

# Data-Driven Modeling and Analysis of Numerical Weather Predictions

**Kevin Höhlein**

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung eines

## **Doktors der Naturwissenschaften (Dr. rer. nat.)**

genehmigten Dissertation.

### **Vorsitz:**

Prof. Dr. Helmut Seidl

### **Prüfende der Dissertation:**

1. Prof. Dr. Rüdiger Westermann
2. Prof. Dr.-Ing. Tobias Günther

Die Dissertation wurde am 23.09.2024 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 11.12.2024 angenommen.



---

## Abstract

---

Weather forecasting relies on large-scale simulation datasets, which are generated by numerical weather prediction (NWP) models. To account for the inherent uncertainties of weather prediction, numerical simulations are repeated multiple times with slightly different initial conditions and model configurations. The resulting forecast ensembles provide forecasters with information about weather trends and prediction uncertainties. Due to the central role of numerical simulation data within this process, recent developments in data-driven modeling, computational technologies, machine learning (ML), and deep learning (DL) have advanced the state of the field. Examples of data-driven inference tasks in NWP include the statistical postprocessing of weather simulations as well as the visualization and visual analysis of multi-dimensional and multivariate forecast ensembles. Such applications can benefit from the adoption of ML and DL techniques and are subject to ongoing research.

In this publication-based dissertation, we compile research results from five articles, which explore the use of data-driven modeling techniques for postprocessing, analyzing, and compressing the outputs of NWP models. A specific focus is put on the design and validation of adequate ML and DL model architectures to meet the requirements and quality criteria of the respective modeling tasks. The proposed DL models are assessed critically regarding their performance advantage over more classical approaches and the interpretability of models' predictions. The developed methods profit from interdisciplinary connections between meteorological applications and data-driven learning tasks in other scientific domains, such as computer vision, image processing, and visualization.

The first group of studies focuses on ML and DL approaches for postprocessing NWP model outputs. Specifically, we address forecast quality limitations in NWP due to model resolution constraints and systematic statistical model errors. Commonly, the finite grid resolution of NWP models constrains the models' ability to resolve physical processes on horizontal spatial scales smaller than several kilometers. In NWP datasets, this induces a lack of prediction accuracy on small spatial scales and statistical error patterns on larger scales. In our first study, we examine how convolutional neural networks (CNNs) can improve the resolution of weather forecasts in these conditions, using near-surface wind field forecasts as a practical application. Drawing parallels to image superresolution, the study compares various CNN architectures and combines aspects of the best-performing architectures to build a new CNN model with high prediction accuracy and low computational

## *Abstract*

footprint compared to costly high-resolution simulations. Our second study focuses on ensemble postprocessing, using DL models with a permutation-invariant neural network (NN) architecture that is natively suited for processing ensemble-valued inputs. Our ensemble-focused approach contrasts previous methods, which often operate on simple ensemble summary statistics and thus discard information potentially prematurely. Case studies with applications to wind gust and surface temperature forecasting demonstrate the utility of the approach, and dedicated model explanation techniques are developed to explore the reasoning processes in the model. In our third study, we develop a simple statistical model for postprocessing surface temperatures in complex terrain. We introduce a simple, physically motivated method that adjusts temperature predictions according to the altitude difference between terrain altitudes in the real world and the approximate coarse representation inside the NWP model. A 3D visualization tool is developed alongside the postprocessing scheme to help understand how terrain variations impact the forecast accuracy.

In the second group of studies, we address the visual analysis and compressed representation of ensemble forecast datasets using DL and visualization techniques. Due to their sheer data volume, large ensemble datasets impose challenges on interactive visualization systems. Motivated by advances in computer vision in representing 3D scenery data in NN-based data structures, our fourth study looks at compressive neural representation for ensemble datasets. We illustrate how neural data representations can be adapted to ensemble datasets and explore the impact of data properties, such as value distributions, on the representation quality. The proposed ensemble representation networks (ERNs) achieve high compression rates while maintaining data quality and facilitating rapid data access. Our fifth and final study explores this line of research further. Neural dependence fields (NDFs) are introduced as a specialized NN-based data structure to enable the interactive exploration of statistical dependencies in volumetric ensemble datasets. Using a dedicated NN architecture, NDFs learn to represent bivariate dependence fields in large ensemble data. Integrating NDFs into an interactive visualization tool enables the visual exploration of correlations and interdependence patterns in large ensemble datasets.

---

## Zusammenfassung

---

Wettervorhersagen stützen sich auf großskalige Simulationsdatensätze, die von numerischen Wettermodellen generiert werden. Um Unsicherheiten zu berücksichtigen, werden Wettersimulationen mehrfach wiederholt und jeweils mit leicht unterschiedlichen Anfangsbedingungen und Modellkonfigurationen ausgewertet. Die daraus resultierenden Ensemble-Vorhersagen liefern Informationen über Wettertrends und zu erwartende Vorhersagefehler. In Anbetracht der zentralen Rolle, die numerischen Simulationsdaten in der Wettervorhersage spielen, motivieren Fortschritte in der Entwicklung von datengetriebenen Modellierungsansätze und in der Verfügbarkeit von computergestützten Technologien neue Forschungsansätze. Insbesondere maschinelles Lernen (ML) und Deep Learning (DL) haben großes Potential, numerische Wettervorhersagen zu verbessern. Beispiele für datengetriebene Forschungsfragen in der numerischen Wettervorhersage sind die statistische Nachbearbeitung von Wettersimulationen und die computergestützte visuelle Analyse von mehrdimensionalen und multivariaten Ensemble-Vorhersagen.

In der vorliegenden publikationsbasierten Dissertation stellen wir Forschungsergebnisse aus fünf Artikeln vor, die sich mit diesen Themen beschäftigen. Insbesondere untersuchen die Studien den Einsatz von datengetriebenen Modellierungstechniken für die Nachbearbeitung, die Analyse und die Komprimierung von Vorhersagedaten aus numerischen Wettermodellen. Ein besonderer Schwerpunkt liegt dabei auf dem Entwurf und der Validierung geeigneter ML- und DL-Modellarchitekturen, um den Anforderungen und Qualitätskriterien der jeweiligen Anwendungen nachzukommen. Die vorgeschlagenen DL-Modelle werden kritisch auf ihre Leistungsvorteile gegenüber klassischeren ML-Ansätzen untersucht und die Interpretierbarkeit ihrer Vorhersagen beurteilt. Die entwickelten Methoden profitieren dabei davon, dass meteorologische Anwendungen oft große Ähnlichkeiten mit Anwendungen in anderen Wissenschaftsbereichen aufweisen. Relevante Methoden aus der vorliegenden Arbeit kommen beispielsweise aus dem maschinellen Sehen, der Bildverarbeitung und der computergestützten Visualisierung.

In der ersten Gruppe von Studien konzentrieren wir uns auf ML- und DL-Ansätze für die Nachbearbeitung von numerischen Wettervorhersagen. Insbesondere befassen wir uns mit Einschränkungen der Vorhersagequalität, die aus der limitierten räumlichen Auflösung von Wettermodellen resultieren, und mit systematischen statistischen Modellfehlern. Die beschränkte Auflösung der numerischen Wettermodelle hat Einfluss auf die Korrektheit der Abbildung von räumlich kleinskaligen physikalischen Prozessen auf

der Größenordnung von einigen Kilometern horizontaler Ausdehnung. In Wettervorhersagen führt dies zu Ungenauigkeiten auf kleinen räumlichen Skalen sowie zu statistischen Fehlern auf größeren Skalen. In unserer ersten Studie untersuchen wir, wie faltungs-basierte neuronale Netze (Convolutional Neural Networks; CNNs) Wettervorhersagen verbessern können. Als Anwendung nutzen wir dabei die Vorhersage von bodennahen Windfeldern. Unter Einbeziehung von Erkenntnissen aus der Bildverarbeitung, insbesondere aus Methoden zur Verbesserung der Auflösung von Bilddaten, vergleicht die Studie verschiedene CNN-Architekturen bezüglich ihrer Vorhersagegenauigkeit und kombiniert Aspekte der leistungsstärksten Architekturen, um ein neues Modell zu konstruieren. Unsere zweite Studie konzentriert sich auf die Nachbearbeitung von Ensemble-Vorhersagen. Dabei verwenden wir DL-Modelle mit einer permutations-invarianten Netzwerkarchitektur, die nativ für die Verarbeitung von Ensembles geeignet sind. Unser ensemble-basierter Ansatz unterscheidet sich von früheren Methoden, die oft mit einfachen Architekturen arbeiten, statistische Zusammenfassungen der Ensemble-Verteilung verwenden und damit potenziell wichtige Informationen in der Ensemble-Verteilung vorzeitig verwerfen. Fallstudien zur Vorhersage von Windböen und Oberflächentemperaturen zeigen den Nutzen des Ansatzes. Weiterhin werden spezielle Techniken zur Erklärung der Modelle entwickelt, die Einblicke in die modell-interne Informationsverarbeitung erlauben. In unserer dritten Studie entwickeln wir ein einfaches statistisches Modell für die Nachbearbeitung von Oberflächentemperaturen in komplexem Gelände. Wir stellen ein einfaches, physikalisch motiviertes Schema vor, nach welchem Temperaturvorhersagen korrigiert werden können, wenn Höhenunterschiede zwischen der realen Welt und der Terrain-Repräsentation im Wettermodell vorliegen. Zusätzlich entwickeln wir ein 3D-Visualisierungstool, um zu verstehen, wie sich Geländeunsicherheiten auf die Vorhersagegenauigkeit auswirken.

In der zweiten Gruppe von Studien befassen wir uns mit der visuellen Analyse und komprimierten Darstellung von Ensemble-Vorhersagen und untersuchen, wie DL und spezialisierte Visualisierungstechniken diese verbessern können. Aufgrund ihrer Datenmenge stellen große Ensemble-Datensätze eine Herausforderung für interaktive Visualisierungssysteme dar. Motiviert durch Fortschritte im maschinellen Sehen und in der Repräsentation von 3D-Szenen-Daten in Form von neuronalen Datenstrukturen, befasst sich unsere vierte Studie mit der komprimierten Darstellung von Ensemble-Datensätzen durch neuronale Repräsentationen. Wir zeigen, wie neuronale Datenrepräsentationen an Ensemble-Daten angepasst werden können und untersuchen die Auswirkungen von Dateneigenschaften, wie z.B. der Werteverteilungen, auf die Qualität der Darstellung. Die vorgeschlagenen Ensemble-Repräsentationsnetzwerke (ERNs) erreichen hohe Kompressionsraten bei guter Datenqualität und erleichtern den schnellen Zugriff auf die gespeicherten Daten. In unserer fünften und letzten Studie führen wir diesen Forschungszweig weiter. Insbesondere stellen wir Neural Dependence Fields (NDFs) als eine spezialisierte neuronale Datenstruktur vor, die die interaktive visuelle Exploration von statistischen Abhängigkeiten in volumetrischen Ensemble-Datensätzen ermöglicht. Unter Verwendung einer speziellen NN-Architektur lernen NDFs, bivariate Abhängigkeitsfelder in großen Ensemble-Daten zu enkodieren. Durch die Integration von NDFs in ein interaktives Visualisierungstool wird die Anwendbarkeit des Ansatzes illustriert.

---

## Acknowledgments

---

I thank my doctoral supervisor, Prof. Dr. Rüdiger Westermann, for his advice and support during my time at the Technical University of Munich (TUM). His openness to new research topics allowed me to develop and follow my project ideas, and his guidance and direction were crucial for bringing the projects to a successful completion. Throughout our discussions, I have admired his ability to identify key contributions in our work and express them concisely, which helped considerably improve the direction and presentation of our research. I am especially grateful for the opportunity to establish research collaborations with scientists at other institutions and the possibility to travel and visit these research groups for scientific and interpersonal exchange.

Furthermore, I want to thank all my coauthors for their timely and valuable contributions, helpful support, critical feedback, and original input. Special thanks go to my senior collaborators, Dr. Tim Hewson at the European Centre for Medium-Range Weather Forecasts (ECMWF) and Dr. Sebastian Lerch at the Karlsruhe Institute of Technology (KIT). Both suggested new research ideas, provided invaluable guidance and domain expertise throughout the project work, and introduced me to their colleagues when further help was needed. Beyond that, I thank Dr. Michael Kern, especially for his support in the early phase of my doctoral studies, and I want to express my appreciation for the efforts and contributions made by Fatemeh Farokhmanesh, Ludwig Leonard, Christoph Neuhauser, Dr. Benedikt Schulz, Josef Stumpfegger, and Dr. Sebastian Weiß. I also thank Dr. Tobias Necker, Prof. Dr. Martin Weissmann, and Prof. Dr. Takemasa Miyoshi for providing the simulation data used in several publications.

Much of the success of this work is due to the pleasant and joyful working environment in which I found myself. A major part of this was the transregional collaborative research center SFB/TRR 165: *Waves to Weather (W2W)*, funded by the German Research Foundation (Deutsche Forschungsgemeinschaft; DFG). The research presented in this thesis was part of the W2W subproject B5: *Data-driven analysis and learning of the temporal evolution of ensemble forecasts*, and the financial support is gratefully acknowledged. Next to that, W2W allowed me to pique into the meteorological research community and connect with collaboration partners and great colleagues inside and outside W2W. Representing all of this, I want to thank Prof. Dr. George Craig and Dr. Audine Laurian for planning and organizing W2W and the early career scientists (ECS) within W2W for making the project a worthwhile experience.

## *Acknowledgments*

Next to this, I am grateful to acknowledge funding by the Munich Center for Machine Learning (MCML) initiated by the Federal Ministry of Education and Research and the State of Bavaria.

On a more personal note, I would like to thank all my colleagues and friends at the chair for computer graphics and visualization at the TUM. Special thanks go to Prof. Dr. Nils Thuerey for mentoring my doctoral project, Susanne Weitz for her continuous support in administrative and organizational matters, Sebastian Wohner for his technical expertise and fun conversations, and all my colleagues and friends who took the time to proofread my doctoral thesis – even on rather short notice. Also, I would like to thank all the students who decided to pursue their thesis and guided research projects under my supervision.

Beyond that, I want to thank my previous colleagues Dr. Dejan Azinović, Dr. Shuvayan Brahmachary, Manuel Dahnert, Dr. Marie-Lena Eckert, Steffen Eckert-Wiewel, Behdad Ghaffari, Dr. Mathias Kanzler, Felix Köhler, Dr. Alexander Kumpf, Han Liang, Björn List, Simon Niedermayr, Dr. Lukas Prantl, Dr. Christian Reinbold, Patrick Schnell, Dr. Junpeng Wang, Dr. Rene Winchenbach, and Dequan Yang, as well as all (former) chair members listed as coauthors. I am thankful for a wonderful and exciting time at the chair, informative and entertaining discussions, and welcome diversions when needed.

Outside of university, I want to thank my friends Dr. Christian Vock, Johannes Harth-Kitzerow, and Nina Wenke for joint activities, valuable discussions on all sorts of matters.

Finally, I thank my family for their constant support throughout my life, during my undergraduate studies and doctoral studies.



---

# Contents

---

<b>Abstract</b>	<b>iii</b>
<b>Zusammenfassung</b>	<b>v</b>
<b>Acknowledgments</b>	<b>vii</b>
<b>Acronyms</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Contributions . . . . .	4
1.2 Outline . . . . .	6
1.3 List of Publications . . . . .	7
<b>2 Atmospheric Dynamics and Weather Forecasting</b>	<b>9</b>
2.1 Physics of the Atmosphere . . . . .	9
2.1.1 Fluid and Thermodynamics . . . . .	9
2.1.2 Approximations and Alternative Parametrizations . . . . .	10
2.1.3 Earth System and Planetary Boundary Layer . . . . .	13
2.2 Probabilistic Weather Forecasting and Postprocessing . . . . .	14
2.2.1 Probabilistic Weather Forecasting . . . . .	15
2.2.2 Ensemble Forecasting . . . . .	15
2.2.3 Verifying Probabilistic Forecasts . . . . .	15
2.2.4 Postprocessing . . . . .	19
2.3 Properties of Numerical Weather Data . . . . .	21
2.3.1 Grid-Based Representation of Spatial Fields . . . . .	22
2.3.2 Multivariate Structure of Numerical Weather Data . . . . .	24
2.3.3 Permutation Symmetry in Ensemble Data . . . . .	26
2.4 Visualizing Numerical Weather Data . . . . .	26
2.4.1 Map-Based Visualization . . . . .	27
2.4.2 3D Visualization . . . . .	28

2.4.3	Topographic Visualization . . . . .	28
<b>3</b>	<b>Deep Learning</b>	<b>31</b>
3.1	Fundamentals of Deep Learning . . . . .	31
3.1.1	Neural Networks . . . . .	31
3.1.2	Modeling Non-Deterministic Functions . . . . .	33
3.1.3	Model Training and Hyperparameter Selection . . . . .	34
3.1.4	Training Objectives . . . . .	35
3.2	Neural Network Architectures . . . . .	37
3.2.1	Fully-Connected Networks and the Multi-Layer Perceptron . . . . .	37
3.2.2	Coordinate-Based Networks . . . . .	38
3.2.3	Convolutional Neural Networks . . . . .	40
3.2.4	Shortcut Connections . . . . .	44
3.2.5	Attention and Transformers . . . . .	45
3.2.6	Neural Networks for Learning from Sets . . . . .	46
3.3	Machine Learning Explainability . . . . .	48
3.4	Deep Learning in Computer Vision . . . . .	49
3.4.1	Superresolution . . . . .	49
3.4.2	Neural Scene Representations . . . . .	50
<b>4</b>	<b>Related Work</b>	<b>53</b>
4.1	Superresolution . . . . .	53
4.1.1	Image Superresolution . . . . .	53
4.1.2	Superresolution for Scientific Data . . . . .	55
4.2	Neural Representations for Scenes and Scientific Data . . . . .	56
4.3	Postprocessing . . . . .	59
4.3.1	Statistical Downscaling . . . . .	59
4.3.2	Ensemble Postprocessing . . . . .	60
4.4	Visual Analysis of Meteorological Data . . . . .	62
4.4.1	Visualizing Weather Data in Their Spatial Context . . . . .	62
4.4.2	Visual Analysis of Ensemble Datasets . . . . .	63
4.4.3	Machine Learning Explanation for Ensemble-Valued Predictors . . . . .	64
<b>5</b>	<b>Publication Summaries</b>	<b>65</b>
5.1	Paper I – A Comparative Study of Convolutional Neural Network Models for Wind Field Downscaling . . . . .	67
5.2	Paper II – Postprocessing of Ensemble Weather Forecasts Using Permutation-Invariant Neural Networks . . . . .	69
5.3	Paper III – Topographic Visualization of Near-Surface Temperatures for Improved Lapse Rate Estimation . . . . .	71
5.4	Paper IV – Evaluation of Volume Representation Networks for Meteorological Ensemble Compression . . . . .	73
5.5	Paper V – Neural Fields for Interactive Visualization of Statistical Dependencies in 3D Simulation Ensembles . . . . .	75

<b>6 Discussion and Future Work</b>	<b>77</b>
6.1 Discussion . . . . .	77
6.1.1 Impact of Network Architectures on Model Performance . . . . .	77
6.1.2 Statistical Reliability Assessment . . . . .	78
6.1.3 Datasets and Model Comparisons . . . . .	79
6.1.4 Modeling Uncertainties . . . . .	79
6.2 Future Work . . . . .	80
6.2.1 Improving the Proposed Models . . . . .	80
6.2.2 Understanding the Information Content of Ensemble Forecasts . . . . .	81
6.2.3 Adopting Foundation Models . . . . .	82
6.2.4 Deep Learning-Based Weather Prediction . . . . .	82
<b>7 Conclusion</b>	<b>85</b>
<b>Bibliography</b>	<b>87</b>
<b>Paper I</b>	<b>119</b>
<b>Paper II</b>	<b>153</b>
<b>Paper III</b>	<b>195</b>
<b>Paper IV</b>	<b>209</b>
<b>Paper V</b>	<b>221</b>



---

## Acronyms

---

CBN	coordinate-based network.
CDF	cumulative distribution function.
CNN	convolutional neural network.
CRPS	continuous ranked probability score.
DGM	deep generative model.
DRN	distribution regression network.
DVR	direct volume rendering.
ERA5	fifth-generation atmospheric reanalysis dataset.
ERN	ensemble representation network.
FCN	fully-connected neural network.
GAN	generative adversarial network.
HR	high-resolution.
ICAO	International Civil Aviation Organization.
iLR	interpolated low resolution.
ISA	international standard atmosphere.
ISR	image superresolution.
LAM	limited area model.
LFN	light field network.
LIC	line integral convolution.
LR	low-resolution.
LRP	layer-wise relevance propagation.

## Acronyms

MAE	mean absolute error.
MAP	maximum a posteriori.
MI	mutual information.
MLE	maximum likelihood estimate.
MLP	multi-layer perceptron.
MSE	mean square error.
NDF	neural dependence fields.
NeRF	neural radiance field.
NN	neural network.
NWP	numerical weather prediction.
OSE	optimum score estimation.
PBL	planetary boundary layer.
PDF	probability density function.
PFI	permutation feature importance.
PI	prediction interval.
PIT	probability integral transform.
PSNR	peak signal-to-noise ratio.
RCM	regional climate model.
RGG	reduced Gaussian grid.
SciSR	superresolution for scientific data.
SISR	single-image superresolution.
SR	superresolution.
SRN	scene representation network.
SSIM	structural similarity metric.
uPIT	unified probability integral transform.
VRN	volume representation network.

# CHAPTER 1

---

## Introduction

---

Weather prediction is inherently a data-driven science. Vast amounts of observation data from weather stations, radiosondes, satellites, radar stations, buoys, and airplane- and vessel-mounted sensors are collected each day to probe the current state of the atmosphere and enable informed projections of what the weather will be like – most likely and subject to inherent uncertainties. Probabilistic future projections are based on weather prediction models that simulate the evolution of the atmosphere and provide insights into weather trends and expected error margins. Weather forecasts are issued daily by national and international weather centers, such as the German Weather Service (Deutscher Wetterdienst; DWD) and the European Centre for Medium-Range Weather Forecasts (ECMWF), covering forecast lead times between a few hours and up to several weeks. Sustaining and extending the current level of forecast skill requires continuous improvements to the quality of the prediction models and the analytical methods used in the evaluation of forecast datasets.

As of today, operational forecasting systems rely on a forecasting pipeline that combines statistical data assimilation techniques with numerical weather prediction (NWP) models and postprocessing methods to optimize the forecast quality. While data assimilation uses statistical techniques and comprehensive observation datasets to assess the current state of the atmosphere, NWP models apply the laws of physics to project the available information into the future. NWP models operate on discretized grid representations of the atmosphere and use numerical integration schemes to simulate the atmospheric dynamics. Monte Carlo methods provide probabilistic forecast information and account for uncertainties about the current state of the atmosphere and its future dynamics. The resulting *ensemble forecasts* represent multi-samples of possible future weather scenarios, allowing insights into the respective likelihoods and interrelations of different events. Postprocessing of the model outputs is required to correct model-specific prediction biases and miscalibrations of the simulation results.

## 1 Introduction

Limitations of operational forecasting systems arise from the interplay between forecast quality, timeliness constraints, and the computational complexity of NWP models. Among the key determinants of achievable forecast quality are the spatial resolution of the model’s simulation grid and the number of independent runs considered in the ensemble (cf., e.g., Palmer 2019). More fine-granular models achieve potentially better representations of small-scale dynamical effects, and larger ensemble sizes facilitate more accurate estimates of event probabilities, especially for weather extremes (e.g., Tempest et al. 2023). However, both come at the cost of larger computational simulation complexity, longer computation times, and increased volume of the generated data. As a compromise between computational complexity and forecast quality, forecasting systems such as the Integrated Forecast System (IFS) at the ECMWF use supercomputing hardware to run global NWP models with kilometer-scale resolution and ensemble sizes of around 100 runs. While the generated data volume amounts to tera- and petabytes each day, the resolution is still too coarse to account for local-scale physical effects below the grid resolution or provide useful forecast information on sub-grid scales. Accordingly, postprocessing methods aim to refine the spatial resolution of forecast products and correct for biases in the predictions due to, e.g., misrepresentations of sub-grid physical effects (e.g., Hewitson and Crane 1996; Vannitsem et al. 2021; Wilby and Wigley 1997). Postprocessing methods with a focus on spatial refinement are called *downscaling methods* and are especially helpful for forecasting atmospheric variables close to the Earth’s surface. *Ensemble postprocessing*, additionally, addresses the calibration and sharpness of ensemble forecasts.

Regarding forecast analysis, challenges arise from the inherent complexity of the generated forecast data. NWP datasets are defined on multi-dimensional simulation domains, may show temporal variability, combine information from multiple different physical parameters, and involve a stochastic component due to the use of Monte Carlo procedures. Notably, ensemble forecasts are stochastic representations of high-dimensional forecast distributions. Spatiotemporal correlation patterns and interrelations of physical variables give rise to intricate interdependencies, which may affect the statistical interpretation of forecasts. For instance, the occurrence of high precipitation in a few simulation runs may provoke limited interest or only careful reactions by forecasters when it is confined to a small area but may justify flood warnings when the extremes extend coherently across a wider region. Similarly, the ability to study interrelations between different kinds of data is a key requirement for improving the prediction quality of existing forecasting systems. Statistical analysis methods based on historical data and interactive visual exploration methods for forecast datasets play a central role in recognizing relevant patterns. The increasing availability and volume of NWP data offer diverse opportunities for new insights but keep pushing existing data storage and analysis solutions to their technical limits. Concerning data storage, for example, common data compression techniques (e.g., Ballester-Ripoll et al. 2020; Di and Cappello 2016; Düben et al. 2019; Lindstrom 2014) enable reductions of the data’s memory footprint but often discard parts of the information and cause compression artifacts (cf., e.g., Baker et al. 2017; Cappello et al. 2019). In visual analysis settings, additionally, extensive decompression or data loading times impede interactive analysis workflows.



In light of the central importance of data in NWP, data-driven modeling and analysis methods offer a wide range of opportunities to improve the achievable forecast quality, computational efficiency, and interpretability of NWP systems. Across a wide range of research domains, developments in computational technology have advanced the abilities of researchers to process and analyze large datasets. Scientific models and analysis procedures are increasingly augmented with data-driven model components using automated machine learning (ML) methods and deep learning (DL). Meteorology and weather forecasting are immaculate examples of these developments. ML and DL techniques are adopted at an increasing pace (see, e.g., Ben-Bouallègue et al. 2024; Dueben et al. 2022; Reichstein et al. 2019) and yield new insights and research methodologies.

Therein, DL has emerged as a particularly powerful and flexible modeling approach, often exceeding the capabilities of more classical ML methods. DL models rely on artificial neural networks (NNs), which are trained to learn functional mappings from comprehensive datasets using scalable gradient-based optimization schemes. The flexibility of DL models is due to a variety of design options and NN *architectures*, i.e., neuron layouts and connectivity structures, that allow adapting the models precisely to the requirements of different learning tasks. Examples of common NN architectures include fully-connected networks (FCNs; e.g., Rosenblatt 1958), which can learn from vector-valued or tabular data inputs, convolutional neural networks (CNNs; e.g., Fukushima 1988; LeCun et al. 1998) for image-like data or data on spatially distributed grids, and transformers (Vaswani et al. 2017) for sequentially ordered or unordered inputs, such as text and set-structured data. Following applications in computer vision (e.g., Krizhevsky et al. 2012), medical imaging (e.g., Ronneberger et al. 2015), and natural language processing (e.g., Radford et al. 2018), DL methods have flourished also in the natural sciences (e.g., Jumper et al. 2021; Kochkov et al. 2024; Reichstein et al. 2019) – including meteorology and weather prediction.

Data-driven learning algorithms promise improvements in various steps of the forecasting pipeline. In postprocessing applications, ML and DL methods offer flexible modeling capabilities that allow the exploration of new data types and sources that were inaccessible to earlier statistical or physics-based approaches (e.g., Hewitson and Crane 1996; Vannitsem et al. 2021; Wilby and Wigley 1997). In applications concerning the evaluation and analysis of probabilistic NWP datasets, ML-based feature extraction methods can help to distill relevant information from large datasets and alleviate complexity and storage constraints (cf., e.g., Han et al. 2021; Lu et al. 2021; Weiss et al. 2022).

A strong benefit of ML and DL techniques is their adaptability to diverse learning tasks and their wide adoption across diverse research domains. Several research questions in meteorology resemble inference problems in other scientific domains. Examples include, e.g., the resolution enhancement of images in computer vision and downscaling of weather data or the representation of 3D scenery information in computer graphics and 3D simulated physical variable fields in meteorology. These interdisciplinary connections give rise to new research opportunities in transferring modeling approaches between scientific domains and help accelerate scientific progress. ML and DL methods developed in computer vision or visualization research can be applied to modeling tasks in meteorology or motivate the development of inherently new data-driven solutions. Research

is required to understand the scientific background of the respective applications and adapt the modeling approaches accordingly. Next to plain model development, statistical analysis and visualization techniques developed in computer vision and visualization research can help to better understand the inference process of data-driven learning models and foster their adoption in meteorology. Research questions for such studies arise from the peculiarities of meteorological modeling tasks due to, e.g., the required domain expertise and the dependence on unique data formats, such as simulation data on NWP model grids and forecast ensembles.

### 1.1 Contributions

In this publication-based dissertation, we compile research results from five articles with a focus on ML and DL applications in meteorology. The work presented in this thesis explores the use of data-driven modeling techniques for postprocessing, analyzing, and compressing the outputs of NWP models. A specific focus is put on the design and validation of adequate DL model architectures to meet the requirements and quality criteria relevant to the respective applications. In close collaboration with domain scientists, the proposed DL models are assessed critically regarding their performance advantage over more classical approaches and the interpretability of models' predictions. Statistically founded and visualization-based analysis methods are presented, which enable the investigation of model predictions and their dependence on the raw NWP data. The studies address two distinct lines of research.

Three of the presented contributions focus on ML- and DL-based statistical postprocessing models that target the downscaling of gridded forecast fields in complex terrain and the postprocessing of ensemble forecasts:

- In Höhle et al. (2020), we examine the use of CNNs for downscaling forecasts of low-level wind fields on extended spatial domains. Four exemplary CNN architectures and a multilinear regression model are compared concerning downscaling accuracy and adaptability of the models to additional predictor variables. Using architectural optimizations, a skillful novel CNN architecture, DeepRU, is developed, which outperforms the remaining models in terms of downscaling quality while maintaining a low computational footprint. Feature relevance metrics are computed to explore how different model architectures combine the information from different sources of predictive information.
- In Höhle et al. (2024b), we design DL models for ensemble postprocessing that operate natively on ensemble-valued input data. Our ensemble-focused approach contrasts previous methods, which often operate on simple ensemble summary statistics and thus discard distribution details before the model has even seen the data. Different model designs are compared with respect to the achievable quality of probabilistic forecasts. Case studies in postprocessing wind gust and surface temperature predictions illustrate the utility of our approach. Additionally, we

present a feature importance analysis for ensemble-valued predictors that generalizes a previous model explanation technique for scalar-valued features. The novel method highlights specific features in the ensemble that are considered important by the trained postprocessing models. Our analysis reveals that most of the relevant information is contained in only a few ensemble-internal degrees of freedom, spawning new research questions on the capabilities and limitations of ensemble forecasts.

- In Höhle et al. (2024a), we develop a simple downscaling scheme for near-surface temperatures in complex terrain. Based on gridded temperature records and elevation information, the method provides case-specific information about local temperature variations due to altitude differences. The information is used in a simple physics-inspired downscaling scheme that corrects temperature predictions for elevation difference effects. To assess forecast improvements and hyperparameter dependencies, the technique is embedded into a 3D topographic visualization system. The tool enables surface- and volume-based visualizations of near-surface temperature predictions and observations in the context of the surrounding terrain. Selected case studies identify and illustrate topographic dependencies of prediction errors. *The results are unpublished and not peer-reviewed.*

The remaining projects address the visual analysis and compressed representation of ensemble forecast datasets using a combination of DL and visualization techniques:

- In Höhle et al. (2022), we evaluate the utility of NN-based data representations for compressing meteorological ensemble data on 3D simulation domains. Inspired by scene representation networks (SRNs) in computer vision and volume representation networks (VRNs) in visualization research, NNs are used as a compressive data structure to encode volumetric ensemble information. The proposed ensemble representation networks (ERNs) exploit similarities between the ensemble members by sharing NN parameters between them, yielding competitive compression ratios relative to traditional compression algorithms. Since meteorological ensembles contain multiple physical parameters with different statistical characteristics, we analyze the impact of data normalization schemes on the representation quality.
- In Farokhmanesh et al. (2023b), we present neural dependence fields (NDFs) to facilitate interactive analysis of bivariate correlation and interdependence relationships in large ensemble datasets. NDFs constitute a specialized NN-based data structure, which stores two-point correlation and interdependence maps in 3D simulation ensembles. NDFs circumvent the evaluation of compute-intensive statistical summary statistics at runtime by precomputing the required values and storing them. The compactness of the representation together with its ability to efficiently provide data access at random locations facilitate the integration of NDFs into an interactive visualization tool.

## **1.2 Outline**

Chapters 2 and 3 introduce fundamentals and relevant background information that enable a detailed understanding of the presented methods. Emphasis is put on the fundamentals of atmospheric dynamics and weather forecasting (chapter 2) as well as on DL methods and their applications in computer vision (chapter 3). The fundamentals of DL are followed by a more detailed discussion of NN architectures and the principles of DL model design, a brief overview of DL model explanation techniques, and DL applications in computer vision that inspired parts of this work. Chapter 4 lists and discusses related work, and chapter 5 provides summaries of the presented articles and original contributions of this thesis. Chapters 6 and 7 conclude the thesis with final discussions and an outlook on future research opportunities. The original publications with license information and supplemental materials are shown in the appendix.

## 1.3 List of Publications

Large parts of the work in this thesis have resulted in peer-reviewed journal articles and conference proceedings. This section lists the publication details and states the articles' relevance for assessing the thesis. Summaries of the contents and author contributions in examination-relevant and additional publications are provided in chapter 5.

**Core publications** The following peer-reviewed journal and conference publications form the core of this thesis and are the most relevant for the examination. The thesis author acted as the lead author of these publications ( $> 50\%$  contribution).

- Kevin Höhlein et al. (2020). “A comparative study of convolutional neural network models for wind field downscaling”. In: *Meteorological Applications* 27.6, e1961.
- Kevin Höhlein et al. (2022). “Evaluation of Volume Representation Networks for Meteorological Ensemble Compression”. In: *Vision, Modeling, and Visualization (VMV 2022)*. The Eurographics Association.
- Kevin Höhlein et al. (2024b). “Postprocessing of Ensemble Weather Forecasts Using Permutation-Invariant Neural Networks”. In: *Artificial Intelligence for the Earth Systems* 3.1, e230070.  
© American Meteorological Society. Used with permission.

**Coauthor publication** The following publication is presented as part of this thesis and is relevant for the examination. The thesis author contributed significant parts to the publication but did not act as the lead author ( $\leq 50\%$  contribution).

- Fatemeh Farokhmanesh et al. (2023b). “Neural Fields for Interactive Visualization of Statistical Dependencies in 3D Simulation Ensembles”. In: *Vision, Modeling, and Visualization (VMV 2023)*. The Eurographics Association;  
**Coauthor contribution.**

**Unpublished work** Contributions from the following unpublished article are presented in this thesis but should not be considered in the examination.

- Kevin Höhlein et al. (2024a). *Topographic Visualization of Near-surface Temperatures for Improved Lapse Rate Estimation*. arXiv: 2406.11894 [physics.ao-ph].  
**Unpublished work, not peer-reviewed, not relevant for the examination.**



---

## Atmospheric Dynamics and Weather Forecasting

---

This section provides an overview of the fundamentals of atmospheric dynamics and weather prediction, introduces meteorology-related terminology, and locates the contributions of the presented publications within the context of weather prediction research.

### 2.1 Physics of the Atmosphere

The atmosphere is a complex dynamical system that evolves according to the laws of physics and chemistry. Current NWP models distinguish between two sorts of atmospheric dynamics: phenomena that can be resolved by a (simplified) fluid and thermodynamic model of the atmosphere, also referred to as *dynamics*, and phenomena that cannot be covered by such models due to insufficient model resolution or non-fluid-dynamical characteristics. The latter are referred to as *physics* (cf. Gross et al. 2018; Inness and Dorling 2012) and must be captured through suitable approximations and parameterization schemes (e.g., Palmer et al. 2009). The subsequent sections introduce the physical background of both kinds of effects and highlight selected aspects that are relevant to the presented studies.

#### 2.1.1 Fluid and Thermodynamics

A large part of atmospheric physics is understood by modeling the atmosphere as a fluid system in 3D space and time, evolving according to the laws of hydro- and thermodynamics. A (macroscopic) dynamical state of the atmosphere is determined through temperature, pressure, flow velocity (i.e., wind speeds), and mass densities for all relevant components of the atmosphere. The variables are commonly denoted as  $T$ ,  $p$ ,  $\mathbf{u}$ , and  $\rho$ , respectively, and are understood as fields in a continuous theory of fluid and thermodynamics. For a single gaseous component of the atmosphere, such as air or water

## 2 Atmospheric Dynamics and Weather Forecasting

vapor, the fundamental equations are derived as partial differential equations from a small number of physical principles (cf. Vallis 2017), namely:

- momentum conservation:

$$\frac{D\mathbf{u}}{Dt} = -\frac{1}{\rho}\nabla p + \frac{\mu}{\rho}\Delta\mathbf{u} + \mathbf{F}_{\text{ext}}, \quad (2.1)$$

wherein  $\mu$  is the viscosity and  $\mathbf{F}_{\text{ext}}$  denotes external forces (per unit mass), such as gravity or drag forces, but also includes the Coriolis and centrifugal forces arising from solving the equations in a rotating reference frame,

- mass conservation:

$$\frac{D\rho}{Dt} + \rho\nabla\mathbf{u} = q_{\text{ext}}, \quad (2.2)$$

with  $q_{\text{ext}}$  denoting mass source and sink terms.

- the first law of thermodynamics:

$$\dot{Q}_{\text{ext}} = c_p \frac{DT}{Dt} - \frac{1}{\rho} \frac{Dp}{Dt}, \quad (2.3)$$

with  $c_p$  denoting specific heat at constant pressure, and  $\dot{Q}_{\text{ext}}$  measuring heat fluxes in and out of the air parcel due to external heating and cooling,

- and the ideal gas law:

$$p = \rho R_s T, \quad (2.4)$$

wherein  $R_s$  is the specific gas constant of the component.

In all of the above equations,  $t$  indicates time, and  $\nabla$  denotes the gradient operator with respect to spatial coordinates (in an inertial frame of reference).  $\Delta$  denotes the Laplace operator, and  $\frac{Df}{Dt} = \frac{\partial f}{\partial t} + (\mathbf{u} \cdot \nabla)f$  is the material time derivative of fields  $f$ . Respecting the spherical symmetry of the Earth, it is common to transform the equations to spherical coordinates. The spherical form of Equations 2.1 to 2.4 form the backbone of operational weather prediction models. For more details, we refer to dedicated textbooks such as Holton (2013) or Vallis (2017).

### 2.1.2 Approximations and Alternative Parametrizations

Solving the equations jointly for all components of the atmosphere generally requires numerical integration procedures. However, qualitative insights on generic dynamical patterns can be obtained from simple approximations, which enable an intuitive understanding of weather situations without invoking costly simulations. Three common approximations are introduced here that address the pressure distribution in the atmosphere, approximate the wind direction based on known pressure fields, and provide an explanation for the temperature distribution in the atmosphere.



**Vertical coordinates and hydrostatic balance** The simplest possible approximation of atmospheric conditions is a static, i.e., time-invariant, atmosphere that is symmetric under rotations around the Earth's rotational axis. In absence of atmospheric motion, Equation 2.1 states that the vertical pressure gradient must balance gravity, i.e.,

$$\frac{\partial p}{\partial z} = -\rho g, \quad (2.5)$$

wherein  $z$  denotes the vertical coordinate, and  $g = g(z)$  is the gravitational acceleration at height  $z$ . Equation 2.5 is called the *hydrostatic balance* condition and is often a reasonable approximation of the true atmospheric conditions, even in the non-static case.

