the simultaneous coverage probability (SCP), which represents the probability that the entire price trajectory lies below (→ upper) or above (→ lower) the band.

More formally, the SCP for the upper prediction band $\boldsymbol{B}^U_{d,h,t_j} \in \mathbb{R}^{10}$ is given by

$$\mathbb{P}\left(X_{d,h,t_j} \leq \boldsymbol{B}^U_{d,h,t_j}, \forall_j\right) = \text{SCP},$$

and for the lower $\boldsymbol{B}^L_{d,h,t_j}$ by

$$\mathbb{P}\left(\boldsymbol{B}^L_{d,h,t_j} \leq X_{d,h,t_j}, \forall_j\right) = \text{SCP}.$$

The algorithm we use to construct prediction bands is based on Staszewska (2007). To satisfy the simultaneous coverage property, which requires that predicted price paths remain within the prediction band at all time points, the procedure involves filtering out simulated trajectories that contain extreme values, specifically maximum values for the upper band and minimum values for the lower band, until only a fraction of the SCP % of the trajectories remain. The prediction band is then formed by selecting the maximum (or minimum) values of the remaining paths at each subsequent time point.

In our fixed-volume scenario for the economic evaluation of path forecasts, we focus on making decisions about when to sell the fixed amount of electricity. The upper prediction band provides information on the highest probable price under a given SCP, while the lower prediction band reflects the lowest probable price. For a risk-seeking decision, we may select the subperiod that achieves the highest value in the upper prediction band as the optimal time for selling. Conversely, for a risk-averse decision, we may select the subperiod time point with the highest value in the lower prediction band, thereby maximizing the lowest expected price. We explore both choices and discuss their implications in more detail later.

Benchmark trading strategies

Naive benchmarks

We use three naive benchmark strategies that do not rely on any generated forecasts as references and simply trade the available fixed amount of electricity at pre-defined points of time. In the $\text{Naive}_{\text{first}}$ benchmark, the RES generator submits a market order in period t_0, just before the 3 hours to delivery trading window starts. The $\text{Naive}_{\text{last}}$ benchmark proceeds by placing market orders in the last period t_{10} before trading get limited to only selected zones. Finally, $\text{Naive}_{\text{avg}}$ benchmark assumes that the total volume is divided into 10 equally sized trades across the 10 periods $t_1, \ldots, t_{10}$. Note that the $\text{Naive}_{\text{avg}}$ benchmark thus deviates from the previous assumption of a fixed-volume selling scenario where the total amount of available electricity in a single subperiod, which might somewhat limit the fairness of the comparisons.

Crystal ball benchmarks and the realized trading potential

The employed fixed-volume selling scenario has a maximum and minimum profit that can theoretically be achieved if the realizing observations of prices were known in advance. Although this would obviously be impossible in any practical application, comparisons against the theoretical optimum might be of interest to assess the capabilities of the proposed forecasting models. To that end, we construct a hypothetical crystal ball (CB) trading strategy, where we assume the future observations to be known and sell the available electricity during the subperiod with the highest (for the maximum profit CB benchmark) or the lowest (for the minimum profit CB benchmark) realized price. We denote the profits from these two benchmarks by CB_{max} and CB_{min}, respectively. We can then define the realized trading potential (RTP) of a given combination of a forecasting model A and a trading strategy as

$$\mathrm{RTP}_A = \frac{\mathrm{Profit}_A - \mathrm{CB}_{\mathrm{min}}}{\mathrm{CB}_{\mathrm{max}} - \mathrm{CB}_{\mathrm{min}}} \times 100, \tag{4.5}$$

where Profit_A is the sum of the trading strategy's profits over the entire 200-day test period when using the predictions of model A. The RTP, which can take values from 0 to 100, can be interpreted as the fraction of the maximum profit that can be achieved (times 100).

Tailoring the conditional generative model to optimize trading profits

As discussed in Section 4.3.1, the CGM can be trained with alternative loss functions. Here, we investigate an adaptation to potentially improve the economic aspects of the CGM model predictions, by combining the previously introduced economic evaluation as a custom loss function with energy score. Specifically, we employ the majority-vote strategy for the fixed-volume trading scenario and integrate an additional loss component that measures the difference between the optimum index derived from the generated path trajectories and the observed optimal subperiod index obtained from the realizing ID prices.

Following Eq. (4.4), we derive the index $\tilde{J}_{d,h}$ of the optimal subperiod for selling based on the path trajectories $\{\tilde{\boldsymbol{X}}^m_{d,h}\}_{m=1}^M$ generated by the CGM by applying the majority vote strategy. Let $J^{\mathrm{obs}}_{d,h}$ denote the index of the subperiod with the highest observed ID price,

$$J^{\mathrm{obs}}_{d,h} = \underset{j\in\{1,\ldots,10\}}{\arg\max}\; X_{d,h,t_j}.$$

The custom loss function for the CGM is then defined as

$$\ell_{d,h} = (1-\omega)\cdot\frac{1}{2}\cdot \mathrm{ES}_{d,h} + \omega\cdot\left(\frac{1}{100}\cdot\left(\tilde{J}_{d,h} - J^{\mathrm{obs}}_{d,h}\right)^2\right), \tag{4.6}$$

where ω controls the weight of each component. The ES component is divided by 2 to ensure a comparable magnitude of typical values encountered during the model optimization.

In the next section, we present results for $\omega = 0.5$, as preliminary experiments suggest that an equally weighted loss results in a better trade-off between statistical and economic performance. The CGM approach trained solely on the energy score is denoted as "CGM (ES loss)", and the one trained with the custom loss that integrates the economic evaluation measure is denoted as "CGM (custom loss)". The performances of both CGM variants are investigated.

4.5 Results

In this section, we present the results of both the statistical and economic evaluation of the generated path forecasts of the CGM and the statistical benchmark models.

4.5.1 Statistical measures

We first compare the univariate performance of the different methods at each margin in terms of the mean absolute error of the median forecast and the CRPS in Figure 4.4. Both evaluation metrics increase as the time subperiod approaches the target delivery time, indicating that it is harder to make accurate forecasts closer to delivery. Figure 4.4(a) shows the absolute error, averaged over all hourly markets in the 200-day test dataset for each subperiod. The LASSO bootstrap benchmark consistently outperforms

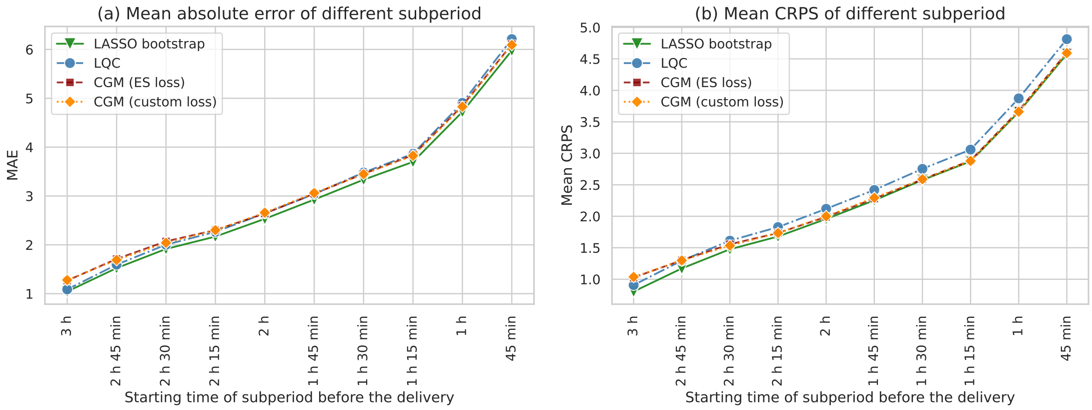


Figure 4.4: Mean absolute error (left) and CRPS (right) of different forecasting methods for each subperiod (margin) of the ID price path.