Due to  $\rho, g > 0$ ,  $p$  decreases monotonically with  $z$ . This fact and the close connection between  $p$  and the remaining state variables motivates the use of *pressure coordinates*, which measure the vertical elevation in terms of pressure  $p$  instead of  $z$ . Replacing  $z$  with  $p$  offers advantages in relating spatiotemporal dynamical equations to the laws of thermodynamics. A related quantity is the *geopotential*,

$$\begin{aligned} \Phi &= \int_{z_0}^z g(z') dz', \\ &= -R_s \int_{p(z_0)}^{p(z)} \frac{T(p')}{p'} dp', \end{aligned}$$

with  $z_0$  denoting the height of the mean sea level. The second equality is obtained by using Equation 2.4. The geopotential relates pressure and elevation coordinate through the *geopotential height*,  $Z = \Phi/g(z_0)$ , which measures vertical differences in terms of the gravitational energy required to elevate a unit mass object by the respective amount. Surfaces of constant geopotential are considered as horizontal in terms of dynamical motion and are perpendicular to the direction of gravity. In particular, geopotential heights on pressure isolevels, such as 500 hPa, are important diagnostics for weather forecasting and flow inference.

**Geostrophic balance and geostrophic winds** In the upper layers of the atmosphere, and far enough from the equator, the dominating forces in Equation 2.1 are the pressure force, gravity, and the Coriolis force, as drag forces are often negligible (Vallis 2017). A steady state solution in the presence of spatial pressure variations amounts to solving Equation 2.1 for  $\frac{D\mathbf{u}}{Dt} = 0$  in a rotating reference frame. The resulting conditions are called *geostrophic balance*.

The wind field in geostrophic balance is called *geostrophic wind* and is oriented *parallel* to isolines of the pressure field, i.e., *perpendicular* to pressure gradients. Since pressure isocontours are commonly shown in meteorological charts and geostrophic balance is often satisfied approximately in mid-latitude regions, geostrophic winds constitute an easily accessible approximation to the true wind field. In hydrostatic balance, the geostrophic wind field can also be expressed in pressure coordinates. The winds are then directed parallel to the isolines of the geopotential height  $Z$  at constant pressure (Holton 2013).

**Temperature variation with height and atmospheric stability** The temperature distribution in the atmosphere is determined jointly by Equations 2.3 and 2.4 (Vallis 2017). Due to potential condensation effects, the air-water content affects temperature variations considerably. In unsaturated air and hydrostatic balance, the equations predict a linear temperature decrease with height at a rate of

$$\frac{dT}{dz} = -\Gamma_{\text{dry}}, \quad \text{where } \Gamma_{\text{dry}} := \frac{g}{c_p}. \quad (2.6)$$

The quantity  $\Gamma_{\text{dry}}$  is called the *dry adiabatic lapse rate* and has a value of  $\Gamma_{\text{dry}} = 9.8 \text{ K km}^{-1}$  for dry air in standard gravity<sup>1</sup>. When air becomes cooler, its ability to hold water vapor decreases until *saturation* is reached and condensation effects start to set in. Beyond this point, the heat release due to condensation decreases the cooling rate. The *moist adiabatic lapse rate*,  $\Gamma_{\text{moist}}$ , accounts for condensation and has values, typically, between  $3 \text{ K km}^{-1}$  and  $6 \text{ K km}^{-1}$ , depending on ambient temperature and pressure.

The *environmental lapse rate*,  $\Gamma_{\text{env}}$ , describes the observed rate of temperature decrease in real weather situations and can be used to assess the dynamical stability of a weather situation. Depending on how  $\Gamma_{\text{env}}$  compares to  $\Gamma_{\text{dry}}$  and  $\Gamma_{\text{moist}}$ , the atmosphere can be in three configurations:

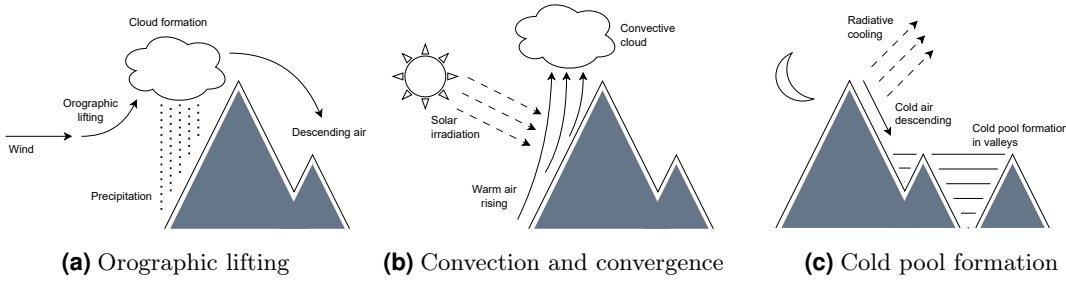
- $\Gamma_{\text{env}} < \Gamma_{\text{moist}}$ : The weather situation is *stable*. Air parcels that rise in the surrounding air cool faster than their environment, resulting in a decelerating force that counteracts further ascent.
- $\Gamma_{\text{moist}} \leq \Gamma_{\text{env}} \leq \Gamma_{\text{dry}}$ : The weather situation is *conditionally unstable*. Unsaturated air masses cool faster than the environment and remain stable. Yet, in saturated air, the heat released due to condensation pushes the air beyond the stability barrier and accelerates the ascend of air parcels.
- $\Gamma_{\text{dry}} < \Gamma_{\text{env}}$ : The weather situation is *unstable*. Air parcels experience an accelerating force when displaced vertically, leading to convective air movements.

The International Civil Aviation Organization (ICAO) defines an *international standard atmosphere* (ISA) as a simplified model of common atmospheric conditions. The ISA assumes a value of  $\Gamma_{\text{env}} = 6.5 \text{ K km}^{-1}$  for the troposphere between mean sea level and 11 km above (International Civil Aviation Organization 1993).

The vertical temperature variation in the free air directly affects temperature values in the vicinity of the Earth's surface. Surface temperatures are among the most important forecast variables in operational weather predictions and exhibit a strong dependence on the reference altitude for which they are computed. In complex terrain, where details of the terrain shape remain unresolved by coarse grid representations in NWP models, this may lead to systematic but dynamically changing prediction errors, depending on the stability conditions in the ambient atmosphere. Prediction errors of this type are addressed by the methods in H ohlein et al. (2024a).

---

<sup>1</sup>Gravity varies only marginally over the relevant elevation differences of a few 100 m to 1000 m and is assumed constant here.



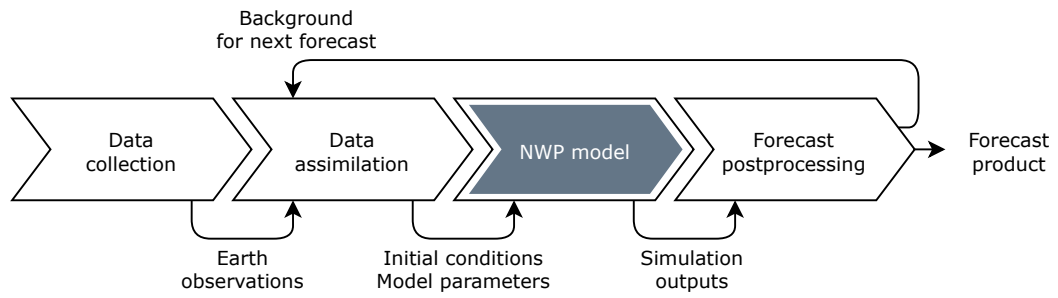
**Figure 2.1:** Simplified examples of orography-mediated interactions between boundary layer atmosphere and Earth's surface.

### 2.1.3 Earth System and Planetary Boundary Layer

In addition to its intrinsic fluid and thermodynamics, the atmosphere is tightly coupled to other Earth system components. In particular, it is driven by solar radiation (captured in  $\dot{Q}_{\text{ext}}$  in Equation 2.3; for details, see Wallace and Hobbs 2006). The energy intake from solar irradiation causes water evaporation and heating of near-surface air, which are precursors, e.g., of convective cloud formation processes.

**The planetary boundary layer** The *planetary boundary layer* (PBL) describes the lowest part of the atmosphere that is in direct contact with the Earth's surface. It takes the role of a regulator of heat and momentum fluxes between the bulk atmosphere and the Earth's surface (Baklanov et al. 2011). Diurnal cycles and details of the Earth's surface, such as local topography or land usage, are important impact factors of the prevalent interaction processes (Foken and Mauder 2008). PBL physics are particularly difficult to express in NWP model equations due to the wealth of relevant phenomena and their dependence on high-resolution terrain and land cover details. Accordingly, PBL physics are a major source of uncertainties and prediction errors in NWP.

**Impact of complex orography** Of particular interest in the context of this work are PBL phenomena due to variations in the local terrain shape, also referred to as *orography*. In areas with complex orography, such as mountain ranges, terrain elevations obstruct movements of air masses and deflect airflow. Depending on the spatial extent of the obstruction, this may lead to the formation of local-scale turbulent wind systems that deviate strongly from the ambient flow and, e.g., geostrophic winds. Figure 2.1 displays a schematic overview of the orography-mediated boundary layer effects. Larger-scale phenomena include orographic lifting situations, in which air masses are forced to ascend above obstructions (Figure 2.1, a). Orographic lifting frequently causes cloud formation and orographically triggered precipitation on the windward side of the obstacle. On the opposite side, it leads to oscillatory vertical air movements and an increase in the air temperature due to latent heat from condensation. Additionally, mountainsides with steep slopes may heat up quicker under solar irradiation than the surroundings, thus leading to unstable atmospheric conditions and potentially triggering the spatially



**Figure 2.2:** Schematic overview of a typical workflow for operational weather forecasting (adapted from Düben et al. 2021). Arrows and labels indicate the direction and type of data transmitted between the sub-steps. The NWP model, forming the core of the process, is highlighted.

confined ascend of warm air masses, i.e., convection (Figure 2.1, b). Inversely, cold air masses may accumulate in topographic depressions, e.g., due to night-time radiative cooling of near-surface air (Figure 2.1, c). Cold air pools in valleys are characterized by a low environmental lapse rate or even an *inversion* of the temperature profile, i.e., an increase of temperature with height. For a more detailed discussion of meteorological phenomena related to boundary-layer effects, see, e.g., Foken and Mauder (2008).

In NWP simulations, PBL physics are approximated through *physics parameterizations*, which emulate (the effect of) the unresolved processes through simplified (stochastic) equations (Berner et al. 2017). Downscaling and postprocessing methods, as discussed in Höhle et al. (2020) and Höhle et al. (2024b), address the resulting prediction errors in a post-hoc step.

## 2.2 Probabilistic Weather Forecasting and Postprocessing

Figure 2.2 displays an overview of a typical forecasting pipeline commonly applied in operational weather forecasting. An NWP model, forming the backbone of the pipeline, starts from information about the current state of the atmosphere and applies numerical integration methods to simulate its spatiotemporal dynamics. In addition to the NWP model, the workflow includes steps for probing the atmosphere’s current state through observations (data collection), providing initial conditions for the NWP model (data assimilation), and postprocessing its outputs to correct for potential deficiencies (forecast postprocessing).

The contributions presented in this thesis address the postprocessing of model outputs (Höhle et al. 2024a, 2020, 2024b) and the subsequent analysis and representation of probabilistic forecast data (Farokhmanesh et al. 2023b; Höhle et al. 2022). This section introduces the required background and formalism behind the proposed methods.

### 2.2.1 Probabilistic Weather Forecasting

Probabilistic weather forecasts account for prediction uncertainties by considering initial and future weather conditions as random variables. The probabilistic formalism enables a statistical assessment of the likelihood of possible future weather scenarios. Formally, a *probabilistic prediction* for a future weather state  $Y$  is issued as a probability distribution  $\hat{\mathcal{P}}$  on the space  $\mathcal{Y}$  of possible weather scenarios.

The distribution information can be conveyed in different forms depending of the type of the target variable. For binary prediction targets – i.e.,  $\mathcal{Y} = \{0, 1\}$  – such as the occurrence of rain or no rain, a prediction can be identified with a scalar probability value  $\pi \in [0, 1]$  of observing one of the outcomes. For real-valued targets – i.e.,  $\mathcal{Y} = \mathbb{R}$  – the prediction distribution can be expressed through its cumulative distribution function (CDF) or, if it is defined, its probability density function (PDF).

### 2.2.2 Ensemble Forecasting

The most common approach for probabilistic forecasting of high-dimensional target variables is *ensemble forecasting*. An ensemble forecast is generated by sampling a set of initial conditions consistent with the observed weather conditions and using different model configurations to propagate these samples independently over time. The result of this process is a set of estimates of plausible future weather conditions. The sample set is called *ensemble* and serves as a stochastic representation of the forecast distribution  $\hat{\mathcal{P}}$ . Each sample forecast separately is called an *ensemble member*.

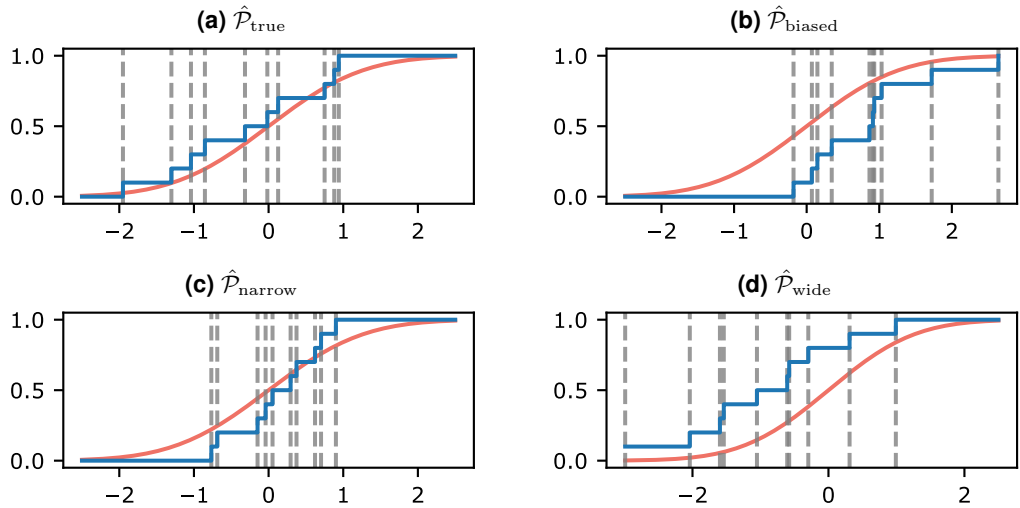
For real-valued forecast targets, an  $M$ -member ensemble forecast  $\hat{\mathcal{E}}_M$  is a set of  $M$  real-valued member forecasts, i.e.,

$$\hat{\mathcal{E}}_M = \{\hat{y}_m \in \mathbb{R} : m \in \{1, \dots, M\}\}, \quad (2.7)$$

with  $\hat{y}_m \in \{1, \dots, M\}$ . The corresponding forecast distribution is best described in terms of its CDF, which is a step function with step increments  $\frac{1}{M}$  at every member location  $\hat{y}_m$ . Figure 2.3 illustrates this for an example with normal distributions for the variable and ensemble members. Failure modes of probabilistic forecasts include the presence of biases (shift between the CDFs) and the over- or underestimation of expected variability of the forecast quantities (differences in the slope of the CDFs).

### 2.2.3 Verifying Probabilistic Forecasts

The assessment of the predictive quality of probabilistic forecasting methods is called *forecast verification* (Jolliffe and Stephenson 2012). The skill of a probabilistic prediction system is determined by the accuracy of the predicted values and the statistical consistency of the predicted distribution with that of the observed weather states. Gneiting and coauthors (Gneiting et al. 2007; Gneiting and Raftery 2005) refer to these aspects as *sharpness* and *calibration*, and claim that an optimal forecasting system should maximize sharpness subject to calibration. A forecasting system is said to be calibrated if the sample distribution of the target variable converges to the predicted distribution in the



**Figure 2.3:** Illustration of ensemble samples relative to a reference distribution  $\mathcal{N}(0,1)$  as groundtruth. 10-member ensembles are sampled from forecast distributions  $\hat{\mathcal{P}}_{\text{true}} = \mathcal{N}(0, 1)$ ,  $\hat{\mathcal{P}}_{\text{biased}} = \mathcal{N}(0.5, 1)$ ,  $\hat{\mathcal{P}}_{\text{narrow}} = \mathcal{N}(0, 0.8)$ ,  $\hat{\mathcal{P}}_{\text{wide}} = \mathcal{N}(0, 1.2)$ . The ensemble members are indicated as dashed vertical lines. The ensembles' empirical CDFs and the reference distribution CDF are shown in blue and red, respectively.

limit of many observations. The prediction is sharper the less variability is predicted on average. Throughout this thesis, the focus is on verification methods for real-valued prediction targets. The subsequent explanations follow the argumentation in Gneiting and Katzfuss (2014)<sup>2</sup>. The presented concepts are used in Höhle et al. (2024b) to evaluate the calibration and sharpness of the presented postprocessing models. For real-valued predictions, we write  $\hat{F}$  to refer to the CDF of the forecast distribution and  $\hat{p}$  to indicate the PDF.

**Probability integral transform** Calibration can be formalized using the *probability integral transform* (PIT). For a real-valued random variable  $Y$ , the PIT is defined as the value that the CDF attains at the materialized value of the observation, i.e.,

$$\text{PIT}(Y) = \hat{F}(Y). \quad (2.8)$$

The value of the PIT is a random variable,  $Z_{\hat{F}} := \hat{F}(Y)$ , that is standard uniform, i.e.,  $Z_{\hat{F}} \sim \mathcal{U}(0, 1)$ , if  $\hat{F}$  is continuous and  $Y \sim \hat{F}$ . The restriction to continuous distributions can be alleviated through randomized generalizations of the PIT (Czado et al. 2009; Rüschemdorf 2009). Motivated by the uniformity property of the PIT, a forecast is said to be *probabilistically calibrated* if  $Z_{\hat{F}} \sim \mathcal{U}(0, 1)$ .

<sup>2</sup>A mathematical framework for probabilistic weather forecasting and verification would be based on probability and prediction spaces (Gneiting and Ranjan 2013) but is omitted here for brevity. An introduction to the mathematical foundations of weather forecasting can be found in Gneiting and Katzfuss (2014).

A related quantity is the *dispersion* of the forecasts, which characterizes the predicted variability relative to the variability of the target variable. If the predicted variability is systematically smaller (larger) than that of the target quantity, relatively more (less) probability weight is associated with PIT values close to the extremes. The variance of the standard uniform distribution,  $\text{Var}(U) = \frac{1}{12}$  for  $U \sim \mathcal{U}(0, 1)$ , motivates the definition of *overdispersion*, *neutral dispersion* and *underdispersion* for forecast systems with  $\text{Var}(Z_{\hat{F}}) < \frac{1}{12}$ ,  $\text{Var}(Z_{\hat{F}}) \approx \frac{1}{12}$  and  $\text{Var}(Z_{\hat{F}}) > \frac{1}{12}$ , respectively.

For ensemble forecasts, the PIT analysis can be replaced with a rank computation. Using the notation from Equation 2.7, the rank  $R(Y)$  of an observation  $Y$  with outcome  $y$  is defined as

$$R(Y) = |\{\hat{y} \in \hat{\mathcal{E}}_M \cup \{y\} : \hat{y} \leq y\}|, \quad (2.9)$$

with  $|\cdot|$  denoting the number of items in the set. Assuming calibration,  $Y$  is indistinguishable from the remaining observations, such that  $R(Y)$  is uniformly distributed on  $\{1, \dots, M + 1\}$ . To align PIT and rank computations, Vogel et al. (2018) introduced the *unified PIT* (uPIT) for ensemble prediction,

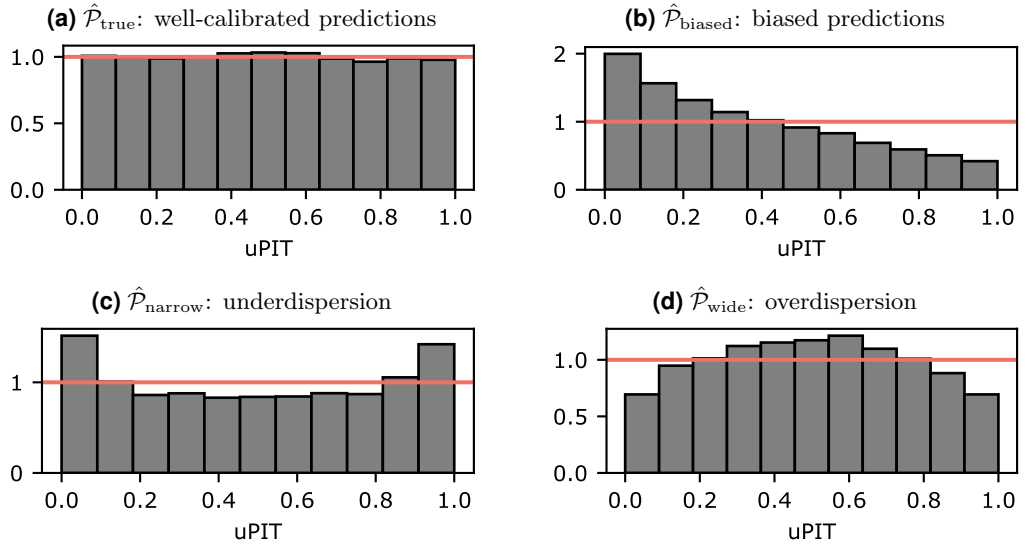
$$\text{uPIT}(Y) = \frac{R(Y) + U - 1}{M + 1}, \quad (2.10)$$

wherein  $U \sim \mathcal{U}(0, 1)$ . Similar to PIT,  $\text{uPIT}(Y) \sim \mathcal{U}(0, 1)$  in the case of neutrally calibrated ensemble forecasts.

**(u)PIT histograms** Given a set of forecast-observations pairs, PIT and uPIT values can be computed and summarized graphically in a *(u)PIT histogram*. Characteristic shapes of the empirical distribution indicate systematic miscalibration patterns. U-shaped and hump-shaped (u)PIT histograms indicate underdispersion and overdispersion, respectively, and skewed histograms suggest a prediction bias. A flat histogram indicates neutral dispersion, in which case the forecasting system is called *well-calibrated*. Analogous to PIT histograms, rank counts for historical ensemble prediction-observation pairs can be shown in *rank verification histograms* to assess calibration. Figure 2.4 shows examples of synthetic uPIT histograms with characteristic shapes.

**Prediction intervals** Calibration and sharpness of probabilistic forecasts can also be evaluated through *prediction intervals* (PIs). PIs are associated with an outlier probability  $\alpha \in (0, 1]$ . A PI at the  $(1 - \alpha)$ -level provides a probabilistic forecast for an interval  $[\hat{y}_{\min}, \hat{y}_{\max}] \subset \mathbb{R}$  that covers the observation outcome with probability  $1 - \alpha$ . Given prediction-observation pairs, calibration can be assessed through the *empirical coverage*, which is the empirical frequency of the observation falling inside the interval and should equal the theoretical coverage probability  $(1 - \alpha)$ . Sharpness can be addressed through the average length of the prediction interval,  $\hat{y}_{\max} - \hat{y}_{\min}$ .

Given a probabilistic forecast for a real-valued observation in terms of a CDF, a *symmetric* PI can be derived from the quantiles at levels  $\frac{\alpha}{2}$  and  $1 - \frac{\alpha}{2}$ . An  $M$ -member ensemble forecast suggests a series of natural outlier probabilities,  $\alpha_l$  for  $l = 1, \dots, \lfloor \frac{M}{2} \rfloor$ ,



**Figure 2.4:** Overview of characteristic shapes in uPIT histograms. Histograms are computed by simulating  $N = 10000$  independent ensemble predictions for target variables with distribution  $\mathcal{N}(0, 1)$ . 10-member ensembles are sampled from forecast distributions  $\hat{\mathcal{P}}_{\text{true}} = \mathcal{N}(0, 1)$ ,  $\hat{\mathcal{P}}_{\text{biased}} = \mathcal{N}(0.5, 1)$ ,  $\hat{\mathcal{P}}_{\text{narrow}} = \mathcal{N}(0, 0.8)$ , and  $\hat{\mathcal{P}}_{\text{wide}} = \mathcal{N}(0, 1.2)$ . The red line indicates the uniform distribution.

for which the order statistics of the ensemble, i.e., the sorted ensemble members in ascending order,  $\{\hat{y}_{(1)}, \dots, \hat{y}_{(M)}\}$ , can be used as interval bounds. For outlier probabilities

$$\alpha_l = \frac{M + 1 - 2l}{M + 1}, \quad (2.11)$$

the symmetric PI at the level  $(1 - \alpha_l)$  is delimited by  $\hat{y}_{\min} = \hat{y}_{(l)}$  and  $\hat{y}_{\max} = \hat{y}_{(M+1-l)}$ .

**Proper scoring rules** An alternative family of verification measures are *scoring rules*. Let  $\Pi_{\mathcal{Y}}$  denote the space of distributions on  $\mathcal{Y}$  and let  $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, \infty\}$  denote the extended real line. A *scoring rule* is a function  $S : \Pi_{\mathcal{Y}} \times \mathcal{Y} \rightarrow \overline{\mathbb{R}}$  that assigns a (potentially infinite) numerical score  $S(\hat{\mathcal{P}}, y)$  to a tuple  $(\hat{\mathcal{P}}, y)$  consisting of a forecast distribution and a materialized observation.  $S(\hat{\mathcal{P}}, \mathcal{P})$  is written to denote the expectation value of  $S(\hat{\mathcal{P}}, \cdot)$  under the distribution  $\mathcal{P} \in \Pi_{\mathcal{Y}}$ . A (negatively oriented) scoring rule is called *proper* (relative to  $\Pi_{\mathcal{Y}}$ ) if  $S(\mathcal{P}, \mathcal{P}) \leq S(\hat{\mathcal{P}}, \mathcal{P})$  for all  $\hat{\mathcal{P}} \in \Pi_{\mathcal{Y}}$ , and *strictly proper* if  $S(\mathcal{P}, \mathcal{P}) = S(\hat{\mathcal{P}}, \mathcal{P})$  implies  $\hat{\mathcal{P}} = \mathcal{P}$  (i.e., the maximum is unique).

Scoring rules are designed to measure the consistency between observations and predicted distributions. They are sensitive to both calibration and sharpness at the same time. Gneiting and Raftery (2007) provide an overview of various proper scoring rules.

One example is the log-score or ignorance score,

$$\text{LogS}(\hat{p}, \mathbf{y}) = -\log \hat{p}(\mathbf{y}), \quad \mathbf{y} \in \mathbb{R}^d, \quad (2.12)$$



which applies to both scalar- and vector-valued random variables and measures consistency through the predicted PDF evaluated at the observed value.

For real-valued prediction targets, the *continuous ranked probability score* (CRPS) is a popular rule that is defined in terms of the cumulative distribution function  $\hat{F}$  of the predicted distribution,

$$\text{CRPS}(\hat{F}, y) = \int_{-\infty}^{\infty} (\hat{F}(z) - \mathbf{1}\{y \leq z\})^2 dz, \quad (2.13)$$

with  $\mathbf{1}\{\cdot\}$  denoting the indicator function. The CRPS replicates the mean absolute error (MAE; cf. also subsection 3.1.4) for deterministic predictions and can thus be seen as a probabilistic generalization of it.

The integral in Equation 2.13 can be solved analytically for several types of probability distributions, including normal, exponential, and logistic distributions, truncated versions thereof<sup>3</sup>, and ensemble forecasts. For examples and an implementation of various scoring rules in R, see Jordan et al. (2018).

**Earth observations and reanalysis data** All verification methods rely on the availability of ground-truth data against which the forecast quality can be gauged. The primary sources for such information are Earth observations and retrospective analysis (*reanalysis*) data.

Earth observations come in various forms, including station observations, radar measurements, satellite observations, or aircraft- and vessel-based measurements. The data format varies between the different modalities, and the quality of the observations depends on the reliability of the measurement procedures (Inness and Dorling 2012).

To obtain a comprehensive representation of the ground truth state of the atmosphere at a given point in time, Earth observations are combined with short-term weather forecasts to yield a best-guess estimate of the atmospheric conditions at a certain point in time (Dee et al. 2014). The result of this process are reanalysis datasets, which have emerged as an important resource in the development of statistical models in meteorology and climate research.

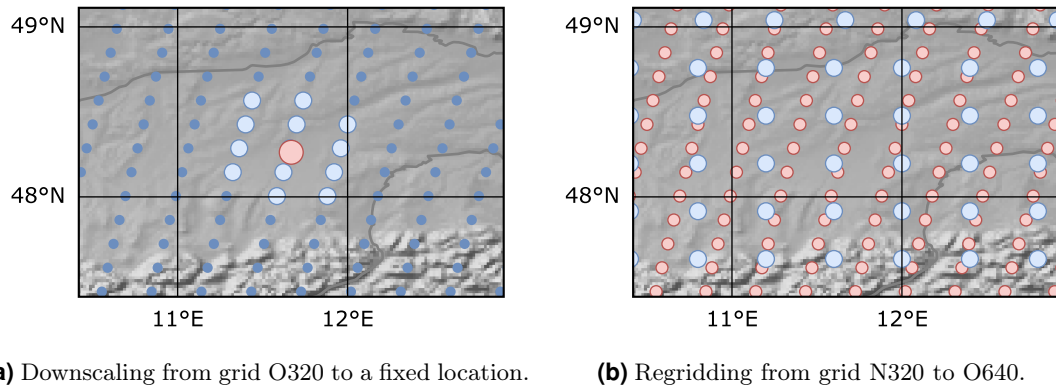
One of the most comprehensive reanalysis datasets, currently, is the fifth generation of the ECMWF atmospheric reanalysis (ERA5), comprising global data records for atmospheric, land surface, and ocean wave variables on an hourly basis from 1950 onwards (Hersbach et al. 2020). The models by H ohlein et al. (2020) use ERA5 data as one of the primary information sources.

## 2.2.4 Postprocessing

*Postprocessing* methods are used to translate raw NWP model outputs into forecast products that are useful for the forecast user. This includes interpolating discretized model outputs (horizontally and vertically) onto the location or area of interest and correcting statistical deficiencies of NWP models.

---

<sup>3</sup>A truncated distribution on the real line is obtained by restricting the support of another distribution to a shorter interval, and reweighting the remaining probabilities accordingly.



**Figure 2.5:** Overview of downscaling configurations. The map sections show the surrounding of Munich. (a) Grid-to-point downscaling predicts the local conditions at a discrete target location (red) based on data from the surrounding grid points (light blue). Data at distant locations (dark blue) is not used. (b) Regridding predicts the conditions at new grid vertices (red) to enhance spatial resolution of low-resolution gridded data (light blue). For definitions of the grid configurations, see subsection 2.3.1. Background graphics from Natural Earth (2012).

**Downscaling** *Downscaling*<sup>4</sup> describes a family of postprocessing techniques that enable the generation of forecast products for local target quantities based on model outputs at coarser spatial resolution. While linear or even nearest neighbor-based interpolation approaches are common for downscaling single-site predictions, more elaborate methods have been developed for downscaling gridded data.

At the methodological level, *dynamical downscaling* approaches are distinguished from *statistical downscaling*. Dynamical downscaling relies on higher-resolution physics-based regional climate models (RCMs) or limited-area models (LAMs), which simulate detailed weather and climate dynamics within regional domains of limited size. The coarse-resolution data serve as boundary conditions for the regional models. RCM and LAM dynamics are driven accordingly while filling in the detailed dynamics consistent with the coarse-scale conditions. Further details are found, e.g., in Giorgi (2019).

Statistical downscaling methods assimilate historical data records to establish a statistical mapping between coarse-scale and fine-scale information (see, e.g., Maraun and Widmann 2018). Statistical downscaling models do not require physical prior knowledge or simulation procedures and are often computationally cheap to evaluate. Once fitted, the models thus offer considerable savings in computation time compared to RCMs. Recent advances in ML and DL have brought algorithmic novelties to statistical downscaling, enabling significant improvements in prediction quality (Baño-Medina et al. 2020; Höhle et al. 2020).

<sup>4</sup>Note that the term *downscaling* in meteorology refers to decreasing the scale of the grid spacing, and corresponds direction-wise to the application of *superresolution* and *upsampling*, as used in computer vision.

Downscaling methods commonly address two different application scenarios, which we call *grid-to-point downscaling* and *regridding*. Figure 2.5 illustrates both configurations. Grid-to-point methods aim to improve the quality of local forecast products, such as scalar-valued or vector-valued prognostic variables at specific weather stations, by minimizing biases or statistical miscalibration of the forecasts. Regridding refers to the downscaling of prediction fields on extended domains and involves a gridded output data format. Regridding generates higher-resolution weather maps, e.g., for local climate assessment, while avoiding the computational complexity of high-resolution NWP simulations.

**Statistical postprocessing** Statistical postprocessing methods leverage statistical models and historical data records of numerical weather predictions and observations to compensate for biases and miscalibrations (Inness and Dorling 2012; Vannitsem et al. 2021). An important aspect is the improvement of probabilistic forecasts, notably ensemble forecasts, as addressed in ensemble postprocessing.

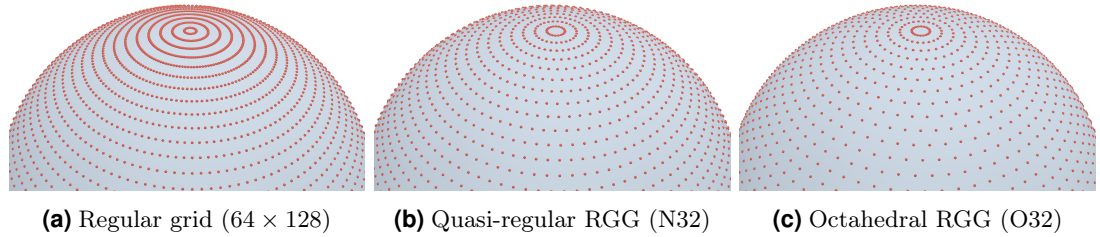
Given suitable metrics of forecast quality, such as strictly proper scoring rules, statistical postprocessing translates to a supervised learning problem, in which the models generate an optimized probabilistic forecast based on the information content of the raw NWP predictions. Depending on the format of the output prediction, statistical postprocessing methods can be classified as distribution-based and distribution-free methods (Vannitsem et al. 2021).

Distribution-based approaches, also called parametric approaches, train statistical models to predict the parameters of prespecified distribution templates, such as the mean and the standard deviation parameters of a normal distribution. While early methods employed simple linear regression models (Gneiting et al. 2005), newer approaches use more expressive model designs, such as tree-based models (e.g., Messner et al. 2017; Schlosser et al. 2019), and NNs (Rasp and Lerch 2018; Schulz and Lerch 2022). Distribution-free methods avoid restrictive distribution assumptions, e.g., by modeling quantile levels or quantile functions of the distribution (e.g., Bremnes 2020; Cannon 2018), parameterizing distribution histograms (Veldkamp et al. 2021), or processing ensemble members separately, *member-by-member*, to generate a calibrated ensemble forecast (e.g., Van Schaeybroeck and Vannitsem 2015, and references therein).

An important design challenge in ensemble postprocessing is how ensemble-valued predictors are supplied to the models. Prominent options are aggregate representations using low-dimensional summary statistics, such as, e.g., the ensemble mean and standard deviation (e.g., Schulz et al. 2021), or sorted ensembles (Bremnes 2020). In H ohlein et al. (2024b), we examine the impact of different representation schemes and suggest a principled DL solution that works natively with ensemble data.

## 2.3 Properties of Numerical Weather Data

NWP datasets are a distinct kind of numerical data with high complexity due to their physical interpretation, multivariate structure, and multi-dimensional variability in space,



**Figure 2.6:** Comparison of a regular horizontal grid with quasi-regular and octahedral reduced Gaussian grids (RGGs) on the northern hemisphere.

time, and the ensemble dimension. As such, NWP data qualify as *scientific data*, which is an umbrella term for datasets that are generated and analyzed in the context of the physical sciences (Card et al. 1999; Tory and Möller 2004). This section summarizes relevant aspects and properties of the scientific data types and formats encountered in NWP and discusses methods for their exploratory analysis.

### 2.3.1 Grid-Based Representation of Spatial Fields

NWP models represent continuous physical variable fields through samples on a discrete spatial grid. The gridded representation enables the computation of spatial derivative fields, e.g., through finite-difference or spectral approaches (see, e.g., Coiffier 2011). Dedicated grid structures are used to realize efficient computing operations or enable compact data storage. Data values on close-by grid points are typically affected by spatial correlations. Commonly, the horizontal discretization is decoupled from the vertical.

**Horizontal discretization** NWP models rely on a horizontal parameterization of the Earth’s surface in spherical coordinates, i.e., using latitude and longitude. Discretization schemes cover (parts of) the spherical domain with 2D grids. Figure 2.6 illustrates different variants used at the ECMWF and encountered in studies presented in this thesis (Höhlein et al. 2024a, 2020).

Regular (rectangular) grids (Figure 2.6, a) consist of grid points with regular spacing in both latitude and longitude. While the angular spacing is constant across the domain, the effective physical grid spacing decreases towards the poles due to spherical distortions, causing inefficiency in the data representation. However, regular grids enable the assignment of independent axes in multidimensional array structures for latitude and longitude dimensions, which simplifies data access in storage applications. Regular grids are thus well-suited for analysis and visualization purposes or for simulations on limited-size domains where distortion effects are negligible.

Irregular grids are preferred for simulations on global domains. Irregular samples allow for a more even distribution of the sample locations and offer increased flexibility for tailoring the discretization to the requirements of the simulation algorithm. Many of the datasets relevant to this study are sourced from the ECMWF and rely heavily on Gaussian grids. Gaussian grids distribute sample points on a fixed number of circles

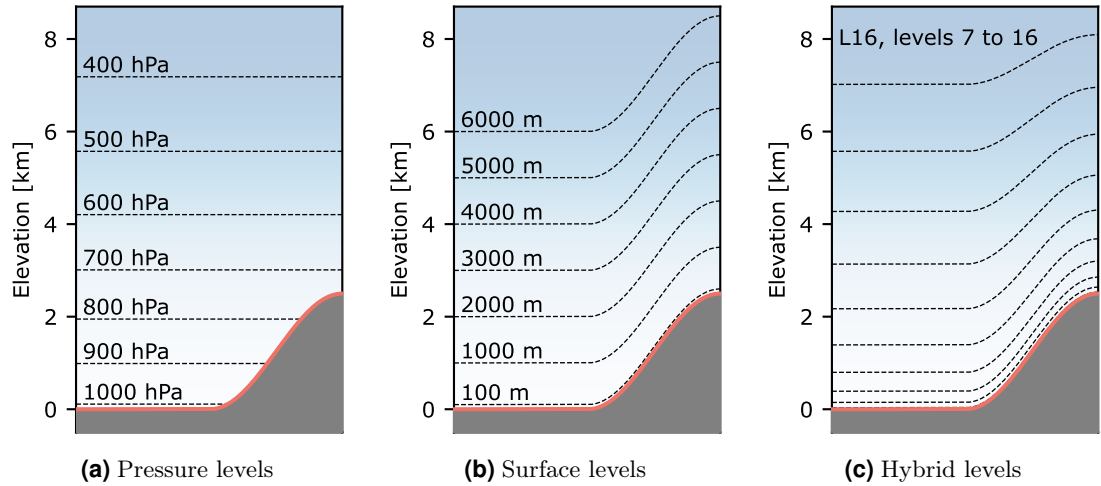
with constant latitude. Denoting the number of circles between the equator and pole on each hemisphere with  $N$ , the latitude values are determined by the roots of the Legendre polynomial of order  $2N$  to facilitate efficient numerical integration procedures. Three types of Gaussian grids are used and annotated with capital letters F, N, and O, respectively, to distinguish the format. The notations FN, NN, and ON (e.g., F320, N640, O1280) are used to distinguish grids of the same type in terms of grid spacing.

In *regular* or *full* Gaussian grids (F-grids; Hortal and Simmons 1991), there are  $4N$  longitude points distributed evenly along each latitude circle, resulting in a total number of  $8N^2$  grid points in a grid FN. Full Gaussian grids are similar to standard regular grids, with slight irregularity due to the uneven spacing between latitude circles.

*Reduced* Gaussian grids (RGGs) are defined to achieve a more even sampling density globally. *Quasi-regular* RGGs (N-grids; Hortal and Simmons 1991) have the same number of latitude circles as the corresponding F-grids but reduce the number of longitude points per circle in discrete steps to obtain a quasi-uniform sampling density for all latitudes (Figure 2.6, b) while maintaining feasibility for fast Fourier transform computations. The circles closest to the equator contain  $4N$  points.

Octahedral RGGs (O-grids; Malardel et al. 2016) reduce the number of points per circle linearly from the equator to the poles (Figure 2.6, c). The procedure is inspired by an octahedron-based projection of the sphere surface. Starting with 20 longitude points on the latitude circle closest to the pole, the point count increases by 4 with every circle. This results in a total count of  $4N(N + 9)$  points for grids ON, enabling a preferable memory-resolution trade-off compared to other grids.