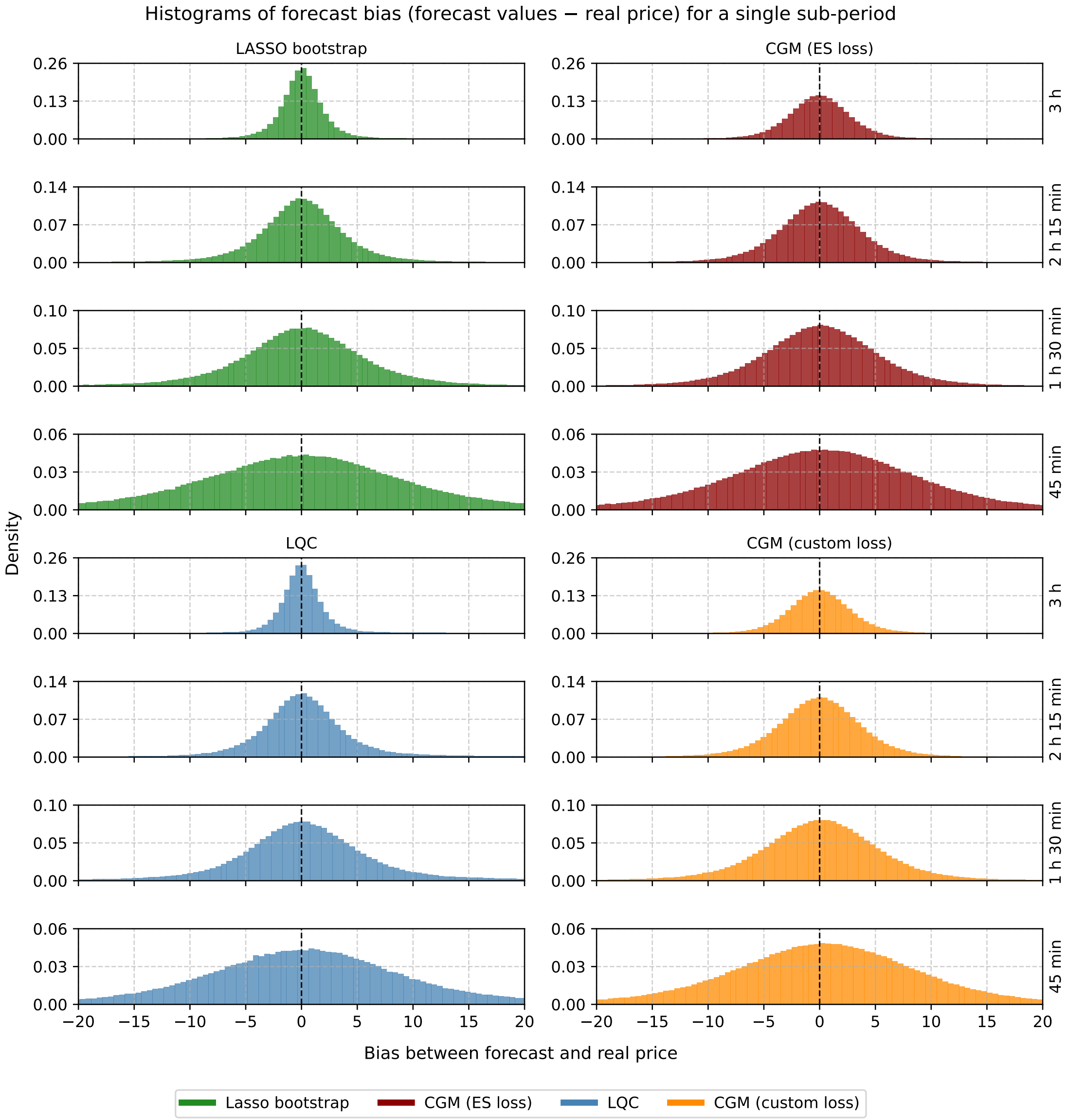


Figure 4.5: Distribution of biases (computed as forecast sample minus observed ID price) for each subperiod of the ID price path, over all hourly markets in the 200-day test period.

Table 4.1: Mean multivariate evaluation scores of path forecasts generated by the LASSO bootstrap, LQC, and two CGM variants trained on different loss functions, separately for the on- and off-peak hours. The best scores in each column are highlighted in bold.

	ES		DSS		VS-1		VS-0.5	
	on-peak	off-peak	on-peak	off-peak	on-peak	off-peak	on-peak	off-peak
LASSO bootstrap	**10.18**	**8.04**	38.43	30.63	34.06	**29.98**	**0.70**	**0.58**
LQC	10.86	8.86	45.10	39.05	41.23	34.96	0.83	0.73
CGM (ES loss)	10.3	8.2	34.76	29.82	**33.46**	30.43	**0.70**	0.64
CGM (custom loss)	10.37	8.2	**34.41**	**29.6**	33.53	30.33	**0.70**	0.63

other approaches across all subperiods closely followed by the LQC benchmark and two CGM variants. In terms of CRPS shown in Figure 4.4(b), the LASSO bootstrap benchmark also performs best among all methods, especially for subperiods further from the delivery time. For subperiods closer to the target delivery, the two CGM variants show comparable performance to the LASSO bootstrap benchmark. The two considered CGM variants consistently show almost no difference in performance over both metrics.

To further investigate the distribution of the forecast errors of these different approaches, we present histograms of the differences between each marginal forecast sample and the observed ID price in Figure 4.5. These histograms are shown for four selected time subperiods. In the first subperiod, which begins three hours before the target delivery, we observe that the bias distributions of the forecasts from two CGM variants are notably wider compared to the two statistical benchmarks. Conversely, in the last subperiod, starting 45 minutes before delivery, the two CGM variants produce slightly narrower bias distributions than the benchmarks. The increasing variance of bias distributions from the first to the last subperiod further underscores the increased complexity of estimating ID prices as the delivery time approaches.

Next, we compare the multivariate performance of the different methods. Table 4.1 shows results for the last 200 days of the out-of-sample period in terms of the mean energy score, the Dawid-Sebastiani score, and two variants of the variogram score (VS-1, VS-0.5). The evaluation is divided into on-peak hours (8:00-19:00) and off-peak hours (the remaining 12 hours of the day).

In general, different evaluation metrics suggest different best-performing methods, and no single method consistently outperforms the others across all metrics. However, the LQC benchmark exhibits the weakest performance throughout, in particular for the DSS and VS[6]. The other three methods are generally comparable, aligning with the univariate evaluation results. The LASSO bootstrap benchmark performs best in terms of the ES, but is outperformed by the two CGM variants when evaluated using VS-1, VS-0.5 during on-peak hours, and in terms the DSS. These results suggest that the CGM variants are better in capturing the temporal dependence structure of the price paths, particularly during on-peak hours, which are more critical periods for trading markets, whereas the LASSO bootstrap benchmark exhibits a smaller bias. That said, the score differences between the CGM models and the LASSO bootstrap benchmark tend to be mostly minor.

Figure 4.6 shows the mean value of the different scoring rules for each hourly market. Across all four panels, two primary peaks are evident, roughly corresponding to 11:00–15:00 and 20:00–23:00, which likely align with high electricity demand during midday and evening hours. During these peak periods, the CGM variants outperform the LASSO bootstrap benchmark in terms of DSS and both VS metrics, while the LASSO bootstrap method achieves better results in terms of the ES. During the remaining periods with lower score values, the LASSO bootstrap method generally performs better than all other methods. The LQC benchmark consistently performs worst over almost all hourly markets, with few exceptions. During the peak periods, which are typically more critical for real-world trading decisions, the CGM variants demonstrate somewhat performance during, particularly in reconstructing temporal dependencies in the path forecasts. For example, the LASSO bootstrap shows notable outliers with the by far worst DSS values of all methods during the few periods with the highest overall forecast errors.

4.5.2 Economic evaluation based on trading profits

We follow the fixed-volume selling scenario introduced in Section 4.4.2, which aims at maximizing the profit from selling 1 MWh in each hourly load period during the 200-day test period, based on the majority vote and prediction band strategies. The trading

[6] The relative performance of the LQC benchmark differs from the results in Serafin et al. (2022) due to a mistake in their code for preprocessing the data.

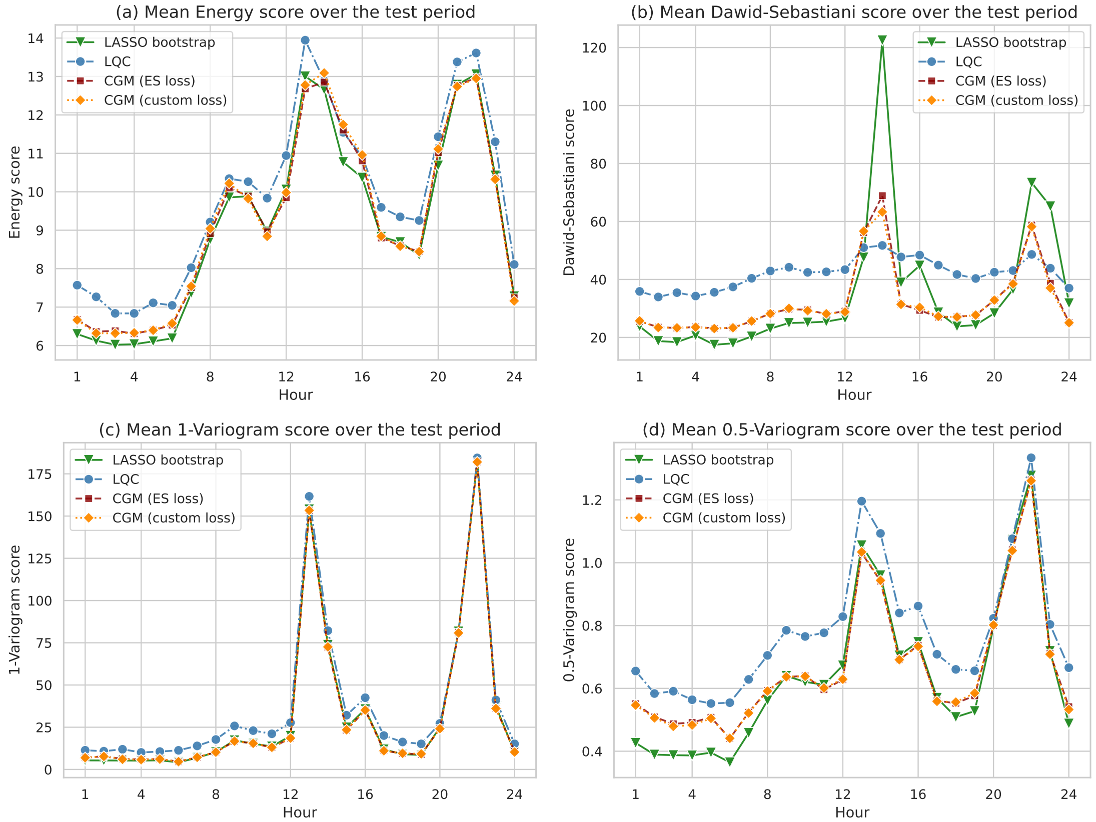


Figure 4.6: Line plots of multivariate evaluation scores averaged over 200 days in the test period for each hourly market.

profits based on the observed ID price at the selected time are computed as the sum of the profits for all hourly markets over the 200-day test period.