**Vertical discretization** NWP models produce diagnostic outputs in three different vertical coordinate systems. Figure 2.7 illustrates the differences. For meteorological analysis of large-scale flow patterns, 2D fields are considered on isosurfaces of constant pressure, exploiting the pressure coordinates introduced in subsection 2.1.2 (e.g., geopotential at 500 hPa). Pressure isosurfaces (Figure 2.7, a) provide faithful information in upper levels of the atmosphere but can intersect with the terrain surface, leading to undefined values for the physical variables close to the Earth’s surface. When storing data in array-based structures, this causes missing data. For near-surface forecasts, it is common, thus, to analyze diagnostic fields on terrain-following elevation surfaces (Figure 2.7, b), which are called surface levels (e.g., temperature 2 m above ground or wind speed 10 m above ground). Surface levels circumvent missing data, but hamper efficiency in the computation of (physically relevant) gradients. To account for both, the simulations at ECMWF are carried out on so-called hybrid model levels, which define the local level height based on the geopotential height corresponding to a weighted average of the surface and the level pressure (Figure 2.7, c). The weighting is based on tabulated coefficients, which guarantee that the model levels follow the terrain geometry near the Earth’s surface and resemble pressure levels higher up in the atmosphere (Simmons and Strüfing 1983). Hybrid level schemes at the ECMWF are denoted as LN, where  $N$  is the number of levels used. Hybrid-level data circumvents the problem of missing data but requires knowl-



**Figure 2.7:** Comparison of vertical coordinate levels in standard atmosphere conditions and with varying orography. The surface level is shown in red.

edge of 3D pressure and humidity fields when conversion between pressure and geometric coordinates is required.

Data on hybrid levels was used in H ohlein et al. (2024a) to probe the volumetric temperature fields. To limit the memory and compute requirements of the presented downscaling approach, the vertical levels heights under the assumption of dry air in ISA conditions.

### 2.3.2 Multivariate Structure of Numerical Weather Data

Meteorological datasets often comprise data from multiple simulated (or measured) variables. Each variable on its own can possess a complex multidimensional (spatio-temporal) substructure. The data associated with distinct variables can have vastly different distribution characteristics, such as varying scales and value ranges of the numerical data and differences in the likelihood of extreme observations (cf., e.g., H ohlein et al. 2022).

Commonly, the different variables are related through multivariate and multidimensional correlation patterns. Understanding the interrelations between variables in NWP data is crucial for uncovering patterns and making informed predictions. *Dependence measures* assess the statistical relationship between two or multiple variables, indicating how changes in one variable coincide with changes in another. Several different dependence measures exist to measure different types of dependence. Variable relations and interdependence patterns in multivariate NWP data can be more intricate than simple pairwise dependencies and may affect the statistical reliability interpretation of the forecast distribution. The work by Farokhmanesh et al. (2023b) addresses the visual analysis of multivariate and spatial interdependence patterns in 3D ensemble forecasts. This requires quantifying the association strength of pairs of real-valued random variables.

**Dependence metrics** Throughout this thesis, a bivariate dependence metric  $\rho$  is a function that, when provided with a pair of real-valued random variables,  $(V_1, V_2)$ , assigns a real-valued dependence score  $\rho(V^{(1)}, V^{(2)})$ . In our work, the random variables reflect the predicted values of physical variables at certain locations in space. We further assume that  $\rho$  is symmetric under exchange of the variables, i.e.,  $\rho(V^{(1)}, V^{(2)}) = \rho(V^{(2)}, V^{(1)})$ , to indicate that the association is mutual and undirected. The statistics literature provides various metrics that assess different kinds of association (e.g., Tjøstheim et al. 2022). Most common variants involve the computation of expectation values, e.g., with respect to the joint distribution of the variables. However, such computations are infeasible in practical applications because the full form of the joint distribution is unknown. In practice, the association is measured based on a set of  $M$  paired samples,

$$\left\{ \left( v_1^{(1)}, v_1^{(2)} \right), \dots, \left( v_M^{(1)}, v_M^{(2)} \right) \right\},$$

and a statistical estimator  $\hat{\rho}$  which converges to  $\rho$  in the limit of infinite observations. We write  $\hat{\rho}(\mathbf{v}^{(1)}, \mathbf{v}^{(2)})$  to denote the finite-sample estimate of  $\rho(V^{(1)}, V^{(2)})$ , wherein

$$\mathbf{v}^{(\cdot)} := \left( v_1^{(\cdot)}, \dots, v_M^{(\cdot)} \right)^T \in \mathbb{R}^M.$$

The required effort for estimating association strengths varies between different metrics. Farokhmanesh et al. (2023b) focus on Pearson’s product-moment correlation coefficient and on mutual information (MI), representing opposite sides of the complexity spectrum (Berenjkoub et al. 2019).

**Pearson product-moment correlation coefficient** The Pearson correlation coefficient, also called Pearson’s  $r$ , measures the linear association between pairs of random variables (e.g., Tjøstheim et al. 2022). It is commonly used in data visualization to explore relationships between variables. Pearson’s  $r$  is defined as

$$r(V^{(1)}, V^{(2)}) := \frac{\text{Cov}(V^{(1)}, V^{(2)})}{\sqrt{\text{Var}(V^{(1)}) \cdot \text{Var}(V^{(2)})}}, \quad (2.14)$$

wherein  $\text{Var}(\cdot)$  and  $\text{Cov}(\cdot, \cdot)$  are variance and covariance of the random variables.

The finite-sample estimate is defined as

$$\hat{r}(\mathbf{v}^{(1)}, \mathbf{v}^{(2)}) := \frac{\mathbf{v}^{(1)} \cdot \mathbf{v}^{(2)}}{\sqrt{\mathbf{v}^{(1)} \cdot \mathbf{v}^{(1)}} \sqrt{\mathbf{v}^{(2)} \cdot \mathbf{v}^{(2)}}}, \quad (2.15)$$

with  $\cdot$  denoting the standard scalar product.

Pearson’s correlation varies between  $-1$  and  $1$ , with  $1$  indicating positive correlation,  $-1$  anti-correlation, and  $0$  the absence of correlation. Pearson-correlated random variables are associated with a linear relation. The metric is easy to compute and interpret. The coefficient presumes that the joint distribution of the random variables is Gaussian with critical importance for the reliability of the estimate. Also, the requirement of a linear relationship is very restrictive and excludes many alternative association patterns.

**Mutual information** MI quantifies more general interdependencies, including nonlinear and nonmonotonic associations (Cover, Thomas, et al. 1991). MI is defined as the difference in uncertainty when considering the random variables separately or jointly. Mathematically, this is expressed as

$$\text{MI}(V^{(1)}, V^{(2)}) := H(V^{(1)}) + H(V^{(2)}) - H(V^{(1)}, V^{(2)}), \quad (2.16)$$

wherein  $H(V^{(1)})$  and  $H(V^{(2)})$  are the entropies of the separate random variables, and  $H(V^{(1)}, V^{(2)})$  is the entropy of the joint distribution. MI values of 0 indicate statistical independence, whereas positive values suggest *some* form of nontrivial statistical association. Information about the details of the association is not conveyed.

Computing estimates  $\hat{\text{MI}}(\mathbf{v}^{(1)}, \mathbf{v}^{(2)})$  from multi-samples of real-valued random variables is challenging. Existing algorithms rely on density estimation and nearest neighbor computations, which do not scale well to large sample sizes (Kraskov et al. 2004; Moon et al. 1995) and are thus unsuited for applications involving very large ensembles (e.g., Necker et al. 2020; Tempest et al. 2023). More recent estimators rely on copulas and NNs (Belghazi et al. 2018; Zeng and Durrani 2011) but are complex, potentially inaccurate, and computationally demanding. The complexity of MI estimators hampers the online computation of MI scores and their integration into interactive analysis workflows. NDFs in Farokhmanesh et al. (2023b) therefore outsource the computation into a preprocessing step and leverage a neural data structure for interactive data analysis.

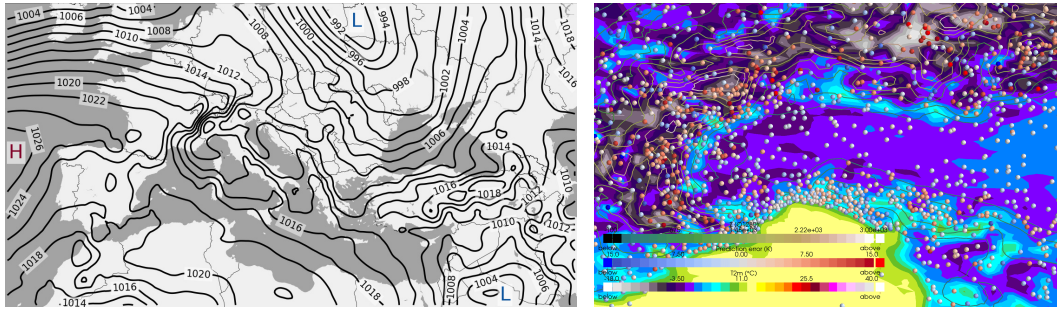
### 2.3.3 Permutation Symmetry in Ensemble Data

In NWP ensemble datasets, each ensemble member constitutes a (potentially multivariate) dataset comprised of field samples of the simulated variables. The level of ensemble composition differs qualitatively from the dimensions associated with spatiotemporal variation and multi-variable composition. Throughout this thesis, the members within an ensemble are interpreted as independent random samples from the predictive distribution of the forecast model. Unlike the space and time dimensions, the ensemble lacks an intrinsic order relation, and unlike in multi-variable data, there is no uniquely determined way of labeling the members distinctly. As a result, any order in which subsequent ensemble members are loaded into memory is arbitrary and should not affect the outcome of analysis procedures. Suitable algorithms must be invariant to permutations of the memory order and should respect the stochastic nature of the data.

## 2.4 Visualizing Numerical Weather Data

Visualization in this thesis focuses on NWP datasets that represent scalar- and vector fields defined on continuous 2D and 3D spatial domains and station observations associated with discrete locations. The visualization methods introduced in this section are by no means complete. Instead, the selection is guided by the methods' relevance to the thesis. We refer to dedicated textbooks, such as Brodlić et al. (2012), or relevant review





(a) Synoptic chart showing pressure isocontours used in Höhle et al. (2020), cf. Figure 13 (b). Used with permission. (b) Map-based representation of a case study in Höhle et al. (2024a), cf. Figure 9 (b). Used with permission.

**Figure 2.8:** Examples of map-based visualizations used in the presented publications illustrating the use of isocontours and colormaps.

articles, such as the works by Rautenhaus et al. (2018) and Afzal et al. (2019), for a more comprehensive discussion.

### 2.4.1 Map-Based Visualization

Map views are popular across many atmospheric forecasting and research tasks (Rautenhaus et al. 2018). Despite the 3D structure of the atmosphere, meteorologists commonly analyze physical parameters on surface or pressure isolevels (cf. subsection 2.3.1), which are effectively 2D surfaces. In map views, the horizontal coordinates are projected to the 2D plane, enabling a visually clear display of information without perspective-related occlusion problems. Examples of map-view visualizations used in the presented publications are shown in Figure 2.8.

Different visualization methods are employed depending on the type of encoded information. Colormaps are commonly used to encode scalar values in the display color, enabling a versatile and intuitive depiction of, e.g., scalar fields. Alternatives include encoding information through ancillary geometry or image textures. For instance, pressure fields or terrain elevation are commonly encoded through polylines representing the fields' isocontours. Higher-dimensional information, such as vectors or general multivariate data, can be encoded through glyphs (cf., e.g., Borgo et al. 2013). Glyph-based representations are used in meteorological applications to embed information about observation data in the spatial context of the observation site (Rautenhaus et al. 2018). Building glyph-based visualizations for spatially continuous data, such as vector fields, requires care due to the visual complexity of the representations and occlusions (e.g., Elmqvist and Tsigas 2008). An example of a texture-based visualization method is line integral convolution (LIC; Cabral and Leedom 1993), which displays the orientation of the flow while circumventing the problems of glyph-based displays. LIC is preferred in Höhle et al. (2020) over glyph-based representations to obtain a spatially dense representation of the downscaled wind fields while avoiding visual clutter.

### 2.4.2 3D Visualization

3D visualizations are preferred when the vertical dimension provides added value, e.g., to combine representations of the real world with representations of abstract data embedded therein (Bleisch 2012). 3D visualizations facilitate a spatial perception of value distributions in the data. This thesis considers visualizations of scalar fields defined on 2D manifolds in 3D space, such as the terrain surface, and native volumetric scalar fields.

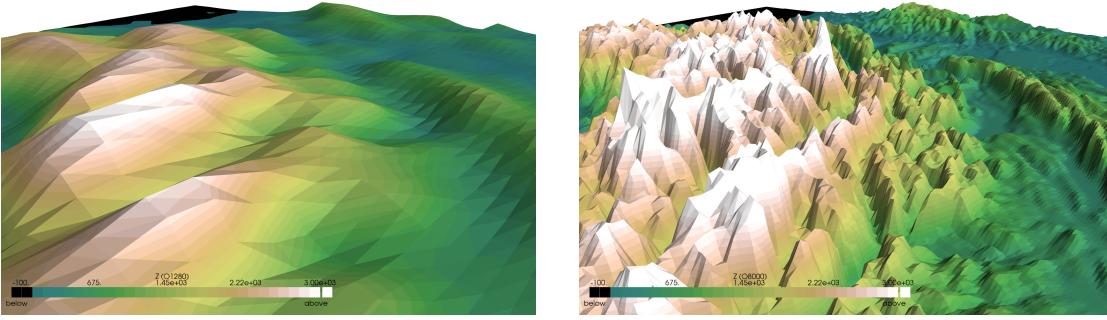
Throughout this thesis, scalar fields on 2D domains admit visualizations similar to map views. Visualizing volumetric fields, however, requires methods that enable the user to recognize features inside the volume without obstructions from the outer volume parts. Common visualization methods for volumetric scalar fields include isosurface rendering, direct volume rendering (DVR), and slicing. In isosurface rendering, user-selected level sets of the scalar field are rendered to indicate the field's value distribution. The rendering techniques are based on rasterization approaches (e.g., Lorensen and Cline 1987; Treece et al. 1999) or raytracing (e.g., Amanatides, Woo, et al. 1987; Levoy 1988, 1990). DVR techniques use volumetric rendering algorithms to visualize scalar fields as translucent objects (e.g., Max 1995). A central component is the *transfer function*, which maps scalar values to the opacity and color values required for rendering. The transfer function requires tuning to emphasize relevant features. In slicing, the volume is intersected with one or several user-configurable 2D planes, and the field information is visualized only at the intersections using surface-based visualization methods. Volumetric visualizations of scalar fields can occlude large parts of the surface geometry, especially if geometries are rendered opaque or if DVR volumes are dense.

Höhlein et al. (2020) and Höhlein et al. (2024a) use 3D visualizations to examine NWP data in a spatial context. Slicing-based visualizations are generally preferred in Höhlein et al. (2024a) to minimize visual obstructions and minimize the required user interactions. In Höhlein et al. (2022), DVR visualizations are preferred due to the dense display of information and for consistency with prior work.

### 2.4.3 Topographic Visualization

Next to standalone data examination, meteorologists require visualizations to assess interrelations between different datasets and data in different meteorological conditions. Due to the relevance of PBL effects for atmospheric dynamics (cf. paragraph 2.1.3), an important concern is the visualization of meteorological datasets in the context of the Earth's surface geometry. The surface terrain is naturally represented in NWP models as a 2D surface in 3D space. The resolution of the terrain geometry depends on the grid spacing of the model grid, which is often too coarse to resolve fine details, especially in global NWP applications. A comparison of terrain representations in two octahedral RGGs (cf. subsection 2.3.1) with different resolutions is shown in Figure 2.9. The resulting misrepresentations and uncertainties, as well as their impact on simulated and measured data values, require exploration.

In Höhlein et al. (2024a), these aspects are explored in the context of temperature downscaling. Important aspects of the visual exploration include:



**(a)** Terrain at O1280 resolution: 9 km grid spacing.      **(b)** Terrain at O8000 resolution: 1 km grid spacing.

**Figure 2.9:** Comparison of terrain representations with different grid resolution. Images reproduced from Hohlein et al. (2024b), Figure 1. Used with permission.

- the spatial distribution of physical variable values on the terrain,
- the relation between variable values on the surface and in the free air,
- the inherent uncertainty of the terrain surface due to sub-grid variability,
- and the allocation of station sites and station-specific variables in the terrain.

Intuitive visualization aspects suggest that an interactive 3D spatial representation is well-suited for displaying the relevant information. However, the joint display of multiple data entities, potentially using several instances of point-, line-, surface-, and volume-like graphics objects, induces visual and analytical complexity.

In Hohlein et al. (2024a), we combine multiple 3D visualization techniques that address the above-mentioned points in an interactive visualization system, enabling meteorologists to explore relevant data interactively while focusing on selected aspects as required.



This chapter introduces key concepts and terminology from ML and statistical inference with a focus on DL and NNs. The chapter concludes with a discussion of selected applications of DL methods in computer vision and a brief overview of DL model explanation techniques.

## 3.1 Fundamentals of Deep Learning

### 3.1.1 Neural Networks

NNs are a family of ML models used to build nonlinear parametric function approximators. Given parameters  $\phi$ , a NN represents a mapping  $\mathbf{h}_\phi : \mathcal{X} \rightarrow \mathcal{Y}$  between (typically vector-valued) inputs  $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^{d_{\text{in}}}$  and outputs  $\mathbf{y} \in \mathcal{Y} \subseteq \mathbb{R}^{d_{\text{out}}}$ . A NN model consists of a set of nodes, called *neurons*, and edges, which connect pairs of neurons, similar to synapses connecting neurons in the brain. Each neuron represents an atomic information processing unit capable of storing data, executing basic data transformations, or responding to input signals from connected neurons. Networks in which information is propagated in a single direction, i.e., networks without closed-loop connections between groups of neurons, are called *feed-forward* networks. In such a network, each neuron responds to the inputs received from its connected input neurons according to a *response function* associated with each neuron. The outputs of the response function are typically real scalar values and are called *activations*.

Commonly, NNs implement affine-linear response functions with a subsequent nonlinear activation function. Mathematically, this can be expressed as

$$a_n := \rho \left( \sum_{m \in \text{In}(n)} w_{nm} a_m + b_n \right), \quad (3.1)$$

Function	$\rho(x)$	$\mathcal{X}$	$\mathcal{Y}$	Remarks
LeakyReLU	$\max(x, \alpha x)$	$\mathbb{R}$	$\mathbb{R}$	parameter $\alpha \in \mathbb{R}$ (default: $\alpha = 0.01$ )
ReLU	$\max(x, 0)$	$\mathbb{R}$	$\mathbb{R}_0^+$	-
Sigmoid	$[1 + \exp(-x)]^{-1}$	$\mathbb{R}$	$[0, 1]$	-
SnakeAlt	$0.5x + \sin(x)^2$	$\mathbb{R}$	$\mathbb{R}$	proposed in Weiss et al. (2022)
Softplus	$\log(1 + \exp(x))$	$\mathbb{R}$	$\mathbb{R}^+$	-

**Table 3.1:** NN activation functions  $\rho : \mathcal{X} \rightarrow \mathcal{Y}$  used throughout the thesis.

wherein  $\text{In}(n)$  denotes the set of input neurons of  $n$  and  $a_n, a_m \in \mathbb{R}$  are the respective activations. The function  $\rho : \mathbb{R} \rightarrow \mathbb{R}$  is a nonlinear activation function. Examples of activation functions relevant to this work are listed in Table 3.1. The parameters  $w_{nm} \in \mathbb{R}$  and  $b_n \in \mathbb{R}$  are the weights and biases of the affine-linear map and can be tuned to change the neuron’s response. Using a nonlinear activation ensures that the NNs can represent nonlinear mappings.

**Network layers and tensor notation** Due to the sequential structure of Equation 3.1, neurons are often organized in a sequence of disjoint *layers*  $\mathcal{L}^{(1)}, \dots, \mathcal{L}^{(L)} \subset \mathcal{N}$ . Assuming a fixed ordering of the neurons in each layer, neuron activations and parameters can be expressed in tensor notation<sup>1</sup>. Equation 3.1 for neurons in layer  $\mathcal{L}^{(l)}$  can be written as

$$\mathbf{a}^{(l)} := \rho \left( \mathbf{W}^{(l)} \mathbf{a}^{(l-1)} + \mathbf{b}^{(l)} \right). \quad (3.2)$$

Therein,  $\mathbf{a}^{(l)} \in \mathbb{R}^{d_l}$  and  $\mathbf{a}^{(l-1)} \in \mathbb{R}^{d_{l-1}}$  denote the activations of neurons in layers  $\mathcal{L}^{(l)}$  and  $\mathcal{L}^{(l-1)}$ , respectively, with  $d_l, d_{l-1} \in \mathbb{N}$  being the number of neurons in  $\mathcal{L}^{(l)}$  and  $\mathcal{L}^{(l-1)}$ . The inputs of the first layer reflect the external inputs to the network, i.e.,  $\mathbf{a}^{(0)} := \mathbf{x}$ .

The parameters  $\mathbf{W}^{(l)} \in \mathbb{R}^{d_l \times d_{l-1}}$  and  $\mathbf{b}^{(l)} \in \mathbb{R}^{d_l}$  are defined element-wise, such that  $[\mathbf{W}^{(l)}]_{nm} := w_{nm}$  and  $[\mathbf{b}^{(l)}]_n := b_n$ . Overloading the notation of Equation 3.1,  $\rho$  denotes an element-wise nonlinear activation function. Though activation functions can be equipped with trainable parameters, the methods presented in this thesis use non-parametric activation functions only.

**Network parameters and hyperparameters** According to Equation 3.2, each network layer  $\mathcal{L}^{(l)}$  defines a parametric transformation  $\mathbf{h}_{\phi^{(l)}}^{(l)}$  with parameters

$$\phi^{(l)} := \{ \mathbf{W}^{(l)}, \mathbf{b}^{(l)} \},$$

<sup>1</sup>The term *tensor* is often used in the DL context to refer to multi-dimensional arrays (e.g., Abadi et al. 2016; Paszke et al. 2019), despite its different meaning in linear algebra (e.g., Brand 2020). This thesis adheres to the DL interpretation, such that vectors and matrices are 1D and 2D tensors. Given a tensor  $\mathbf{T}$ , the notation  $[\mathbf{T}]$  is used to express indexing of tensor elements.

such that

$$\mathbf{a}^{(l)} = \mathbf{h}_{\phi^{(l)}}^{(l)}(\mathbf{a}^{(l-1)}). \quad (3.3)$$

The function that is represented by the complete feed-forward network, i.e., the actual ML model  $\mathbf{h}_\phi$ , is obtained by concatenating the layer-wise mappings sequentially,

$$\mathbf{h}_\phi = \mathbf{h}_{\phi^{(L)}}^{(L)} \circ \dots \circ \mathbf{h}_{\phi^{(1)}}^{(1)}. \quad (3.4)$$

Therein,  $\phi := \{\phi^{(l)} : l \in \{1, \dots, L\}\}$  refers to the full set of *model parameters*, which together determine the represented function. Model parameters can usually be represented in tensorial form and admit the computation of gradients  $\nabla_\phi \mathbf{h}_\phi(\mathbf{x})$  of the model outputs wrt. the parameters. Other design choices, such as the number of layers, the number of neurons per layer, or properties of the activation function, affect the model outputs non-differentiably and are usually considered separate from the model parameters. These are called *hyperparameters*. Problem-specific constraints on the connectivity between neurons or systematic deviations from the layered network structure are considered as alternative *network architectures* and are discussed in more detail in section 3.2.

### 3.1.2 Modeling Non-Deterministic Functions

Probabilistic models represent parametric functions with non-deterministic outcomes. Models of this kind are well suited for capturing uncertainties or ambiguities. Instead of generating deterministic outputs, probabilistic models return random variables with a distribution conditioned on the model inputs. The output distribution is often denoted in terms of a conditional probability density function (cf., e.g., Prince 2023). Following this convention, we write  $p_{\mathbf{h}_\phi}(\mathbf{y}|\mathbf{x})$  to denote the output distribution of a probabilistic model that is parametrized through a NN  $\mathbf{h}_\phi$  and evaluated on inputs  $\mathbf{x}$ .

Distribution regression networks (DRNs; Rasp and Lerch 2018) and deep generative models (DGMs) are two probabilistic model classes that are important for this work.

**Distribution regression networks** In DRNs, NNs are used to predict parameters  $\boldsymbol{\theta} = \mathbf{h}_\phi(\mathbf{x})$  that control the shape of a predefined distribution template  $p_\theta(\mathbf{y})$ . The choice of the template distribution depends on the characteristics of the target quantity. In this thesis, the target quantities are scalar- or vector-valued continuous random variables. Examples of distributions for scalar-valued continuous random variables, characterized by probability density function, support, and parameters, are listed in Table 3.2. Distributions for vector-valued variables can be constructed by joining multiple distributions for the vector components – one component at a time – or using natively multivariate template distributions (for examples see, e.g., Bishop 2006).

**Deep generative models** In real-world applications, parametric assumptions about the shape of the predictive distribution may be overly restrictive. Generative models avoid the limitations of a fixed-shape output distribution by expressing the predictive distribution without an explicit functional form. DGMs are a class of ML techniques that leverage NNs and DL to learn free-form representations of complex data distributions.

Distribution	Probability density function	Support	Parameters
Normal distribution	$\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$	$x \in \mathbb{R}$	$\mu \in \mathbb{R}, \sigma \in \mathbb{R}^+$
Laplace distribution	$\frac{\lambda}{2} e^{-\lambda x-\mu }$	$x \in \mathbb{R}$	$\mu \in \mathbb{R}, \lambda \in \mathbb{R}^+$
Logistic distribution	$\frac{e^{-(x-\mu)/s}}{s(1+e^{-(x-\mu)/s})^2}$	$x \in \mathbb{R}$	$\mu \in \mathbb{R}, s \in \mathbb{R}^+$
Log-normal distribution	$\frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\log(x)-\mu}{\sigma}\right)^2}$	$x \in \mathbb{R}^+$	$\mu \in \mathbb{R}, \sigma \in \mathbb{R}^+$

**Table 3.2:** Parametric distributions for scalar-valued continuous random variables.

DGMs are trained to generate outputs indistinguishable from samples of the target distribution. The sampling process involves randomness, which is captured in non-deterministic *latent variables*. Latent variables are potentially high-dimensional unobserved random variables that are introduced artificially in statistical models to encapsulate sources of randomness. The values of the latent variables are sampled randomly at runtime and are supplied to the model as auxiliary inputs. Current DGM algorithms can be coarsely subdivided into four categories: likelihood-based variational models, generative adversarial networks (GANs), energy-based models, and normalizing flows. A detailed discussion and a comparison of DGM approaches can be found in Bond-Taylor et al. (2022).

GANs and energy-based models, specifically diffusion models, are used most commonly in applications related to this work. GANs (Arjovsky et al. 2017; Goodfellow et al. 2014; Gulrajani et al. 2017) consist of a pair of neural networks, which take the role of a sample generator and a discriminator, respectively. While the generator learns to translate latent variables into samples that are similar to those from the target distribution, the discriminator learns to distinguish synthetic samples from real data and provides a training signal for the generator to improve the sample quality. Diffusion models implement sequential sampling strategies that mimic the step-wise inversion of a noisy diffusion process and are subject to ongoing research. A review of recent progress has been presented by Yang et al. (2023).

### 3.1.3 Model Training and Hyperparameter Selection

The process of fitting deep learning models to data involves the selection of suitable model parameters and hyperparameters, such that a performance metric is optimized. In the following, we write  $P[\mathbf{h}_\phi; \mathcal{D}]$  to denote the performance score of a model  $\mathbf{h}_\phi$  evaluated on a set of available data samples, denoted as  $\mathcal{D}$ .

Practical optimization procedures often operate iteratively and in two stages. An inner optimization cycle using gradient-based optimization techniques is invoked to learn the model parameters, whereas an outer optimization with more general optimization procedures is used to select the hyperparameters. To achieve the best results, the available data  $\mathcal{D}$  is commonly split into a training dataset,  $\mathcal{D}_{\text{train}}$ , and a validation dataset  $\mathcal{D}_{\text{val}}$ ,



such that  $\mathcal{D} = \mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{val}}$ , and  $\mathcal{D}_{\text{train}} \cap \mathcal{D}_{\text{val}} = \emptyset$ . The process of learning model parameters  $\phi$  is called *model training* and involves a numerical optimization of the objective function  $P(\phi) := P[\mathbf{h}_\phi; \mathcal{D}_{\text{train}}]$ . In DL applications, the optimization objective is commonly called the *loss function* and is assumed to be differentiable wrt.  $\phi$ . The outer optimization is referred to as *hyperparameter selection* and focuses on finding hyperparameters, such that the validation performance  $P[\mathbf{h}_\phi; \mathcal{D}_{\text{val}}]$  of the model is optimized. Jointly, the two stages guarantee that the final model performs as well as possible on data not seen during training. Model intercomparisons are commonly based on independent datasets, called test data,  $\mathcal{D}_{\text{test}}$ , that must not be used for optimization purposes. A more comprehensive discussion of DL training methods can be found in dedicated DL textbooks (e.g., Murphy 2022; Prince 2023).

**Regularization** *Regularization* methods are employed to counteract overfitting and improve generalization. Frequently used regularization methods rely on loss-based penalization schemes or architectural design patterns. A common approach from the first category is weight decay, which penalizes the magnitude of model parameters such that smoother function representations are learned. A popular architectural regularization scheme is Dropout (Srivastava et al. 2014), which introduces noise into the neuron activations, thus limiting the information flow between layers and enforcing redundancy and smoothness in the learned activation patterns. Both weight decay and Dropout have been used in various experiments presented in this thesis.

### 3.1.4 Training Objectives

Training objectives for DL are based on task-specific empirical performance metrics or probability-based quality criteria, originating from statistical inference theory (Murphy 2022). This section discusses examples of objective functions used in the presented studies. We focus on performance metrics for regression tasks, in which the models learn to emulate a mapping between input items  $\mathbf{x} \in \mathcal{X}$ , and target outputs,  $\mathbf{t} \in \mathcal{Y}$ , with  $\mathcal{X}$  and  $\mathcal{Y}$  as introduced in subsection 3.1.1. The datasets for training, validation, and testing, accordingly, are of the form

$$\mathcal{D} = \left\{ \left( \mathbf{x}^{(n)}, \mathbf{t}^{(n)} \right) \right\}_{n=1}^{|\mathcal{D}|} \subset \mathcal{X} \times \mathcal{Y},$$

where  $|\mathcal{D}| \in \mathbb{N}$  is the number of data points in the respective set.

**Mean squared error and mean absolute error** In many studies, ML and DL models are trained by optimizing the agreement between vector-valued model predictions and targets (e.g., Höhle et al. 2022; Lu et al. 2021; Weiss et al. 2022). Prominent examples are the *mean squared error* (MSE),

$$P_{\text{MSE}}[\mathbf{h}_\phi; \mathcal{D}] := \frac{1}{|\mathcal{D}|} \sum_{(\mathbf{x}, \mathbf{t}) \in \mathcal{D}} \|\mathbf{t} - \mathbf{h}_\phi(\mathbf{x})\|_2^2,$$

and the *mean absolute error* (MAE),

$$P_{\text{MAE}}[\mathbf{h}_\phi; \mathcal{D}] := \frac{1}{|\mathcal{D}|} \sum_{(\mathbf{x}, \mathbf{t}) \in \mathcal{D}} \|\mathbf{t} - \mathbf{h}_\phi(\mathbf{x})\|_1,$$

wherein  $\|\cdot\|_{1/2}$  denote 1- and 2-norm in the respective vector space.

**Maximum likelihood and maximum a posteriori** For probabilistic models, a common training approach derives from maximizing the so-called model posterior, i.e., the likelihood  $p(\mathbf{h}_\phi|\mathcal{D})$  of a model  $\mathbf{h}_\phi$  explaining the observed pattern, given observed data  $\mathcal{D}$ . Bayes' theorem states that this likelihood is proportional to a product of the likelihood  $p(\mathcal{D}|\mathbf{h}_\phi)$  of observing  $\mathcal{D}$  under model  $\mathbf{h}_\phi$  and a model prior  $p(\mathbf{h}_\phi)$ , measuring the marginal plausibility of  $\mathbf{h}_\phi$  without knowing any data. Depending on the choice of  $p(\mathbf{h}_\phi)$ , this results in one of two optimization problems.

If the model prior encodes model preferences that are considered in the optimization, this motivates the *maximum a posteriori* (MAP) approach,

$$\phi_{\text{MAP}} = \arg \max_{\phi \in \Phi} p(\mathbf{h}_\phi|\mathcal{D}), \quad (3.5)$$

which searches for the set of model parameters that is the most likely – in terms of  $P_{\text{MAP}}[\mathbf{h}_\phi; \mathcal{D}] = p(\mathbf{h}_\phi|\mathcal{D})$  – to have generated the observed data.

If the prior assigns equal likelihood to all models, it can be neglected in the likelihood optimization. The resulting model estimate,

$$\phi_{\text{MLE}} = \arg \max_{\phi \in \Phi} p(\mathcal{D}|\mathbf{h}_\phi), \quad (3.6)$$

optimizes the model to agree with the observed data and is called the *maximum-likelihood estimate* (MLE) with loss function  $P_{\text{MLE}}[\mathbf{h}_\phi; \mathcal{D}] = p(\mathcal{D}|\mathbf{h}_\phi)$ .

In both cases, the relevant probabilities can be expressed in terms of the output distributions of the probabilistic model. Furthermore, optimizing the raw probabilities is equivalent to optimizing the logarithmic probabilities, which are often easier and numerically more stable to compute. The optimization problems arising from MSE and MAE are special cases of Equation 3.6 when the model predictions are interpreted as the parameters of a probability distribution  $p_{\mathbf{h}_\phi}(\mathbf{y}|\mathbf{x})$ . MSE and MAE objectives correspond to component-wise normal distribution or Laplace distribution, respectively (Murphy 2022).

Among the contributions presented in this thesis, Höhle et al. (2020), Höhle et al. (2022), Farokhmanesh et al. (2023b), and Höhle et al. (2024a) train models, which are based on MAP-like and MLE-like objectives.

**Optimum score estimation** Both MAP and MLE require a quantitative evaluation of the predicted distribution density  $p_{\mathbf{h}_\phi}(\mathbf{y}|\mathbf{x})$ . Optimum score estimation (OSE) has been proposed by Gneiting and Raftery (2007) as an alternative to (and generalization of) MLE for more general probabilistic models. The performance assessment of different models relies on strictly proper scoring rules, as introduced in subsection 2.2.3.

For compatibility with the previous notation, let  $p_{\mathbf{h}_\phi}(\cdot | \mathbf{x}) \in \Pi_{\mathcal{Y}}$  denote the predicted distribution of a NN model  $\mathbf{h}_\phi$  with inputs  $\mathbf{x} \in \mathcal{X}$ , and let  $S : \Pi_{\mathcal{Y}} \times \mathcal{Y} \rightarrow \overline{\mathbb{R}}$  be a proper scoring rule. Given a training dataset  $\mathcal{D} \subset \mathcal{X} \times \mathcal{Y}$  as introduced above, the mean score  $P_S[\mathbf{h}_\phi; \mathcal{D}]$  is defined as

$$P_S[\mathbf{h}_\phi; \mathcal{D}] = \frac{1}{|\mathcal{D}|} \sum_{(\mathbf{x}, \mathbf{t}) \in \mathcal{D}} S(p_{\mathbf{h}_\phi}(\cdot | \mathbf{x}), \mathbf{t}), \quad (3.7)$$

and provides an objective for the optimum-score optimization,

$$\phi_S := \arg \min_{\phi \in \Phi} P_S[\mathbf{h}_\phi; \mathcal{D}]. \quad (3.8)$$

OSE based on the CRPS (cf. Equation 2.13) is used in Höhle et al. (2024b) for training parametric probabilistic DL models for statistical postprocessing.

## 3.2 Neural Network Architectures

NNs are flexible parametric function approximations. Universal approximation properties have been proved for several types of network configurations (Cybenko 1989; Hornik 1991; Lu et al. 2017). In practical learning tasks, the quality of the learned functions is determined by the *architecture* of a NN, i.e., the arrangement of neurons in the network, their respective connectivity, and the choice of activation functions. The network architecture is critical for the model performance in learning tasks with scarce training data, in applications where training and inference are subject to memory and time constraints, or if the input data format requires special considerations. The quality of a trained network is reflected in its ability to interpolate between training examples and generalize to unseen data. The flexibility of NNs comes with a risk of overfitting the training data (Zhang et al. 2021a). Different network architectures come with different *inductive biases* (cf. Mitchell 1980), which guide the optimization towards different solutions. This may be caused by implicit regularization induced by the gradient-based optimization (Smith and Le 2018; Zhang et al. 2021a) or by explicit functional constraints encoded in the network design.

The subsequent sections introduce several families of NN architectures that have advantages in certain learning tasks or are tailored to special types of input data. The inductive biases of the architectures are discussed and related to applications in the presented publications.

### 3.2.1 Fully-Connected Networks and the Multi-Layer Perceptron

Fully-connected networks (FCNs) are NNs where the neurons in a certain layer receive input from all neurons in the previous layer. The response function corresponds to a dense matrix-vector product with subsequent offset addition and nonlinear activation (cf. Equation 3.2). A multi-layer perceptron (MLP) is an FCN with at least three layers. The first and last layers serve as input and output neurons, respectively, and additional

layers in between are called *hidden* layers. The number of neurons in the input layer is fixed and determined by the dimension of the (vector-valued) input signal. Similarly, the number of output neurons determines the output dimension. MLPs are used as standalone regression models or as building blocks in more complex network architectures.

MLPs are well suited for mapping continuous scalar- or vector-valued inputs to scalar- or vector-valued outputs. Applications often involve tabular data in which a data item represents a row in a data table, and the table columns contain the attributes for each item (Borisov et al. 2024). The concatenated attribute values are used as inputs or targets in the learning pipeline. MLPs are able to process categorical inputs if they are transformed into a vector-like format before the processing. MLP-based models are used in Hühlein et al. (2024b) for postprocessing weather predictions.

**Encoding categorical data** Categorical predictors require encoding to embed discrete category information in the context of the continuous-valued vector space of model inputs. A popular example is *entity encoding* (Guo and Berkhahn 2016), which maps category labels  $c \in \{c_1, c_2, \dots\}$  to  $d_{\text{emb}}$ -dimensional embedding vectors  $\gamma_c \in \mathbb{R}^{d_{\text{emb}}}$ . Each embedding takes the role of a pseudo-label for its respective category. The vector is treated as a learnable parameter and updated iteratively during training. A review of further encodings for categorical data can be found in Hancock and Khoshgoftaar (2020).

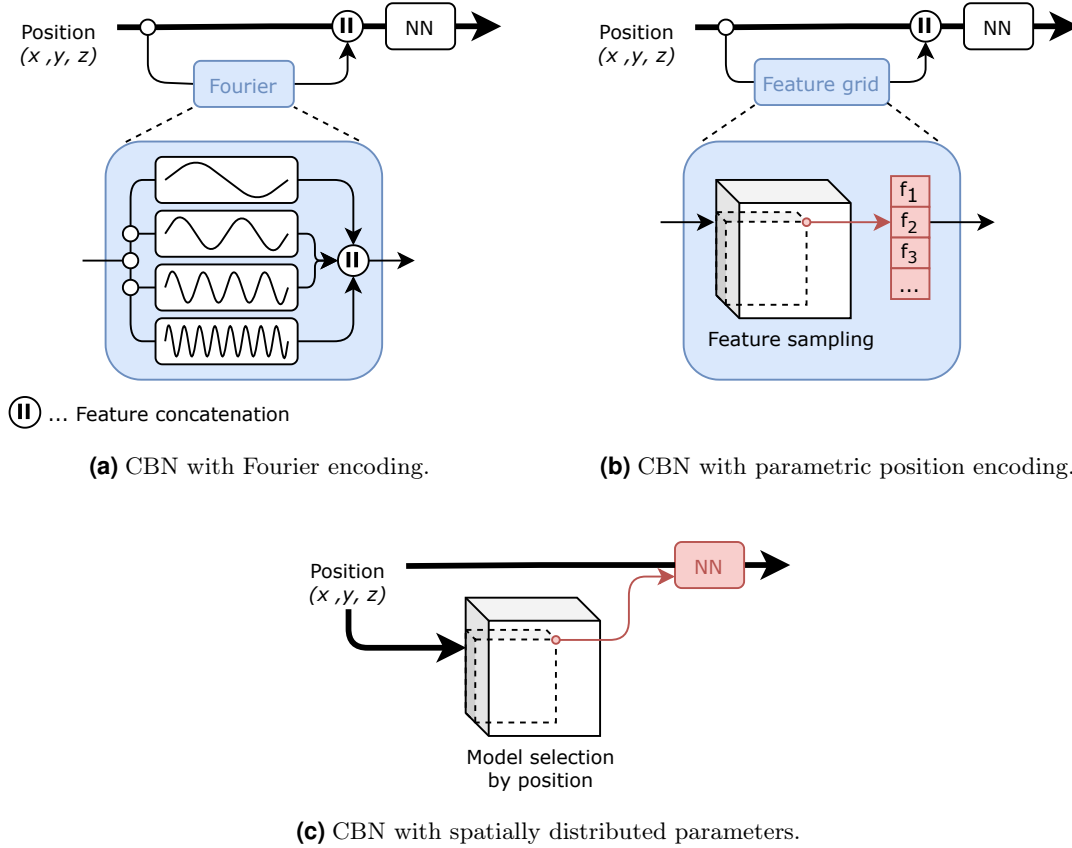
Rasp and Lerch (2018) use entity encoding in postprocessing models to represent weather station identifiers, which help the respective models distinguish between the stations. The presented strategy was adopted in Hühlein et al. (2024b).