Figure 4.7(a) compares the total profit gains of the different path forecasting methods using the majority-vote strategy. For reference, the results from three naive benchmarks presented in Section 4.4.2 are included as baselines. Among the four forecasting methods, the CGM trained with the ES performs best, closely followed by the CGM trained with the custom loss function. This is somewhat unexpected as the custom loss function was explicitly designed to incorporate economic evaluation during the model training. The LASSO bootstrap benchmark outperforms the LQC method and all three naive benchmarks, and is only slightly worse than the two CGM variants. Further, the relative

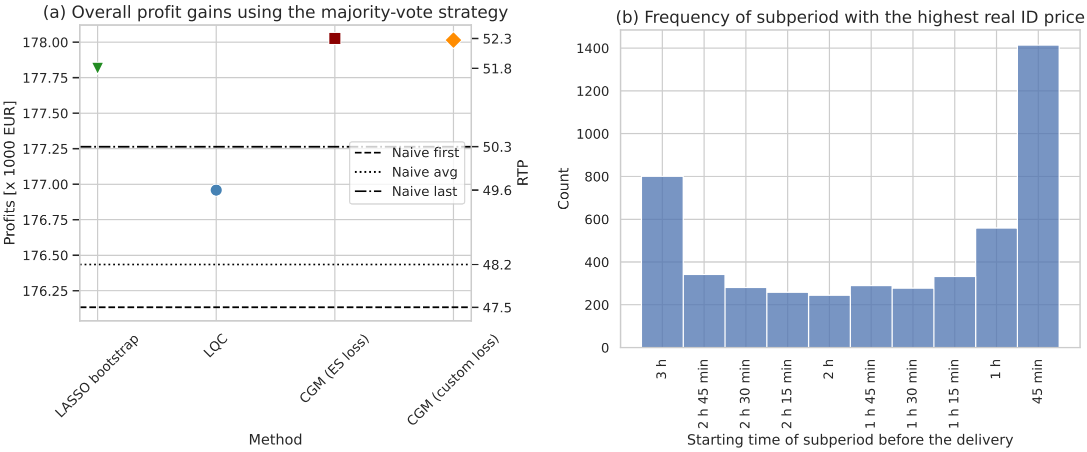


Figure 4.7: Left: Overall profit gains using the majority-vote strategy described in Section 4.4.2 in terms of the nominal profit (left axis) and the realized trading potential (right axis; see Section 4.4.2). Right: Frequency of the 10 considered subperiods for achieving the highest price in real ID paths within the test set.

differences in the overall trading profits remain relatively small across all considered methods and benchmarks.

The realized trading potential indicated that the best naive strategy, $\text{Naive}_{\text{last}}$, achieves an RTP of 50.3. In comparison, the CGM variants yield RTP values of approximately 52.3, representing a 4% improvement over $\text{Naive}_{\text{last}}$. The LASSO bootstrap benchmark, with an RTP of 51.8, provides a 3% improvement. Relative to $\text{Naive}_{\text{first}}$ with an RTP of 47.5, corresponding to a simple market sell order submitted 3 hours before delivery, the CGM variants achieve a 10% improvement, while the LASSO bootstrap shows a 9% improvement.

Interestingly, the $\text{Naive}_{\text{last}}$ benchmark performs well, even outperforming the LQC benchmark. To investigate this further, we analyzed the observed prices within considered subperiods over the test period. Figure 4.7(b) shows the frequency of indices with the highest observed ID price for all hourly markets, indicating the highest price for a given hourly market most frequently occurs during the last subperiod (i.e., from 45 to 30 minutes before delivery). This pattern explains the strong performance of $\text{Naive}_{\text{last}}$ relative to other naive benchmarks.

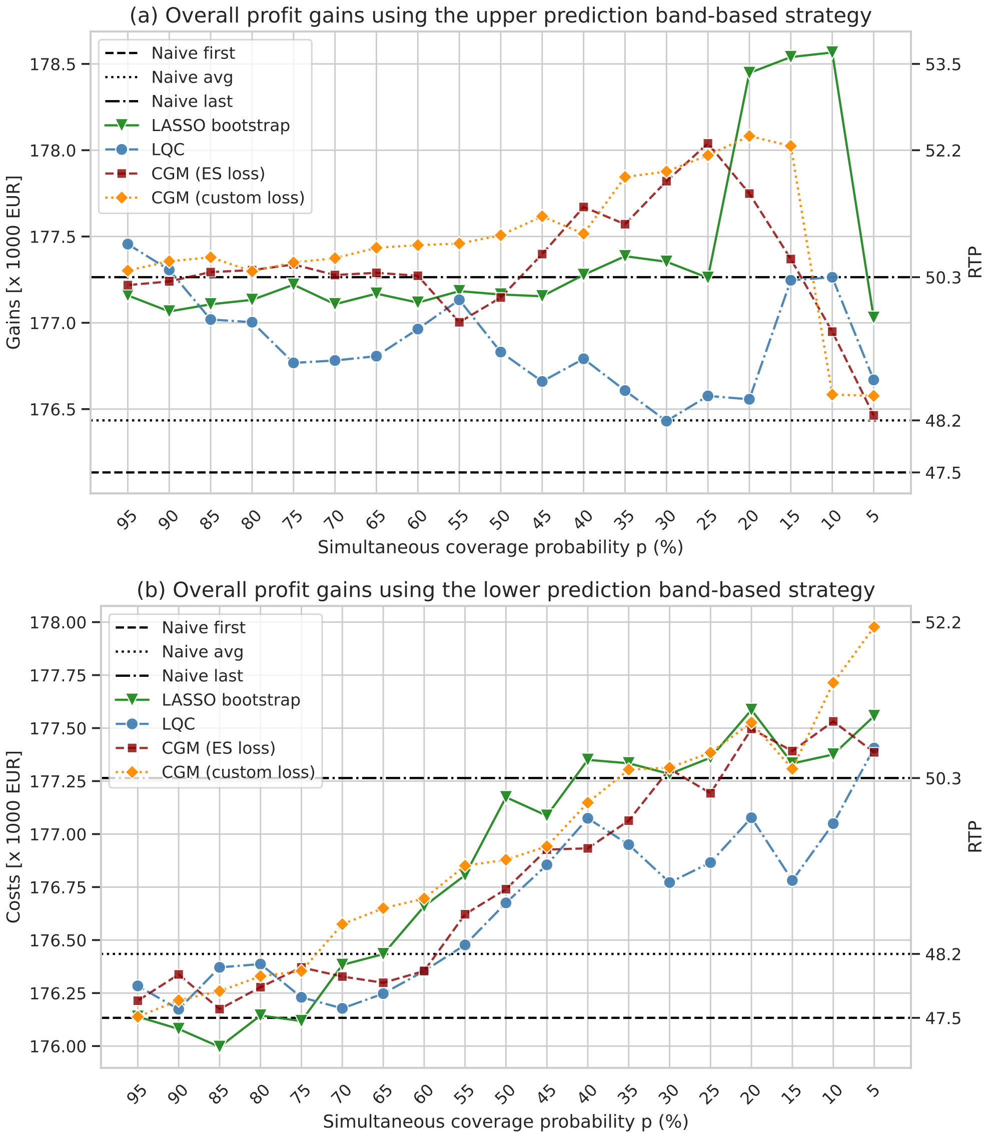


Figure 4.8: Overall profit gains using the prediction band-based strategy described in Section 4.4.2, based on the upper prediction band (upper plot) and the lower prediction band (lower plot), in terms of the nominal profit (left axis) and the realized trading potential (right axis).

As discussed in Section 4.4.2, prediction bands can be derived from a collection of path forecasts. For evaluation and comparison, we first need to specify the simultaneous coverage probability. Here, SCP values ranging from 5% to 95% are considered as ex-post selected thresholds for a more generalized analysis. In real-time trading, the optimal SCP value leading to the highest profits varies over time and of course needs to be selected ex-ante, for example based on historical data, as suggested in Serafin et al. (2022).