**Low-frequency bias** When learning from data, FCNs and MLPs show a bias toward learning functions that vary slowly with changing input signals (Basri et al. 2020, 2019; Rahaman et al. 2019). Signal components with higher frequency are susceptible to noise and are extracted only after longer training times. Together with the biases of gradient-based training algorithms, which prefer wider local minima over narrower ones (Smith and Le 2018), this leads MLPs to implicitly prefer smoother explanations of the target function, potentially at the cost of reduced accuracy.

The smoothing bias is beneficial in applications where the input data are known to be noisy, enabling MLPs to generalize to unseen inputs despite considerable deviations from the training data. MLPs perform worse if the predictors are sharp and contain informative low-frequency signals that map to high-frequency components in the target function. MLPs may yield suboptimal representations even though the available predictors provide perfect information about the learning task (Rahaman et al. 2019). Preprocessing steps, such as feature rescaling, nonlinear feature transformations, or input embeddings for continuous predictors, can help boost the performance (Gorishniy et al. 2022).

#### 3.2.2 Coordinate-Based Networks

Coordinate-based networks (CBNs) are an MLP-based NN design that was developed in the computer vision community to learn compact representations for spatiotemporal signals on multi-dimensional domains (Mescheder et al. 2019; Müller et al. 2021; Sitzmann



**Figure 3.1:** Overview of common CBN architectures. (a) Basic MLP architecture with Fourier encoding, (b) MLP architecture with position encoding, and (c) distributed architecture with separate sub-models for different subdomains.

et al. 2019). CBNs predict scalar- or vector-valued target signals based on coordinate vectors that denote point locations in the scenery domain – e.g.,  $(x_1, x_2, x_3)^T \in \mathbb{R}^3$  for 3D spatial scenes. The outputs are field samples corresponding to the location of the input samples. CBNs rely heavily on MLP architectures to represent the target signal as a function of the input coordinates. As a consequence, CBNs inherit the low-frequency bias of MLPs. Scenery details, such as corners or sharp edges, are difficult to learn because discontinuous signals involve high-frequency components, which are dampened by the CBN’s bias. Dedicated activation functions, position encodings, and model decompositions have been developed to overcome these limitations. Position encodings translate the raw position coordinates into a set of higher-dimensional coordinates, which are provided to the MLP instead of or in addition to the raw coordinate vector. An illustration of the resulting CBN architectures is shown in Figure 3.1.

**Fourier encoding** A popular class of position encodings is based on Fourier modes (Mildenhall et al. 2021; Tancik et al. 2020). In 3D space, the Fourier feature  $\mathbf{f}_k(\mathbf{x}) \in \mathbb{R}^2$

with wave vector  $\mathbf{k} \in \mathbb{R}^3$  of a point  $\mathbf{x} \in \mathbb{R}^3$  is computed as

$$\mathbf{f}_{\mathbf{k}}(\mathbf{x}) = (\cos(\mathbf{k} \cdot \mathbf{x}), \sin(\mathbf{k} \cdot \mathbf{x}))^T, \quad (3.9)$$

with  $\cdot$  denoting the scalar product. The Fourier encoding combines Fourier features for different wave vectors, i.e., Fourier modes with different directions and frequencies (cf. Figure 3.1, a).

Given an orthogonal coordinate system in the scene domain, i.e., orthogonal axes with labels x, y, and z, respectively, Mildenhall et al. (2021) introduced the axis-aligned Fourier encoding with exponential frequencies. The wave vectors for these modes are

$$\mathbf{k}_j^{(i)} = 2^i \pi \hat{\mathbf{e}}_j, \quad (3.10)$$

wherein  $i \in \{0, 1, 2, \dots\}$ , and  $\hat{\mathbf{e}}_j \in \mathbb{R}^3$  are the axis-aligned unit vectors along  $j \in \{x, y, z\}$ .

Other authors reported better performance with randomized wave vectors  $\mathbf{k} \sim \mathcal{N}(0, \sigma^2)$  sampled from a zero-centered normal distribution and variance  $\sigma^2$ , selected as a hyperparameter of the model architecture (Tancik et al. 2020).

**Parametric position encoding and domain decomposition** In addition to non-parametric position encodings, empirical evidence suggests that CBNs can be trained faster if the model parameters are associated more directly with specific locations in space (Chabra et al. 2020). A variety of model designs have emerged, which place parts of the trainable model parameters in a spatially distributed data structure (parametric position encodings, cf. Figure 3.1, b), or decompose the scene domain and train separate models for the non-overlapping subdomains. For inference, the relevant model is sampled from a spatial data structure according to the input coordinates and evaluated accordingly (cf. Figure 3.1, c). Notably, the model layouts depend on the domain properties, and most work has focused on model designs for 3D spatial domains with or without time. A more detailed overview of the available architectures is deferred to section 4.2.

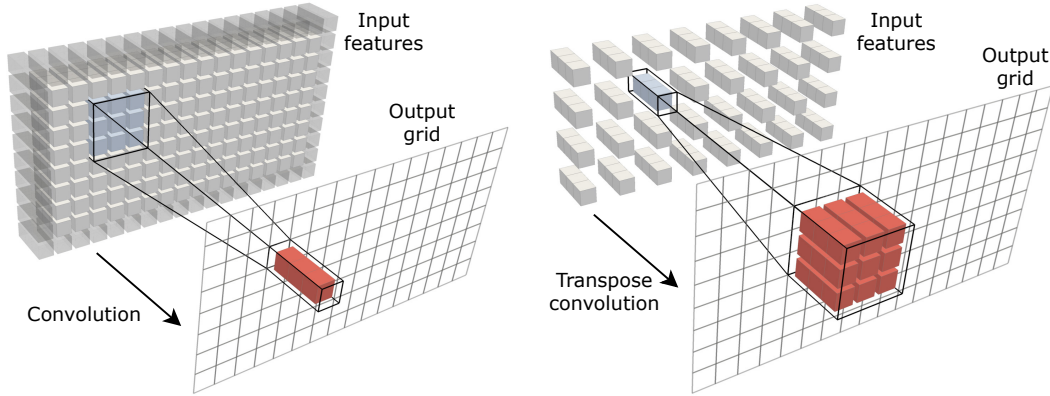
Fourier encoding and parametric encoding schemes are used in H3hlein et al. (2022) and Farokhmanesh et al. (2023b) to improve the models' capability of learning high-frequent features in the respective target signals.

### 3.2.3 Convolutional Neural Networks

Convolutional neural networks (CNNs) impose a spatial ordering on the neurons in each layer according to a rectangular grid. Neurons in subsequent layers are connected through convolution or pooling operations, which respect the spatial context. The dimension and size of the spatial grid depend on the learning task. Throughout this thesis, CNNs are used on 2D domains<sup>2</sup>.

For the 2D case, let  $N_x \times N_y$  denote the dimensions of the spatial grid, with  $N_x, N_y \in \mathbb{N}$ . Each grid position holds a fixed number of neurons, called the layer's channel number.

<sup>2</sup>CNNs work well also in 1D and 3D domains with applications, e.g., in time series processing (1D), video processing (2D and time), and learning in volumetric domains (3D). We focus on CNNs for 2D domains due to their relevance for the present work.



(a) Convolution with kernel size  $3 \times 3$ , stride 1, and one row of padding on all input sides (translucent boxes). (b) Transpose convolution with kernel size  $3 \times 3$  and stride 2.

**Figure 3.2:** Neuron arrangement and information propagation in 2D CNNs. Boxes illustrate groups of neurons at specific grid locations. Blue boxes indicate the inputs of a single kernel application, red boxes the outputs.

The grid dimensions and channel numbers may vary between layers. Denoting the channel number with  $C$ , the neuron activations in a CNN layer can be summarized in a multi-dimensional tensor  $\mathbf{A} \in \mathbb{R}^{N_x \times N_y \times C}$ , such that element-wise the value  $a_{ijc} := [\mathbf{A}]_{ijc} \in \mathbb{R}$  represents the activation of the  $c$ -th channel neuron at grid position  $(i, j)$ . The 2D activation maps per channel are called *feature maps*.

**Convolution layer** Considering now the  $l$ -th layer in a CNN, the activations  $\mathbf{A}^{(l)}$  of this layer are connected to the activations  $\mathbf{A}^{(l-1)}$  of the previous one. Similar to Equation 3.2, the neurons' response function is parameterized as a convolution operation, denoted  $*$ , with subsequent nonlinear activation:

$$\mathbf{A}^{(l)} = \rho\left(\mathbf{K}^{(l)} * \mathbf{A}^{(l-1)} + \mathbf{b}^{(l)}\right). \quad (3.11)$$

Therein,  $\mathbf{b}^{(l)} \in \mathbb{R}^{1 \times 1 \times C^{(l)}}$  is a bias vector, and  $\mathbf{K}^{(l)} \in \mathbb{R}^{K_x \times K_y \times C^{(l-1)} \times C^{(l)}}$  denotes the *kernel tensor* of the convolution. The kernel slices,  $[\mathbf{K}^{(l)}]_{\dots c}$ , for  $c \in \{1, \dots, C^{(l)}\}$  are  $K_x \times K_y$ -sized,  $C^{(l-1)}$ -dimensional *filter kernels* which are slid across incoming activation maps and correlated against the local signal<sup>3</sup>. Each activation depends on the inputs within a  $K_x \times K_y$ -sized window on the previous layer's grid. The kernel size is also referred to as the *receptive field* size of the convolution. Mathematical formulas are omitted here for brevity and can be found in textbooks like Goodfellow et al. (2016). An illustration of a single kernel application is shown in Figure 3.2 (a).

<sup>3</sup>The use of the term *convolution* in convolution layers involves is actually inaccurate in many DL library implementations, as convolutional layer are designed to execute a cross-correlation. The operations differ in the relative orientation of the filter and the data tensor.

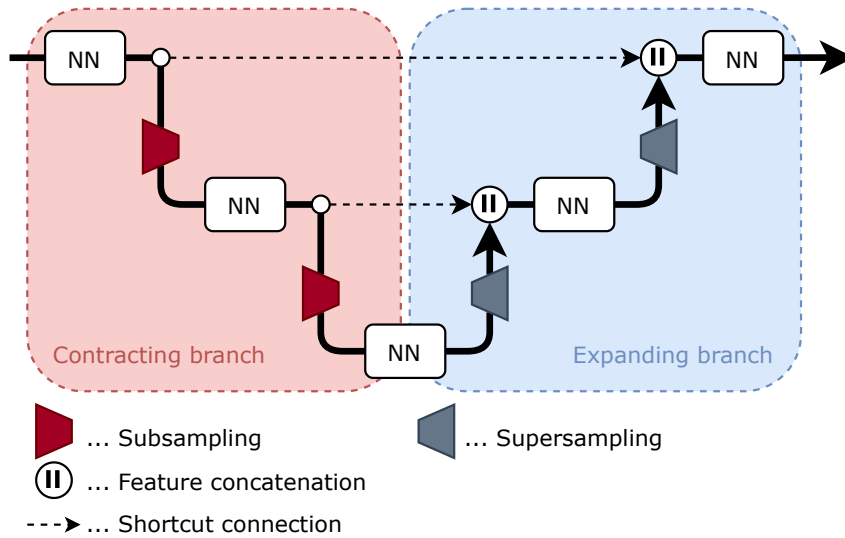
Notably, the shape of the activation tensor after the convolution depends on the convolution hyperparameters, namely the kernel dimensions, the *padding size*, and the *stride*. Convolution inputs are commonly padded to compensate for resolution losses at the grid boundary. Popular padding methods include zero-padding, i.e., extending the grid with nodes with value zero; replication padding, i.e., using information from neurons at the grid boundary to extend the grid; or padding with periodic boundary conditions. Kernel size and padding affect the loss of grid nodes at grid boundaries due to sample locations falling outside the grid. The *stride* of the convolution determines the size of steps between successive evaluations of the correlation between the kernel and the input activation maps. A stride parameter of  $s = 1$  evaluates the kernel correlation at every possible grid position, resulting in an activation map with roughly the same dimensions as the input tensor (up to boundary effects). Stride  $s > 1$  evaluates the correlations only at every  $s$ -th grid position, leading to a *resolution reduction* by a factor of roughly  $s$ . Strided convolutions help CNNs to resample activation maps, often accompanied by increasing the channel number. This allows the model to extract more abstract features at coarser spatial scales.

**Transpose convolution** Transpose convolutions (Zeiler et al. 2010), also called deconvolution or reverse convolutions, invert the information flow of a regular convolution operation. The kernel tensor has the same shape as in a regular convolution but is applied reversely. Each neuron in the input layer *broadcasts* its activation state to neurons in a window of the size of the convolution’s receptive field (cf. Figure 3.2, a). Striding and padding are applied in the convolution’s output layer, such that a stride of  $s > 1$  in the transpose setting leads to a *resolution enhancement* by a factor of roughly  $s$ .

**Pooling and resampling** In addition to convolutional layers, CNNs often comprise intermediate processing steps that allow for resolution adaptations between convolution layers without invoking learned parameters. Non-parametric resolution reduction is achieved through moving-window pooling operations, such as average or maximum pooling, with stride  $s > 1$ . Resolution enhancement can be realized by resampling the activation maps along the spatial dimensions, e.g., through nearest-neighbor-based, bilinear, or bicubic interpolation.

**Translation invariance and the local inductive bias** All processing operations in CNNs are designed to enforce (approximate) equivariance of the learned mapping wrt. spatial translations. Aside from potential information loss at the grid boundary or due to resolution changes, input transformations of the form  $a_{ijc} \mapsto a_{(i-\delta_x)(j-\delta_y)c}$ , for integer shift sizes  $\delta_x$  and  $\delta_y$ , result in an equivalent shift of the activation maps in each layer, but leave the channel-wise information unchanged. The finite kernel and window size of convolution and pooling operations further induce a local inductive bias on the learned mappings. Together, these properties allow CNNs to generalize well in learning tasks where spatial translations should not affect the prediction outcome. The use of fixed-size kernels additionally limits the number of trainable parameters. CNNs thus



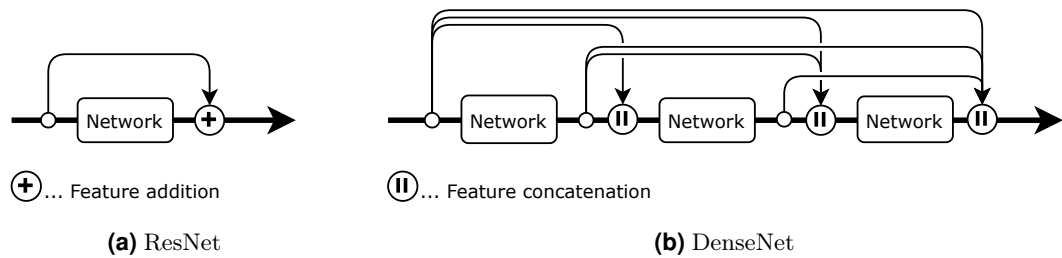


**Figure 3.3:** Schematic illustration of the U-Net principle with three resolution stages. Solid lines and arrows mark the information flow. NN modules operate at different resolution stages, separated by subsampling and supersampling operations, implemented, e.g., through strided and transpose convolutions. High-resolution information is preserved by forwarding feature tensors through shortcut connections between the contracting and expanding branches.

consume less memory space than MLP-based architectures with a comparable number of neurons and are computationally more efficient. Accordingly, CNNs have performed well in image-based reasoning tasks, such as image classification, image segmentation, or superresolution (see, e.g., Khan et al. 2020, for a survey on CNNs and their applications).

**Multi-scale architectures and U-Net** On the downside, CNNs face limitations in learning from *global* features in the input data. Due to the limited receptive field of each convolution layer, many layers, i.e., deep architectures, are required to relate information from different locations in grid space. To increase the receptive field of CNNs, studies have developed multi-scale CNNs that work with activation maps at different spatial resolutions. While high-resolution activation maps ( $N_{x/y}$  large) allow information extraction from local details, low-resolution activation maps ( $N_{x/y}$  small) enable information to propagate over larger spatial distances.

The most prominent implementation of the multi-scale approach is the *U-Net* architecture proposed by Ronneberger et al. (2015) for image segmentation. Inputs and outputs of the U-Net have approximately the same spatial resolution. To exploit the multi-scale idea, the U-Net architecture consists of two mirrored processing branches: the contracting (or encoding) and the expansive (or decoding) branch. A schematic illustration of the U-Net-like multi-resolution architecture is shown in Figure 3.3. The contracting branch (left side of the U) applies a series of small-kernel convolution operations with intermittent subsampling to successively reduce the spatial resolution of the activation maps.



**Figure 3.4:** Information flow in ResNet and DenseNet. NN blocks (Network) are wrapped by shortcut connections. While feature addition in the ResNet architecture maintains the dimensionality of the representation, repeated concatenation operations in the DenseNet architecture successively accumulate additional dimensions

With decreasing resolution of the activation maps, the spatial extent of a convolution’s receptive field increases. In the expansive branch, regular convolutions are interleaved with transposed convolutions that increase the resolution again. Each resolution level on the contracting branch has a corresponding level on the expansive branch, such that information from the former branch can be forwarded to the latter via a shortcut connection. These connections disentangle the information content of the different resolution levels and improve the model’s training efficiency.

Since its inception, U-Net-based architectures have been adopted in various applications. Specifically, the plain convolution operations have been replaced with more elaborate functional mappings, such as residual or attention blocks (see Khan et al. 2020, for examples), and U-Net-like subnetworks are frequently used as backbones in advanced learning frameworks, such as diffusion models (Croitoru et al. 2023).

The U-Net architecture was used in Höhle et al. (2020) as an alternative to other CNN architectures and outperformed the competitors.

### 3.2.4 Shortcut Connections

Deeper NNs with more layers are theoretically more expressive than shallower networks and often perform better in practical applications (Prince 2023). With increasing depth, however, the stability and effectiveness of the common training procedures decrease leading to training difficulties with very deep NNs (He et al. 2016). Shortcut connections in NN are a common way to resolve training instabilities and enable training of very deep and expressive networks (He et al. 2016; Huang et al. 2017).

Shortcuts deviate from the sequential layer structure of NNs and establish alternative paths on which information can propagate through the network while bypassing parts of it. The most prominent implementations of shortcuts are residual connections and dense shortcuts. Both approaches are illustrated in Figure 3.4.

**Residual connections** Residual connections establish a fast-forward connection between the network inputs and outputs. Given a network function  $h_\phi$ , the corresponding

residual network is defined through the function  $\bar{\mathbf{h}}_\phi$ , where for inputs  $\mathbf{x}$ ,

$$\bar{\mathbf{h}}_\phi(\mathbf{x}) = \mathbf{x} + \mathbf{h}_\phi(\mathbf{x}). \quad (3.12)$$

The network  $\mathbf{h}_\phi$  may be arbitrarily complex as long as its outputs have the same format as the inputs. In practice,  $\mathbf{h}_\phi$  often denotes a part of a larger network structure that is wrapped by a shortcut connection to obtain a *residual block*. Each block comprises a small number of densely connected or convolution layers and nonlinear activations. Normalization layers, such as batch normalization (Ioffe and Szegedy 2015), are sometimes added to improve the training stability further (Prince 2023). Deep residual networks are constructed by chaining multiple residual blocks.

**Dense shortcuts** Dense shortcuts operate on a series of sequential fully-connected or convolutional layers. Given a series of layer response functions,  $\{\mathbf{h}_{\phi^{(l)}}^{(l)}\}_{l=1}^L$ , as defined in Equation 3.2 and an input  $\mathbf{x}$ , a *dense block* is defined through the mapping

$$\begin{aligned} \mathbf{a}^{(1)} &= \mathbf{h}_{\phi^{(1)}}^{(1)}(\mathbf{x}) \\ \mathbf{a}^{(2)} &= \mathbf{h}_{\phi^{(2)}}^{(2)}\left(\text{concat}\left(\mathbf{x}, \mathbf{a}^{(1)}\right)\right) \\ &\dots \\ \mathbf{a}^{(L)} &= \mathbf{h}_{\phi^{(L)}}^{(L)}\left(\text{concat}\left(\mathbf{x}, \mathbf{a}^{(1)}, \dots, \mathbf{a}^{(L-1)}\right)\right), \end{aligned}$$

wherein  $\mathbf{a}^{(L)}$  is the final output, and  $\text{concat}(\cdot)$  denotes concatenation of the respective inputs. Dense blocks admit more extensive reuse of extracted features but require large amounts of memory and compute resources.

**Relevance of shortcuts** While the reasons for the efficiency of shortcut connections are not understood in detail, a key factor seems to be the shattered gradient phenomenon (Prince 2023). The term *shattered gradients* refers to the observation that the objective function gradients with respect to network parameters become increasingly noisy the earlier the parameters appear in the model. The noisy gradients carry less information and limit the efficiency of gradient-based training. Shortcuts enable a more direct propagation of gradient information and better parameter updates (Li et al. 2018).

Due to the advantages of shortcut architectures over purely sequential networks, shortcuts are commonly applied in various applications, including image superresolution (e.g., Kim et al. 2016; Lim et al. 2017) and volumetric scene representations (Lu et al. 2021). Residual connections are often preferred due to their ease of implementation and computational efficiency compared to dense shortcuts.

### 3.2.5 Attention and Transformers

Attention is a form of nonlinear activation function for learning tasks on inputs of varying sizes, such as sequences or sets of input items. To obtain a uniform representation format,

each information piece is decomposed into a sequence of *tokens*. Attention enables the translation of a sequence of input tokens of length  $N_{\text{in}}$  into a sequence of output tokens of length  $N_{\text{out}}$ . Each input token is associated with a  $D$ -dimensional *key* vector and a  $C$ -dimensional *value* vector. Outputs are computed as an attention-weighted average of the input *values* and require a  $D$ -dimensional *query* vector per generated token.

Given a tensor of token values  $\mathbf{V} \in \mathbb{R}^{N_{\text{in}} \times C}$  with associated keys  $\mathbf{K} \in \mathbb{R}^{N_{\text{in}} \times D}$  and queries  $\mathbf{Q} \in \mathbb{R}^{N_{\text{out}} \times D}$ , the attention activation is defined as

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) := \mathbf{A}\mathbf{V} \in \mathbb{R}^{N_{\text{out}} \times C}, \quad (3.13)$$

$$\text{where } \mathbf{A} = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{D}}\right) \in [0, 1]^{N_{\text{out}} \times N_{\text{in}}}.$$

Therein,  $\mathbf{A}$  denotes the tensor of attention weights, and Softmax is applied to normalize the weights along the input dimension. To increase the flexibility of the activation mapping, multiple key-query pairs are combined to obtain  $H$ -fold multi-head attention,

$$\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) := \text{concat}(\mathbf{H}_1, \dots, \mathbf{H}_i, \dots, \mathbf{H}_H) \mathbf{W}^O, \quad (3.14)$$

$$\text{where } \mathbf{H}_i = \text{Attention}\left(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V\right),$$

with learnable parameters  $\mathbf{W}_i^Q, \mathbf{W}_i^K \in \mathbb{R}^{D \times D/H}$ ,  $\mathbf{W}_i^V \in \mathbb{R}^{C \times C/H}$ , and  $\mathbf{W}^O \in \mathbb{R}^{C \times C}$ , and concat indicating concatenation along the channel dimension. Self-attention refers to a special case of input configuration where  $\mathbf{Q}$  and  $\mathbf{K}$  equal  $\mathbf{V}$ , i.e.,  $\text{MultiHead}(\mathbf{V}, \mathbf{V}, \mathbf{V})$ .

Compared to MLPs and CNNs, attention-based NNs are not limited to specific input sizes (as MLPs are) while still admitting densely connected information flow between all input tokens (as opposed to CNNs, which enforce locality in the grid domain). Attention-based modeling originated in natural language processing (Vaswani et al. 2017), where the tokens represent parts of words or grammatical structure descriptors in texts. The corresponding NN architecture that uses attention is called a *transformer*. Subsequent work has adapted transformers for other applications, such as computer vision (cf., e.g., Khan et al. 2022). In such contexts, the tokens represent image patches.

### 3.2.6 Neural Networks for Learning from Sets

Set-structured predictors,  $\mathbf{x} = \{\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_{|\mathbf{x}|}\}$ , that consist of multiple distinct but interchangeable items are challenging to represent in NN-based learning tasks. The number  $|\mathbf{x}|$  of items in each sample may vary and might not be known at training time. Furthermore, each item  $\boldsymbol{\xi} \in \mathbf{x}$  may possess an intrinsic structure, such as a vector or tensor format. This thesis considers the items as  $C$ -dimensional vectors.

**Permutation symmetries** As a defining property, set-structured predictors convey the same information, irrespective of the order in which the set items are processed. The output of an ML model should thus be the same for each ordering of the items or, if the model predictions are associated with specific input items, reflect the interchange of the items identically. The corresponding symmetries are called *permutation invariance*

if the model outputs remain unchanged under permutations of the predictor items and *permutation equivariance* if the permutation is applied identically to the outputs. This thesis focuses on permutation-invariant models that generate outputs not associated with specific input items.

Several strategies have been suggested to account for set-structured inputs in DL, including purely data-driven approaches, such as training on permuted predictors and algorithmic adaptations that enforce permutation invariance by design. Algorithmic solutions are often favorable and lead to better generalization and higher training efficiency of the models (Lyle et al. 2020). Two algorithmic approaches, *DeepSets* (Zaheer et al. 2017) and *set transformers* (Lee et al. 2019), are described here in detail and adapted in (Höhlein et al. 2024b) for learning from meteorological ensemble data.

**DeepSets** *DeepSets*, also called *set pooling architectures*, achieve permutation invariance by extracting permutation-invariant summary features from the predictor set. The features are obtained by first applying an encoder network  $\mathbf{f}_{\phi(f)}$  to all items separately, followed by a permutation-invariant pooling and subsequent interpretation of the pooled features through a decoder network  $\mathbf{g}_{\phi(g)}$ . The full model can be written as

$$\mathbf{h}_{\phi}(\mathbf{x}) = \mathbf{g}_{\phi(g)}\left(\text{pool}\left(\left\{\mathbf{f}_{\phi(f)}(\boldsymbol{\xi}) : \boldsymbol{\xi} \in \mathbf{x}\right\}\right)\right). \quad (3.15)$$

wherein pool is a permutation-invariant pooling function.

Pooling-type network architectures were introduced by Edwards and Storkey (2017) and investigated in more detail by Zaheer et al. (2017) and Sannai et al. (2019), proving that pooling architectures with additive pooling are universal approximators of functions on sets. More expressive pooling functions may enhance the performance (Soelch et al. 2019). An additional determinant of the model capacity is the dimension of the outputs of  $\mathbf{f}_{\phi(f)}$  and the resulting summary features. Higher-dimensional summary features allow the model to extract more information from the input set but support overfitting (Wagstaff et al. 2019).

**Set transformer** Self-attention is equivariant with respect to permutations of the input tokens. *Set transformers* (Lee et al. 2019) exploit this symmetry to model permutation-invariant interactions between set items via self-attention. Specifically, the models combine multi-head attention with an item-wise neural network  $\mathbf{h}_{\phi}$  and LayerNorm (Ba et al. 2016) to build a permutation-invariant set-attention block as

$$\begin{aligned} \text{SetAttention}(\mathbf{X}) &:= \text{LayerNorm}(\mathbf{R} + \mathbf{h}_{\phi}(\mathbf{R})), \\ \text{where } \mathbf{R} &= \text{LayerNorm}(\mathbf{X} + \text{MultiHead}(\mathbf{X}, \mathbf{X}, \mathbf{X})). \end{aligned}$$

Therein,  $\mathbf{X} := \text{concat}(\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_{|\mathbf{x}|})^T \in \mathbb{R}^{|\mathbf{x}| \times C}$  denotes the matrix of concatenated predictor items in arbitrary order. A set transformer is obtained by stacking multiple set-attention blocks. Set transformers constitute a flexible architecture for extracting item-wise feature vectors. In contrast to pooling architectures, the set structure is maintained throughout the inference process.

To obtain vector-valued predictions from set-valued inputs, Lee et al. (2019) propose an attention-based pooling mechanism in which the output query vectors are implemented as learnable parameters. An additional MLP can be added after pooling to increase the flexibility of the model outputs.

## 3.3 Machine Learning Explainability

While ML methods excel at extracting informative features and learning complex mappings from large amounts of data, human analysts often have difficulties understanding their reasoning principles. This is particularly true for complex DL models, which resist simple explanation methods due to their high-dimensional and nonlinear nature. ML explainability methods have been developed to address such concerns and widen the scope of ML deployment. Detailed reviews of methods for different model classes and data configurations have been conducted, e.g., by Linardatos et al. (2021), Sahakyan et al. (2021), Burkart and Huber (2021), and Zhang et al. (2021b). Model explanation aims to understand black-box algorithms by assessing their general logic or outcomes of specific predictions. Depending on the purpose of the explanation, different granularity levels are addressed (Guidotti et al. 2018).

**Sample-wise explanations** Sample-wise explanations are used, e.g., to assess the impact of different predictors within the context of one specific prediction. Among these approaches are Shapley values, which allow statements about the strength and direction with which different predictors contribute to a final prediction. Shapley values are rooted in game theory (Shapley 1951) and provide a unified view on various other popular explanation techniques (Lundberg and Lee 2017). For reasons of computational complexity, practical applications often rely on more targeted methods. Layer-wise relevance propagation (LRP; Ba et al. 2016), for instance, enables the computation of 2D relevance maps in CNN applications by exploiting the sequential structure of NN inference for efficient computations. Applications of such methods in the earth-system sciences include the works by Labe and Barnes (2021), Farokhmanesh et al. (2023a), or Rampal et al. (2022).

While attribution-based approaches are well suited for an in-depth investigation of the model inference, they usually come at high computational cost and provide information that is too detailed for higher-level tasks, such as comparing different ML algorithms or assessing the overall importance of different predictors.

**Model-level explanations** The DL applications considered in this study require statements about the relevance of different model inputs to the model performance. For such tasks, averaged importance scores of certain predictors offer more information. Feature permutation importance (PFI; Breiman 2001) is a popular method to access this kind of information. PFI is applied to models after training and works by measuring the performance reduction of models when withholding information about specific predictors. PFI methods have been used previously in meteorological applications (e.g., Rasp and Lerch 2018). In this work, PFI methods are adapted to measure the importance of 2D

field-valued predictors in CNN-based statistical downscaling (Höhlein et al. 2020) and to probe the relevance of features within ensemble-valued predictors (Höhlein et al. 2024b).

## 3.4 Deep Learning in Computer Vision

DL methods and NNs have advanced the state of the art in many fields of science. Many new methods were proposed initially to solve tasks in computer vision. This section introduces selected application areas that have impacted the research in this thesis.

### 3.4.1 Superresolution

Superresolution (SR) refers to a family of tasks concerned with increasing the resolution, i.e., the sampling density, of pixelated or voxelized visual data, such as rasterized images or discretely sampled scientific fields. Common to all these tasks is that some form of low-resolution (LR) data,  $\mathbf{x}^{(\text{LR})} \in \mathcal{X}^{(\text{LR})}$ , is used to generate a high-resolution (HR) version of it,  $\mathbf{x}^{(\text{HR})} \in \mathcal{X}^{(\text{HR})}$ . The case in which  $\mathbf{x}^{(\text{LR})}$  and  $\mathbf{x}^{(\text{HR})}$  are images is called single-image superresolution (SISR). This is opposed to multi-image superresolution, in which multiple LR images are used jointly to infer  $\mathbf{x}^{(\text{HR})}$ . Both types of methods are collectively referred to as image superresolution (ISR).

**Inverse problem formulation** Mathematically, SR is often phrased as an inverse problem, in which the unknown  $\mathbf{x}^{(\text{HR})}$  and the available  $\mathbf{x}^{(\text{LR})}$  are related through a (formally unknown) sub-sampling or degradation process,  $\text{Degrade} : \mathcal{X}^{(\text{HR})} \rightarrow \mathcal{X}^{(\text{LR})}$ , such that

$$\mathbf{x}^{(\text{LR})} = \text{Degrade}\left(\mathbf{x}^{(\text{HR})}\right). \quad (3.16)$$

Superresolution methods try to invert  $\text{Degrade}$  and provide estimates of what  $\mathbf{x}^{(\text{HR})}$  might have looked like. The degradation process usually involves a loss of information, such that it is impossible to restore  $\mathbf{x}^{(\text{HR})}$  perfectly. In the case of considerable information loss, e.g., due to a large resolution difference between  $\mathbf{x}^{(\text{LR})}$  and  $\mathbf{x}^{(\text{HR})}$ , even a *best guess* estimate may not be uniquely defined. To account for ambiguities, superresolution methods apply regularization schemes, use generative modeling, or incorporate prior information on the detail structure of  $\mathbf{x}^{(\text{HR})}$  to fill the estimate with plausible and realistic-looking details. In ISR applications, such prior information is called image priors (Tappen et al. 2003; Ulyanov et al. 2018).

**Methods** SR algorithm can be classified as deterministic or probabilistic approaches (Nasrollahi and Moeslund 2014). Deterministic approaches produce HR image estimates based on a deterministic function of the input images and can be described as MLE or MAP estimators. Common quality metrics are optimized for image data, including the peak signal-to-noise ratio (PSNR) and the structural similarity metric (SSIM, Wang et al. 2003). Probabilistic approaches use generative models, such as GANs, to emulate sampling from the posterior distribution. State-of-the-art methods rely heavily on DL

models, which are trained on extensive image datasets, containing paired examples of HR and LR images (cf., e.g., Dong et al. 2016a; Wang et al. 2021). A more detailed review of relevant methods is given in section 4.1.

**Superresolution for scientific data** In contrast to images, scientific data are concerned with scientific scalar or vector fields. The fields are commonly defined on continuous spatial or spatiotemporal domains, and scientific data are stored as discretely sampled values at predefined locations in space and time. Interpolation and resampling are required in various processing tasks to enable data access at arbitrary locations. SR methods for scientific data will be called scientific superresolution (SciSR) methods. Approaches in this direction originate commonly from the visualization or data science community (e.g., Jakob et al. 2021; Tang and Wang 2024; Wurster et al. 2023; Zhou et al. 2017) and are designed to resample general scientific data for analysis and visualization.

#### 3.4.2 Neural Scene Representations

Computer vision and graphics applications rely heavily on digital representations of real-world 3D scenery information. Scenery descriptions contain information about the location, shape, and appearance of objects within the scene and are crucial for digital image generation, scene understanding, or scene editing. Neural scene representations are a promising alternative to classical representations based on explicit 3D geometry and lighting and object appearance models. While classical representations are often memory intensive and require detailed supervision by artists in the capturing or design process, neural scene representations enable concise representations of complicated 3D scenes and facilitate streamlined optimization-based procedures during scene generation.

**Basic algorithms** Neural scene representations encode 3D scenery information as a mapping between algorithm-specific scenery coordinates and scene properties at the specified location. The mapping is implemented through a NN, which is often a CBN. Scene information is stored implicitly within the parameters of the NN, and the model outputs are used in classical or dedicated neural rendering algorithms to visualize the scene. Neural scene representations were pioneered by Sitzmann et al. (2019), who trained CBN-based Scene Representation Networks (SRNs) to encode the shape and appearance information of objects in 3D visual scenes. Their work was published concurrently with similar approaches (Chen and Zhang 2019; Mescheder et al. 2019; Michalkiewicz et al. 2019; Park et al. 2019), which used CBNs as generative models for 3D shapes.

SRNs output high-dimensional feature vectors, which are interpreted in a subsequent neural rendering step to yield images of the encoded scenery. Alternative rendering methods were proposed by Mildenhall et al. (2021) and Sitzmann et al. (2021), leading to the development of Neural Radiance Fields (NeRFs) and Light Field Networks (LFNs), respectively. In NeRFs, the CBN generates four-dimensional outputs, interpreted as three color channels and a scalar opacity value. Volume rendering is applied to the resulting color-opacity field to obtain visual output. The rendering procedures require many sequential network evaluations per pixel, which are computationally costly and



Representation	SRN	NeRF	LFN
Model input	3D coords.	3D coords. & 2D view dir.	6D ray coords.
Model outputs	high-dim. features	3D color & opacity	3D color
Rendering algorithm	neural rendering	raycasting	model query

**Table 3.3:** Comparison of neural scene representations wrt. model configurations used in the different approaches.

memory-intensive. LFNs alleviate such problems by emulating a representation of the light field in the scene and reducing the number of CBN evaluations. A comparison of the respective algorithms regarding model inputs, outputs, and rendering algorithms is shown in Table 3.3. A more detailed review can be found, e.g., in the survey article by Xie et al. (2022).

Neural scene representations can be fitted to classical digital scenery models or real-world scenes. During the training process for existing digital models, both scenes are rendered, and the renderings are compared through suitable loss functions. The parameters of the neural scene are tuned to minimize the deviation between both renderings. Differentiable rendering algorithms are required for the neural scene to enable gradient-based parameter optimization. For real-world scenes, the optimization is commonly based on photographic image data. Neural scene representations commonly encode one scene at a time, i.e., they are overfitted to a specific scene, or require additional parametric inputs to determine the rendered scene.

**Efficiency considerations** The representation quality, runtime performance, and memory efficiency of neural scene representations are critically determined by the architecture of the used CBNs and by the practical implementation of the model evaluation on computing hardware.

The representation quality of neural scene representations was shown to profit considerably from using sinusoidal activation functions (e.g., Sitzmann et al. 2020; Weiss et al. 2021; Ziyin et al. 2020) and from positional encodings applied to the spatial coordinate inputs (e.g., Mildenhall et al. 2021; Tancik et al. 2020). Both adaptations improve the models’ ability to represent high-frequency visual details in the scene by circumventing the MLP-specific low-frequency bias (cf. subsections 3.2.1 and 3.2.2). The parameter and learning efficiency, as well as the evaluation time, have been improved through parametric coordinate embeddings (cf., e.g., Chabra et al. 2020; Martel et al. 2021; Müller et al. 2022; Takikawa et al. 2021, and subsection 3.2.2). Parametric embeddings enable the use of very small MLPs, which, in turn, better exploit acceleration structures on GPU hardware (Müller et al. 2022; Weiss et al. 2021). When combined into one model, such improvements can enable real-time rendering speeds of the neural representation with a reduced memory footprint compared to traditional representations.



## CHAPTER 4

---

### Related Work

---

In this chapter, we review related work relevant to the context of the presented publications. The topics are grouped by research area, covering research on SR, neural scene representations, postprocessing, and the visual analysis of meteorological data. The relevance of the different subsections to the presented publications is outlined in Table 4.1.

### 4.1 Superresolution

SR methods for images and scientific data are closely related to the work in Höhle et al. (2020). Following the orientation of the work therein, we focus on the use of CNNs for SR and highlight advancements and current trends in the design of the employed models.

#### 4.1.1 Image Superresolution

Following the far-reaching adoption of DL methods in computer vision, the landscape of ISR algorithms has witnessed significant changes. While classical approaches focused on image interpolation, kernel methods, or simple regression schemes, virtually all recent ISR approaches rely on DL. The comprehensive reviews by Nasrollahi and Moeslund (2014) and Anwar et al. (2020) illustrate the shift in focus. A turning point is marked by the introduction of CNN-based SISR algorithms (Dong et al. 2016a,b; Kim et al. 2016), which lead to substantial improvements in image quality and runtime compared to earlier methods. The exceptional performance of CNNs in SR tasks was attributed to their inductive bias, which provides a favorable image prior (Chen and Zhang 2019; Ulyanov et al. 2018).

**CNNs for image superresolution** SRCNN (Dong et al. 2016a) was the first CNN-based model for SISR. SRCNN uses a shallow 3-layer network architecture with kernel

#### 4 Related Work

Article	Topic	Subsection
Höhlein et al. (2020)	Superresolution	section 4.1
	Statistical downscaling	subsection 4.3.1
Höhlein et al. (2024b)	Ensemble postprocessing	subsection 4.3.2
	Model explanation	subsection 4.4.3
Höhlein et al. (2024a)	Statistical downscaling	subsection 4.3.1
	Topographic visualization	subsection 4.4.1
Höhlein et al. (2022)	Neural scene representations	section 4.2
	Ensemble visualization	subsection 4.4.2
Farokhmanesh et al. (2023b)	Neural scene representations	section 4.2
	Ensemble visualization	subsection 4.4.2

**Table 4.1:** Overview of relevant sections in the related work for the presented articles.

sizes of  $9 \times 9$ ,  $1 \times 1$ , and  $5 \times 5$  pixels, respectively, and operates on a bicubic interpolated version of the original LR image (iLR) to predict the HR image. VDSR (Kim et al. 2016) improved on SRCNN by training deeper networks with up to 20 layers and uniform kernel size  $3 \times 3$  while copying the strategy of postprocessing the iLR image. The smaller kernels help save compute operations, resulting in better runtime performance. To account for problems with exploding and vanishing gradients, Kim et al. (2016) suggest *residual learning*. For example, instead of predicting the HR image directly, VDSR is trained to predict a residual difference between HR and iLR. FSRCNN (Dong et al. 2016b), subsequently, used similarly deep networks together with additional architectural refinements, such as pixel-wise convolutions, to improve data efficiency and runtime.

Subsequent research optimized the architectures further or explored additional design patterns. Notable examples include the works by Lim et al. (2017) and Ahn et al. (2018), who use ResNet-like residual blocks for ISR. Lai et al. (2017) process features in multiple resolution stages, thus splitting the superresolution process into simpler steps with a smaller upsampling factor. REDNet (Mao et al. 2016) employs a U-Net-like encoder-decoder architecture, which allows the model to infer feature representations at multiple resolution levels and improves runtime performance.

Anwar et al. (2020) identify residual connections as an important aspect in ISR model design and distinguish two types of residual learning. Global residual learning refers to VDSR-style prediction of HR image residuals, whereas local residual learning concerns ResNet-like shortcut connections inside the network architecture.

The above work guided the selection of network architectures in H ohlein et al. (2020). The presented model architectures exploit residual connections, residual learning, and multi-scale information processing, which are also reflected in the downscaling architectures in our comparison.

**Generative image superresolution** All of the above architectures generate *deterministic* HR estimates and are trained using MAE- or MSE-based loss functions. Beyond this, several studies have explored using DGMs for SR tasks. In particular, GANs have been a popular choice. Examples of such models include Enhancenet (Sajjadi et al. 2017), SRGAN (Ledig et al. 2017), ESRGAN (Wang et al. 2019b), or SRFeat (Park et al. 2018). The GAN-based approaches use generator architectures similar to those of the deterministic models. SRGAN and Enhancenet, for instance, are based on ResNet blocks. Similar approaches have been used in visualization research to achieve image-space SR for scientific visualizations (e.g., Weiss et al. 2021).

To avoid excessive deviations from the target image distribution, GAN training is often regularized using global residual learning (e.g., Sajjadi et al. 2017) or by adding additional pixel-wise or feature-based loss components to the training objective. More recently, studies have explored alternative generative model classes, such as normalizing flows (Lugmayr et al. 2020) or diffusion models (Li et al. 2022). Training methods and model architectures of these approaches are determined by the requirements of the learning algorithms and deviate significantly from the CNN-based regression methods. GAN-based approaches were therefore excluded from the comparison in H ohlein et al. (2020), but have been used frequently in subsequent downscaling studies.

#### 4.1.2 Superresolution for Scientific Data

Compared to image data, scientific data vary largely wrt. statistical properties and value ranges. Scientific data are also multi-faceted (Kehrer and Hauser 2013), i.e., datasets may be multi-dimensional (often 2D or 3D space and time) or multivariate, and the data may comprise scalar- and vector-valued components. SciSR approaches have to account for such properties or may even take advantage of them. This section thus discusses SciSR methods for scalar-valued and vector field data.

**Superresolution for scalar fields** Zhou et al. (2017) were the first to train a CNN to increase the resolution of volumetric scalar fields. Their approach is close in spirit to SRCNN (Dong et al. 2016a) while replacing 2D image convolutions with 3D convolutions to account for the volumetric domain. In a series of works, Han and Wang suggested GAN-based models for spatial (Han and Wang 2022) and temporal (Han and Wang 2020) superresolution of time-varying vector fields, as well as a framework for joint superresolution in both space and time (Han et al. 2022). All works focus specifically on the temporal coherence of the SR estimates, which is achieved through recurrent network architectures or dedicated regularization losses. Notably, many of the presented studies operate on volumetric fields and time-varying data, whereas the data in H ohlein et al. (2020) is two-dimensional.

## 4 Related Work

Several more recent studies have shifted the focus away from CNN architectures. Examples include the hierarchical model by Wurster et al. (2023), the normalizing flow model by Shen and Shen (2023), and the CBN-based model by Tang and Wang (2024).

**Superresolution for vector fields** While vector fields – such as wind fields – are not univariate, in terms of data dimension, the vector components, e.g., in flow fields, are often strongly correlated. Vector fields also possess additional theoretical structure, such as topological features, and admit a more profound analysis than a plain collection of scalar fields, e.g., in terms of streamlines, divergences, or rotation fields. The work presented in this section is closely related to the setting of Höhle et al. (2020), which has been an early contribution to the field.

Similar to our study, Guo et al. (2020) present SSR-VFD, a CNN-based solution for super-sampling volumetric vector fields. While each vector component is processed by a separate model branch (one branch per vector component), the objective function considers the predictions for all components jointly. The loss function is based on a linear combination of magnitude and cosine-based angle losses, as used in the evaluation of our models. Several authors have followed up on their work by proposing other refined loss functions (An et al. 2021; Sahoo and Berger 2021).

Among vector fields, fluid flows have attracted particular attention due to their high relevance in many areas of science and engineering. Contributions include the work by Xie et al. (2018), who explore the use of targeted GAN architectures for temporally coherent fluid flow SR, as well as Pant and Farimani (2021) and Fukami et al. (2021), who study multi-scale CNN and U-Net-based architectures for fluid flow SR. A more comprehensive review of machine learning approaches in fluid flow superresolution has been conducted by Fukami et al. (2023). However, contributions from the fluid flow community often focus on idealized flow settings in closed systems, where boundary conditions and obstacles strongly determine the structure of the flow. Such assumptions do not necessarily apply to atmospheric flows on length scales of tens or hundreds of kilometers and with sparse data sampling on kilometer-scale grid resolutions.

## 4.2 Neural Representations for Scenes and Scientific Data

Neural scene representations constitute an important inspiration for the development of ERNs (Höhle et al. 2022) and NDFs (Farokhmanesh et al. 2023b) as efficient data representations of 3D ensembles. Overviews on the fundamentals of neural scene representations and neural rendering have been provided, e.g., by Xie et al. (2022), Gao et al. (2023), and Tewari et al. (2022). A comprehensive review of progress in the field is beyond the scope of this work. Instead, we emphasize work addressing the memory and runtime efficiency of neural scene representation, and studies applying neural scene representations to scientific data.

**Parametric positional encodings and spatial decomposition schemes** Key improvements to the training and runtime efficiency of neural scene representations are

due to the introduction of parametric coordinate embeddings. Early neural shape representations, such as the models by Park et al. (2019) and Mescheder et al. (2019), used high-dimensional feature vectors to generate different shapes with a single model. The feature vectors were learned during training in parallel with the MLP model parameters, laying the foundation for auto-decoder CBNs with parametric coordinate encodings. Chabra et al. (2020) and Jiang et al. (2020) used multiple feature vectors per object to encode information about the object’s shape in local subdomains. The locality-specific feature vectors were stored in a regular sparse voxel grid, thus constituting an early form of parametric positional encoding. However, the required grid resolution grows quickly for fine-granular shapes or sceneries with dense output fields. Müller et al. (2022) overcome this limitation through multi-resolution hash grids, which provide a fixed budget of feature vectors, which are reused in different grid locations. The feature vectors are stored in a fixed-size lookup table, and a hashing function is used to connect the features to the 3D-indexed grid locations. While the non-unique grid assignment leads to potentially ambiguous update signals for the feature vectors during training, the procedure was found to help distribute the model capacity locally where needed. The expressive feature grid enables the authors to shrink the subsequent feature decoders considerably, enabling the exploitation of hardware acceleration structures for evaluating the neural scene representations (Müller et al. 2021).

Some of the above concepts are adopted in the work presented as part of this thesis. In Höhle et al. (2022), we employ fixed-resolution regular feature grids, similar to those by Chabra et al. (2020) and Jiang et al. (2020). In Farokhmanesh et al. (2023b), multi-resolution hash grids together with a small MLP-based feature encoder improved the model performance and were adopted in the model design.

**Parametric encodings for higher-dimensional coordinates** High-dimensional coordinate spaces are difficult to cover with grid-based parametric encodings. For instance, dynamic scenes involve 4D space-time coordinates, and radiance fields require the specification of 3D spatial and 2D (or more) view direction coordinates. While feature grids have been applied in such scenarios (e.g., Fridovich-Keil et al. 2022; Yu et al. 2021), the resulting feature grids require large amounts of memory, even in sparse configurations. Hash grids alleviate the memory problem but remain vulnerable to insufficient sample coverage during training (Müller et al. 2022).

Chen et al. (2022) realized that feature grids in high-dimensional coordinate spaces can be interpreted as multi-dimensional tensors. To reduce the memory size and number of degrees of freedom, the authors apply a tensor decomposition (Kolda and Bader 2009) and represent the full-grid feature tensor as a sum of lower-rank tensor products. A 3D tensor is obtained from the outer product of three 1D vectors or a 2D matrix and a 1D vector, resulting in linear or quadratic memory complexity, respectively, when increasing the grid resolution. Jang and Kim (2022) use tensor products of four 1D vectors or two 2D matrices, respectively, to generalize the approach to 4D feature tensors and dynamic scene representations. Similar approaches have been suggested by Fridovich-Keil et al. (2023) and Cao and Johnson (2023). Both works project high-dimensional coordinates to a set

## 4 Related Work

of axis-aligned 2D planes and extract features from 2D grids – i.e., matrices – embedded in these planes. The features from different planes are combined using multiplication. A similar representation was used by Chan et al. (2022) to represent data on 3D volumetric grids but with an additive feature combination. To further reduce memory consumption, Fridovich-Keil et al. (2023) suggest using multi-scale grids in all planes, similar to the multi-resolution hash grids by Müller et al. (2022).

In Farokhmanesh et al. (2023b), we follow the idea of tensor decompositions to parameterize the feature grid for the 6D feature space spanned by a pair of 3D point coordinates – the query points for the two-point dependencies. The features are sampled independently for both query points from a 3D multi-resolution hash grid, then transformed by a small MLP, and subsequently combined by multiplication. The strategy corresponds to a tensor decomposition of the 6D grid into a pair of 3D grids. This decomposition scheme simplifies the implementation of symmetries into the learned two-point dependence fields. Symmetric model outputs are obtained by reusing the same 3D feature grid for both query locations. 2D-based decompositions, as suggested by Fridovich-Keil et al. (2023) and Cao and Johnson (2023), require more grid evaluations and additional symmetry constraints on some of the planar grids.

**Neural representations for scientific data** Early DL approaches for scientific data representation used NNs as feature extractors. Jain et al. (2017), for instance, derived compressed representations for time-varying volumetric data using an encoder-decoder NN architecture, achieving compression ratios between 10 and 20. Larger compression ratios of up to 200 can be achieved through subsampling and subsequent superresolution (Han et al. 2022; Wurster et al. 2023), but remain less performant than CBN methods.

Other approaches skip the representation of the volumetric dataset and generate visual output directly. For instance, Berger et al. (2019) use GANs to directly generate rendered images, given view and transfer function parameters for color and opacity, and He et al. (2020) facilitate parameter space exploration in numerical simulation experiments by predicting renderings of the data. Similarly, Shi et al. (2023) support the generation of visuals based only on simulation hyperparameters.

CBN-based representations for volumetric scalar fields were introduced by Lu et al. (2021). The authors use a monolithic MLP architecture called Neurcomp without parametric positional encoding. Despite slow evaluation performance, the authors demonstrate good representation quality at compression ratios up to 1000 and more. Weiss et al. (2022) introduced fV-SRN as an accelerated alternative to Neurcomp. Key improvements include a positional encoding based on Fourier features, a dense parametric feature grid, and an optimized implementation of the neural representations on recent GPU hardware. This enables real-time volumetric rendering directly out of the compact representation. The work in Höhle et al. (2022) is inspired strongly by the contributions by Weiss et al. (2022) and reuses parts of the code base of this work.

Neural representations have also been developed for ensemble datasets. Notably, Shi et al. (2022) present a model to emulate the output of ocean simulations based on simulation input parameters, and Han and Wang (2023) present a parameter-conditioned



volume representation with applications to SciSR and volume visualization. We note that several studies with a focus on ensemble parameter space exploration assume an injective relationship between simulation hyperparameters and ensemble members (e.g., He et al. 2020; Shi et al. 2023; Wu et al. 2023). The methods presented in these studies do not account for the stochastic sampling dimension of NWP ensemble forecasts and thus address a different notion of ensemble datasets compared to H ohlein et al. (2022) and Farokhmanesh et al. (2023b).

## 4.3 Postprocessing

### 4.3.1 Statistical Downscaling

ML-inspired downscaling approaches have been adapted for multiple application scenarios, such as the downscaling of coarse-scale numerical model outputs (Maraun and Widmann 2018) or remote sensing data (Atkinson 2013; Sdraka et al. 2022). Relevant to this work are approaches that enhance the resolution of gridded NWP or climate model outputs, approaches for downscaling wind predictions, and grid-to-point approaches focusing on temperature downscaling in complex terrain.

**CNNs for regridding prediction data** Due to the striking similarity between ISR and statistical downscaling, many authors have taken inspiration from ISR to design CNN-based downscaling models. DeepSD (Vandal et al. 2017) was the first of these approaches, using a CNN in the style of SRCNN to downscale gridded precipitation predictions. To adapt the architecture for meteorological purposes, Vandal and coauthors use multiple stacked instances of SRCNN and add high-resolution orography information as additional predictors into the model. Similar studies include the works by Passarella et al. (2022), who adapt FSRCNN to downscale precipitation, temperature, and solar irradiation maps, Stengel et al. (2020), who build upon SRGAN to downscale wind fields and solar irradiation, Cheng et al. (2022), using ESRGAN for precipitation downscaling, and Serifi et al. (2021), who generalize REDNet to downscale precipitation and temperature predictions. Other authors deviate from the established ISR architectures. The works by Pan et al. (2019) and Ba no-Medina et al. (2020) use convolution layers for feature extraction and apply dense layers to generate the downscaled prediction. Mu et al. (2020) propose a CNN-based multi-scale model with residual connections to account for multi-scale spatial correlation patterns.

The work by H ohlein et al. (2020) follows this line of research and compares a selection of ISR-inspired CNN-based downscaling models regarding their prediction accuracy. The best-performing model is a U-Net-like multi-scale architecture with residual blocks. U-Net-based model designs have been popular in several subsequent studies, as well (Doury et al. 2023; Serifi et al. 2021; Sha et al. 2020a,b).

**Downscaling wind fields in complex terrain** While many studies downscale temperature and precipitation, the work presented in H ohlein et al. (2020) addresses wind field

## 4 Related Work

downscaling in complex terrain. Due to the tight coupling between terrain shape and wind dynamics, wind downscaling requires carefully integrating high-resolution terrain information into the downscaling models. Traditional downscaling approaches, such as the works by Fiddes and Gruber (2014) and Forthofer et al. (2014), apply physics-based models for downscaling winds. Earlier statistical methods, such as the approaches by Winstral et al. (2017) and Helbig et al. (2017), use carefully engineered feature descriptors to encode local terrain shapes. Höhle et al. (2020), instead, leverage the feature extraction capabilities of deep CNN architectures. Other CNN-based approaches follow similar strategies (e.g., Stengel et al. 2020; Zhang and Li 2021). WindTopo by Dujardin and Lehning (2022) is another CNN-based approach but operates in a grid-to-point setting. Nevertheless, the authors put particular emphasis on feature extraction from auxiliary data. Next to terrain information, the model receives several additional meteorological predictors, and features are extracted separately for each predictor.

**Downscaling temperatures in complex terrain** The work presented in this thesis is inspired by the earlier study by Sheridan et al. (2010). The authors study the occurrence and estimation of non-adiabatic ambient lapse rates and suggest a linear height-based correction scheme similar to the one by Höhle et al. (2024b). The original downscaling scheme has been combined with more elaborate methods that focus on physical parameterizations for cold-pooling effects in valleys and on parameterizations for hill-site locations (Sheridan et al. 2014; Sheridan et al. 2018; Smith et al. 2010; Vosper and Brown 2008). Physics-based parameterizations are not used in the dynamical scheme discussed here to keep the approach as simple and scalable as possible.

Downscaling of surface temperatures has been addressed independently in several other works. Frei (2014) and Hiebl and Frei (2016), e.g., suggest fitting nonlinear temperature profiles from which temperatures are interpolated at the target altitude. While the approach offers greater flexibility for the shape of the vertical temperature dependence, the methods rely on observation data, require intricate fitting and clustering procedures to work, and do not scale to global reanalysis applications. Uboldi et al. (2008) and Lussana et al. (2018) describe 3D spatial interpolation methods for surface temperatures in regional-scale reanalysis and climate applications. The approach fits multiple parametric functions to local subregions, which is computationally more involved than ours. All such approaches, furthermore, use observation data, which is scarce in many regions of the world. Volumetric temperature model data on pressure levels are used by Luo et al. (2019) and Fiddes and Gruber (2014) but require potentially expensive memory access operations when deployed for global domains. Fiddes and Gruber (2014) and, subsequently, Fiddes et al. (2022) focus on very-high-resolution downscaling down to hillslope-scale resolution and operate at the border between empirical statistical downscaling and full dynamical modeling.

### 4.3.2 Ensemble Postprocessing

Statistical postprocessing is an important component of operational forecasting pipelines, enabling the delivery of seamless forecasts for end users. A broad literature base covers

application studies of postprocessing across different variables and geographic regions. A comprehensive review of studies in the field has been conducted by Vannitsem et al. (2021) but is beyond the scope of the present work. Instead, we focus on methodological contributions and studies that suggest original postprocessing algorithms based on ML and DL techniques related to the methods in H ohlein et al. (2024b).

Two of the first statistical methods for postprocessing ensemble forecasts are ensemble model output statistics (EMOS, Gneiting et al. 2005) and Bayesian model averaging (BMA, Raftery et al. 2005). While EMOS performs a distributional regression based on a suitable family of parametric distributions and summary statistics of the ensemble, BMA generates a mixture distribution based on the individual ensemble members. Due to its simplicity, EMOS has been applied to a wide range of weather variables (Pantillon et al. 2018; Scheuerer 2014; Schulz et al. 2021; Thorarinsdottir and Gneiting 2010) and is commonly considered as a baseline approach in comparison studies (e.g., Demaeyer et al. 2023; Schulz and Lerch 2022). Basic statistical methods were succeeded by machine learning approaches, e.g., based on regression trees (Messner et al. 2017; Taillardat et al. 2016), which incorporate information about auxiliary meteorological variables or spatio-temporal context.

The first approaches based on NNs are DRNs (cf. subsection 3.1.2) for postprocessing as an extension of the EMOS framework (Rasp and Lerch 2018), and the Bernstein quantile network (BQN; Bremnes 2020), which use a more flexible parameterization of the output distribution and an adapted loss function. A comprehensive selection of standard ML-based and NN-based postprocessing approaches was evaluated by (Schulz and Lerch 2022) in the context of wind gust postprocessing. A broader review of statistical postprocessing techniques for weather forecasts is found in Vannitsem et al. (2021). Following the success of the first NN-based postprocessing methods, research has focused on further developing these approaches. An important aspect has been integrating multivariate and spatial information for postprocessing. Several common NN architectures have been explored in this context, including CNNs (Gr onquist et al. 2021; Horat and Lerch 2024; Scheuerer et al. 2020; Veldkamp et al. 2021) and graph-based NNs (Feik et al. 2024).

Concurrently with our work, several authors explored transformer NNs for ensemble postprocessing. Most similar to our work, Mlakar et al. (2024) evaluate an attention-based NN architecture and achieve the overall best postprocessing performance in a comparison study by Demaeyer et al. (2023). Their work focuses on a single hand-crafted architecture rather than a comparison of different symmetry-motivated architectures as considered in H ohlein et al. (2024b).

All works discussed above postprocess scalar-valued predictions of meteorological parameters at station sites. Finn (2021) and Ben-Bouall egue et al. (2023), instead, postprocess gridded forecast fields as a special case of multivariate forecasts. Ongoing work extends the focus on this aspect (e.g. Chen et al. 2024; Lerch et al. 2020)

## 4.4 Visual Analysis of Meteorological Data

Several of the articles in this thesis involve the use of visual analysis approaches or develop methods to facilitate the visual analysis of meteorology-related datasets. Visualization methods for meteorological and climate data have been discussed comprehensively in several textbooks (e.g., Andrienko and Andrienko 2006; Hoffman et al. 2023; Middleton et al. 2005; Monmonier 2000), surveys, and review articles (e.g., Afzal et al. 2019; Nocke et al. 2008; Rautenhaus et al. 2018; Röber et al. 2021). A comprehensive review is beyond the scope of this work. In this section, we highlight selected aspects of visualization and visual analytics research that have influenced the presented contributions.

### 4.4.1 Visualizing Weather Data in Their Spatial Context

**Guidelines for visualizing weather and climate data** Based on a user study with domain scientists, Dasgupta et al. (2015) discuss trade-offs between design problems in visualizing weather and climate data and provide guidelines for designing effective visualizations. Notably, the considerations about the encoding of data items apply to the work in Hühlein et al. (2024a). Dübel et al. (2014) categorize visualization methods for spatially referenced data based on the dimensionality of the visual representations, i.e., 2D or 3D visualizations of the spatial reference space and 2D or 3D embeddings of abstract data therein. Both 2D and 3D views are commonly used by weather and climate researchers (cf., e.g., Afzal et al. 2019; Rautenhaus et al. 2018) and are supported in the presented visualization tool. Helbig et al. (2014) discuss the advantages of 3D visualizations for weather and climate data and suggest a workflow for composing informative visualizations. Emphasis is put on how 2D and 3D datasets are composed and represented visually, as well as on how color schemes can help to encode relevant information. The authors provide guidelines for representing volumetric scalar fields and recommend slice-based representations as used in Hühlein et al. (2024a).

Guidelines for using colors in meteorological visualizations have been provided by the American Meteorological Society (1993) and refined later by, e.g., Stauffer et al. (2015). Important considerations include the familiarity of the visualization users with the utilized color schemes and the consistency of the colors, e.g., among different variables in the data with corresponding value ranges and physical interpretation. The latter is especially relevant in 3D visualizations, where colors can be distorted due to lighting and shading effects. Hühlein et al. (2024a) account for this via consistent default settings of color schemes for different physical variables and by providing options to disable lighting and shading on demand.

**Visualizing terrain uncertainty** The communication of data uncertainties is an important topic in visualization, often requiring the depiction of uncertainty information jointly with the actual data values. Comprehensive reviews of methods and taxonomies in uncertainty visualization are given in summary articles, e.g., by Griethe and Schumann (2006), Bonneau et al. (2014), and Kamal et al. (2021). MacEachren et al. (2005) review uncertainty visualization methods specifically for geospatial data and distinguish uncer-

tainty in the spatial allocation of the data from uncertainty in the data attributes. The uncertainty of the model terrain in Höhle et al. (2024a) falls into the latter category.

Diverse methods exist to visualize such uncertainties. Retchless and Brewer (2016) review and compare map-based visualizations for temperature change projections in climate studies and visualized value uncertainty jointly with the data values by using color and texture variations. Dübel et al. (2017) study methods for visualizing data in its terrain context while also communicating uncertainty in the data through intrinsic and extrinsic encodings. In both studies, uncertainties concern data attributes that are not related to the terrain itself. In our work, the elevation coordinate itself is uncertain, and uncertainty bounds possess a spatial context themselves.

Kao et al. (2002) examine visualization methods for uncertain data distributed across a 2D domain. The authors suggest density-based visualizations of local data distributions in a 3D view and present visualizations of 3D probability volumes using linked views, cutting planes, PDF isosurfaces, and DVR. The approach is similar to the one in Höhle et al. (2024a). Local elevation distributions are obtained by aggregating elevation samples within a specified radius and can be visualized using volumetric representations. While Kao et al. (2002) focus on the PDF and probability isovalues to represent the distributions, we prefer the CDF and quantile isovalues as the basis of the presented elevation summary plots.

Alternative visualization methods for uncertain surfaces include point-based probabilistic surfaces (Grigoryan and Rheingans 2004), the representation of uncertain surfaces through crossing probabilities (Pfaffelmoser et al. 2011), or occurrence probabilities (Pothkow and Hege 2011), and the use of confidence surfaces for bounding the extent of a surface uncertainty interval (Zehner et al. 2010).

#### 4.4.2 Visual Analysis of Ensemble Datasets

As a special form of multi-faceted data (Kehrer and Hauser 2013), ensemble datasets are conceptually complex and offer varied visualization options. A comprehensive review of visualization methods for ensemble datasets has been conducted by Wang et al. (2019a), who identify analytical tasks for visual ensemble analysis and provide references with specific visualization methods.

Next to efficiency improvements to the in-memory representation of ensemble data through neural representations, as discussed in section 4.2, the presented studies focus on methods for assessing correlations and dependence structures in ensemble datasets and are related to studies within this context. Correlations and interdependence effects impact the reliability of uncertainty estimates and parameter sensitivity studies (e.g., Ancell and Hakim 2007; Molteni et al. 1996). Visual analysis methods facilitate the study of relevant patterns on global, local, and multi-variable levels. On the global level, clustering approaches are used to identify groups of similar ensemble members, which helps to structure the ensemble information and reduces the complexity of required visualizations (e.g., Bordoloi et al. 2004; Ferranti and Corti 2011; Kumpf et al. 2018). Local dependencies, such as autocorrelations within forecast fields or correlations between ensemble forecasts at different locations in space, affect the statistical interpretation of

## 4 Related Work

the forecast and the likelihood of extreme events, e.g., due to spatially correlated extreme precipitation. Visualizing local dependencies comprehensively is considered challenging due to the complexity of the dependence fields and is listed as an open problem in the review article by Wang et al. (2019a). Several approaches have visualized dependence and correlation fields through clustering approaches (e.g., Kumpf et al. 2019; Liebmann et al. 2018; Pfaffelmoser and Westermann 2012), using dedicated glyph representations (e.g., Pfaffelmoser and Westermann 2013), or using spatial chord diagrams (Neuhauser et al. 2024). Multi-variable dependencies have similar effects on the statistical validity of forecast probabilities and have been examined, e.g., using correlation graphs (e.g., Liu and Shen 2016; Sauber et al. 2006; Zhang et al. 2015).

In Farokhmanesh et al. (2023b), spatial dependence fields and multi-variable dependencies have been visualized through interactive DVR. Using DVR as a particularly costly and sample-intensive approach demonstrates the feasibility of NDFs as an efficient data structure for storing the precomputed fields. NDFs as a storage method are independent of the specific downstream visualization and can be combined arbitrarily with other postprocessing and correlation visualization techniques.

### 4.4.3 Machine Learning Explanation for Ensemble-Valued Predictors

In Höhle et al. (2024b), we propose a conditional PFI measure for ensemble-valued predictors to assess the importance of different aspects of the ensemble-internal variability. The proposed method is based on a conditional permutation scheme, which preserves specific aspects of the distribution information while destroying the remaining aspects. Conditional PFI measures have been considered in earlier works (e.g., Molnar et al. 2024; Strobl et al. 2008), aiming to assess interaction effects between different predictors. For this, the importance of specific predictors is evaluated in the context of the remaining predictors. In these studies, each predictor is understood as a scalar-valued feature that can be distinguished unambiguously from the remaining ones. In contrast, the approach in Höhle et al. (2024b) addresses ensemble-valued predictors, in which each predictor channel receives a set of scalar samples as input. Features refer to specific aspects of the ensemble distribution. For a broader review of model explainability techniques in meteorological applications, we refer to the review article by McGovern et al. (2019).

## CHAPTER 5

---

### Publication Summaries

---

The following sections contain summaries of the presented publications in line with the guidelines for publication-based dissertations of the TUM School of Computation, Information and Technology at the Technical University of Munich and the doctoral regulations of the Technical University of Munich as of 23 August 2021.





## 5.1 Paper I – A Comparative Study of Convolutional Neural Network Models for Wind Field Downscaling

**Summary** Low-level winds, i.e., wind speed and direction in the lowermost layers of the atmosphere, are important forecast products, e.g., for energy planning, aviation, and marine transport. Close to the Earth’s surface, winds are affected by the large-scale atmospheric pressure distribution and local-scale interactions with the terrain (cf. subsection 2.1.3). Numerical weather models commonly operate on discrete grid representations of the atmosphere and terrain, which are too coarse to capture such interactions in detail. This constrains the quality of forecast products on scales below the spacing of the simulation grid. Improving the resolution of wind field forecasts in regions with complex terrain requires regridding methods (cf. subsection 2.2.4) that emulate local turbulence effects while maintaining computational efficiency.

In Höhle et al. (2020), we examine the utility of convolutional neural networks (CNNs; cf. subsection 3.2.3) for increasing the spatial resolution of wind field simulations at the 100 m-level. The models are trained to infer the outputs of high-resolution simulations with an average grid spacing of 9 km based on data from a simulation with 31 km grid spacing. CNNs are selected for their exceptional interpolation capabilities on spatially distributed data, as demonstrated in image superresolution and superresolution for scientific datasets (cf. subsection 3.4.1). Based on a review of prior work in these fields, we select three representative CNN architectures, which are evaluated in a comprehensive comparison. To assess the skill advantage of CNN-based models over classical downscaling approaches, the CNNs are also compared against linear baseline models. Finally, we use the acquired insights to design a new model with optimized architecture and performance.

Next to architectural improvements, we study the capabilities of CNNs to incorporate auxiliary predictor information as a means to improve prediction accuracy. Based on meteorological prior knowledge, we select a set of auxiliary predictor fields, including boundary layer height, low-resolution, and high-resolution orography, and study the performance improvements achieved when using the predictors for inference. The importance of different predictor channels is assessed using model explanation methods based on permutation feature importance (PFI; cf. section 3.3), which we adapt for 2D field-valued predictors. The results suggest that both architectural complexity and the inclusion of auxiliary predictors benefit the prediction accuracy.

**Author contribution** The first author was responsible for the model design, the implementation and execution of experiments, and the final evaluation. Michael Kern contributed to the implementation and execution of the experiments. Architectural optimizations and the overall structure of the paper were developed jointly by the first author, Michael Kern, and Rüdiger Westermann. Timothy Hewson provided the dataset, suggested meteorological quality metrics, and contributed discussions of potential applications of the models in the meteorological context.

**Copyright** © 2020 The Authors. Reprint as published in *Meteorological Applications* by John Wiley & Sons Ltd on behalf of the Royal Meteorological Society under the terms of the CC BY Attribution License 4.0 (<https://creativecommons.org/licenses/by/4.0/>). Used with permission.

## 5.2 Paper II – Postprocessing of Ensemble Weather Forecasts Using Permutation-Invariant Neural Networks

**Summary** Raw ensemble forecasts based on numerical weather simulations often show systematic errors and miscalibrations that must be corrected through ensemble postprocessing (cf. subsection 2.2.3). A key question in designing effective models for this is how the models are informed about the distribution information conveyed in the ensemble. Common postprocessing methods aggregate the ensemble information in early inference stages or sort the ensemble members, leading to constrained representations and potential information loss.

Extending earlier work on distribution regression networks (DRNs; cf. subsection 3.1.2) for ensemble postprocessing, we suggest permutation-invariant neural network (NN) architectures (cf. subsection 3.2.6) to overcome such bottlenecks. In contrast to traditional postprocessing models, permutation-invariant NNs are natively suited for learning from set-structured data and are more flexible in extracting features from the sample distribution. Using DeepSets and Set Transformers as representative examples, we compare different permutation-invariant NN architectures regarding their utility for ensemble postprocessing. Our results demonstrate that the proposed models achieve state-of-the-art quality in postprocessing wind gust and temperature forecast ensembles.

We also study the relevance of selected features in the ensemble distribution for the prediction quality in different model architectures. Classical model explainability techniques work well only for scalar-valued predictors and require adaptations to work in the high-dimensional ensemble setting. Inspired by permutation feature importance (PFI), we develop a permutation-based method for studying the importance of ensemble-valued predictors through an inverse reasoning trick. While the common PFI measure quantifies the information loss after perturbing specific aspects of the input signal, the proposed method quantifies the information gain when leaving certain aspects unperturbed. Analyzing the models with ensemble PFI methods provides insights into which weather variables or ensemble features influence the postprocessed forecast quality.

**Author contribution** The first author was responsible for developing the permutation-invariant postprocessing models, implementing and executing the relevant experiments, and developing the importance analysis for ensemble-valued predictors. Benedikt Schulz contributed experimental data for the evaluation of the baseline models and suggested suitable methods for evaluating the postprocessing quality. The structure and final form of the paper were composed in close collaboration between all authors. Sebastian Lerch further provided access to the data used in the study.

**Copyright** © American Meteorological Society. Used with permission.



## 5.3 Paper III – Topographic Visualization of Near-Surface Temperatures for Improved Lapse Rate Estimation

**Summary** A key challenge in forecasting surface temperatures is the variation of temperatures with height. Predictions of surface temperatures need to account for this by compensating for temperature deviations due to differences in the prediction reference altitude. Modeling the change of temperatures with height is complicated, especially in complex terrain, due to strong variations of the relation in different weather conditions (cf. subsection 2.1.3). Operationally, temperatures are corrected by adding an offset to the model-predicted temperature, which is proportional to the height difference with a fixed lapse rate coefficient as a multiplicative factor. Due to the fixed coefficient, this method can cause significant forecast errors in certain weather situations. In a localized study over Great Britain, Sheridan et al. (2010) suggest estimating locally varying lapse rates by fitting a linear model to paired samples of point-wise model predictions and model terrain altitudes in the vicinity of the target location, resulting in higher prediction accuracy.

In H ohlein et al. (2024a), we adopt this approach and generalize it to larger domains and coarser grids by using an improved importance-weighted estimation procedure. While the original method was applied to data on a limited domain, the novel method is applied to global near-surface temperature forecasts and evaluated on a comprehensive dataset of global temperature observations.

To facilitate interactive exploration of the benefits and parameter dependencies of the method in different weather situations, the scheme is embedded into an interactive visualization tool. The presented tool enables joint visualizations of temperatures and temperature gradients on the model terrain surface and in the surrounding atmosphere using common surface-based and volume visualizations (cf. section 2.4). Predictions for weather stations and corresponding observation data can be visualized additionally and explored in the context of the model terrain. The integration of a higher-resolution terrain model allows the assessment of sub-grid terrain variability using a dedicated visualization based on summary statistics for terrain uncertainties.

*The study is unpublished, not peer-reviewed, and not relevant for the examination.*

**Author contribution** The first author was responsible for implementing and evaluating the downscaling scheme and the visualization tool. Timothy Hewson brought up the original idea, provided the required data, and suggested test cases for the downscaling scheme. The visualization tool was developed in close collaboration between the first author and R udiger Westermann. The final paper was composed jointly by all authors.

**Copyright**   2024 The Authors. Reprint as hosted on arXiv.org under the terms of the CC BY Attribution License 4.0 (<https://creativecommons.org/licenses/by/4.0/>). Used with permission.



## 5.4 Paper IV – Evaluation of Volume Representation Networks for Meteorological Ensemble Compression

**Summary** Meteorological ensemble datasets consist of a set of weather forecast simulations, which can differ in initial conditions, model parameters, or model assumptions. Researchers have kept pushing ensemble sizes to larger scales while, at the same time, extending the spatial domain size and grid resolution. The resulting ensemble datasets can become extremely large, making any attempt to analyze such datasets intrinsically difficult. Interactive analytics applications require rapid data access, optimally at random domain locations. In-memory data representations are often preferred due to I/O bandwidth restrictions when loading data from disk. While existing data compression approaches would be able to compress the data sufficiently to load large ensembles into the working memory, the subsequent decompression times are frequently incompatible with timing constraints in interactive applications. Fast volume scene representations (fV-SRNs) have been proposed as an alternative neural network-based representation for volumetric scalar fields, offering both high compression ratios and rapid data access. Extending fV-SRNs to the case of ensemble data introduces new challenges due to the multivariate and set-structured format of ensemble forecasts.

In Höhle et al. (2022), we explore the use of neural data representations for representing meteorological ensemble datasets. Targeting visual analysis applications, the proposed ensemble representation networks (ERNs) are trained to store an approximate representation of the ensemble dataset within the network parameters of a coordinate-based neural network (CBN, cf. subsection 3.2.2). We suggest and evaluate two design approaches for ERN architectures, which facilitate ensemble data compression via network parameter sharing between the neural representations of different ensemble members and physical variable fields. Numerical experiments address the impact of distribution characteristics of the physical variables on the achievable reconstruction accuracy and the models’ ability to exploit coherence between ensemble members. Our results demonstrate en-par or better compression-reconstruction trade-offs in comparison with classical compressors and highlight the different error and memory characteristics of the architectures.

**Author contributions** The first author conceptualized the proposed model architectures, implemented and executed the majority of the experiments and method comparisons, and acted as the lead author of the publication. Sebastian Weiss contributed an initial code basis, originating from prior work, which included a GPU-accelerated implementation of neural volume representations and a volume visualization tool, which was used in the paper to generate the visualizations. The code base was used and extended by the first author to implement the ensemble models. Rüdiger Westermann developed the original idea for the project and supervised the study. The objectives of the study as well as the final form of the paper were composed in close collaboration between Sebastian Weiss, Rüdiger Westermann and the first author. Tobias Necker, Martin Weissmann, and Takemasa Miyoshi provided the dataset used in the study.

**Copyright** © 2022 The Authors. Eurographics Proceedings © 2022 The Eurographics Association. Reprint as published in the proceedings of VMV: Vision, Modeling, and Visualization 2022 under the terms of the CC BY Attribution License 4.0 (<https://creativecommons.org/licenses/by/4.0/>) with extended authors list according to the article's corrigendum as of 28 February 2024 (details available at the article's permanent online presence accessible through the digital object identifier link: <https://doi.org/10.2312/vmv.20221198>). Used with permission.



## 5.5 Paper V – Neural Fields for Interactive Visualization of Statistical Dependencies in 3D Simulation Ensembles

**Summary** Uncertainty assessments in ensemble forecasting often involve the computational and visual exploration of spatial and spatiotemporal correlation patterns within the ensemble distribution. For nonlinear dependence measures, such as mutual information (cf. subsection 2.3.2), estimating the dependencies or even keeping the required ensemble dataset in the working memory can be computationally challenging. For instance, the visual exploration of bivariate dependencies, e.g., between the values of physical variables in an ensemble, involves estimating bivariate dependence metrics between the simulated physical variables at different locations in space and time. Spatial or spatiotemporal context visualizations commonly require access to many such dependence scores in parallel. Computing the dependencies on the fly may be computationally too costly in interactive settings, and storing all bivariate dependencies leads to an explosion in required memory.

In Farokhmanesh et al. (2023b), we address these challenges through neural dependence fields (NDFs) as a novel representation of correlation and dependence structures in large-scale multi-variable ensembles. The required dependence estimates are interpreted as the values of a scalar field defined over a multi-dimensional query domain. The resulting dependence field is encoded in a coordinate-based neural network (CBN; cf. subsection 3.2.2). Once trained, the NDF can be queried at interactive speeds and integrated into GPU-accelerated visualization workflows. By design, the proposed model architecture respects the symmetries of the underlying dependence fields. This is achieved by combining efficient feature-based positional encodings with feature combination approaches for high-dimensional encodings inspired by tensor decompositions. To demonstrate the utility of the approach, the proposed representations are integrated into an interactive direct volume renderer, which enables volumetric visualizations of the encoded fields. Extensive experiments demonstrate the feasibility of the approach and illustrate the visual exploration of dependence fields in univariate and multivariate use cases.