Figure 4.8 illustrates the profit gains achieved using the prediction band-based strategy, based on both the upper and lower prediction bands with selected SCP values. As discussed in Section 4.4.2, the decision to determine the optimal selling time based on either the upper or lower prediction band reflects the trader's risk preference. Our observations indicate that no forecasting method consistently outperforms the others across all SCP values in both cases. However, the profit gains associated with the upper prediction band are generally higher than those from the lower prediction band, suggesting that taking on a relatively higher level of risk may yield better returns.

In the lower prediction band-related results shown in Figure 4.8(b), there is a clear trend of increasing profits as the SCP decreases. In contrast, the upper prediction band-related results shown in Figure 4.8(a) do not exhibit a clear trend. At very low SCP values, the remaining path trajectories for deriving lower prediction bands are those with consistently high predicted ID prices, while for the upper prediction bands, the remaining path trajectories correspond to those with consistently low predicted ID prices. This makes the results of profit gains more diverse, as observed for both types of bands in the illustrations, compared with other SCP values.

Focusing on the upper prediction band results that yield higher profits in Figure 4.8(a), we observe that within the middle SCP range (25%–75%), which is more commonly used, the CGM trained with the custom loss function consistently achieves the best performance. This aligns with the intended purpose of the custom loss design. The CGM trained with the ES closely follows, with both CGM variants outperforming all benchmark methods. Although the LASSO bootstrap approach performs reasonably well, its performance under this strategy is not as strong as in the majority vote strategy, and it fails to outperform the best naive baseline. The LQC method performs worst throughout.

4.6 Conclusions

We propose a new approach for electricity price forecasting in continuous intraday markets, utilizing conditional generative machine learning models to produce probabilistic path forecasts. The proposed CGM approach generates multivariate path trajectories directly as the output of a generative neural network, trained using the energy score, which is a mathematically principled loss function for multivariate probabilistic forecasts. A key advantage of this approach is the ability to bypass the separate modeling of marginal distributions and temporal dependencies, which is the cornerstone of many alternative multivariate forecasting approaches. By conditioning on exogenous input variables, such as wind and load data, CGMs can flexibly incorporate information from additional predictors in both the marginal distributions as well as the temporal dependencies. Further, the CGMs can be trained with custom loss functions, for example aiming to integrate specific economic objectives related to trading profits in electricity markets.

An important aspect for evaluating multivariate EPF models is to not only apply commonly used statistical evaluation metrics in the form of suitable multivariate proper scoring rules, but also to evaluate the forecasts from a practically oriented, economic perspective. To that end, we proposed two tailored trading strategies based on multivariate probabilistic information, the majority vote strategy and the prediction band-based strategy, to evaluate the economic performance of path forecasts in a fixed-volume selling scenario.

The results show that while no single model consistently outperforms all others across all statistical and economic evaluation metrics, the CGM framework demonstrates good performance in both aspects compared to two state-of-the-art statistical benchmark methods. Specifically, the CGM is better able to capture temporal dependencies, particularly during peak electricity usage hours. In terms of the economic evaluations, a naive benchmark approach of placing sell orders always at the last time subperiod performed well due to the typical trends of observed ID price paths. Nevertheless, CGMs improved profit gains over this benchmark by 4% in the majority-vote strategy, and yield the highest overall trading profits across all considered approaches. In the prediction band-based strategy, CGMs showed clear advantages, particularly when trained with a custom loss that integrates economic objectives, further highlighting their potential benefits for trading scenarios.

To the best of our knowledge, our work is the first to introduce generative machine learning methods for forecasting ID electricity price paths. A promising avenue for future work lies in advancing economic evaluation methodologies. Realistic trading scenarios provide valuable insights into model performance from a decision-maker's perspective, serving as a practical complement to traditional statistical metrics. By bridging the gap between forecasting accuracy and economic impact, this study contributes to the literature on the economic evaluation of forecasts (Maciejowska et al., 2023; Yardley and Petropoulos, 2021). Moving beyond the fixed-volume selling scenario explored here, it would be interesting to investigate other scenarios in realistic trading markets. Some of the ideas for making trading decisions proposed in this study could be adapted and may also require substantial modifications for certain scenarios, underscoring the need for further research in this domain. Beyond the specific trading scenario, the optimal use of multivariate probabilistic forecasts in deriving optimal trading strategies represents another interesting topic for future research.

Further, while our attempts to integrate the economic aspects into the loss function for training the generative models showed some promise in terms of the realized trading profits, the overall improvements over naive benchmark strategies remain limited. From a methodological perspective, it would be interesting to further investigate the role of the loss function in training generative models for multivariate probabilistic forecasting, with possible choices including a plethora not only of available multivariate proper scoring rules (Pacchiardi et al., 2024), but also of potential ways to incorporate economic aspects.

Bibliography

ADEL, T., Z. GHAHRAMANI, AND A. WELLER (2018): "Discovering Interpretable Representations for Both Deep Generative and Discriminative Models," in *Proceedings of the 35th International Conference on Machine Learning*, PMLR, vol. 80, 50–59.

ALEXANDER, C., M. COULON, Y. HAN, AND X. MENG (2022): "Evaluating the discrimination ability of proper multi-variate scoring rules," *Annals of Operations Research*, 1–27.

ALLEN, S., G. R. EVANS, P. BUCHANAN, AND F. KWASNIOK (2021): "Incorporating the North Atlantic Oscillation into the post-processing of MOGREPS-G wind speed forecasts," *Quarterly Journal of the Royal Meteorological Society*, 147, 1403–1418.

ALLEN, S., D. GINSBOURGER, AND J. ZIEGEL (2023): "Evaluating forecasts for high-impact events using transformed kernel scores," *SIAM/ASA Journal on Uncertainty Quantification*, 11, 906–940.

AN, J. AND S. CHO (2015): "Variational autoencoder based anomaly detection using reconstruction probability," *Special lecture on IE*, 2, 1–18.

ARJOVSKY, M., S. CHINTALA, AND L. BOTTOU (2017): "Wasserstein Generative Adversarial Networks," in *Proceedings of the 34th International Conference on Machine Learning*, ed. by D. Precup and Y. W. Teh, PMLR, vol. 70 of *Proceedings of Machine Learning Research*, 214–223.

BACCOUCHE, M., F. MAMALET, C. WOLF, C. GARCIA, AND A. BASKURT (2012): "Spatio-Temporal Convolutional Sparse Auto-Encoder for Sequence Classification." in *BMVC*, 1–12.

Backer, M., D. Keles, and E. Kraft (2023): "The economic impacts of integrating European balancing markets: The case of the newly installed aFRR energy market-coupling platform PICASSO," *Energy Economics*, 107124.

Baran, S. and S. Lerch (2015): "Log-normal distribution based Ensemble Model Output Statistics models for probabilistic wind-speed forecasting," *Quarterly Journal of the Royal Meteorological Society*, 141, 2289–2299.

——— (2016): "Mixture EMOS model for calibrating ensemble forecasts of wind speed," *Environmetrics*, 27, 116–130.

Baran, S. and A. Möller (2015): "Joint probabilistic forecasting of wind speed and temperature using Bayesian model averaging," *Environmetrics*, 26, 120–132.

Baran, S., P. Szokol, and M. Szabó (2021): "Truncated generalized extreme value distribution-based ensemble model output statistics model for calibration of wind speed ensemble forecasts," *Environmetrics*, 32, e2678.

Bauer, P., A. Thorpe, and G. Brunet (2015): "The quiet revolution of numerical weather prediction," *Nature*, 525, 47–55.

Benjamini, Y. and Y. Hochberg (1995): "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing," *Journal of the Royal Statistical Society: Series B (Methodological)*, 57, 289–300.

Bergstra, J., D. Yamins, and D. Cox (2013): "Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures," in *Proceedings of the 30th International Conference on Machine Learning*, Atlanta, Georgia, USA: PMLR, vol. 28, 115–123.

Bouallegue, Z. B., T. Haiden, N. J. Weber, T. M. Hamill, and D. S. Richardson (2020): "Accounting for Representativeness in the Verification of Ensemble Precipitation Forecasts," *Monthly Weather Review*, 148, 2049–2062.

Bouallègue, Z. B., T. Heppelmann, S. E. Theis, and P. Pinson (2016): "Generation of Scenarios from Calibrated Ensemble Forecasts with a Dual-Ensemble Copula-Coupling Approach," *Monthly Weather Review*, 144, 4737–4750.

Bougeault, P., Z. Toth, et al. (2010): "The THORPEX Interactive Grand Global Ensemble," *Bulletin of the American Meteorological Society*, 91, 1059–1072.

Bourlard, H. and Y. Kamp (1988): "Auto-association by multilayer perceptrons and singular value decomposition," *Biological Cybernetics*, 59, 291–294.

Bremnes, J. B. (2020): "Ensemble Postprocessing Using Quantile Function Regression Based on Neural Networks and Bernstein Polynomials," *Monthly Weather Review*, 148, 403–414.

Browell, J. (2018): "Risk constrained trading strategies for stochastic generation with a single-price balancing market," *Energies*, 11, 1345.

Browell, J. and C. Gilbert (2022): "Predicting electricity imbalance prices and volumes: Capabilities and opportunities," *Energies*, 15, 3645.