**Author contributions** The author of this thesis suggested the idea underlying the publication, provided an initial proof of concept implementation of the proposed models, and contributed significant aspects of the model design. Fatemeh Farokhmanesh acted as the lead author of the paper, prepared the final implementation of the models, and carried out the experiments, leading to the results shown in the paper. Christoph Neuhauser developed a GPU-based visualization tool in which the proposed models are used to store the dependence field information. The final model and the written form of the paper were developed in close collaboration between Fatemeh Farokhmanesh, Christoph Neuhauser, Rüdiger Westermann, and the author of this thesis. Tobias Necker, Martin Weissmann, and Takemasa Miyoshi provided the dataset used in the study.

**Copyright** © 2023 The Authors. Reprint as published in the proceedings of VMV: Vision, Modeling, and Visualization 2023 under the terms of the CC BY Attribution License 4.0 (<https://creativecommons.org/licenses/by/4.0/>) with extended authors

## *5 Publication Summaries*

list according to the article's corrigendum as of 28 February 2024 (details available at the article's permanent online presence, accessible through the digital object identifier link: <https://doi.org/10.2312/vmv.20231229>). Used with permission.

### 6.1 Discussion

In this thesis, we have outlined the contributions from several articles that examine applications of ML- and DL-based modeling approaches to postprocessing and ensemble data analysis in meteorology. This section summarizes observations and key insights from the presented studies and discusses the findings in a coherent context.

#### 6.1.1 Impact of Network Architectures on Model Performance

A common scheme in several of the presented publications has been the comparative evaluation of different model architectures for specific learning tasks. A central motivation for this has been the search for guidelines that facilitate the design of skillful DL models. Our findings support the view that model architectures strongly impact the model quality, but the performance also depends on the modeling task and available data.

In H ohlein et al. (2020), we found that deeper and more complex CNN architectures achieve a higher accuracy of the downscaled wind fields, and overly simplistic nonlinear models may perform even worse than baseline interpolation schemes. Similar findings apply to the neural data representations in H ohlein et al. (2022) and Farokhmanesh et al. (2023b), where a higher model complexity improved the representation quality. Despite the potential improvements, more complex models tend to overfit the training data. While neural data representations are trained to overfit by design, overfitting requires regularization, e.g., in the downscaling models, or implies a need for more training data. Simple linear or physics-based models, as employed in H ohlein et al. (2024a), can operate in more data-scarce scenarios but may not exploit the full potential of the available data.

Across all our experiments, we recognize strong dependencies of the model quality on the amount and information content of the available data. The findings concerning the auxiliary predictors in H ohlein et al. (2020) and H ohlein et al. (2024b) as well as

the relevance of positional encodings in neural representations in Höhle et al. (2022) and Farokhmanesh et al. (2023b) imply that the performance of DL models is highly sensitive to the input data. This is backed by the PFI analyses applied in Höhle et al. (2020) and Höhle et al. (2024b), which imply that the highest quality models are obtained when all available predictor variables are considered in the model. Omitting certain parameters during training and inference in both studies caused stronger accuracy losses than architectural adaptations. Accordingly, several studies have explored ways to incorporate auxiliary information and physical prior knowledge in learning tasks related to our work (e.g., Dujardin and Lehning 2022; Mlakar et al. 2024; Mu et al. 2020; Rampal et al. 2022). Systematic research on the impact of different pieces of information on the learning performance of data-driven models in meteorology may help to build a better understanding of the field.

Interpreting the results of the model comparison in Höhle et al. (2024b) is less clear. Despite operating on notably different representations of the input data, no significant differences were observed between the baseline models using summary statistics and the more complex permutation-invariant ensemble models. While the PFI analysis suggests that most of the relevant information in the ensemble predictors is conveyed in the ensemble mean and standard deviation, it remains unclear why this is the case. One hypothesis would be that, in fact, the ensemble mean and standard deviation together are sufficient statistics that summarize all relevant information that the input ensemble contains about the target variable. This appears unlikely, however, in light of the potential complexity of distribution shapes. An alternative hypothesis is that potentially more information is contained in the ensemble but cannot be resolved due to its finite size and stochastic character. This view is supported by ongoing research, demonstrating that large ensemble sizes may be required to accurately capture certain distribution features and extreme weather events (Craig et al. 2022; Necker et al. 2020; Tempest et al. 2023). Further research is needed to explore the underlying reasons for the observed model performance in detail.

### 6.1.2 Statistical Reliability Assessment

Well-founded verification procedures are required to identify the models' abilities and limitations. In our downscaling and postprocessing studies, we have tried to achieve this by adopting standard verification procedures using, e.g., cross-validation methods and independent test datasets and evaluating the models with comprehensive sets of quality metrics. Nevertheless, data limitations affect the results, and the choice of model quality metrics requires expert knowledge and depends on the learning task. The choice of quality metrics in our studies was guided by comprehensive literature research and the expert knowledge of collaborating domain scientists. Data availability, however, has been a central topic in several projects.

Meteorological datasets commonly comprise a sequence of example cases observed over time. When sampled at high frequency, subsequent examples in the dataset become correlated, and temporal data may exhibit trends and periodic patterns on a global level. In Höhle et al. (2020), we used numerical simulation data from three consecutive

years, with example cases sampled hourly. Daily and annual weather cycles, as well as an expected lifetime of weather situations in central Europe of several days, imply that subsequent example cases are, in fact, not fully independent. The effect of temporal dependencies has been minimized by splitting the dataset appropriately. Still, the limited time span considered in the experiment, covering only three annual cycles, rules out conclusive statements about the long-term reliability of the models, which would be important, e.g., for assessing the models' utility in climate applications. The results in H ohlein et al. (2024b) are less vulnerable to such effects due to the longer considered time span in the datasets. In H ohlein et al. (2024a), the data is not split along the temporal dimension but in the spatial domain. Spatial correlations are likely less pronounced due to the distance between neighboring sample locations.

### 6.1.3 Datasets and Model Comparisons

A key challenge in our comparative studies (H ohlein et al. 2020, 2024b, 2022) has been comparing the proposed methods against prior and concurrent work. Postprocessing and compression methods are often developed for specific application scenarios and tested on unique data configurations. Proprietary data policies and data access impediments hamper informative model comparisons. The work in this thesis is largely based on proprietary datasets and custom evaluation procedures. Best efforts were made to guarantee fair model comparisons by applying competitor methods identically to the available data or reusing results from prior work, where applicable.

Motivated by the success of *benchmark datasets* for model comparisons in computer vision, such as MNIST (LeCun et al. 1998) and ImageNet (Deng et al. 2009), similar comparison projects have been established in meteorology (e.g., Ashkboos et al. 2022; Demaeyer et al. 2023; Rasp et al. 2020, 2024) and scientific data processing (Jakob et al. 2021; Zhao et al. 2020). Next to publicly accessible data, the benchmarks offer guidelines for evaluating novel models. Currently, the available benchmarks cover only a few learning tasks and some are subject to ongoing work or data restrictions (e.g., Demaeyer et al. 2023; Rasp et al. 2024). Specifically, analysis and compression tasks on spatial and spatiotemporal ensemble datasets are not covered appropriately, yet. A broader adoption of publicly available benchmarks will foster progress in the field.

### 6.1.4 Modeling Uncertainties

All models in this study are based on empirical methods that exploit finite, historical data records to infer predictions about the future. Uncertainties are an inherent part of the considered prediction problems. In H ohlein et al. (2024b) and H ohlein et al. (2024a), postprocessing and downscaling uncertainties were addressed explicitly using appropriate statistical quality metrics and visualization methods.

The work in H ohlein et al. (2020) and H ohlein et al. (2022) was limited to deterministic downscaling models and compression approaches. The proposed models, thus, do not admit a direct assessment of the involved uncertainties due to downscaling or compression reconstruction. Subsequent studies have explored the use of DGMs for downscaling,

using GANs (cf., e.g., Sun et al. 2024, and references therein) and normalizing flows (Groenke et al. 2021; Shen and Shen 2023; Winkler and Rolnick 2024). DGMs could, in principle, enable a targeted assessment of downscaling and reconstruction uncertainties through probabilistic predictions. However, reliable verification of the output statistics remains challenging due to the high-dimensional character of the predicted fields. Future research must explore suitable verification methods, e.g., based on appropriate scoring rules (Gneiting and Raftery 2007; Gneiting et al. 2008; Scheuerer and Hamill 2015).

NDFs, proposed in Farokhmanesh et al. (2023b), are subject to very specific uncertainties. Next to the investigated model errors in the reconstructed dependence field, the raw bivariate dependence estimates are inherently uncertain due to the finite size of the underlying ensemble dataset (cf. subsection 2.3.2). For the dataset considered in our study, such uncertainties should be minor due to the size of the underlying ensemble. For more general applications, especially with smaller ensembles, the uncertainty of the estimates should be considered. Correlated error patterns in the estimates could lead to a misinterpretation of the significance of spatially distributed dependence patterns (Wilks 2016). Future research should address such limitations and adapt the model architecture and visualization procedures accordingly.

## 6.2 Future Work

The work presented in this thesis offers several starting points for future research. This section highlights selected research directions that appear particularly promising.

### 6.2.1 Improving the Proposed Models

While the proposed models used state-of-the-art architectures at the time of publication, architectural novelties developed in the meantime and design patterns excluded from the studies could lead to more skillful models.

Specifically, for H ohlein et al. (2020), potential extensions include the use of attention mechanisms, which have gained popularity in ISR and computer vision research (cf., e.g., Anwar et al. 2020; Khan et al. 2022), and geometric and graph-based DL methods (cf., e.g., Bronstein et al. 2017; Wu et al. 2021), which adapt more easily to irregular grid structures and the spherical geometry of the Earth’s surface. Similarly, neural operator learning (Guibas et al. 2022; Kovachki et al. 2024; Li et al. 2021) has emerged as an alternative to convolution-based model designs. Applications of neural operators to ISR (Wei and Zhang 2023), fluid dynamics (Li et al. 2024), and DL-based weather prediction (Pathak et al. 2022) suggest that such models can benefit downscaling workflows for spatially distributed prediction fields. Methodological improvements could be achieved by adopting uncertainty-aware modeling approaches, e.g., based on DGMs. While existing studies have explored the utility of GANs (cf. Sun et al. 2024, and references therein) and normalizing flow models (e.g., Groenke et al. 2021; Shen and Shen 2023), diffusion models have not been considered in detail, so far. SR-related studies involving diffusion models (e.g., Li et al. 2022) suggest that diffusion models are a viable alternative to other DGM classes.

Neural data representations, as examined in Höhle et al. (2022) and Farokhmanesh et al. (2023b), offer great compression opportunities for meteorological simulation datasets due to their flexibility and independence on grid specifications. Yet, the current implementations are still in a proof-of-concept phase and require further improvements to become ready for adoption in the domain sciences. Current neural data representations often require long training times and hyperparameter tuning to adapt the models to the peculiarities of diverse data configurations. Future work could provide application-ready compression interfaces with improved training times and automated routines for hyperparameter tuning and architecture search. Computer vision research suggests that generative *hypermodels* can learn to predict the network parameters of neural representations directly without costly iterative optimizations (Erkoç et al. 2023). Similar strategies could lower the compression times when ported to neural data representations. Recent work is starting to explore the use of hypermodels for predicting the parameters of neural data representations but does not adopt the generative approach yet (Wu et al. 2023). Hyperparameter selection could be addressed through self-tuning model architectures, which automatically adapt the model size and memory requirements to the complexity of the considered dataset. Automated model size selection and compression are subject to active research (cf., e.g., Mishra et al. 2020; Ren et al. 2021), and their adoption for neural data representations would elevate the models' practical utility.

Potential improvements appear less obvious in the settings of Höhle et al. (2024a) and Höhle et al. (2024b). In both cases, the modeling task involves learning from tabular-like datasets, where each physical predictor variable represents an attribute column in a table. As opposed to computer vision applications, DL models are known to offer limited benefits over more straightforward ML approaches on tabular-like datasets (cf., e.g., Borisov et al. 2024; Shwartz-Ziv and Armon 2022). Therefore, an alternative focus has been on exploiting new sources of forecast skill, such as spatial context information (Feik et al. 2024). Future work could follow up in this direction and investigate the inclusion of, e.g., temporal context information or examine the information content of ensemble forecasts in more detail.

## 6.2.2 Understanding the Information Content of Ensemble Forecasts

Statistical postprocessing of ensemble forecasts is intrinsically an intriguing environment for ML research due to the very nature of the learning task. Using probabilistic distribution representations to infer a different notion of probabilistic information constitutes a unique learning setting. The findings in Höhle et al. (2024b) raise questions about the nature of relevant features in the input ensemble, effective methods to represent them, and ways to use them efficiently in ML and DL models. Addressing such questions is complicated because both the raw input ensembles and the postprocessing models may be subject to uncertainties and deficiencies that reduce the achievable prediction accuracy. Separating the deficiencies of postprocessing models from those of the input ensembles could benefit both the design of future postprocessing models and the understanding of prediction capabilities and biases of NWP models.

The PFI analysis for ensemble-valued predictors, proposed in Höhle et al. (2024b), constitutes a first step in this direction and highlights important characteristics of the ensemble distribution and information overlap between different summary statistics. A key limitation, however, is its reliance on hand-selected summary statistics and the qualitative character of the provided information. A more quantitative and automated approach, e.g., based on information theoretic methods (e.g., Schulz et al. 2020; Tishby et al. 2000), unsupervised feature learning, and targeted visual analysis solutions, could yield more detailed insights. Throughout such experiments, the development and use of ensemble-oriented toy datasets may help to limit the complexity of the analysis task.

### 6.2.3 Adopting Foundation Models

A key limitation to the adoption of elaborate DL methods in meteorological research lies in the immense amounts of computing resources required to build skillful models. Currently, the best-performing models across various prediction tasks are trained on large datasets and use very deep model architectures. Naturally, training large DL models *from scratch* is compute-intensive and potentially inefficient when specializing the model to one single learning task. Foundation models have become popular as a way to reuse a model’s capabilities in different learning settings. For this, foundation models are trained once on large datasets and for multiple different prediction tasks, and refined subsequently for specific applications. Refining pre-trained models for new learning tasks requires less computational effort than training from scratch. The idea of generalist foundation models originates in computer vision and natural language processing (Bommasani et al. 2022; Radford et al. 2021) and is now also adopted in meteorology and climate research (e.g., Nguyen et al. 2023). In the long run, efforts to develop and improve the adoption of foundation models might – in many applications – promise more sustainable progress than pushing the capabilities of dedicated models for specific tasks.

### 6.2.4 Deep Learning-Based Weather Prediction

Beyond data-driven postprocessing and analysis models, recent years have brought rapid advancements in developing inherently DL-based weather prediction models. An overview of relevant works can be found, e.g., in the article by Olivetti and Messori (2024). Such models are trained to generate weather forecasts based on historical data records, often bypassing the need for physical modeling entirely. The field is developing rapidly, and various approaches have demonstrated promising forecasting capabilities (e.g., Bi et al. 2022; Lam et al. 2023; Pathak et al. 2022). Even operational DL models for weather predictions have entered a testing phase (Lang et al. 2024).

The work in this thesis is unaffected by this form of weather prediction. In particular, all datasets used in this study are based on traditional physics-based numerical weather simulations. Nevertheless, the developed postprocessing and ensemble analysis methods remain applicable, and the proposed feature importance methods can be adapted, potentially, for the new class of DL-based prediction models. Ongoing studies are examining the physical and statistical reliability of the DL-based models (e.g., Ben-Bouallègue et al.



2024; Selz and Craig 2023). Model explanation and visual analysis techniques similar to those presented in this thesis will play a vital role in developing a better understanding of the new models' characteristics and building trust in the new technology.



---

## Conclusion

---

In this thesis, we have compiled the results from a series of studies applying data-driven machine learning (ML) and deep learning (DL) techniques to improve numerical weather prediction (NWP) and meteorological data analysis. The proposed methods address key challenges in postprocessing numerical weather forecasts, meteorological data compression, and visualization. Our research demonstrates the potential of recent neural network (NN) models to handle complex meteorological datasets, such as simulation datasets on unstructured grids and ensemble forecasts, in diverse applications.

In Höhle et al. (2020), we explored the use of convolutional neural networks (CNNs) for downscaling near-surface wind fields on extended spatial domains. The study demonstrates that deeper, more complex models can exploit nonlinear relations between physical predictor variables and the target wind fields, benefiting particularly from high-resolution predictors like orography. The findings highlight the potential of CNNs for improving downscaling while suggesting applications in operational forecasting systems and future research directions.

In Höhle et al. (2024b), we applied permutation-invariant NN architectures for postprocessing ensemble forecasts of wind gusts and surface temperatures. The proposed models achieve state-of-the-art prediction accuracy and offer the potential to exploit rich features in multi-variate predictor ensembles. A feature importance analysis, developed specifically for ensemble-valued predictors, suggests that the key informative features relevant to the final prediction are confined mainly to the ensemble mean and standard deviation. Our findings motivate future work on the information content and representation capabilities of ensemble forecasts and the use of ensemble-based models in postprocessing.

In Höhle et al. (2024a), we developed a physically motivated downscaling scheme for surface temperatures using data-driven estimates of the local ambient lapse rate. Our work shows substantial improvements in the accuracy of surface temperature predictions

## 7 Conclusion

under complex topographic conditions. The technique is embedded into a 3D topographic visualization system, enabling surface-based and volumetric visualizations of near-surface temperature predictions and observations in the context of the surrounding terrain.

In Höhle et al. (2022), we introduced ensemble representation networks (ERNs) for compressing meteorological ensembles in visualization applications. The study illustrates that ERNs perform on par with or outperform traditional compression methods while maintaining fast reconstruction capabilities. ERNs offer promising capabilities for interactive data analysis, paving the way for future work to improve their usability in large-scale visual analysis applications.

Finally, in Farokhmanesh et al. (2023b), neural dependence fields (NDFs) were introduced as a novel method for encoding and visualizing statistical dependencies in 3D meteorological ensemble simulations. NDFs facilitate the interactive exploration of complex correlation structures in ensemble datasets, which would otherwise be intractable due to computation time and memory constraints. NDFs use targeted coordinate-based network architectures, which enable a promising trade-off between memory savings and reconstruction accuracy.

Collectively, the presented studies advance the field of numerical weather forecasting and meteorological data analysis through the development of new modeling and model analysis techniques. Notably, our work adapted model architectures and took inspiration from adjacent research domains, such as computer vision and visualization research. In light of the increasing importance of data-driven methods in meteorology and climate science, the interdisciplinary fusion of expertise and modeling techniques between computer science and meteorology has turned out as a productive research approach. We hope that the developed model explanation techniques and insights gained in the presented studies facilitate a deeper understanding of the emergent technologies and help to build more reliable data-driven ML and DL systems for future applications in weather forecasting and meteorological data analysis.

---

## Bibliography

---

- Abadi, Martín, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mane, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, et al. (2016). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*. arXiv: 1603.04467 [cs.DC].
- Afzal, Shehzad, Mohamad Mazen Hittawe, Sohaib Ghani, Tahira Jamil, Omar Knio, Markus Hadwiger, and Ibrahim Hoteit (2019). “The State of the Art in Visual Analysis Approaches for Ocean and Atmospheric Datasets”. In: *Computer Graphics Forum* 38.3, pp. 881–907.
- Ahn, Namhyuk, Byungkon Kang, and Kyung-Ah Sohn (2018). “Fast, Accurate, and Lightweight Super-Resolution with Cascading Residual Network”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 252–268.
- Amanatides, John, Andrew Woo, et al. (1987). “A fast voxel traversal algorithm for ray tracing.” In: *Eurographics*. Vol. 87. 3. Citeseer, pp. 3–10.
- American Meteorological Society (1993). “Guidelines for Using Color to Depict Meteorological Information: IIPS Subcommittee for Color Guidelines”. In: *Bulletin of the American Meteorological Society* 74.9, pp. 1709–1713.
- An, Yifei, Han-Wei Shen, Guihua Shan, Guan Li, and Jun Liu (2021). “STSRNet: Deep Joint Space–Time Super-Resolution for Vector Field Visualization”. In: *IEEE Computer Graphics and Applications* 41.6, pp. 122–132.
- Ancell, Brian and Gregory J. Hakim (2007). “Comparing Adjoint- and Ensemble-Sensitivity Analysis with Applications to Observation Targeting”. In: *Monthly Weather Review* 135.12, pp. 4117–4134.
- Andrienko, Natalia and Gennady Andrienko (2006). *Exploratory analysis of spatial and temporal data: a systematic approach*. Springer Science & Business Media.

## Bibliography

- Anwar, Saeed, Salman Khan, and Nick Barnes (2020). “A Deep Journey into Super-resolution: A Survey”. In: *ACM Computing Surveys* 53.3, pp. 1–34.
- Arjovsky, Martin, Soumith Chintala, and Léon Bottou (2017). “Wasserstein Generative Adversarial Networks”. In: *Proceedings of the 34th International Conference on Machine Learning*. Vol. 70. PMLR, pp. 214–223.
- Ashkboos, Saleh, Langwen Huang, Nikoli Dryden, Tal Ben-Nun, Peter Dueben, Lukas Gianinazzi, Luca Kummer, and Torsten Hoeffler (2022). “ENS-10: A Dataset For Post-Processing Ensemble Weather Forecasts”. In: *Advances in Neural Information Processing Systems*. Vol. 35. Curran Associates, Inc., pp. 21974–21987.
- Atkinson, Peter M. (2013). “Downscaling in remote sensing”. In: *International Journal of Applied Earth Observation and Geoinformation* 22. Spatial Statistics for Mapping the Environment, pp. 106–114.
- Ba, Jimmy Lei, Jamie Ryan Kiros, and Geoffrey E. Hinton (2016). *Layer Normalization*. arXiv: 1607.06450 [stat.ML].
- Baker, Allison H., Haiying Xu, Dorit M. Hammerling, Shaomeng Li, and John P. Clyne (2017). “Toward a Multi-method Approach: Lossy Data Compression for Climate Simulation Data”. In: *High Performance Computing*. Cham: Springer International Publishing, pp. 30–42.
- Baklanov, Alexander A., Branko Grisogono, Robert Bornstein, Larry Mahrt, Sergej S. Zilitinkevich, Peter Taylor, Søren E. Larsen, Mathias W. Rotach, and H. J. S. Fernando (2011). “The Nature, Theory, and Modeling of Atmospheric Planetary Boundary Layers”. In: *Bulletin of the American Meteorological Society* 92.2, pp. 123–128.
- Ballester-Ripoll, Rafael, Peter Lindstrom, and Renato Pajarola (2020). “TTHRESH: Tensor Compression for Multidimensional Visual Data”. In: *IEEE Transactions on Visualization and Computer Graphics* 26.9, pp. 2891–2903.
- Baño-Medina, Jorge, Rodrigo Manzananas, and José Manuel Gutiérrez (2020). “Configuration and intercomparison of deep learning neural models for statistical downscaling”. In: *Geoscientific Model Development* 13.4, pp. 2109–2124.
- Basri, Ronen, Meirav Galun, Amnon Geifman, David Jacobs, Yoni Kasten, and Shira Kritchman (2020). “Frequency bias in neural networks for input of non-uniform density”. In: *Proceedings of the 37th International Conference on Machine Learning*. ICML’20. JMLR.org, pp. 685–694.
- Basri, Ronen, David Jacobs, Yoni Kasten, and Shira Kritchman (2019). “The Convergence Rate of Neural Networks for Learned Functions of Different Frequencies”. In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc.
- Belghazi, Mohamed Ishmael, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm (2018). “Mutual Information Neural Es-

- timation”. In: *Proceedings of the 35th International Conference on Machine Learning*. Vol. 80. PMLR, pp. 531–540.
- Ben-Bouallègue, Zied, Mariana C. A. Clare, Linus Magnusson, Estibaliz Gascón, Michael Maier-Gerber, Martin Janoušek, Mark Rodwell, Florian Pinault, Jesper S. Dramsch, Simon T. K. Lang, Baudouin Raoult, Florence Rabier, Matthieu Chevallier, Irina Sandu, Peter Dueben, Matthew Chantry, and Florian Pappenberger (2024). “The Rise of Data-Driven Weather Forecasting: A First Statistical Assessment of Machine Learning–Based Weather Forecasts in an Operational-Like Context”. In: *Bulletin of the American Meteorological Society* 105.6, E864–E883.
- Ben-Bouallègue, Zied, Jonathan A. Weyn, Mariana C. A. Clare, Jesper Dramsch, Peter Dueben, and Matthew Chantry (2023). *Improving medium-range ensemble weather forecasts with hierarchical ensemble transformers*. arXiv: 2303.17195 [physics.ao-ph].
- Berenjkoub, Marzieh, Rodolfo Ostilla Monico, Robert S. Laramee, and Guoning Chen (2019). “Visual Analysis of Spatia-temporal Relations of Pairwise Attributes in Unsteady Flow”. In: *IEEE Transactions on Visualization and Computer Graphics* 25.1, pp. 1246–1256.
- Berger, Matthew, Jixian Li, and Joshua A. Levine (2019). “A Generative Model for Volume Rendering”. In: *IEEE Transactions on Visualization and Computer Graphics* 25.4, pp. 1636–1650.
- Berner, Judith, Ulrich Achatz, Lauriane Batté, Lisa Bengtsson, Alvaro de la Cámara, Hannah M. Christensen, Matteo Colangeli, Danielle R. B. Coleman, Daan Crommelin, Stamen I. Dolaptchiev, Christian L. E. Franzke, Petra Friederichs, Peter Imkeller, Heikki Järvinen, Stephan Juricke, Vassili Kitsios, François Lott, Valerio Lucarini, Salil Mahajan, Timothy N. Palmer, Cécile Penland, Mirjana Sakradzija, Jin-Song von Storch, Antje Weisheimer, Michael Weniger, et al. (2017). “Stochastic Parameterization: Toward a New View of Weather and Climate Models”. In: *Bulletin of the American Meteorological Society* 98.3, pp. 565–588.
- Bi, Kaifeng, Lingxi Xie, Hengheng Zhang, Xin Chen, Xiaotao Gu, and Qi Tian (2022). *Pangu-Weather: A 3D High-Resolution Model for Fast and Accurate Global Weather Forecast*. arXiv: 2211.02556 [physics.ao-ph].
- Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag.
- Bleisch, S. (2012). “3D Geovisualization - Definition and Structures for the Assessment of Usefulness”. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* I-2, pp. 129–134.
- Bommasani, Rishi, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue,

## Bibliography

- Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, et al. (2022). *On the Opportunities and Risks of Foundation Models*. arXiv: 2108.07258 [cs.LG].
- Bond-Taylor, Sam, Adam Leach, Yang Long, and Chris G. Willcocks (2022). “Deep Generative Modelling: A Comparative Review of VAEs, GANs, Normalizing Flows, Energy-Based and Autoregressive Models”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.11, pp. 7327–7347.
- Bonneau, Georges-Pierre, Hans-Christian Hege, Chris R. Johnson, Manuel M. Oliveira, Kristin Potter, Penny Rheingans, and Thomas Schultz (2014). “Overview and State-of-the-Art of Uncertainty Visualization”. In: *Scientific Visualization: Uncertainty, Multi-field, Biomedical, and Scalable Visualization*. London: Springer London, pp. 3–27.
- Bordoloi, Udepta D., David L. Kao, and Han-Wei Shen (2004). “Visualization techniques for spatial probability density function data”. In: *Data Science Journal* 3, pp. 153–162.
- Borgo, Rita, Johannes Kehler, David H. S. Chung, Eamonn Maguire, Robert S. Laramée, Helwig Hauser, Matthew Ward, and Min Chen (2013). “Glyph-based Visualization: Foundations, Design Guidelines, Techniques and Applications”. In: *Eurographics 2013 - State of the Art Reports*. The Eurographics Association, pp. 39–63.
- Borisov, Vadim, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci (2024). “Deep Neural Networks and Tabular Data: A Survey”. In: *IEEE Transactions on Neural Networks and Learning Systems* 35.6, pp. 7499–7519.
- Brand, Louis (2020). *Vector and Tensor Analysis*. Dover Books on Mathematics. Dover Publications.
- Breiman, Leo (2001). “Random Forests”. In: *Machine Learning* 45.1, pp. 5–32.
- Bremnes, John Bjørnar (2020). “Ensemble Postprocessing Using Quantile Function Regression Based on Neural Networks and Bernstein Polynomials”. In: *Monthly Weather Review* 148.1, pp. 403–414.
- Brodie, Ken W., Lesley Ann Carpenter, Rae A. Earnshaw, Julian R. Gallop, Roger J. Hubbold, Anne M. Mumford, Chris D. Osland, and Peter Quarendon (2012). *Scientific visualization: techniques and applications*. Springer Science & Business Media.
- Bronstein, Michael M., Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst (2017). “Geometric Deep Learning: Going beyond Euclidean data”. In: *IEEE Signal Processing Magazine* 34.4, pp. 18–42.
- Burkart, Nadia and Marco F. Huber (2021). “A Survey on the Explainability of Supervised Machine Learning”. In: *Journal of Artificial Intelligence Research* 70, pp. 245–317.
- Cabral, Brian and Leith Casey Leedom (1993). “Imaging vector fields using line integral convolution”. In: *Proceedings of the 20th Annual Conference on Computer Graphics*



- and Interactive Techniques*. SIGGRAPH '93. Anaheim, CA: Association for Computing Machinery, pp. 263–270.
- Cannon, Alex J. (2018). “Non-crossing nonlinear regression quantiles by monotone composite quantile regression neural network, with application to rainfall extremes”. In: *Stochastic Environmental Research and Risk Assessment* 32.11, pp. 3207–3225.
- Cao, Ang and Justin Johnson (2023). “HexPlane: A Fast Representation for Dynamic Scenes”. In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, pp. 130–141.
- Cappello, Franck, Sheng Di, Sihuan Li, Xin Liang, Ali Murat Gok, Dingwen Tao, Chun Hong Yoon, Xin-Chuan Wu, Yuri Alexeev, and Frederic T Chong (2019). “Use cases of lossy compression for floating-point data in scientific data sets”. In: *The International Journal of High Performance Computing Applications* 33.6, pp. 1201–1220.
- Card, Stuart K., Jock D. Mackinlay, and Ben Shneiderman (1999). *Readings in information visualization: using vision to think*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- Chabra, Rohan, Jan E. Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard Newcombe (2020). “Deep Local Shapes: Learning Local SDF Priors for Detailed 3D Reconstruction”. In: *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX*. Glasgow, United Kingdom: Springer-Verlag, pp. 608–625.
- Chan, Eric R., Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J. Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein (2022). “Efficient Geometry-Aware 3D Generative Adversarial Networks”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16123–16133.
- Chen, Anpei, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su (2022). “TensorRF: Tensorial Radiance Fields”. In: *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII*. Tel Aviv, Israel: Springer-Verlag, pp. 333–350.
- Chen, Jieyu, Tim Janke, Florian Steinke, and Sebastian Lerch (2024). “Generative machine learning methods for multivariate ensemble postprocessing”. In: *The Annals of Applied Statistics* 18.1, pp. 159–183.
- Chen, Zhiqin and Hao Zhang (2019). “Learning Implicit Fields for Generative Shape Modeling”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5939–5948.
- Cheng, Jianxin, Jin Liu, Qiuming Kuang, Zhou Xu, Chenkai Shen, Wang Liu, and Kang Zhou (2022). “DeepDT: Generative Adversarial Network for High-Resolution Climate Prediction”. In: *IEEE Geoscience and Remote Sensing Letters* 19, pp. 1–5.

## Bibliography

- Coiffier, Jean (2011). *Fundamentals of Numerical Weather Prediction*. Cambridge University Press.
- Cover, Thomas M., Joy A. Thomas, et al. (1991). “Entropy, relative entropy and mutual information”. In: *Elements of information theory* 2.1, pp. 12–13.
- Craig, George C., Matjaž Puh, Christian Keil, Kirsten Tempest, Tobias Necker, Juan Ruiz, Martin Weissmann, and Takemasa Miyoshi (2022). “Distributions and convergence of forecast variables in a 1,000-member convection-permitting ensemble”. In: *Quarterly Journal of the Royal Meteorological Society* 148.746, pp. 2325–2343.
- Croitoru, Florinel-Alin, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah (2023). “Diffusion Models in Vision: A Survey”. In: *IEEE Transactions on Pattern Analysis & Machine Intelligence* 45.09, pp. 10850–10869.
- Cybenko, G. (1989). “Approximation by superpositions of a sigmoidal function”. In: *Mathematics of Control, Signals and Systems* 2.4, pp. 303–314.
- Czado, Claudia, Tilmann Gneiting, and Leonhard Held (2009). “Predictive Model Assessment for Count Data”. In: *Biometrics* 65.4, pp. 1254–1261.
- Dasgupta, Aritra, Jorge Poco, Yaxing Wei, Robert Cook, Enrico Bertini, and Claudio T. Silva (2015). “Bridging Theory with Practice: An Exploratory Study of Visualization Use and Design for Climate Model Comparison”. In: *IEEE Transactions on Visualization & Computer Graphics* 21.09, pp. 996–1014.
- Dee, D. P., M. Balsaseda, G. Balsamo, R. Engelen, A. J. Simmons, and J.-N. Thépaut (2014). “Toward a Consistent Reanalysis of the Climate System”. In: *Bulletin of the American Meteorological Society* 95.8, pp. 1235–1248.
- Demaeyer, Jonathan, Jonas Bhend, Sebastian Lerch, Cristina Primo, Bert Van Schaeybroeck, Aitor Atencia, Zied Ben Bouallègue, Jieyu Chen, Markus Dabernig, Gavin Evans, Jana Faganelli Pucer, Ben Hooper, Nina Horat, David Jobst, Janko Merše, Peter Mlakar, Annette Möller, Olivier Mestre, Maxime Taillardat, and Stéphane Vanitsem (2023). “The EUPPBench postprocessing benchmark dataset v1.0”. In: *Earth System Science Data* 15.6, pp. 2635–2653.
- Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei (2009). “ImageNet: A large-scale hierarchical image database”. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255.
- Di, Sheng and Franck Cappello (2016). “Fast Error-Bounded Lossy HPC Data Compression with SZ”. In: *2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pp. 730–739.
- Dong, Chao, Chen Change Loy, Kaiming He, and Xiaoou Tang (2016a). “Image Super-Resolution Using Deep Convolutional Networks”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.2, pp. 295–307.

- Dong, Chao, Chen Change Loy, and Xiaoou Tang (2016b). “Accelerating the Super-Resolution Convolutional Neural Network”. In: *Computer Vision – ECCV 2016*. Cham: Springer International Publishing, pp. 391–407.
- Doury, Antoine, Samuel Somot, Sebastien Gadat, Aurélien Ribes, and Lola Corre (2023). “Regional climate model emulator based on deep learning: concept and first evaluation of a novel hybrid downscaling approach”. In: *Climate Dynamics* 60.5, pp. 1751–1779.
- Dübel, Steve, Martin Röhlig, Heidrun Schumann, and Matthias Trapp (2014). “2D and 3D presentation of spatial data: A systematic review”. In: *2014 IEEE VIS International Workshop on 3DVis (3DVis)*, pp. 11–18.
- Dübel, Steve, Martin Röhlig, Christian Tominski, and Heidrun Schumann (2017). “Visualizing 3D Terrain, Geo-Spatial Data, and Uncertainty”. In: *Informatics* 4.1. 6.
- Düben, Peter, Umberto Modigliani, Alan Geer, Stephan Siemen, Florian Pappenberger, Peter Bauer, Andy Brown, Martin Palkovic, Baudouin Raoult, Nils Wedi, and Vasileios Baousis (2021). “Machine learning at ECMWF: A roadmap for the next 10 years”. In: *ECMWF Technical Memoranda* 878.
- Düben, Peter D., Martin Leutbecher, and Peter Bauer (2019). “New Methods for Data Storage of Model Output from Ensemble Simulations”. In: *Monthly Weather Review* 147.2, pp. 677–689.
- Dueben, Peter D., Martin G. Schultz, Matthew Chantry, David John Gagne, David Matthew Hall, and Amy McGovern (2022). “Challenges and Benchmark Datasets for Machine Learning in the Atmospheric Sciences: Definition, Status, and Outlook”. In: *Artificial Intelligence for the Earth Systems* 1.3, e210002.
- Dujardin, Jérôme and Michael Lehning (2022). “Wind-Topo: Downscaling near-surface wind fields to high-resolution topography in highly complex terrain with deep learning”. In: *Quarterly Journal of the Royal Meteorological Society* 148.744, pp. 1368–1388.
- Edwards, Harrison and Amos Storkey (2017). *Towards a Neural Statistician*. arXiv: 1606.02185 [stat.ML].
- Elmqvist, Niklas and Philippas Tsigas (2008). “A Taxonomy of 3D Occlusion Management for Visualization”. In: *IEEE Transactions on Visualization and Computer Graphics* 14.5, pp. 1095–1109.
- Erkoç, Ziya, Fangchang Ma, Qi Shan, Matthias Nießner, and Angela Dai (2023). “HyperDiffusion: Generating Implicit Neural Fields with Weight-Space Diffusion”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14300–14310.
- Farokhmanesh, Fatemeh, Kevin Höhlein, Tobias Necker, Martin Weissmann, Takemasa Miyoshi, and Rüdiger Westermann (2023a). “Deep Learning–Based Parameter Transfer in Meteorological Data”. In: *Artificial Intelligence for the Earth Systems* 2.1, e220024.

## Bibliography

- Farokhmanesh, Fatemeh, Kevin Höhlein, Christoph Neuhauser, Tobias Necker, Martin Weissmann, Takemasa Miyoshi, and Rüdiger Westermann (2023b). “Neural Fields for Interactive Visualization of Statistical Dependencies in 3D Simulation Ensembles”. In: *Vision, Modeling, and Visualization (VMV 2023)*. The Eurographics Association.
- Feik, Moritz, Sebastian Lerch, and Jan Stühmer (2024). *Graph Neural Networks and Spatial Information Learning for Post-Processing Ensemble Weather Forecasts*. arXiv: 2407.11050 [cs.LG].
- Ferranti, Laura and S. Corti (2011). “New clustering products”. In: *ECMWF Newsletter* 127, pp. 6–11.
- Fiddes, J., K. Aalstad, and M. Lehning (2022). “TopoCLIM: rapid topography-based downscaling of regional climate model output in complex terrain v1.1”. In: *Geoscientific Model Development* 15.4, pp. 1753–1768.
- Fiddes, J. and S. Gruber (2014). “TopoSCALE v.1.0: downscaling gridded climate data in complex terrain”. In: *Geoscientific Model Development* 7.1, pp. 387–405.
- Finn, Tobias Sebastian (2021). *Self-Attentive Ensemble Transformer: Representing Ensemble Interactions in Neural Networks for Earth System Models*. arXiv: 2106.13924 [cs.LG].
- Foken, Thomas and Matthias Mauder (2008). *Micrometeorology*. Vol. 2. Springer.
- Forthofer, Jason M., Bret W. Butler, and Natalie S. Wagenbrenner (2014). “A comparison of three approaches for simulating fine-scale surface winds in support of wildland fire management. Part I. Model formulation and comparison against measurements”. In: *International Journal of Wildland Fire* 23.7, pp. 969–981.
- Frei, Christoph (2014). “Interpolation of temperature in a mountainous region using non-linear profiles and non-Euclidean distances”. In: *International Journal of Climatology* 34.5, pp. 1585–1605.
- Fridovich-Keil, Sara, Giacomo Meanti, Frederik Rahbaek Warburg, Benjamin Recht, and Angjoo Kanazawa (2023). “K-Planes: Explicit Radiance Fields in Space, Time, and Appearance”. In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, pp. 12479–12488.
- Fridovich-Keil, Sara, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa (2022). “Plenoxels: Radiance Fields Without Neural Networks”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5501–5510.
- Fukami, Kai, Koji Fukagata, and Kunihiro Taira (2021). “Machine-learning-based spatio-temporal super resolution reconstruction of turbulent flows”. In: *Journal of Fluid Mechanics* 909, A9.
- (2023). “Super-resolution analysis via machine learning: a survey for fluid flows”. In: *Theoretical and Computational Fluid Dynamics* 37.4, pp. 421–444.