Bülte, C., N. Horat, J. Quinting, and S. Lerch (2024): "Uncertainty quantification for data-driven weather models," Preprint, available at https://arxiv.org/abs/2403.13458.

Bunn, D., A. Gianfreda, and S. Kermer (2018): "A trading-based evaluation of density forecasts in a real-time electricity market," *Energies*, 11, 2658.

Burda, Y., R. Grosse, and R. Salakhutdinov (2016): "Importance Weighted Autoencoders," Preprint, available at https://arxiv.org/abs/1509.00519.

Chaloulos, G. and J. Lygeros (2007): "Effect of Wind Correlation on Aircraft Conflict Probability," *Journal of Guidance, Control, and Dynamics*, 30, 1742–1752.

Chapman, W. E., L. D. Monache, S. Alessandrini, A. C. Subramanian, F. M. Ralph, S.-P. Xie, S. Lerch, and N. Hayatbini (2022): "Probabilistic Predictions from Deterministic Atmospheric River Forecasts with Deep Learning," *Monthly Weather Review*, 150, 215 – 234.

Chen, J., T. Janke, F. Steinke, and S. Lerch (2024): "Generative machine learning methods for multivariate ensemble postprocessing," *The Annals of Applied Statistics*, 18, 159–183.

Chen, X., Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel (2016): "InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets," in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., vol. 29.

Chitsaz, H., P. Zamani-Dehkordi, H. Zareipour, and P. Parikh (2018): "Electricity price forecasting for operational scheduling of behind-the-meter storage systems," *IEEE Transactions on Smart Grid*, 9, 6612–6622.

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.

Clevert, D.-A. (2015): "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)," Preprint, available at `https://arxiv.org/abs/1511.07289`.

Collins, M. (2007): "Ensembles and probabilities: a new era in the prediction of climate change," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365, 1957–1970.

Courty, N., R. Flamary, D. Tuia, and A. Rakotomamonjy (2016): "Optimal Transport for Domain Adaptation," *IEEE transactions on pattern analysis and machine intelligence*, 39, 1853–1865.

Cramer, E., D. Witthaut, A. Mitsos, and M. Dahmen (2023): "Multivariate probabilistic forecasting of intraday electricity prices using normalizing flows," *Applied Energy*, 346, 121370.

Cramton, P. (2017): "Electricity market design," *Oxford Review of Economic Policy*, 33, 589–612.

Cuturi, M. (2013): "Sinkhorn Distances: Lightspeed Computation of Optimal Transport," in *Advances in Neural Information Processing Systems*, ed. by C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, Curran Associates, Inc., vol. 26.

Dai, Y. and S. Hemri (2021): “Spatially Coherent Postprocessing of Cloud Cover Ensemble Forecasts,” *Monthly Weather Review*, 149, 3923–3937.

Dawid, A. P. and P. Sebastiani (1999): “Coherent dispersion criteria for optimal experimental design,” *Annals of Statistics*, 65–81.

Diebold, F. X. and R. S. Mariano (1995): “Comparing Predictive Accuracy,” *Journal of Business & Economic Statistics*, 13, 253–263.

Dorninger, M., E. Gilleland, B. Casati, M. P. Mittermaier, E. E. Ebert, B. G. Brown, and L. J. Wilson (2018): “The Setup of the MesoVICT Project,” *Bulletin of the American Meteorological Society*, 99, 1887 – 1906.

Dosovitskiy, A., L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby (2021): “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,” Preprint, available at `https://arxiv.org/abs/2010.11929`.

Dziugaite, G. K., D. M. Roy, and Z. Ghahramani (2015): “Training generative neural networks via maximum mean discrepancy optimization,” in *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, Arlington, Virginia, USA: AUAI Press, UAI’15, 258–267.

Düben, P. D., M. Leutbecher, and P. Bauer (2019): “New Methods for Data Storage of Model Output from Ensemble Simulations,” *Monthly Weather Review*, 147, 677 – 689.

Eisenberger, M., A. Toker, L. Leal-Taixé, F. Bernard, and D. Cremers (2022): “A Unified Framework for Implicit Sinkhorn Differentiation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 509–518.

EPEX (2023): “EPEX SPOT Annual Market Review 2022,” https://www.epex-spot.com/en/news/power-markets-deliver-transparent-price-signals-under-increased-supply-pressure, date accessed: 03.08.2023.

Feik, M., S. Lerch, and J. Stühmer (2024): “Graph Neural Networks and Spatial Information Learning for Post-Processing Ensemble Weather Forecasts,” International Conference on Machine Learning 2024 - Machine Learning for Earth System Modeling Workshop. Available at https://arxiv.org/abs/2407.11050.

Feldmann, K., M. Scheuerer, and T. L. Thorarinsdottir (2015): “Spatial Post-processing of Ensemble Forecasts for Temperature Using Nonhomogeneous Gaussian Regression,” *Monthly Weather Review*, 143, 955–971.

Frogner, C., C. Zhang, H. Mobahi, M. Araya-Polo, and T. Poggio (2015): “Learning with a Wasserstein Loss,” in *Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 2*, Cambridge, MA, USA: MIT Press, NIPS’15, 2053–2061.

Gilleland, E., D. A. Ahijevych, B. G. Brown, and E. E. Ebert (2010): “Verifying Forecasts Spatially,” *Bulletin of the American Meteorological Society*, 91, 1365 – 1376.

Glachant, J.-M., P. Joskow, and M. Pollitt (2021): *Handbook on Electricity Markets*, Edward Elgar Publishing Ltd.

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,” *The 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., A. E. Raftery, A. H. Westveld, and T. Goldman (2005): “Calibrated Probabilistic Forecasting Using Ensemble Model Output Statistics and Minimum CRPS Estimation,” *Monthly Weather Review*, 133, 1098–1118.

Gondara, L. (2016): “Medical Image Denoising Using Convolutional Denoising Autoencoders,” in *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, 241–246.

Goodfellow, I. J., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio (2014): "Generative Adversarial Nets," in *NIPS'14: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, Cambridge, MA, USA: MIT Press, 2672–2680.

Grams, C. M., R. Beerli, S. Pfenninger, I. Staffell, and H. Wernli (2017): "Balancing Europe's wind-power output through spatial deployment informed by weather regimes," *Nature Climate Change*, 7, 557–562.

Gretton, A., K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola (2012): "A Kernel Two-Sample Test," *Journal of Machine Learning Research*, 13, 723–773.

Grooms, I. (2021): "Analog ensemble data assimilation and a method for constructing analogs with variational autoencoders," *Quarterly Journal of the Royal Meteorological Society*, 147, 139–149.

Grönquist, P., C. Yao, T. Ben-Nun, N. Dryden, P. Dueben, S. Li, and T. Hoefler (2021): "Deep learning for post-processing ensemble weather forecasts," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 379, 20200092.

Gui, J., Z. Sun, Y. Wen, D. Tao, and J. Ye (2021): "A Review on Generative Adversarial Networks: Algorithms, Theory, and Applications," *IEEE Transactions on Knowledge and Data Engineering*, 1–1.

Harris, L., A. T. T. McRae, M. Chantry, P. D. Dueben, and T. N. Palmer (2022): "A Generative Deep Learning Approach to Stochastic Downscaling of Precipitation Forecasts," *Journal of Advances in Modeling Earth Systems*, 14, e2022MS003120.

Haupt, S. E., W. Chapman, S. V. Adams, C. Kirkwood, J. S. Hosking, N. H. Robinson, S. Lerch, and A. C. Subramanian (2021): "Towards implementing artificial intelligence post-processing in weather and climate: proposed actions from the Oxford 2019 workshop," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 379, 20200091.

He, K., X. Zhang, S. Ren, and J. Sun (2016): “Deep Residual Learning for Image Recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hess, P., M. Drüke, S. Petri, F. M. Strnad, and N. Boers (2022): “Physically constrained generative adversarial networks for improving precipitation fields from Earth system models,” *Nature Machine Intelligence*, 4, 828–839.

Hinton, G. E. and R. R. Salakhutdinov (2006): “Reducing the Dimensionality of Data with Neural Networks,” *Science*, 313, 504–507, publisher: American Association for the Advancement of Science.

Hinton, G. E. and R. Zemel (1993): “Autoencoders, Minimum Description Length and Helmholtz Free Energy,” in *Advances in Neural Information Processing Systems*, ed. by J. Cowan, G. Tesauro, and J. Alspector, Morgan-Kaufmann, vol. 6.

Hirsch, S. and F. Ziel (2024): “Multivariate simulation-based forecasting for intraday power markets: Modeling cross-product price effects,” *Applied Stochastic Models in Business and Industry*, 40, 1571–1595.

Horat, N. and S. Lerch (2024): “Deep Learning for Postprocessing Global Probabilistic Forecasts on Subseasonal Time Scales,” *Monthly Weather Review*, 152, 667–687.

Hu, Y., M. J. Schmeits, S. J. van Andel, J. S. Verkade, M. Xu, D. P. Solomatine, and Z. Liang (2016): “A Stratified Sampling Approach for Improved Sampling from a Calibrated Ensemble Forecast Distribution,” *Journal of Hydrometeorology*, 17, 2405–2417.