- Fukushima, Kunihiko (1988). “Neocognitron: A hierarchical neural network capable of visual pattern recognition”. In: *Neural Networks* 1.2, pp. 119–130.
- Gao, Kyle, Yina Gao, Hongjie He, Dening Lu, Linlin Xu, and Jonathan Li (2023). *NeRF: Neural Radiance Field in 3D Vision, A Comprehensive Review*. arXiv: 2210.00379 [cs.CV].
- Giorgi, Filippo (2019). “Thirty Years of Regional Climate Modeling: Where Are We and Where Are We Going next?”. In: *Journal of Geophysical Research: Atmospheres* 124.11, pp. 5696–5723.
- Gneiting, Tilmann, Fadoua Balabdaoui, and Adrian E. Raftery (2007). “Probabilistic forecasts, calibration and sharpness”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 69.2, pp. 243–268.
- Gneiting, Tilmann and Matthias Katzfuss (2014). “Probabilistic Forecasting”. In: *Annual Review of Statistics and Its Application* 1. Volume 1, 2014, pp. 125–151.
- Gneiting, Tilmann and Adrian E Raftery (2007). “Strictly Proper Scoring Rules, Prediction, and Estimation”. In: *Journal of the American Statistical Association* 102.477, pp. 359–378.
- (2005). “Weather Forecasting with Ensemble Methods”. In: *Science* 310.5746, pp. 248–249.
- Gneiting, Tilmann, Adrian E. Raftery, Anton H. Westveld, and Tom Goldman (2005). “Calibrated Probabilistic Forecasting Using Ensemble Model Output Statistics and Minimum CRPS Estimation”. In: *Monthly Weather Review* 133.5, pp. 1098–1118.
- Gneiting, Tilmann and Roopesh Ranjan (2013). “Combining predictive distributions”. In: *Electronic Journal of Statistics* 7.none, pp. 1747–1782.
- Gneiting, Tilmann, Larissa I. Stanberry, Eric P. Grimit, Leonhard Held, and Nicholas A. Johnson (2008). “Assessing probabilistic forecasts of multivariate quantities, with an application to ensemble predictions of surface winds”. In: *TEST* 17.2, pp. 211–235.
- Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep Learning*. <http://www.deeplearningbook.org>. MIT Press.
- Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio (2014). “Generative Adversarial Nets”. In: *Advances in Neural Information Processing Systems*. Vol. 27. Curran Associates, Inc.
- Gorishniy, Yury, Ivan Rubachev, and Artem Babenko (2022). “On Embeddings for Numerical Features in Tabular Deep Learning”. In: *Advances in Neural Information Processing Systems*. Vol. 35. Curran Associates, Inc., pp. 24991–25004.
- Griethe, Henning and Heidrun Schumann (2006). “The Visualization of Uncertain Data: Methods and Problems”. In: *Proceedings of SimVis 2006*.

## Bibliography

- Grigoryan, Gevorg and Penny Rheingans (2004). “Point-based probabilistic surfaces to show surface uncertainty”. In: *IEEE Transactions on Visualization and Computer Graphics* 10.5, pp. 564–573.
- Groenke, Brian, Luke Madaus, and Claire Monteleoni (2021). “ClimAlign: Unsupervised statistical downscaling of climate variables via normalizing flows”. In: *Proceedings of the 10th International Conference on Climate Informatics*. CI2020. virtual, United Kingdom: Association for Computing Machinery, pp. 60–66.
- Grönquist, Peter, Chengyuan Yao, Tal Ben-Nun, Nikoli Dryden, Peter Dueben, Shigang Li, and Torsten Hoefler (2021). “Deep learning for post-processing ensemble weather forecasts”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 379.2194. 20200092.
- Gross, Markus, Hui Wan, Philip J. Rasch, Peter M. Caldwell, David L. Williamson, Daniel Klocke, Christiane Jablonowski, Diana R. Thatcher, Nigel Wood, Mike Cullen, Bob Beare, Martin Willett, Florian Lemarié, Eric Blayo, Sylvie Malardel, Piet Termomia, Almut Gassmann, Peter H. Lauritzen, Hans Johansen, Colin M. Zarzycki, Koichi Sakaguchi, and Ruby Leung (2018). “Physics–Dynamics Coupling in Weather, Climate, and Earth System Models: Challenges and Recent Progress”. In: *Monthly Weather Review* 146.11, pp. 3505–3544.
- Guibas, John, Morteza Mardani, Zongyi Li, Andrew Tao, Anima Anandkumar, and Bryan Catanzaro (2022). *Adaptive Fourier Neural Operators: Efficient Token Mixers for Transformers*. arXiv: 2111.13587 [cs.CV].
- Guidotti, Riccardo, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi (2018). “A Survey of Methods for Explaining Black Box Models”. In: *ACM Computing Surveys* 51.5, pp. 1–42.
- Gulrajani, Ishaan, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C. Courville (2017). “Improved Training of Wasserstein GANs”. In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc.
- Guo, Cheng and Felix Berkhahn (2016). *Entity Embeddings of Categorical Variables*. arXiv: 1604.06737 [cs.LG].
- Guo, Li, Shaojie Ye, Jun Han, Hao Zheng, Han Gao, Danny Z. Chen, Jian-Xun Wang, and Chaoli Wang (2020). “SSR-VFD: Spatial Super-Resolution for Vector Field Data Analysis and Visualization”. In: *2020 IEEE Pacific Visualization Symposium (PacificVis)*, pp. 71–80.
- Han, Jun and Chaoli Wang (2020). “TSR-TVD: Temporal Super-Resolution for Time-Varying Data Analysis and Visualization”. In: *IEEE Transactions on Visualization and Computer Graphics* 26.1, pp. 205–215.
- (2022). “SSR-TVD: Spatial Super-Resolution for Time-Varying Data Analysis and Visualization”. In: *IEEE Transactions on Visualization and Computer Graphics* 28.6, pp. 2445–2456.

- (2023). “CoordNet: Data Generation and Visualization Generation for Time-Varying Volumes via a Coordinate-Based Neural Network”. In: *IEEE Transactions on Visualization and Computer Graphics* 29.12, pp. 4951–4963.
- Han, Jun, Hao Zheng, Danny Z. Chen, and Chaoli Wang (2022). “STNet: An End-to-End Generative Framework for Synthesizing Spatiotemporal Super-Resolution Volumes”. In: *IEEE Transactions on Visualization and Computer Graphics* 28.1, pp. 270–280.
- Han, Jun, Hao Zheng, Yunhao Xing, Danny Z. Chen, and Chaoli Wang (2021). “V2V: A Deep Learning Approach to Variable-to-Variable Selection and Translation for Multivariate Time-Varying Data”. In: *IEEE Transactions on Visualization and Computer Graphics* 27.2, pp. 1290–1300.
- Hancock, John T. and Taghi M. Khoshgoftaar (2020). “Survey on categorical data for neural networks”. In: *Journal of Big Data* 7.1. 28.
- He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778.
- He, Wenbin, Junpeng Wang, Hanqi Guo, Ko-Chih Wang, Han-Wei Shen, Mukund Raj, Youssef S. G. Nashed, and Tom Peterka (2020). “InSituNet: Deep Image Synthesis for Parameter Space Exploration of Ensemble Simulations”. In: *IEEE Transactions on Visualization & Computer Graphics* 26.01, pp. 23–33.
- Helbig, Carolin, Hans-Stefan Bauer, Karsten Rink, Volker Wulfmeyer, Michael Frank, and Olaf Kolditz (2014). “Concept and workflow for 3D visualization of atmospheric data in a virtual reality environment for analytical approaches”. In: *Environmental Earth Sciences* 72.10, pp. 3767–3780.
- Helbig, N., R. Mott, A. van Herwijnen, A. Winstral, and T. Jonas (2017). “Parameterizing surface wind speed over complex topography”. In: *Journal of Geophysical Research: Atmospheres* 122.2, pp. 651–667.
- Hersbach, Hans, Bill Bell, Paul Berrisford, Shoji Hirahara, András Horányi, Joaquín Muñoz-Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Dinand Schepers, Adrian Simmons, Cornel Soci, Saleh Abdalla, Xavier Abellan, Gianpaolo Balsamo, Peter Bechtold, Gionata Biavati, Jean Bidlot, Massimo Bonavita, Giovanna De Chiara, Per Dahlgren, Dick Dee, Michail Diamantakis, Rossana Dragani, Johannes Flemming, et al. (2020). “The ERA5 global reanalysis”. In: *Quarterly Journal of the Royal Meteorological Society* 146.730, pp. 1999–2049.
- Hewitson, Bruce C. and Robert George Crane (1996). “Climate downscaling: techniques and application”. In: *Climate Research* 07.2, pp. 85–95.
- Hiebl, Johann and Christoph Frei (2016). “Daily temperature grids for Austria since 1961—concept, creation and applicability”. In: *Theoretical and Applied Climatology* 124.1, pp. 161–178.

## Bibliography

- Hoffman, Robert R., Daphne S. LaDue, H. Michael Mogil, Paul J. Roebber, and J. Gregory Trafton (2023). *Minding the Weather: How Expert Forecasters Think*. The MIT Press.
- Höhlein, Kevin, Timothy Hewson, and Rüdiger Westermann (2024a). *Topographic Visualization of Near-surface Temperatures for Improved Lapse Rate Estimation*. arXiv: 2406.11894 [physics.ao-ph].
- Höhlein, Kevin, Michael Kern, Timothy Hewson, and Rüdiger Westermann (2020). “A comparative study of convolutional neural network models for wind field downscaling”. In: *Meteorological Applications* 27.6, e1961.
- Höhlein, Kevin, Benedikt Schulz, Rüdiger Westermann, and Sebastian Lerch (2024b). “Postprocessing of Ensemble Weather Forecasts Using Permutation-Invariant Neural Networks”. In: *Artificial Intelligence for the Earth Systems* 3.1, e230070.
- Höhlein, Kevin, Sebastian Weiss, Tobias Necker, Martin Weissmann, Takemasa Miyoshi, and Rüdiger Westermann (2022). “Evaluation of Volume Representation Networks for Meteorological Ensemble Compression”. In: *Vision, Modeling, and Visualization (VMV 2022)*. The Eurographics Association.
- Holton, James R. (2013). *An introduction to dynamic meteorology*. Elsevier Academic Press Boston.
- Horat, Nina and Sebastian Lerch (2024). “Deep Learning for Postprocessing Global Probabilistic Forecasts on Subseasonal Time Scales”. In: *Monthly Weather Review* 152.3, pp. 667–687.
- Hornik, Kurt (1991). “Approximation capabilities of multilayer feedforward networks”. In: *Neural Networks* 4.2, pp. 251–257.
- Hortal, M. and A. J. Simmons (1991). “Use of Reduced Gaussian Grids in Spectral Models”. In: *Monthly Weather Review* 119.4, pp. 1057–1074.
- Huang, Gao, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger (2017). “Densely Connected Convolutional Networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4700–4708.
- Inness, Peter Michael and Steve Dorling (2012). *Operational weather forecasting*. John Wiley & Sons.
- International Civil Aviation Organization (1993). *Manual of the ICAO Standard Atmosphere: extended to 80 kilometres (262 500 feet)*. 3rd Edition. ICAO.
- Ioffe, Sergey and Christian Szegedy (2015). “Batch normalization: accelerating deep network training by reducing internal covariate shift”. In: *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37. ICML’15*. Lille, France: JMLR.org, pp. 448–456.



- Jain, Somay, Wesley Griffin, Afzal Godil, Jeffrey W Bullard, Judith Terrill, and Amitabh Varshney (2017). “Compressed volume rendering using deep learning”. In: *Proceedings of the Large Scale Data Analysis and Visualization Symposium*, pp. 1187–1194.
- Jakob, Jakob, Markus Gross, and Tobias Günther (2021). “A Fluid Flow Data Set for Machine Learning and its Application to Neural Flow Map Interpolation”. In: *IEEE Transactions on Visualization and Computer Graphics* 27.2, pp. 1279–1289.
- Jang, Hankyu and Daeyoung Kim (2022). *D-TensoRF: Tensorial Radiance Fields for Dynamic Scenes*. arXiv: 2212.02375 [cs.CV].
- Jiang, Chiyu "Max", Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Niessner, and Thomas Funkhouser (2020). “Local Implicit Grid Representations for 3D Scenes”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6001–6010.
- Jolliffe, Ian T. and David B. Stephenson (2012). *Forecast verification: a practitioner's guide in atmospheric science*. John Wiley & Sons.
- Jordan, Alexander, Fabian Krüger, and Sebastian Lerch (2018). *Evaluating probabilistic forecasts with scoringRules*. arXiv: 1709.04743 [stat.CO].
- Jumper, John, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, et al. (2021). “Highly accurate protein structure prediction with AlphaFold”. In: *Nature* 596.7873, pp. 583–589.
- Kamal, Aasim, Parashar Dhakal, Ahmad Y. Javaid, Vijay K. Devabhaktuni, Devinder Kaur, Jack Zaiantz, and Robert Marinier (2021). “Recent advances and challenges in uncertainty visualization: a survey”. In: *Journal of Visualization* 24.5, pp. 861–890.
- Kao, David, Alison Luo, Jennifer L. Dungan, and Alex Pang (2002). “Visualizing spatially varying distribution data”. In: *Proceedings Sixth International Conference on Information Visualisation*, pp. 219–225.
- Kehrer, Johannes and Helwig Hauser (2013). “Visualization and Visual Analysis of Multifaceted Scientific Data: A Survey”. In: *IEEE Transactions on Visualization and Computer Graphics* 19.3, pp. 495–513.
- Khan, Asifullah, Anabia Sohail, Umme Zahoora, and Aqsa Saeed Qureshi (2020). “A survey of the recent architectures of deep convolutional neural networks”. In: *Artificial Intelligence Review* 53.8, pp. 5455–5516.
- Khan, Salman, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah (2022). “Transformers in Vision: A Survey”. In: *ACM Computing Surveys* 54.10s, pp. 1–41.

## Bibliography

- Kim, Jiwon, Jung Kwon Lee, and Kyoung Mu Lee (2016). “Accurate Image Super-Resolution Using Very Deep Convolutional Networks”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, pp. 1646–1654.
- Kochkov, Dmitrii, Janni Yuval, Ian Langmore, Peter Norgaard, Jamie Smith, Griffin Mooers, Milan Klöwer, James Lottes, Stephan Rasp, Peter Düben, Sam Hatfield, Peter Battaglia, Alvaro Sanchez-Gonzalez, Matthew Willson, Michael P. Brenner, and Stephan Hoyer (2024). “Neural general circulation models for weather and climate”. In: *Nature* 632.8027, pp. 1060–1066.
- Kolda, Tamara G. and Brett W. Bader (2009). “Tensor Decompositions and Applications”. In: *SIAM Review* 51.3, pp. 455–500.
- Kovachki, Nikola, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhatnatharya, Andrew Stuart, and Anima Anandkumar (2024). “Neural operator: learning maps between function spaces with applications to PDEs”. In: *Journal of Machine Learning Research* 24.89, pp. 1–97.
- Kraskov, Alexander, Harald Stögbauer, and Peter Grassberger (2004). “Estimating mutual information”. In: *Physical Review E* 69 (6), p. 066138.
- Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems*. Vol. 25. Curran Associates, Inc.
- Kumpf, Alexander, Marc Rautenhaus, Michael Riemer, and Rüdiger Westermann (2019). “Visual Analysis of the Temporal Evolution of Ensemble Forecast Sensitivities”. In: *IEEE Transactions on Visualization & Computer Graphics* 25.01, pp. 98–108.
- Kumpf, Alexander, Bianca Tost, Marlene Baumgart, Michael Riemer, Rüdiger Westermann, and Marc Rautenhaus (2018). “Visualizing Confidence in Cluster-Based Ensemble Weather Forecast Analyses”. In: *IEEE Transactions on Visualization and Computer Graphics* 24.1, pp. 109–119.
- Labe, Zachary M. and Elizabeth A. Barnes (2021). “Detecting Climate Signals Using Explainable AI With Single-Forcing Large Ensembles”. In: *Journal of Advances in Modeling Earth Systems* 13.6, e2021MS002464.
- Lai, Wei-Sheng, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang (2017). “Deep Laplacian Pyramid Networks for Fast and Accurate Super-Resolution”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 624–632.
- Lam, Remi, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, Alexander Merose, Stephan Hoyer, George Holland, Oriol Vinyals, Jacklynn Stott, Alexander Pritzel, Shakir Mohamed, and Peter Battaglia (2023). “Learning skillful medium-range global weather forecasting”. In: *Science* 382.6677, pp. 1416–1421.

- Lang, Simon, Mihai Alexe, Matthew Chantry, Jesper Dramsch, Florian Pinault, Baudouin Raoult, Mariana C. A. Clare, Christian Lessig, Michael Maier-Gerber, Linus Magnusson, Zied Ben Bouallègue, Ana Prieto Nemesio, Peter D. Dueben, Andrew Brown, Florian Pappenberger, and Florence Rabier (2024). *AIFS – ECMWF’s data-driven forecasting system*. arXiv: 2406.01465 [physics.ao-ph].
- LeCun, Yann, Léon Bottou, Yoshua Bengio, and Patrick Haffner (1998). “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11, pp. 2278–2324.
- Ledig, Christian, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. (2017). “Photo-realistic single image super-resolution using a generative adversarial network”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4681–4690.
- Lee, Juho, Yoonho Lee, Jungtaek Kim, Adam Kosiosek, Seungjin Choi, and Yee Whye Teh (2019). “Set Transformer: A Framework for Attention-based Permutation-Invariant Neural Networks”. In: *Proceedings of the 36th International Conference on Machine Learning*. Vol. 97. PMLR, pp. 3744–3753.
- Lerch, Sebastian, Sándor Baran, Annette Möller, Jürgen Groß, Roman Schefzik, Stephan Hemri, and Maximiliane Graeter (2020). “Simulation-based comparison of multivariate ensemble post-processing methods”. In: *Nonlinear Processes in Geophysics* 27.2, pp. 349–371.
- Levoy, Marc (1988). “Display of surfaces from volume data”. In: *IEEE Computer Graphics and Applications* 8.3, pp. 29–37.
- (1990). “Efficient ray tracing of volume data”. In: *ACM Transactions on Graphics* 9.3, pp. 245–261.
- Li, Hao, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein (2018). “Visualizing the Loss Landscape of Neural Nets”. In: *Advances in Neural Information Processing Systems*. Vol. 31. Curran Associates, Inc.
- Li, Haoying, Yifan Yang, Meng Chang, Shiqi Chen, Huajun Feng, Zhihai Xu, Qi Li, and Yueting Chen (2022). “SRDiff: Single image super-resolution with diffusion probabilistic models”. In: *Neurocomputing* 479, pp. 47–59.
- Li, Zongyi, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhat-tacharya, Andrew Stuart, and Anima Anandkumar (2021). *Fourier Neural Operator for Parametric Partial Differential Equations*. arXiv: 2010.08895 [cs.LG].
- Li, Zongyi, Hongkai Zheng, Nikola Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Kamyar Azizzadenesheli, and Anima Anandkumar (2024). “Physics-Informed Neural Operator for Learning Partial Differential Equations”. In: *ACM / IMS Journal of Data Science* 1.3, pp. 1–27.

## Bibliography

- Liebmann, Tom, Gunther H. Weber, and Gerik Scheuermann (2018). “Hierarchical Correlation Clustering in Multiple 2D Scalar Fields”. In: *Computer Graphics Forum* 37.3, pp. 1–12.
- Lim, Bee, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee (2017). “Enhanced Deep Residual Networks for Single Image Super-Resolution”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 136–144.
- Linardatos, Pantelis, Vasilis Papastefanopoulos, and Sotiris Kotsiantis (2021). “Explainable AI: A Review of Machine Learning Interpretability Methods”. In: *Entropy* 23.1. 18.
- Lindstrom, Peter (2014). “Fixed-Rate Compressed Floating-Point Arrays”. In: *IEEE Transactions on Visualization and Computer Graphics* 20.12, pp. 2674–2683.
- Liu, Xiaotong and Han-Wei Shen (2016). “Association Analysis for Visual Exploration of Multivariate Scientific Data Sets”. In: *IEEE Transactions on Visualization and Computer Graphics* 22.1, pp. 955–964.
- Lorensen, William E. and Harvey E. Cline (1987). “Marching cubes: A high resolution 3D surface construction algorithm”. In: *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '87. New York, NY, USA: Association for Computing Machinery, pp. 163–169.
- Lu, Yuzhe, Kairong Jiang, Joshua A. Levine, and Matthew Berger (2021). “Compressive Neural Representations of Volumetric Scalar Fields”. In: *Computer Graphics Forum* 40.3, pp. 135–146.
- Lu, Zhou, Hongming Pu, Feicheng Wang, Zhiqiang Hu, and Liwei Wang (2017). “The Expressive Power of Neural Networks: A View from the Width”. In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc.
- Lugmayr, Andreas, Martin Danelljan, Luc Van Gool, and Radu Timofte (2020). “SR-Flow: Learning the Super-Resolution Space with Normalizing Flow”. In: *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V*. Glasgow, United Kingdom: Springer-Verlag, pp. 715–732.
- Lundberg, Scott M and Su-In Lee (2017). “A Unified Approach to Interpreting Model Predictions”. In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc.
- Luo, Haolin, Fei Ge, Kangquan Yang, Shoupeng Zhu, Ting Peng, Wenyue Cai, Xiaoran Liu, and Weiwei Tang (2019). “Assessment of ECMWF reanalysis data in complex terrain: Can the CERA-20C and ERA-Interim data sets replicate the variation in surface air temperatures over Sichuan, China?” In: *International Journal of Climatology* 39.15, pp. 5619–5634.

- Lussana, Cristian, Ole Einar Tveito, and Francesco Uboldi (2018). “Three-dimensional spatial interpolation of 2 m temperature over Norway”. In: *Quarterly Journal of the Royal Meteorological Society* 144.711, pp. 344–364.
- Lyle, Clare, Mark van der Wilk, Marta Kwiatkowska, Yarin Gal, and Benjamin Bloem-Reddy (2020). *On the Benefits of Invariance in Neural Networks*. arXiv: 2005.00178 [cs.LG].
- MacEachren, Alan M., Anthony Robinson, Susan Hopper, Steven Gardner, Robert Murray, Mark Gahegan, and Elisabeth Hetzler (2005). “Visualizing Geospatial Information Uncertainty: What We Know and What We Need to Know”. In: *Cartography and Geographic Information Science* 32.3, pp. 139–160.
- Malardel, S., Nils Wedi, Willem Deconinck, Michail Diamantakis, Christian Kuehnlein, G. Mozdzyński, M. Hamrud, and Piotr Smolarkiewicz (2016). “A new grid for the IFS”. In: *ECMWF Newsletter* 146, pp. 23–28.
- Mao, Xiaojiao, Chunhua Shen, and Yu-Bin Yang (2016). “Image Restoration Using Very Deep Convolutional Encoder-Decoder Networks with Symmetric Skip Connections”. In: *Advances in Neural Information Processing Systems*. Vol. 29. Curran Associates, Inc.
- Maraun, Douglas and Martin Widmann (2018). *Statistical Downscaling and Bias Correction for Climate Research*. Cambridge University Press.
- Martel, Julien N. P., David B. Lindell, Connor Z. Lin, Eric R. Chan, Marco Monteiro, and Gordon Wetzstein (2021). *ACORN: Adaptive Coordinate Networks for Neural Scene Representation*. arXiv: 2105.02788 [cs.CV].
- Max, Nelson (1995). “Optical models for direct volume rendering”. In: *IEEE Transactions on Visualization and Computer Graphics* 1.2, pp. 99–108.
- McGovern, Amy, Ryan Lagerquist, David John Gagne, G. Eli Jergensen, Kimberly L. Elmore, Cameron R. Homeyer, and Travis Smith (2019). “Making the Black Box More Transparent: Understanding the Physical Implications of Machine Learning”. In: *Bulletin of the American Meteorological Society* 100.11, pp. 2175–2199.
- Mescheder, Lars, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger (2019). “Occupancy networks: Learning 3d reconstruction in function space”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4460–4470.
- Messner, Jakob W., Georg J. Mayr, and Achim Zeileis (2017). “Nonhomogeneous Boosting for Predictor Selection in Ensemble Postprocessing”. In: *Monthly Weather Review* 145.1, pp. 137–147.
- Michalkiewicz, Mateusz, Jhony K. Pontes, Dominic Jack, Mahsa Baktashmotlagh, and Anders Eriksson (2019). *Deep Level Sets: Implicit Surface Representations for 3D Shape Inference*. arXiv: 1901.06802 [cs.CV].

## Bibliography

- Middleton, Don, Tim Scheitlin, and Bob Wilhelmson (2005). “44 - Visualization in Weather and Climate Research”. In: *Visualization Handbook*. Burlington: Butterworth-Heinemann, pp. 845–871.
- Mildenhall, Ben, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng (2021). “NeRF: representing scenes as neural radiance fields for view synthesis”. In: *Commun. ACM* 65.1, pp. 99–106.
- Mishra, Rahul, Hari Prabhat Gupta, and Tanima Dutta (2020). *A Survey on Deep Neural Network Compression: Challenges, Overview, and Solutions*. arXiv: 2010.03954 [cs.LG].
- Mitchell, Tom M. (1980). *The need for biases in learning generalizations*. Tech. rep. CBM-TR-117. Rutgers University, New Brunswick.
- Mlakar, Peter, Janko Merše, and Jana Faganeli Pucer (2024). “Ensemble weather forecast post-processing with a flexible probabilistic neural network approach”. In: *Quarterly Journal of the Royal Meteorological Society* 150.764, pp. 4156–4177.
- Molnar, Christoph, Gunnar König, Bernd Bischl, and Giuseppe Casalicchio (2024). “Model-agnostic feature importance and effects with dependent features: a conditional subgroup approach”. In: *Data Mining and Knowledge Discovery* 38.5, pp. 2903–2941.
- Molteni, F., R. Buizza, T. N. Palmer, and T. Petroliaigis (1996). “The ECMWF Ensemble Prediction System: Methodology and validation”. In: *Quarterly Journal of the Royal Meteorological Society* 122.529, pp. 73–119.
- Monmonier, Mark (2000). *Air apparent: How meteorologists learned to map, predict, and dramatize weather*. University of Chicago Press.
- Moon, Young-II, Balaji Rajagopalan, and Upmanu Lall (1995). “Estimation of mutual information using kernel density estimators”. In: *Physical Review E* 52 (3), pp. 2318–2321.
- Mu, Bin, Bo Qin, Shijin Yuan, and Xiaoyun Qin (2020). “A Climate Downscaling Deep Learning Model considering the Multiscale Spatial Correlations and Chaos of Meteorological Events”. In: *Mathematical Problems in Engineering* 2020.1. 7897824.
- Müller, Thomas, Alex Evans, Christoph Schied, and Alexander Keller (2022). “Instant neural graphics primitives with a multiresolution hash encoding”. In: *ACM Transactions on Graphics* 41.4, pp. 1–15.
- Müller, Thomas, Fabrice Rousselle, Jan Novák, and Alexander Keller (2021). “Real-time neural radiance caching for path tracing”. In: *ACM Transactions on Graphics* 40.4.
- Murphy, Kevin P. (2022). *Probabilistic Machine Learning: An introduction*. MIT Press.
- Nasrollahi, Kamal and Thomas B. Moeslund (2014). “Super-resolution: a comprehensive survey”. In: *Machine Vision and Applications* 25.6, pp. 1423–1468.

- Natural Earth (2012). *Gray Earth with Shaded Relief, Hypsography, and Flat Water (Version 3.2.0)*. [https://www.naturalearthdata.com/http://www.naturalearthdata.com/download/10m/raster/GRAY\\_HR\\_SR\\_W.zip](https://www.naturalearthdata.com/http://www.naturalearthdata.com/download/10m/raster/GRAY_HR_SR_W.zip). Accessed: 2024-09-17.
- Necker, Tobias, Stefan Geiss, Martin Weissmann, Juan Ruiz, Takemasa Miyoshi, and Guo-Yuan Lien (2020). “A convective-scale 1,000-member ensemble simulation and potential applications”. In: *Quarterly Journal of the Royal Meteorological Society* 146.728, pp. 1423–1442.
- Neuhauser, Christoph, Josef Stumpfegger, and Rüdiger Westermann (2024). “Adaptive Sampling of 3D Spatial Correlations for Focus+Context Visualization”. In: *IEEE Transactions on Visualization & Computer Graphics* 30.02, pp. 1608–1623.
- Nguyen, Tung, Johannes Brandstetter, Ashish Kapoor, Jayesh K. Gupta, and Aditya Grover (2023). *ClimaX: A foundation model for weather and climate*. arXiv: 2301.10343 [cs.LG].
- Nocke, Thomas, Till Sterzel, Michael Böttinger, Markus Wrobel, et al. (2008). “Visualization of climate and climate change data: An overview”. In: *Digital earth summit on geoinformatics*, pp. 226–232.
- Olivetti, Leonardo and Gabriele Messori (2024). “Advances and prospects of deep learning for medium-range extreme weather forecasting”. In: *Geoscientific Model Development* 17.6, pp. 2347–2358.
- Palmer, Tim (2019). “The ECMWF ensemble prediction system: Looking back (more than) 25 years and projecting forward 25 years”. In: *Quarterly Journal of the Royal Meteorological Society* 145.S1, pp. 12–24.
- Palmer, Tim N., Roberto Buizza, F. Doblas-Reyes, T. Jung, Martin Leutbecher, G.J. Shutts, M. Steinheimer, and Antje Weisheimer (2009). “Stochastic parametrization and model uncertainty”. In: *ECMWF Technical Memoranda* 598, p. 42.
- Pan, Baoxiang, Kuolin Hsu, Amir AghaKouchak, and Soroosh Sorooshian (2019). “Improving Precipitation Estimation Using Convolutional Neural Network”. In: *Water Resources Research* 55.3, pp. 2301–2321.
- Pant, Pranshu and Amir Barati Farimani (2021). *Deep Learning for Efficient Reconstruction of High-Resolution Turbulent DNS Data*. arXiv: 2010.11348 [physics.flu-dyn].
- Pantillon, Florian, Sebastian Lerch, Peter Knippertz, and Ulrich Corsmeier (2018). “Forecasting wind gusts in winter storms using a calibrated convection-permitting ensemble”. In: *Quarterly Journal of the Royal Meteorological Society* 144.715, pp. 1864–1881.
- Park, Jeong Joon, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove (2019). “DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation”. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 165–174.

## Bibliography

- Park, Seong-Jin, Hyeongseok Son, Sunghyun Cho, Ki-Sang Hong, and Seungyong Lee (2018). “Srfeat: Single image super-resolution with feature discrimination”. In: *Proceedings of the European conference on computer vision (ECCV)*, pp. 439–455.
- Passarella, Linsey S., Salil Mahajan, Anikesh Pal, and Matthew R. Norman (2022). “Reconstructing High Resolution ESM Data Through a Novel Fast Super Resolution Convolutional Neural Network (FSRCNN)”. In: *Geophysical Research Letters* 49.4, e2021GL097571.
- Paszke, Adam, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala (2019). “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc.
- Pathak, Jaideep, Shashank Subramanian, Peter Harrington, Sanjeev Raja, Ashesh Chattopadhyay, Morteza Mardani, Thorsten Kurth, David Hall, Zongyi Li, Kamyar Azizzadenesheli, Pedram Hassanzadeh, Karthik Kashinath, and Animashree Anandkumar (2022). *FourCastNet: A Global Data-driven High-resolution Weather Model using Adaptive Fourier Neural Operators*. arXiv: 2202.11214 [physics.ao-ph].
- Pfaffelmoser, Tobias, Matthias Reitering, and Rüdiger Westermann (2011). “Visualizing the Positional and Geometrical Variability of Isosurfaces in Uncertain Scalar Fields”. In: *Computer Graphics Forum* 30.3, pp. 951–960.
- Pfaffelmoser, Tobias and Rüdiger Westermann (2012). “Visualization of Global Correlation Structures in Uncertain 2D Scalar Fields”. In: *Computer Graphics Forum* 31.3pt2, pp. 1025–1034.
- (2013). “Correlation Visualization for Structural Uncertainty Analysis”. In: *International Journal for Uncertainty Quantification* 3.2, pp. 171–186.
- Pothkow, Kai and Hans-Christian Hege (2011). “Positional Uncertainty of Isocontours: Condition Analysis and Probabilistic Measures”. In: *IEEE Transactions on Visualization and Computer Graphics* 17.10, pp. 1393–1406.
- Prince, Simon J.D. (2023). *Understanding Deep Learning*. <http://udlbook.com>. The MIT Press.
- Radford, Alec, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever (2021). “Learning Transferable Visual Models From Natural Language Supervision”. In: *Proceedings of the 38th International Conference on Machine Learning*. Vol. 139. PMLR, pp. 8748–8763.
- Radford, Alec, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever (2018). *Improving Language Understanding by Generative Pre-Training*. OpenAI.



- Raftery, Adrian E., Tilmann Gneiting, Fadoua Balabdaoui, and Michael Polakowski (2005). “Using Bayesian Model Averaging to Calibrate Forecast Ensembles”. In: *Monthly Weather Review* 133.5, pp. 1155–1174.
- Rahaman, Nasim, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville (2019). “On the Spectral Bias of Neural Networks”. In: *Proceedings of the 36th International Conference on Machine Learning*. Vol. 97. PMLR, pp. 5301–5310.
- Rampal, Neelesh, Peter B. Gibson, Abha Sood, Stephen Stuart, Nicolas C. Fauchereau, Chris Brandolino, Ben Noll, and Tristan Meyers (2022). “High-resolution downscaling with interpretable deep learning: Rainfall extremes over New Zealand”. In: *Weather and Climate Extremes* 38. 100525.
- Rasp, Stephan, Peter D. Dueben, Sebastian Scher, Jonathan A. Weyn, Soukayna Mouatadid, and Nils Thuerey (2020). “WeatherBench: A Benchmark Data Set for Data-Driven Weather Forecasting”. In: *Journal of Advances in Modeling Earth Systems* 12.11, e2020MS002203.
- Rasp, Stephan, Stephan Hoyer, Alexander Merose, Ian Langmore, Peter Battaglia, Tyler Russell, Alvaro Sanchez-Gonzalez, Vivian Yang, Rob Carver, Shreya Agrawal, Matthew Chantry, Zied Ben Bouallegue, Peter Dueben, Carla Bromberg, Jared Sisk, Luke Barrington, Aaron Bell, and Fei Sha (2024). “WeatherBench 2: A Benchmark for the Next Generation of Data-Driven Global Weather Models”. In: *Journal of Advances in Modeling Earth Systems* 16.6, e2023MS004019.
- Rasp, Stephan and Sebastian Lerch (2018). “Neural Networks for Postprocessing Ensemble Weather Forecasts”. In: *Monthly Weather Review* 146.11, pp. 3885–3900.
- Rautenhaus, Marc, Michael Böttinger, Stephan Siemen, Robert Hoffman, Robert M. Kirby, Mahsa Mirzargar, Niklas Röber, and Rüdiger Westermann (2018). “Visualization in Meteorology—A Survey of Techniques and Tools for Data Analysis Tasks”. In: *IEEE Transactions on Visualization and Computer Graphics* 24.12, pp. 3268–3296.
- Reichstein, Markus, Gustau Camps-Valls, Bjorn Stevens, Martin Jung, Joachim Denzler, Nuno Carvalhais, and Prabhat (2019). “Deep learning and process understanding for data-driven Earth system science”. In: *Nature* 566.7743, pp. 195–204.
- Ren, Pengzhen, Yun Xiao, Xiaojun Chang, Po-yao Huang, Zhihui Li, Xiaojiang Chen, and Xin Wang (2021). “A Comprehensive Survey of Neural Architecture Search: Challenges and Solutions”. In: *ACM Computing Surveys* 54.4, pp. 1–34.
- Retchless, David P. and Cynthia A. Brewer (2016). “Guidance for representing uncertainty on global temperature change maps”. In: *International Journal of Climatology* 36.3, pp. 1143–1159.
- Röber, Niklas, Michael Böttinger, and Bjorn Stevens (2021). “Visualization of Climate Science Simulation Data”. In: *IEEE Computer Graphics and Applications* 41.1, pp. 42–48.

## Bibliography

- Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (2015). “U-Net: Convolutional Networks for Biomedical Image Segmentation”. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Cham: Springer International Publishing, pp. 234–241.
- Rosenblatt, F. (1958). “The perceptron: A probabilistic model for information storage and organization in the brain.” In: *Psychological Review* 65.6, pp. 386–408.
- Rüschendorf, Ludger (2009). “On the distributional transform, Sklar’s theorem, and the empirical copula process”. In: *Journal of Statistical Planning and Inference* 139.11. Special Issue: The 8th Tartu Conference on Multivariate Statistics & The 6th Conference on Multivariate Distributions with Fixed Marginals, pp. 3921–3927.
- Sahakyan, Maria, Zeyar Aung, and Talal Rahwan (2021). “Explainable Artificial Intelligence for Tabular Data: A Survey”. In: *IEEE Access* 9, pp. 135392–135422.
- Sahoo, Saroj and Matthew Berger (2021). “Integration-Aware Vector Field Super Resolution”. In: *EuroVis 2021 - Short Papers*. The Eurographics Association, pp. 49–53.
- Sajjadi, Mehdi SM, Bernhard Scholkopf, and Michael Hirsch (2017). “Enhancenet: Single image super-resolution through automated texture synthesis”. In: *Proceedings of the IEEE international conference on computer vision*, pp. 4491–4500.
- Sannai, Akiyoshi, Yuuki Takai, and Matthieu Cordonnier (2019). *Universal approximations of permutation invariant/equivariant functions by deep neural networks*. arXiv: 1903.01939 [cs.LG].
- Sauber, Natascha, Holger Theisel, and Hans-peter Seidel (2006). “Multifield-Graphs: An Approach to Visualizing Correlations in Multifield Scalar Data”. In: *IEEE Transactions on Visualization and Computer Graphics* 12.5, pp. 917–924.
- Scheuerer, M. (2014). “Probabilistic quantitative precipitation forecasting using Ensemble Model Output Statistics”. In: *Quarterly Journal of the Royal Meteorological Society* 140.680, pp. 1086–1096.
- Scheuerer, Michael and Thomas M. Hamill (2015). “Variogram-Based Proper Scoring Rules for Probabilistic Forecasts of Multivariate Quantities”. In: *Monthly Weather Review* 143.4, pp. 1321–1334.
- Scheuerer, Michael, Matthew B. Switanek, Rochelle P. Worsnop, and Thomas M. Hamill (2020). “Using Artificial Neural Networks for Generating Probabilistic Subseasonal Precipitation Forecasts over California”. In: *Monthly Weather Review* 148.8, pp. 3489–3506.
- Schlosser, Lisa, Torsten Hothorn, Reto Stauffer, and Achim Zeileis (2019). “Distributional regression forests for probabilistic precipitation forecasting in complex terrain”. In: *The Annals of Applied Statistics* 13.3, pp. 1564–1589.

- Schulz, Benedikt, Mehrez El Ayari, Sebastian Lerch, and Sándor Baran (2021). “Post-processing numerical weather prediction ensembles for probabilistic solar irradiance forecasting”. In: *Solar Energy* 220, pp. 1016–1031.
- Schulz, Benedikt and Sebastian Lerch (2022). “Machine Learning Methods for Postprocessing Ensemble Forecasts of Wind Gusts: A Systematic Comparison”. In: *Monthly Weather Review* 150.1, pp. 235–257.
- Schulz, Karl, Leon Sixt, Federico Tombari, and Tim Landgraf (2020). *Restricting the Flow: Information Bottlenecks for Attribution*. arXiv: 2001.00396 [stat.ML].
- Sdraka, Maria, Ioannis Papoutsis, Bill Psomas, Konstantinos Vlachos, Konstantinos Ioannidis, Konstantinos Karantzas, Ilias Gialampoukidis, and Stefanos Vrochidis (2022). “Deep Learning for Downscaling Remote Sensing Images: Fusion and super-resolution”. In: *IEEE Geoscience and Remote Sensing Magazine* 10.3, pp. 202–255.
- Selz, Tobias and George C. Craig (2023). “Can Artificial Intelligence-Based Weather Prediction Models Simulate the Butterfly Effect?” In: *Geophysical Research Letters* 50.20, e2023GL105747.
- Serifi, Agon, Tobias Günther, and Nikolina Ban (2021). “Spatio-Temporal Downscaling of Climate Data Using Convolutional and Error-Predicting Neural Networks”. In: *Frontiers in Climate* 3. 656479.
- Sha, Yingkai, David John Gagne II, Gregory West, and Roland Stull (2020a). “Deep-Learning-Based Gridded Downscaling of Surface Meteorological Variables in Complex Terrain. Part I: Daily Maximum and Minimum 2-m Temperature”. In: *Journal of Applied Meteorology and Climatology* 59.12, pp. 2057–2073.
- (2020b). “Deep-Learning-Based Gridded Downscaling of Surface Meteorological Variables in Complex Terrain. Part II: Daily Precipitation”. In: *Journal of Applied Meteorology and Climatology* 59.12, pp. 2075–2092.
- Shapley, Lloyd S. (1951). *Notes on the N-Person Game – II: The Value of an N-Person Game*. Santa Monica, CA: RAND Corporation.
- Shen, Jingyi and Han-Wei Shen (2023). “PSRFlow: Probabilistic Super Resolution with Flow-Based Models for Scientific Data”. In: *IEEE Transactions on Visualization and Computer Graphics* 30.1, pp. 986–996.
- Sheridan, P. F., S. B. Vosper, and A. R. Brown (2014). “Characteristics of cold pools observed in narrow valleys and dependence on external conditions”. In: *Quarterly Journal of the Royal Meteorological Society* 140.679, pp. 715–728.
- Sheridan, Peter, Samantha Smith, Andrew Brown, and Simon Vosper (2010). “A simple height-based correction for temperature downscaling in complex terrain”. In: *Meteorological Applications* 17.3, pp. 329–339.