Höhlein, K., B. Schulz, R. Westermann, and S. Lerch (2024): “Postprocessing of Ensemble Weather Forecasts Using Permutation-Invariant Neural Networks,” *Artificial Intelligence for the Earth Systems*, 3, e230070.

Höhlein, K., S. Weiss, T. Necker, M. Weissmann, T. Miyoshi, and R. Westermann (2022): “Evaluation of Volume Representation Networks for Meteorological Ensemble Compression,” in *Vision, Modeling, and Visualization*, ed. by J. Bender, M. Botsch, and D. A. Keim, The Eurographics Association.

Janczura, J. and A. Puć (2023): “ARX-GARCH probabilistic price forecasts for diversification of trade in electricity markets – Variance stabilizing transformation and financial risk-minimizing portfolio allocation,” *Energies*, 16, 807.

Janczura, J. and E. Wójcik (2022): “Dynamic short-term risk management strategies for the choice of electricity market based on probabilistic forecasts of profit and risk measures. The German and the Polish market case study,” *Energy Economics*, 110, 106015.

Janke, T., M. Ghanmi, and F. Steinke (2021): “Implicit Generative Copulas,” in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., vol. 34, 26028–26039.

Janke, T. and F. Steinke (2019): “Forecasting the price distribution of continuous intraday electricity trading,” *Energies*, 12, 4262.

——— (2020): “Probabilistic multivariate electricity price forecasting using implicit generative ensemble post-processing,” in *2020 International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*, IEEE, 1–6.

Jolliffe, I. T. and J. Cadima (2016): “Principal component analysis: a review and recent developments,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374, 20150202.

Jordan, A., F. Krüger, and S. Lerch (2019): “Evaluating Probabilistic Forecasts with scoringRules,” *Journal of Statistical Software*, 90, 1–37.

Kantorovich, L. V. (1960): “Mathematical Methods of Organizing and Planning Production,” *Management science*, 6, 366–422.

Kath, C. and F. Ziel (2018): “The value of forecasts: Quantifying the economic gains of accurate quarter-hourly electricity price forecasts,” *Energy Economics*, 76, 411–423.

Kiefer, S. M., S. Lerch, P. Ludwig, and J. G. Pinto (2023): “Can Machine Learning Models Be a Suitable Tool for Predicting Central European Cold Winter Weather on Subseasonal to Seasonal Time Scales?” *Artificial Intelligence for the Earth Systems*, 2, e230020.

——— (2024): "Random Forests' Postprocessing Capability of Enhancing Predictive Skill on Subseasonal Time Scales — A Flow-Dependent View on Central European Winter Weather," *Artificial Intelligence for the Earth Systems*, 3, e240014.

Kingma, D. P. (2013): "Auto-Encoding Variational Bayes," Preprint, available at `https://arxiv.org/abs/1312.6114`.

——— (2014): "Adam: A Method for Stochastic Optimization," Preprint, available at `https://arxiv.org/abs/1412.6980`.

Kingma, D. P. and M. Welling (2019): "An Introduction to Variational Autoencoders," *Found. Trends Mach. Learn.*, 12, 307–392.

Klein, N., M. S. Smith, and D. J. Nott (2023): "Deep distributional time series models and the probabilistic forecasting of intraday electricity prices," *Journal of Applied Econometrics*, 38, 493–511.

Kochkov, D., J. Yuval, I. Langmore, P. Norgaard, J. Smith, G. Mooers, M. Klöwer, J. Lottes, S. Rasp, P. Düben, S. Hatfield, P. Battaglia, A. Sanchez-Gonzalez, M. Willson, M. P. Brenner, and S. Hoyer (2024): "Neural general circulation models for weather and climate," *Nature*, 632, 1060–1066.

Koenker, R. W. (2005): *Quantile Regression*, Cambridge University Press.

Kolassa, S. (2020): "Why the "best" point forecast depends on the error or accuracy measure," *International Journal of Forecasting*, 36, 208–211.

Kolouri, S., S. R. Park, M. Thorpe, D. Slepcev, and G. K. Rohde (2017): "Optimal Mass Transport: Signal processing and machine-learning applications," *IEEE signal processing magazine*, 34, 43–59.

Kramer, M. A. (1991): "Nonlinear principal component analysis using autoassociative neural networks," *AIChE Journal*, 37, 233–243.

Kuppelwieser, T. and D. Wozabal (2021): "Liquidity costs on intraday power markets: Continuous trading versus auctions," *Energy Policy*, 154, 112299.

——— (2023): "Intraday power trading: toward an arms race in weather forecasting?" *OR Spectrum*, 45, 57–83.

Kwok, J.-Y. and I.-H. Tsang (2004): "The pre-image problem in kernel methods," *IEEE Transactions on Neural Networks*, 15, 1517–1525.

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.

Lakatos, M., S. Lerch, S. Hemri, and S. Baran (2023): "Comparison of multivariate post-processing methods using global ECMWF ensemble forecasts," *Quarterly Journal of the Royal Meteorological Society*, 149, 856–877.

Lakshminarayanan, B., A. Pritzel, and C. Blundell (2017): "Simple and Scalable Predictive Uncertainty Estimation Using Deep Ensembles," in *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems*, Red Hook, NY, USA: Curran Associates Inc., 6405–6416.

Lang, M. N., S. Lerch, G. J. Mayr, T. Simon, R. Stauffer, and A. Zeileis (2020): "Remember the past: a comparison of time-adaptive training schemes for non-homogeneous regression," *Nonlinear Processes in Geophysics*, 27, 23–34.

Lang, M. N., G. J. Mayr, R. Stauffer, and A. Zeileis (2019): "Bivariate Gaussian models for wind vectors in a distributional regression framework," *Advances in Statistical Climatology, Meteorology and Oceanography*, 5, 115–132.

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.

Lauret, P., M. David, and P. Pinson (2019): "Verification of solar irradiance probabilistic forecasts," *Solar Energy*, 194, 254–271.

LEINONEN, J., D. NERINI, AND A. BERNE (2021): “Stochastic Super-Resolution for Downscaling Time-Evolving Atmospheric Fields With a Generative Adversarial Network,” *IEEE Transactions on Geoscience and Remote Sensing*, 59, 7211–7223.

LERCH, S. AND S. BARAN (2017): “Similarity-based semilocal estimation of post-processing models,” *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 66, 29–51.

LERCH, S., S. BARAN, A. MÖLLER, J. GROSS, R. SCHEFZIK, S. HEMRI, AND M. GRAETER (2020): “Simulation-based comparison of multivariate ensemble post-processing methods,” *Nonlinear Processes in Geophysics*, 27, 349–371.

LERCH, S. AND K. L. POLSTERER (2022): “Convolutional autoencoders for spatially-informed ensemble post-processing,” International Conference on Learning Representations (ICLR) 2022 - AI for Earth and Space Science Workshop. Available at https://arxiv.org/abs/2204.05102.

LERCH, S. AND T. L. THORARINSDOTTIR (2013): “Comparison of non-homogeneous regression models for probabilistic wind speed forecasting,” *Tellus A: Dynamic Meteorology and Oceanography*, 65, 21206.

LERCH, S., T. L. THORARINSDOTTIR, F. RAVAZZOLO, AND T. GNEITING (2017): “Forecaster’s Dilemma: Extreme Events and Forecast Evaluation,” *Statistical Science*, 32, 106 – 127.

LI, L., K. JAMIESON, G. DESALVO, A. ROSTAMIZADEH, AND A. TALWALKAR (2018): “Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization,” *Journal of Machine Learning Research*, 18, 1–52.

LI, Y., K. SWERSKY, AND R. ZEMEL (2015): “Generative Moment Matching Networks,” in *Proceedings of the 32nd International Conference on Machine Learning*, Lille, France: PMLR, vol. 37, 1718–1727.

LIAW, R., E. LIANG, R. NISHIHARA, P. MORITZ, J. E. GONZALEZ, AND I. STOICA (2018): “Tune: A Research Platform for Distributed Model Selection and Training,” Preprint, available at https://arxiv.org/abs/1807.05118.

LOSHCHILOV, I. AND F. HUTTER (2019): "Decoupled Weight Decay Regularization," Preprint, available at https://arxiv.org/abs/1711.05101.

LUCAS, J., G. TUCKER, R. GROSSE, AND M. NOROUZI (2019): "Understanding Posterior Collapse in Generative Latent Variable Models," ICLR 2019 Workshop DeepGenStruct. Available at https://openreview.net/forum?id=r1xaVLUYuE.

MACIEJOWSKA, K. (2022): "Portfolio management of a small RES utility with a structural vector autoregressive model of electricity markets in Germany," *Operations Research and Decisions*, 32, 75–90.

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.

MACIEJOWSKA, K., B. UNIEJEWSKI, AND R. WERON (2023): "Forecasting Electricity Prices," in *Oxford Research Encyclopedia of Economics and Finance*, Oxford University Press.

MAHESH, A., W. COLLINS, B. BONEV, N. BRENOWITZ, Y. COHEN, J. ELMS, P. HARRINGTON, K. KASHINATH, T. KURTH, J. NORTH, T. O'BRIEN, M. PRITCHARD, D. PRUITT, M. RISSER, S. SUBRAMANIAN, AND J. WILLARD (2024): "Huge Ensembles Part I: Design of Ensemble Weather Forecasts using Spherical Fourier Neural Operators," Preprint, available at https://arxiv.org/abs/2408.03100.

MATHESON, J. E. AND R. L. WINKLER (1976): "Scoring Rules for Continuous Probability Distributions," *Management Science*, 22, 1087–1096.

MAYER, K. AND S. TRÜCK (2018): "Electricity markets around the world," *Journal of Commodity Markets*, 9, 77–100.

MCGOVERN, A., R. LAGERQUIST, D. J. GAGNE, G. E. JERGENSEN, K. L. ELMORE, C. R. HOMEYER, AND T. SMITH (2019): "Making the Black Box More Transparent: Understanding the Physical Implications of Machine Learning," *Bulletin of the American Meteorological Society*, 100, 2175–2199.

Mika, S., B. Schölkopf, A. Smola, K.-R. Müller, M. Scholz, and G. Rätsch (1998): "Kernel PCA and De-Noising in Feature Spaces," in *Advances in Neural Information Processing Systems*, ed. by M. Kearns, S. Solla, and D. Cohn, MIT Press, vol. 11.

Mockert, F., C. M. Grams, S. Lerch, M. Osman, and J. Quinting (2024): "Multivariate post-processing of probabilistic sub-seasonal weather regime forecasts," *Quarterly Journal of the Royal Meteorological Society*, 150, 4771–4787.

Mohamed, S. and B. Lakshminarayanan (2017): "Learning in Implicit Generative Models," Preprint, available at `https://arxiv.org/abs/1610.03483`.

Möller, A., A. Lenkoski, and T. L. Thorarinsdottir (2013): "Multivariate probabilistic forecasting using ensemble Bayesian model averaging and copulas," *Quarterly Journal of the Royal Meteorological Society*, 139, 982–991.

Monteiro, C., I. Ramirez-Rosado, L. Fernandez-Jimenez, and P. Conde (2016): "Short-term price forecasting models based on artificial neural networks for intraday sessions in the Iberian electricity market," *Energies*, 9, 721.

Muschinski, T., M. N. Lang, G. J. Mayr, J. W. Messner, A. Zeileis, and T. Simon (2022): "Predicting power ramps from joint distributions of future wind speeds," *Wind Energy Science*, 7, 2393–2405.

Narajewski, M. and F. Ziel (2020a): "Econometric modelling and forecasting of intraday electricity prices," *Journal of Commodity Markets*, 19, 100107.

——— (2020b): "Ensemble forecasting for intraday electricity prices: Simulating trajectories," *Applied Energy*, 279, 115801.

Nelsen, R. B. (2006): *An Introduction to Copulas*, Springer New York, NY, 2 ed.

Nowotarski, J. and R. Weron (2015): "Computing electricity spot price prediction intervals using quantile regression and forecast averaging," *Computational Statistics*, 30, 791–803.

Oksuz, I. and U. Ugurlu (2019): "Neural network based model comparison for intraday electricity price forecasting," *Energies*, 12, 4557.

Pacchiardi, L., R. A. Adewoyin, P. Dueben, and R. Dutta (2024): "Probabilistic forecasting with generative networks via scoring rule minimization," *Journal of Machine Learning Research*, 25, 1–64.

Pantillon, F., S. Lerch, P. Knippertz, and U. Corsmeier (2018): "Forecasting wind gusts in winter storms using a calibrated convection-permitting ensemble," *Quarterly Journal of the Royal Meteorological Society*, 144, 1864–1881.

Patrini, G., R. van den Berg, P. Forré, M. Carioni, S. Bhargav, M. Welling, T. Genewein, and F. Nielsen (2020): "Sinkhorn AutoEncoders," in *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, ed. by R. P. Adams and V. Gogate, PMLR, vol. 115 of *Proceedings of Machine Learning Research*, 733–743.

Pearson, K. (1901): "LIII. On lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2, 559–572.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay (2011): "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, 12, 2825–2830.

Pennington, J., R. Socher, and C. Manning (2014): "GloVe: Global Vectors for Word Representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar: Association for Computational Linguistics, 1532–1543.

Perrone, E., I. Schicker, and M. N. Lang (2020): "A case study of empirical copula methods for the statistical correction of forecasts of the ALADIN-LAEF system," *Meteorologische Zeitschrift*, 29, 277–288.

Petropoulos, F. et al. (2022): "Forecasting: theory and practice," *International Journal of Forecasting*, 38, 705–871.

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

Pinson, P., H. Madsen, H. A. Nielsen, G. Papaefthymiou, and B. Klöckl (2009): "From probabilistic forecasts to statistical scenarios of short-term wind power production," *Wind Energy*, 12, 51–62.

Pinson, P. and J. W. Messner (2018): "Chapter 9 - Application of Postprocessing for Renewable Energy," in *Statistical Postprocessing of Ensemble Forecasts*, Elsevier, 241–266.

Price, I. and S. Rasp (2022): "Increasing the accuracy and resolution of precipitation forecasts using deep generative models," in *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, PMLR, vol. 151, 10555–10571.

Price, I., A. Sanchez-Gonzalez, F. Alet, T. Ewalds, A. El-Kadi, J. Stott, S. Mohamed, P. Battaglia, R. Lam, and M. Willson (2023): "GenCast: Diffusion-based ensemble forecasting for medium-range weather," Preprint, available at https://arxiv.org/abs/2312.15796.

Pu, Y., Z. Gan, R. Henao, X. Yuan, C. Li, A. Stevens, and L. Carin (2016): "Variational Autoencoder for Deep Learning of Images, Labels and Captions," 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.

Rasp, S. and S. Lerch (2018): "Neural Networks for Postprocessing Ensemble Weather Forecasts," *Monthly Weather Review*, 146, 3885–3900.

Ravuri, S., K. Lenc, M. Willson, et al. (2021): "Skilful precipitation nowcasting using deep generative models of radar," *Nature*, 597, 672–677.

Rizzo, M. L. and G. J. Székely (2016): "Energy distance," *WIREs Computational Statistics*, 8, 27–38.

Roberts, N. M. and H. W. Lean (2008): "Scale-Selective Verification of Rainfall Accumulations from High-Resolution Forecasts of Convective Events," *Monthly Weather Review*, 136, 78 – 97.

Rodwell, M. J., D. S. Richardson, D. B. Parsons, and H. Wernli (2018): "Flow-Dependent Reliability: A Path to More Skillful Ensemble Forecasts," *Bulletin of the American Meteorological Society*, 99, 1015 – 1026.

Roweis, S. T. and L. K. Saul (2000): "Nonlinear Dimensionality Reduction by Locally Linear Embedding," *Science*, 290, 2323–2326.

Russo, M., E. Kraft, V. Bertsch, and D. Keles (2022): "Short-term risk management of electricity retailers under rising shares of decentralized solar generation," *Energy Economics*, 109, 105956.

Sakurada, M. and T. Yairi (2014): "Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction," in *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, New York, NY, USA: Association for Computing Machinery, MLSDA'14, 4–11.

Schefzik, R. (2016): "A Similarity-Based Implementation of the Schaake Shuffle," *Monthly Weather Review*, 144, 1909–1921.

Schefzik, R. and A. Möller (2018): "Chapter 4 - Ensemble Postprocessing Methods Incorporating Dependence Structures," in *Statistical Postprocessing of Ensemble Forecasts*, Elsevier, 91–125.

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

Scheuerer, M. and T. M. Hamill (2015): "Variogram-Based Proper Scoring Rules for Probabilistic Forecasts of Multivariate Quantities," *Monthly Weather Review*, 143, 1321–1334.

Scheuerer, M., T. M. Hamill, B. Whitin, M. He, and A. Henkel (2017): "A method for preferential selection of dates in the Schaake shuffle approach to constructing spatiotemporal forecast fields of temperature and precipitation," *Water Resources Research*, 53, 3029–3046.

Scheuerer, M., C. Heinrich-Mertsching, T. K. Bahaga, M. Gudoshava, and T. L. Thorarinsdottir (2024): "Applications of machine learning to predict seasonal precipitation for East Africa," Preprint, available at `https://arxiv.org/abs/2409.06238`.

Scheuerer, M. and D. Möller (2015): "Probabilistic Wind Speed Forecasting on a Grid Based on Ensemble Model Output Statistics," *The Annals of Applied Statistics*, 9, 1328–1349.

Scheuerer, M., M. B. Switanek, R. P. Worsnop, and T. M. Hamill (2020): "Using Artificial Neural Networks for Generating Probabilistic Subseasonal Precipitation Forecasts over California," *Monthly Weather Review*, 148, 3489–3506.

Schölkopf, B., A. Smola, and K.-R. Müller (1997): "Kernel principal component analysis," in *Artificial Neural Networks — ICANN'97*, ed. by W. Gerstner, A. Germond, M. Hasler, and J.-D. Nicoud, Berlin, Heidelberg: Springer Berlin Heidelberg, 583–588.

Schuhen, N., T. L. Thorarinsdottir, and T. Gneiting (2012): "Ensemble Model Output Statistics for Wind Vectors," *Monthly Weather Review*, 140, 3204–3219.

Schulz, B. and S. Lerch (2022a): "Aggregating distribution forecasts from deep ensembles," Preprint, available at `https://arxiv.org/abs/2204.02291`.

——— (2022b): "Machine Learning Methods for Postprocessing Ensemble Forecasts of Wind Gusts: A Systematic Comparison," *Monthly Weather Review*, 150, 235 – 257.

Sejdinovic, D., B. Sriperumbudur, A. Gretton, and K. Fukumizu (2013): "Equivalence of distance-based and RKHS-based statistics in hypothesis testing," *The Annals of Statistics*, 41, 2263–2291.

Serafin, T., G. Marcjasz, and R. Weron (2022): "Trading on short-term path forecasts of intraday electricity prices," *Energy Economics*, 112, 106125.

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.

SLOUGHTER, J. M., T. GNEITING, AND A. E. RAFTERY (2010): “Probabilistic wind speed forecasting using ensembles and Bayesian model averaging,” *Journal of the American Statistical Association*, 105, 25–35.

SONG, Y. AND S. ERMON (2020): “Improved Techniques for Training Score-Based Generative Models,” in *NIPS’20: Proceedings of the 34th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA: Curran Associates Inc., 1043, 12438–12448.

STASZEWSKA, A. (2007): “Representing uncertainty about response paths: The use of heuristic optimisation methods,” *Computational Statistics & Data Analysis*, 52, 121–132.

SZÉKELY, G. J. (2003): “E-statistics: The energy of statistical samples,” *Bowling Green State University, Department of Mathematics and Statistics Technical Report*, 3, 1–18.

SZÉKELY, G. J. AND M. L. RIZZO (2013): “Energy statistics: A class of statistics based on distances,” *Journal of Statistical Planning and Inference*, 143, 1249–1272.

TAILLARDAT, M., O. MESTRE, M. ZAMO, AND P. NAVEAU (2016): “Calibrated Ensemble Forecasts Using Quantile Regression Forests and Ensemble Model Output Statistics,” *Monthly Weather Review*, 144, 2375–2393.

TENENBAUM, J. B., V. DE SILVA, AND J. C. LANGFORD (2000): “A Global Geometric Framework for Nonlinear Dimensionality Reduction,” *Science*, 290, 2319–2323.

THORARINSDOTTIR, T. L. AND T. GNEITING (2010): “Probabilistic forecasts of wind speed: ensemble model output statistics by using heteroscedastic censored regression,” *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 173, 371–388.

THORARINSDOTTIR, T. L., T. GNEITING, AND N. GISSIBL (2013): “Using Proper Divergence Functions to Evaluate Climate Models,” *SIAM/ASA Journal on Uncertainty Quantification*, 1, 522–534.

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.

Tibshirani, R. (1996): "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society B*, 58, 267–288.

Timmermann, A. (2000): "Density forecasting in economics and finance," *Journal of Forecasting*, 19, 231.

Tolstikhin, I., O. Bousquet, S. Gelly, and B. Schoelkopf (2019): "Wasserstein Auto-Encoders," Preprint, available at `https://arxiv.org/abs/1711.01558`.

Tschora, L., E. Pierre, M. Plantevit, and C. Robardet (2022): "Electricity price forecasting on the day-ahead market using machine learning," *Applied Energy*, 313, 118752.

Uniejewski, B., G. Marcjasz, and R. Weron (2019): "Understanding intraday electricity markets: Variable selection and very short-term price forecasting using LASSO," *International Journal of Forecasting*, 35, 1533–1547.

Uniejewski, B., R. Weron, and F. Ziel (2018): "Variance stabilizing transformations for electricity spot price forecasting," *IEEE Transactions on Power Systems*, 33, 2219–2229.

Van Schaeybroeck, B. and S. Vannitsem (2015): "Ensemble post-processing using member-by-member approaches: theoretical aspects," *Quarterly Journal of the Royal Meteorological Society*, 141, 807–818.

Vannitsem, S., J. B. Bremnes, J. Demaeyer, G. R. Evans, J. Flowerdew, S. Hemri, S. Lerch, N. Roberts, S. Theis, A. Atencia, Z. B. Bouallègue, J. Bhend, M. Dabernig, L. D. Cruz, L. Hieta, O. Mestre, L. Moret, I. O. Plenković, M. Schmeits, M. Taillardat, J. V. den Bergh, B. V. Schaeybroeck, K. Whan, and J. Ylhaisi (2021): "Statistical Postprocessing for Weather Forecasts: Review, Challenges, and Avenues in a Big Data World," *Bulletin of the American Meteorological Society*, 102, E681–E699.

VELDKAMP, S., K. WHAN, S. DIRKSEN, AND M. SCHMEITS (2021): "Statistical Postprocessing of Wind Speed Forecasts Using Convolutional Neural Networks," *Monthly Weather Review*, 149, 1141–1152.

WANG, W., Y. HUANG, Y. WANG, AND L. WANG (2014): "Generalized Autoencoder: A Neural Network Framework for Dimensionality Reduction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops.*

WANG, Y., H. YAO, AND S. ZHAO (2016): "Auto-encoder based dimensionality reduction," *Neurocomputing*, 184, 232–242.

WANG, Z., A. BOVIK, H. SHEIKH, AND E. SIMONCELLI (2004): "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, 13, 600–612.

WERON, R. (2014): "Electricity price forecasting: A review of the state-of-the-art with a look into the future," *International Journal of Forecasting*, 30, 1030–1081.

WILKS, D. S. (2015): "Multivariate ensemble Model Output Statistics using empirical copulas," *Quarterly Journal of the Royal Meteorological Society*, 141, 945–952.

——— (2020): "Regularized Dawid–Sebastiani score for multivariate ensemble forecasts," *Quarterly Journal of the Royal Meteorological Society*, 146, 2421–2431.

WORSNOP, R. P., M. SCHEUERER, T. M. HAMILL, AND J. K. LUNDQUIST (2018): "Generating wind power scenarios for probabilistic ramp event prediction using multivariate statistical post-processing," *Wind Energy Science*, 3, 371–393.

YANG, L. M. AND I. GROOMS (2021): "Machine learning techniques to construct patched analog ensembles for data assimilation," *Journal of Computational Physics*, 443, 110532.

YARDLEY, E. AND F. PETROPOULOS (2021): "Beyond error measures to the utility and cost of the forecasts," *Foresight*, Q4, 36–45.

Zaheer, M., S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola (2017): “Deep Sets,” 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.

Zhang, Z. and M. Wu (2022): “Predicting real-time locational marginal prices: A GAN-based approach,” *IEEE Transactions on Power Systems*, 37, 1286–1296.

Zhong, X., L. Chen, H. Li, J. Feng, and B. Lu (2024): “FuXi-ENS: A machine learning model for medium-range ensemble weather forecasting,” Preprint, available at `https://arxiv.org/abs/2405.05925`.

Zhou, B. (2022): “Interpreting Generative Adversarial Networks for Interactive Image Generation,” in *xxAI - Beyond Explainable AI: International Workshop, Held in Conjunction with ICML 2020, July 18, 2020, Vienna, Austria, Revised and Extended Papers*, Cham: Springer International Publishing, 167–175.

Zhou, C. and R. C. Paffenroth (2017): “Anomaly Detection with Robust Deep Autoencoders,” in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA: Association for Computing Machinery, KDD ’17, 665–674.

Ziel, F. and K. Berk (2019): “Multivariate Forecasting Evaluation: On Sensitive and Strictly Proper Scoring Rules,” Preprint, available at `https://arxiv.org/abs/1910.07325`.

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.

Eidesstattliche Versicherung

gemäß § 13 Abs. 2 Ziff. 3 der Promotionsordnung des Karlsruher Instituts für Technologie für die KIT-Fakultät für Wirtschaftswissenschaften

1. Bei der eingereichten Dissertation zu dem Thema *Generative Machine Learning Methods for Multivariate Probabilistic Forecasting* handelt es sich um meine eigenständig erbrachte Leistung.

2. Ich habe nur die angegebenen Quellen und Hilfsmittel benutzt und mich keiner unzulässigen Hilfe Dritter bedient. Insbesondere habe ich wörtlich oder sinngemäß aus anderen Werken übernommene Inhalte als solche kenntlich gemacht.

3. Die Arbeit oder Teile davon habe ich bislang nicht an einer Hochschule des In- oder Auslands als Bestandteil einer Prüfungs- oder Qualifikationsleistung vorgelegt.

4. Die Richtigkeit der vorstehenden Erklärungen bestätige ich.

5. Die Bedeutung der eidesstattlichen Versicherung und die strafrechtlichen Folgen einer unrichtigen oder unvollständigen eidesstattlichen Versicherung sind mir bekannt. Ich versichere an Eides statt, dass ich nach bestem Wissen die reine Wahrheit erklärt und nichts verschwiegen habe.

Karlsruhe, 20. Jan 2025

(Jieyu Chen)