## Bibliography

- Sheridan, Peter, Simon Vosper, and Samantha Smith (2018). “A Physically Based Algorithm for Downscaling Temperature in Complex Terrain”. In: *Journal of Applied Meteorology and Climatology* 57.8, pp. 1907–1929.
- Shi, Neng, Jiayi Xu, Haoyu Li, Hanqi Guo, Jonathan Woodring, and Han-Wei Shen (2023). “VDL-Surrogate: A View-Dependent Latent-based Model for Parameter Space Exploration of Ensemble Simulations”. In: *IEEE Transactions on Visualization & Computer Graphics* 29.01, pp. 820–830.
- Shi, Neng, Jiayi Xu, Skylar W. Wurster, Hanqi Guo, Jonathan Woodring, Luke P. Van Roekel, and Han-Wei Shen (2022). “GNN-Surrogate: A Hierarchical and Adaptive Graph Neural Network for Parameter Space Exploration of Unstructured-Mesh Ocean Simulations”. In: *IEEE Transactions on Visualization and Computer Graphics* 28.6, pp. 2301–2313.
- Shwartz-Ziv, Ravid and Amitai Armon (2022). “Tabular data: Deep learning is not all you need”. In: *Information Fusion* 81, pp. 84–90.
- Simmons, A. J. and R. Strüfing (1983). “Numerical forecasts of stratospheric warming events using a model with a hybrid vertical coordinate”. In: *Quarterly Journal of the Royal Meteorological Society* 109.459, pp. 81–111.
- Sitzmann, Vincent, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein (2020). “Implicit Neural Representations with Periodic Activation Functions”. In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., pp. 7462–7473.
- Sitzmann, Vincent, Semon Rezhikov, Bill Freeman, Josh Tenenbaum, and Fredo Durand (2021). “Light Field Networks: Neural Scene Representations with Single-Evaluation Rendering”. In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., pp. 19313–19325.
- Sitzmann, Vincent, Michael Zollhoefer, and Gordon Wetzstein (2019). “Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations”. In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc.
- Smith, S. A., A. R. Brown, S. B. Vosper, P. A. Murkin, and A. T. Veal (2010). “Observations and Simulations of Cold Air Pooling in Valleys”. In: *Boundary-Layer Meteorology* 134.1, pp. 85–108.
- Smith, Samuel L. and Quoc V. Le (2018). *A Bayesian Perspective on Generalization and Stochastic Gradient Descent*. arXiv: 1710.06451 [cs.LG].
- Soelch, Maximilian, Adnan Akhundov, Patrick van der Smagt, and Justin Bayer (2019). “On Deep Set Learning and the Choice of Aggregations”. In: *Artificial Neural Networks and Machine Learning – ICANN 2019: Theoretical Neural Computation*. Cham: Springer International Publishing, pp. 444–457.

- Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov (2014). “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. In: *Journal of Machine Learning Research* 15.56, pp. 1929–1958.
- Stauffer, Reto, Georg J. Mayr, Markus Dabernig, and Achim Zeileis (2015). “Somewhere Over the Rainbow: How to Make Effective Use of Colors in Meteorological Visualizations”. In: *Bulletin of the American Meteorological Society* 96.2, pp. 203–216.
- Stengel, Karen, Andrew Glaws, Dylan Hettinger, and Ryan N. King (2020). “Adversarial super-resolution of climatological wind and solar data”. In: *Proceedings of the National Academy of Sciences* 117.29, pp. 16805–16815.
- Strobl, Carolin, Anne-Laure Boulesteix, Thomas Kneib, Thomas Augustin, and Achim Zeileis (2008). “Conditional variable importance for random forests”. In: *BMC Bioinformatics* 9.1. 307.
- Sun, Yongjian, Kefeng Deng, Kaijun Ren, Jia Liu, Chongjiu Deng, and Yongjun Jin (2024). “Deep learning in statistical downscaling for deriving high spatial resolution gridded meteorological data: A systematic review”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 208, pp. 14–38.
- Taillardat, Maxime, Olivier Mestre, Michaël Zamo, and Philippe Naveau (2016). “Calibrated Ensemble Forecasts Using Quantile Regression Forests and Ensemble Model Output Statistics”. In: *Monthly Weather Review* 144.6, pp. 2375–2393.
- Takikawa, Towaki, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler (2021). “Neural geometric level of detail: Real-time rendering with implicit 3d shapes”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11358–11367.
- Tancik, Matthew, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng (2020). “Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains”. In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., pp. 7537–7547.
- Tang, Kaiyuan and Chaoli Wang (2024). “STSR-INR: Spatiotemporal super-resolution for multivariate time-varying volumetric data via implicit neural representation”. In: *Computers & Graphics* 119. 103874.
- Tappen, Marshall F., Bryan C. Russell, and William T Freeman (2003). “Exploiting the sparse derivative prior for super-resolution and image demosaicing”. In: *Proceedings of the Third International Workshop Statistical and Computational Theories of Vision*, pp. 1–28.
- Tempest, Kirsten I., George C. Craig, and Jonas R. Brehmer (2023). “Convergence of forecast distributions in a 100,000-member idealised convective-scale ensemble”. In: *Quarterly Journal of the Royal Meteorological Society* 149.752, pp. 677–702.

## Bibliography

- Tewari, Ayush, Justus Thies, Ben Mildenhall, Pratul Srinivasan, Edgar Tretschk, Yifan Wang, Christoph Lassner, Vincent Sitzmann, Ricardo Martin-Brualla, Stephen Lombardi, Tomas Simon, Christian Theobalt, Matthias Nießner, Jonathan T. Barron, Gordon Wetzstein, Michael Zollhöfer, and Vladislav Golyanik (2022). “Advances in Neural Rendering”. In: *Computer Graphics Forum* 41.2, pp. 703–735.
- Thorarinsdottir, Thordis L. and Tilmann Gneiting (2010). “Probabilistic forecasts of wind speed: ensemble model output statistics by using heteroscedastic censored regression”. In: *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 173.2, pp. 371–388.
- Tishby, Naftali, Fernando C. Pereira, and William Bialek (2000). *The information bottleneck method*. arXiv: physics/0004057 [physics.data-an].
- Tjøstheim, Dag, Håkon Otneim, and Bård Støve (2022). “Statistical Dependence: Beyond Pearson’s  $\rho$ ”. In: *Statistical Science* 37.1, pp. 90–109.
- Tory, Melanie and Torsten Möller (2004). “Human factors in visualization research”. In: *IEEE Transactions on Visualization and Computer Graphics* 10.1, pp. 72–84.
- Treese, G. M., R. W. Prager, and A. H. Gee (1999). “Regularised marching tetrahedra: improved iso-surface extraction”. In: *Computers & Graphics* 23.4, pp. 583–598.
- Uboldi, Francesco, Cristian Lussana, and Marta Salvati (2008). “Three-dimensional spatial interpolation of surface meteorological observations from high-resolution local networks”. In: *Meteorological Applications* 15.3, pp. 331–345.
- Ulyanov, Dmitry, Andrea Vedaldi, and Victor Lempitsky (2018). “Deep Image Prior”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9446–9454.
- Vallis, Geoffrey K. (2017). *Atmospheric and oceanic fluid dynamics*. Cambridge University Press.
- Van Schaeybroeck, Bert and Stéphane Vannitsem (2015). “Ensemble post-processing using member-by-member approaches: theoretical aspects”. In: *Quarterly Journal of the Royal Meteorological Society* 141.688, pp. 807–818.
- Vandal, Thomas, Evan Kodra, Sangram Ganguly, Andrew Michaelis, Ramakrishna Nemani, and Auroop R. Ganguly (2017). “DeepSD: Generating High Resolution Climate Change Projections through Single Image Super-Resolution”. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’17. Halifax, NS, Canada: Association for Computing Machinery, pp. 1663–1672.
- Vannitsem, Stéphane, John Bjørnar Bremnes, Jonathan Demaeyer, Gavin R. Evans, Jonathan Flowerdew, Stephan Hemri, Sebastian Lerch, Nigel Roberts, Susanne Theis, Aitor Atencia, Zied Ben Bouallègue, Jonas Bhend, Markus Dabernig, Lesley De Cruz, Leila Hieta, Olivier Mestre, Lionel Moret, Iris Odak Plenković, Maurice Schmeits,

- Maxime Taillardat, Joris Van den Bergh, Bert Van Schaeybroeck, Kirien Whan, and Jussi Ylhaisi (2021). “Statistical Postprocessing for Weather Forecasts: Review, Challenges, and Avenues in a Big Data World”. In: *Bulletin of the American Meteorological Society* 102.3, E681–E699.
- Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin (2017). “Attention is All you Need”. In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., pp. 5998–6008.
- Veldkamp, Simon, Kirien Whan, Sjoerd Dirksen, and Maurice Schmeits (2021). “Statistical Postprocessing of Wind Speed Forecasts Using Convolutional Neural Networks”. In: *Monthly Weather Review* 149.4, pp. 1141–1152.
- Vogel, Peter, Peter Knippertz, Andreas H. Fink, Andreas Schlueter, and Tilmann Gneiting (2018). “Skill of Global Raw and Postprocessed Ensemble Predictions of Rainfall over Northern Tropical Africa”. In: *Weather and Forecasting* 33.2, pp. 369–388.
- Vosper, S. B. and A. R. Brown (2008). “Numerical Simulations of Sheltering in Valleys: The Formation of Nighttime Cold-Air Pools”. In: *Boundary-Layer Meteorology* 127.3, pp. 429–448.
- Wagstaff, Edward, Fabian B. Fuchs, Martin Engelcke, Ingmar Posner, and Michael Osborne (2019). *On the Limitations of Representing Functions on Sets*. arXiv: 1901.09006 [cs.LG].
- Wallace, John M. and Peter V. Hobbs (2006). *Atmospheric science: an introductory survey*. Vol. 2. Elsevier.
- Wang, Junpeng, Subhashis Hazarika, Cheng Li, and Han-Wei Shen (2019a). “Visualization and Visual Analysis of Ensemble Data: A Survey”. In: *IEEE Transactions on Visualization and Computer Graphics* 25.9, pp. 2853–2872.
- Wang, Xintao, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy (2019b). “ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks”. In: *Computer Vision – ECCV 2018 Workshops: Munich, Germany, September 8-14, 2018, Proceedings, Part V*. Munich, Germany: Springer-Verlag, pp. 63–79.
- Wang, Z., E.P. Simoncelli, and A.C. Bovik (2003). “Multiscale structural similarity for image quality assessment”. In: *The Thirtieth-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*. Vol. 2, pp. 1398–1402.
- Wang, Zhihao, Jian Chen, and Steven C. H. Hoi (2021). “Deep Learning for Image Super-Resolution: A Survey”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.10, pp. 3365–3387.

## Bibliography

- Wei, Min and Xuesong Zhang (2023). “Super-Resolution Neural Operator”. In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, pp. 18247–18256.
- Weiss, S., P. Hermüller, and R. Westermann (2022). “Fast Neural Representations for Direct Volume Rendering”. In: *Computer Graphics Forum* 41.6, pp. 196–211.
- Weiss, Sebastian, Mengyu Chu, Nils Thuerey, and Rudiger Westermann (2021). “Volumetric Isosurface Rendering with Deep Learning-Based Super-Resolution”. In: *IEEE Transactions on Visualization & Computer Graphics* 27.06, pp. 3064–3078.
- Wilby, R. L. and T. M. L. Wigley (1997). “Downscaling general circulation model output: a review of methods and limitations”. In: *Progress in Physical Geography: Earth and Environment* 21.4, pp. 530–548.
- Wilks, D. S. (2016). “The Stippling Shows Statistically Significant Grid Points: How Research Results are Routinely Overstated and Overinterpreted, and What to Do about It”. In: *Bulletin of the American Meteorological Society* 97.12, pp. 2263–2273.
- Winkler, Christina and David Rolnick (2024). *Towards Climate Variable Prediction with Conditioned Spatio-Temporal Normalizing Flows*. arXiv: 2311.06958 [cs.LG].
- Winstral, Adam, Tobias Jonas, and Nora Helbig (2017). “Statistical Downscaling of Gridded Wind Speed Data Using Local Topography”. In: *Journal of Hydrometeorology* 18.2, pp. 335–348.
- Wu, Qi, David Bauer, Yuyang Chen, and Kwan-Liu Ma (2023). *HyperINR: A Fast and Predictive Hypernetwork for Implicit Neural Representations via Knowledge Distillation*. arXiv: 2304.04188 [cs.GR].
- Wu, Zonghan, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu (2021). “A Comprehensive Survey on Graph Neural Networks”. In: *IEEE Transactions on Neural Networks and Learning Systems* 32.1, pp. 4–24.
- Wurster, Skylar W., Hanqi Guo, Han-Wei Shen, Tom Peterka, and Jiayi Xu (2023). “Deep Hierarchical Super Resolution for Scientific Data”. In: *IEEE Transactions on Visualization and Computer Graphics* 29.12, pp. 5483–5495.
- Xie, Yiheng, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar (2022). “Neural Fields in Visual Computing and Beyond”. In: *Computer Graphics Forum* 41.2, pp. 641–676.
- Xie, You, Erik Franz, Mengyu Chu, and Nils Thuerey (2018). “tempoGAN: a temporally coherent, volumetric GAN for super-resolution fluid flow”. In: *ACM Transactions on Graphics* 37.4, pp. 1–15.
- Yang, Ling, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang (2023). “Diffusion Models: A Comprehensive Survey of Methods and Applications”. In: *ACM Computing Surveys* 56.4, pp. 1–39.



- Yu, Alex, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa (2021). “Plenoptrees for real-time rendering of neural radiance fields”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5752–5761.
- Zaheer, Manzil, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola (2017). “Deep Sets”. In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., pp. 3391–3401.
- Zehner, Björn, Norihiro Watanabe, and Olaf Kolditz (2010). “Visualization of gridded scalar data with uncertainty in geosciences”. In: *Computers & Geosciences* 36.10, pp. 1268–1275.
- Zeiler, Matthew D., Dilip Krishnan, Graham W. Taylor, and Rob Fergus (2010). “Deconvolutional networks”. In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2528–2535.
- Zeng, X. and T. S. Durrani (2011). “Estimation of mutual information using copula density function”. In: *Electronics Letters* 47 (8), pp. 493–494.
- Zhang, Chiyuan, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals (2021a). “Understanding deep learning (still) requires rethinking generalization”. In: *Communications of the ACM* 64.3, pp. 107–115.
- Zhang, Shuangyi and Xichen Li (2021). “Future projections of offshore wind energy resources in China using CMIP6 simulations and a deep learning-based downscaling method”. In: *Energy* 217. 119321.
- Zhang, Yu, Peter Tiño, Aleš Leonardis, and Ke Tang (2021b). “A Survey on Neural Network Interpretability”. In: *IEEE Transactions on Emerging Topics in Computational Intelligence* 5.5, pp. 726–742.
- Zhang, Zhiyuan, Kevin T. McDonnell, Erez Zadok, and Klaus Mueller (2015). “Visual Correlation Analysis of Numerical and Categorical Data on the Correlation Map”. In: *IEEE Transactions on Visualization and Computer Graphics* 21.2, pp. 289–303.
- Zhao, Kai, Sheng Di, Xin Lian, Sihuan Li, Dingwen Tao, Julie Bessac, Zizhong Chen, and Franck Cappello (2020). “SDRBench: Scientific Data Reduction Benchmark for Lossy Compressors”. In: *2020 IEEE International Conference on Big Data (Big Data)*, pp. 2716–2724.
- Zhou, Zhenglei, Yule Hou, Qirui Wang, Guangxiang Chen, Jiawei Lu, Yubo Tao, and Hai Lin (2017). “Volume upscaling with convolutional neural networks”. In: *Proceedings of the Computer Graphics International Conference. CGI '17*. Yokohama, Japan: Association for Computing Machinery, pp. 1–6.
- Ziyin, Liu, Tilman Hartwig, and Masahito Ueda (2020). “Neural Networks Fail to Learn Periodic Functions and How to Fix It”. In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., pp. 1583–1594.



# **Publications and Paper Drafts**



# A comparative study of convolutional neural network models for wind field downscaling

Kevin Höhle<sup>1</sup>  | Michael Kern<sup>1</sup>  | Timothy Hewson<sup>2</sup>  |  
Rüdiger Westermann<sup>1</sup> 

<sup>1</sup>TUM Department of Informatics,  
Technical University of Munich,  
Garching, Germany

<sup>2</sup>Forecast Department, European Center  
for Medium-Range Weather Forecasts,  
Reading, UK

## Correspondence

Kevin Höhle, TUM Department of  
Informatics, Technical University of  
Munich, Garching, DE-85748, Germany.  
Email: kevin.hoehlein@tum.de

## Funding information

Deutsche Forschungsgemeinschaft, Grant/  
Award Number: CRC/Transregio 165,  
Waves to Weather, Projects A7 and B5;  
Open access funding enabled and  
organized by Projekt DEAL

## Abstract

We analyze the applicability of convolutional neural network (CNN) architectures for downscaling of short-range forecasts of near-surface winds on extended spatial domains. Short-range wind forecasts (at the 100 m level) from European Centre for Medium Range Weather Forecasts ERA5 reanalysis initial conditions at 31 km horizontal resolution are downscaled to mimic high resolution (HRES) (deterministic) short-range forecasts at 9 km resolution. We evaluate the downscaling quality of four exemplary CNN architectures and compare these against a multilinear regression model. We conduct a qualitative and quantitative comparison of model predictions and examine whether the predictive skill of CNNs can be enhanced by incorporating additional atmospheric variables, such as geopotential height and forecast surface roughness, or static high-resolution fields, like land–sea mask and topography. We further propose DeepRU, a novel U-Net-based CNN architecture, which is able to infer situation-dependent wind structures that cannot be reconstructed by other models. Inferring a target 9 km resolution wind field from the low-resolution input fields over the Alpine area takes less than 10 ms on our graphics processing unit target architecture, which compares favorably to an overhead in simulation time of minutes or hours between low- and high-resolution forecast simulations.

## KEYWORDS

convolutional neural network (CNN), deep learning, statistical downscaling, wind field simulation

## 1 | INTRODUCTION AND CONTRIBUTION

Accurate prediction of near-surface wind fields is a topic of central interest in various fields of science and industry. Severe memory and performance costs of numerical

weather simulations, however, limit the availability of fine-scale (high-resolution) predictions, especially when forecast data are required for extended spatial domains. While running global reanalyses and forecasts with a spatial resolution of around 30 km is computationally affordable (e.g., Hersbach *et al.*, 2020), these models are unable

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2020 The Authors. Meteorological Applications published by John Wiley & Sons Ltd on behalf of the Royal Meteorological Society.

to reproduce wind climatology accurately in regions with complex orography, such as mountain ranges. Since wind speed and direction are determined by localized interactions between airflow and surface topography, with sometimes the added complication of thermal forcing, accurate numerical simulation requires information on significantly finer length scales, particularly in regions that are topographically complex. For instance, (sub-grid-scale) topographic features such as steep slopes, valleys, mountain ridges or cliffs may induce wind shear, turbulence, acceleration and deceleration patterns that cannot be resolved by global models that lack information on these factors. Moreover, meteorologically relevant factors such as the vertical stability, snow cover or the presence of nearby lakes, river beds or sea can strongly influence local wind conditions (e.g., McQueen *et al.*, 1995; Holtslag *et al.*, 2013). In these regions, finer-resolution regional numerical models with grid spacings of the order of kilometers or less need to be applied in order to obtain reliable low-level winds (e.g., Salvador *et al.*, 1999; Mass *et al.*, 2002).

One approach to circumvent costly high-resolution simulations over extended spatial scales is known as downscaling, that is, inferring information on physical quantities at local scale from readily available low-resolution simulation data using suitable refinement processes. Downscaling is a long-standing topic of interest in many scientific disciplines, and in particular in meteorological research there exists a large variety of methods to downscale physical parameters. Such methods can be broadly classified into dynamical and empirical-statistical approaches (e.g., Hewitson and Crane, 1996; Rummukainen, 1997; Wilby and Wigley, 1997).

In dynamical downscaling (e.g., Räisänen *et al.*, 2004; Rummukainen, 2010; Radić and Clarke, 2011; Kotlarski *et al.*, 2014; Xue *et al.*, 2014), high-resolution numerical models are used over limited sub-domains of the area of interest, and numerical model outputs on coarser scales provide boundary conditions for the simulations on a finer scale. While the restricted size of the model domain leads to a significant reduction of computational costs compared to global domain simulations, dynamical downscaling still remains computationally demanding and time-consuming.

Statistical downscaling, on the other hand, aims to avoid simulation at the finer scales by using a coarse-scale simulation (referred to as predictor data) to infer predictions at fine scale (referred to as predictand data). Correlations between the quantities at fine and coarse scales are learned by training statistical models on a set of known predictor–predictand data pairs.

Over time, a large number of empirical-statistical downscaling approaches have been developed, which

apply statistical regression methods for downscaling purposes, such as (generalized) multilinear regression methods (e.g., Chandler, 2005) or quantile mapping approaches (e.g., Wood *et al.*, 2004). With recent developments in data-driven machine learning and computer science, however, more powerful modeling techniques have become available, which may have the potential to outperform previous methods in terms of both accuracy and efficiency. Only a few studies have examined the use of nonlinear regression methods or more recent non-classical machine learning techniques (e.g., Eccel *et al.*, 2007; Gaitan *et al.*, 2014; Vandal *et al.*, 2019). Specifically, the extent to which nonlinear machine learning approaches can provide additional value over classical methods is a question that has not been answered conclusively, as yet.

Deep learning methods are among the most prominent examples of state-of-the-art machine learning techniques (e.g., LeCun *et al.*, 2015; Goodfellow *et al.*, 2016). In particular, convolutional neural networks (CNNs) have found manifold applications in complex image processing and understanding tasks (e.g., Guo *et al.*, 2016; Yang *et al.*, 2019). One of these is single-image super-resolution, that is, the generation of high-resolution images from low-resolution images (e.g., Yang *et al.*, 2019), which, formally, can be thought of as a very similar task to downscaling of climate variables.

CNNs rely on expressing regression models that operate on an extended spatial domain as a set of localized linear models (localized filter kernels), which are applied repeatedly at varying spatial positions across the domain through convolution operations. The restriction of the model parametrization to local filter kernels effectively limits the number of trainable parameters, and thus reduces the tendency of the model to overfit spurious patterns in the data, while increasing model efficiency. While also applicable to irregular graph-based data structures (Kipf and Welling, 2016), for example data defined on irregular grids, CNNs work most effectively with regular-gridded data in multi-dimensional array representations, facilitating an efficient parallel computation of optimization tasks on graphics processing unit (GPU) based computer hardware. Computational efficiency through parallelization is one of the major selling points of CNNs and should be considered as an important aspect during model design and data preparation. Furthermore, more complex mappings can be learned by stacking multiple layers of convolution operations (increasing the depth of the models) and applying these successively to generate more abstract feature representations. Similar to standard artificial neural networks, applying nonlinear activation functions between successive convolution layers can enable the model to

learn nonlinear mappings. Beyond purely sequential feature processing, more elaborate model design patterns, like skip connections between pairs of convolution layers (Srivastava *et al.*, 2015), residual learning (e.g., He *et al.*, 2016) or changes in the spatial resolution of internal feature representations (e.g., Ronneberger *et al.*, 2015), can be leveraged to improve model performance.

CNNs are thus particularly well suited for learning tasks involving spatially distributed data, which are often encountered in meteorology. Although CNN-based model architectures are increasingly adopted also in Earth-system sciences (e.g., Shen, 2018; Reichstein *et al.*, 2019; Vannitsem *et al.*, 2020), their use for downscaling applications has rarely been discussed (e.g., Vandal *et al.*, 2018; Baño-Medina *et al.*, 2019). In particular, earlier studies focused on simple CNN architectures which do not make use of recent model design patterns and thus do not exploit the full potential of state-of-the-art CNN architectures.

## 1.1 | Contribution

In this work, we perform a study of fully-convolutional neural network architectures for statistical downscaling of near-surface wind vector fields. The results are compared to those obtained by a multilinear regression model, with respect to both quality and performance. We train models to predict the most likely outcome of a high-resolution simulation of near-surface winds 100 m above ground, based on low-resolution short-range wind field forecasts as primary predictors. The data are defined on irregular octahedral and triangular reduced Gaussian grids with 9 km and 31 km horizontal resolution, respectively. To enable efficient processing of the data with CNNs and to avoid destroying local detail via interpolation, the data are mapped to regular grids through suitable padding. We view this work as an initial “proof of concept” step, to pave the way to using finer resolutions, for both predictor and predictand. If the predictand scale could reach 1 or 2 km we would envisage a much greater range of practical applications emerging.

We compare the capabilities of different existing models, which reflect varying degrees of model complexity and elaboration. Starting with a multilinear regression model and a light-weight linear convolutional model, we continue the comparison with nonlinear convolutional models of increasing complexity. By incorporating beneficial design patterns identified beforehand, in combination with adaptations in architectural design and training methodology, we propose DeepRU—a U-Net-based CNN model that improves the reconstruction quality of existing architectures.

For all models, we analyze whether incorporating additional climate variables and high-resolution

topography like surface altitude and land–sea mask (LSM) improves the network's inference capabilities. We further train the models on sub-regions of the domain, to avoid learning relationships between low- and high-resolution winds purely based on geographical location, that is, to avoid overfitting to a particular domain. The reconstruction quality of all downscaling models is compared to the high-resolution simulations of real-world weather situations for a topographically complex region in central and southern Europe for the period between March 2016 and September 2019 (Figure 1). Our key finding is that thought-out architecture design and appropriate model tuning enable network-based downscaling methods to generate high-resolution wind fields efficiently in which local- and global-scale structures are reproduced with high fidelity.

To further analyze the usability of network-based downscaling, the relationships between model complexity, network performance and computational requirements such as memory consumption and prediction time are evaluated. We show how the model depth as well as the design patterns used, that is, residual connections across successive convolution layers and U-shaped encoder–decoder architectures, are leveraged to balance between model complexity and prediction quality.

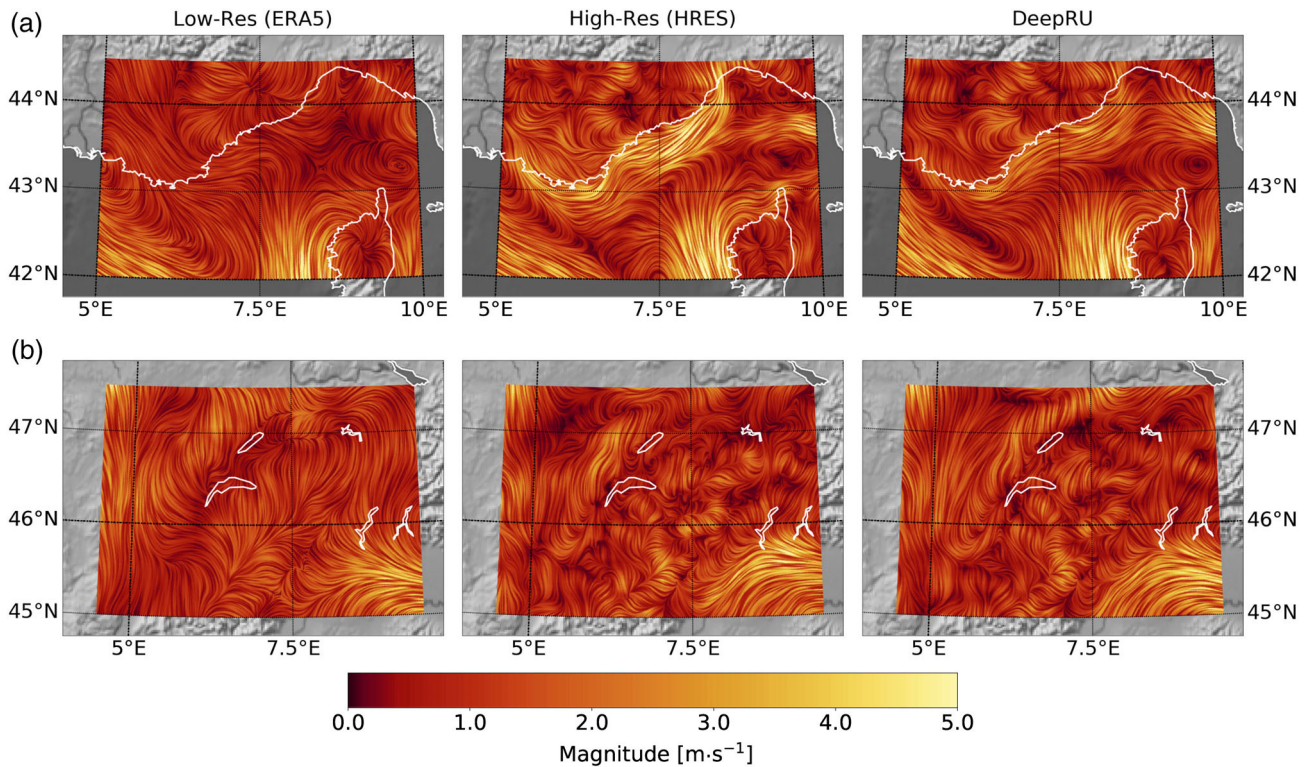
We have made our implementations publicly available at Höhle and Kern (2020).

## 2 | RELATED WORK

### 2.1 | Empirical-statistical downscaling

In describing downscaling options available at the time, Wilby and Wigley (1997) distinguish between regression methods, weather typing approaches and stochastic weather generators. Regression-based methods build upon the construction of parametric models, which are trained in an optimization procedure to establish a transfer function between low-resolution predictor variables and high-resolution predictands. Weather typing approaches, in contrast, rely on finding a suitable match between a set of predictor values and predictor value sets contained in the training data, in order to select out the most appropriate weather pattern analogue (e.g., Zorita and von Storch, 1999). Stochastic weather generators provide a probabilistic approach and are trained to replicate spatio-temporal sample statistics, as implied by the training data (e.g., Wilks, 2010; 2012).

A comprehensive review and comparison of empirical-statistical models for downscaling climate variables has been conducted by Maraun *et al.* (2015; 2019) and Gutiérrez *et al.* (2019), who showed that many of the approaches perform well generally but leave space for improvement. For instance, realistic replication of spatial variability in



**FIGURE 1** Wind field on December 5, 2018, at 1200 UTC. Left: Low-resolution simulation based on ERA5 reanalysis data. Middle: High-resolution simulation based on HRES. Right: Prediction from the low-resolution field, our proposed convolutional neural network DeepRU. Streamlines are color coded with wind magnitude. (a) Coastal region enclosing the French Riviera and Corsica. (b) Highly varying winds over part of the Swiss Alps

the high-resolution predictand variables remains a major challenge for many of the models (Maraun *et al.*, 2019).

Specifically addressing the problem of wind field downscaling and forecasting, Pryor (2005) and Michelangeli *et al.* (2009) proposed distribution-based approaches for wind field inference, and Huang *et al.* (2015) proposed a physical-statistical hybrid method for downscaling.

The question of what methods provide additional value over classical approaches has only been addressed by a number of smaller model comparison studies—with varying results. While Eccel *et al.* (2007), Mao and Monahan (2018) and Vandal *et al.* (2019) found hardly any or no advantage in applying non-classical machine learning methods, Gaitan *et al.* (2014) show non-classical methods outperforming classical ones, with artificial neural networks being a particular method example. More recently, Buzzi *et al.* (2019) used neural networks for nowcasting wind in the Swiss Alps and achieved very skillful models. These apparently contradictory findings raise the question of when, and under what conditions, deep learning methods can be profitably employed for downscaling.

Within meteorology, only a small number of studies have dealt with using CNNs for downscaling applications.

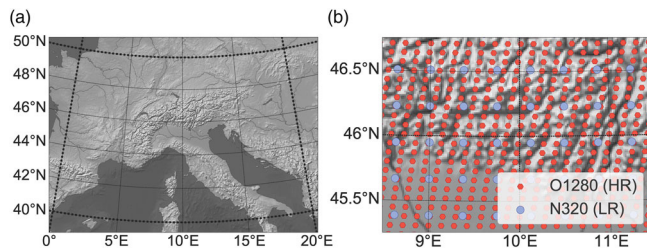
For example, Vandal *et al.* (2018) proposed “DeepSD,” a simple CNN for downscaling precipitation over extended spatial domains, and more recently Baño-Medina *et al.* (2019) studied the performance of a set of CNNs for downscaling temperature and precipitation over Europe. Pan *et al.* (2019) proposed a similar architecture, again with a focus on precipitation.

While the influence of model complexity has been examined by Baño-Medina *et al.* (2019) in terms of model depth, that is, the number of convolution layers, the models in use did not exploit recent design patterns, like skip or residual connections (e.g., Srivastava *et al.*, 2015; He *et al.*, 2016) or the fully-convolutional U-Net-like architecture (Ronneberger *et al.*, 2015), which enable network models to achieve state-of-the-art results in computer vision tasks.

## 2.2 | Upscaling of images and physical fields

Computer vision, being the origin of a large number of technological developments in machine learning, provides a problem setting, which is closely related to downscaling in meteorology and climatology—single-image super-resolution. There, the goal is to identify mappings





**FIGURE 2** (a) Map of the surface topography in Europe representing the data domain. (b) Low-resolution (N320) and high-resolution octahedral Gaussian simulation grid (O1280) used by ERA5 and HRES respectively. Over our domain the high-resolution grid comprises about three times more grid points in longitude and about four times more in latitude

which allow the resolution of single low-resolution input images to be increased, while maintaining visual quality and avoiding pixel artifacts and blurriness. Within this context, the use of deep learning has led to remarkable improvements compared to standard statistical models (e.g., Yang *et al.*, 2019). In particular, CNNs were found to be particularly successful (e.g., Dong *et al.*, 2014; 2016a; 2016b; Sajjadi *et al.*, 2017).

Also in scientific data visualization researchers have begun to explore the capabilities of CNNs for upscaling and reconstruction of 2D/3D steady and time-varying scientific data, including both scalar and vector fields. Zhou *et al.* (2017) presented a CNN-based solution that down-scales a volumetric dataset using three hidden layers designed for feature extraction, nonlinear mapping and reconstruction, respectively. Han *et al.* (2019) took a two-stage approach for vector field reconstruction via deep learning. The first stage initializes a low-resolution vector field based on the input streamline set. The second stage refines the low-resolution vector field to a high-resolution one via a CNN. The use of neural-network-based inference of data samples in the context of in situ visualization was demonstrated by Han and Wang (2020), by letting networks learn to infer missing time steps between 3D simulation results. Guo *et al.* (2020) designed a deep learning framework that produces spatial super-resolution of 3D vector field data. They demonstrate the downsampling of vector field data at simulation time and upsample the reduced data back to the original resolution. Weiss *et al.* (2019) extend image upscaling to geometry images of isosurfaces in 3D scalar fields by including depth and normal information.

### 3 | TRAINING DATA

For model training and evaluation, we use short-range weather forecast data, which include near-surface wind

field simulations at different scales. The data are taken from the European Center for Medium Range Weather Forecasts (ECMWF) Meteorological Archival and Retrieval System (MARS) (Maass, 2019) and cover a spatial domain in central and southern Europe.

#### 3.1 | Domain description

The training domain is restricted to  $40^{\circ}$ – $50^{\circ}$  N and  $0^{\circ}$ – $20^{\circ}$  E (Figure 2a) and is composed of sub-regions with varying orographic properties. Specifically, the domain contains high mountains of the Alps, some smaller mountain ranges in central Europe, flat areas in France, parts of the Mediterranean Sea and southwest-facing coastal regions of the Adriatic, to confront the employed models with challenging scenarios where winds are highly influenced by the topography. In particular, in the Dinaric Alps, situated in the eastern part of the domain, topographically forced gap flows are known to be an important phenomenon (e.g., Lee *et al.*, 2005; Belušić *et al.*, 2013). Significant differences between the low- and high-resolution numerical simulation results are most commonly observed in and around mountain ranges and coast lines, leading to the question of whether downscaling techniques can learn these differences and accurately predict the high-resolution fields from the low-resolution versions.

#### 3.2 | Low- and high-resolution simulations

As “low-resolution” input to our models, we use data derived from the ERA5 reanalysis product suite (Hersbach *et al.*, 2020). ERA5 is the fifth in the series of ECMWF global reanalyses and provides estimates of the 3D global atmospheric state (climate) over time, based on a 4D variational data assimilation of past observations into a recent version of the operational ECMWF numerical forecast model. Output is provided on a regular reduced Gaussian grid with a horizontal resolution of 31 km ( $0.28125^{\circ}$ ). In this study we use hourly forecast fields, from data times of 0600 and 1800 UTC, at time steps of  $T + 1, 2, \dots, 12$  hr. We use these short-range forecasts instead of the true reanalysis fields to avoid systematic small jumps in low-level winds seen in the latter at 0900 and 2100 UTC (documented in Hersbach *et al.*, 2020).

The higher-resolution target dataset was provided by operational short-range forecasts from ECMWF’s high-resolution (HRES) model, also at hourly intervals, initialized twice per day. HRES is a component of the ECMWF

Integrated Forecast System that can provide relatively accurate forecast products into the medium ranges ( $\geq 72$  hr ahead) (ECMWF, 2017). HRES is the highest available resolution model at ECMWF ( $\sim 9$  km) and, as with reanalyses, incorporates observations and information about the Earth-system as a prior for simulation runs. The output is provided on an octahedral reduced Gaussian grid (O1280). Forecast time steps used were  $T + 7, 8, 9, \dots, 18$  hr from the 0000 UTC and 1200 UTC runs. These were chosen as a compromise between being long enough to reduce any contamination from model spin-up and short enough to retain forecast accuracy. The different spatial resolutions of ERA5 and HRES are illustrated in Figure 2b.

Products for HRES on the O1280 grid were first introduced operationally in March 2016 and so are only available from that point onwards. Therefore, we restrict our analysis to time periods between March 2016 and October 2019.

### 3.3 | Predictor and predictand variables

Both the low-resolution predictors and the high-resolution predictands provide two wind variables, which contain spatio-temporal information on the horizontal wind components 100 m above ground. The wind variables are denoted by  $U$  (meridional wind) and  $V$  (zonal wind). At the same locations (i.e., grid points), land surface elevation (altitude, ALT) and a binary LSM are available in low- and high-resolution variants. These are used as static predictors.

From the low-resolution dataset, supplementary predictor variables are obtained and used as dynamical, that is, time-varying, predictors. The additional variables were manually selected according to the following considerations:

- Boundary layer height (BLH) is a model diagnostic that describes the vertical extent of the lowest layer of the atmosphere within which interactions take place between the Earth's surface and the atmosphere (Stull, 2017). Its value typically ranges between about 0.3 and 3 km and it is essentially a metric for low-level stability, with larger values implying deeper layers of instability-driven mixing. Earlier studies (e.g., Holtstag *et al.*, 2013) found that boundary-layer effects can have a significant impact on model performance in numerical temperature and wind predictions. Therefore, BLH may encode information that affects the matching between the low- and high-resolution variants. Also, BLH can provide the model with information about diurnal cycles. For these various reasons there was clear potential for

this standard model output variable to be a useful predictor.

- Forecast surface roughness (FSR) denotes the surface roughness as represented in the forecast and thereby provides information on the frictional retardation of the near-surface airflow. Contributory factors are vegetation types and land cover such as soil or snow. The only dynamic component in the ECMWF modeling architecture is snow cover; other aspects are fixed year-round. We expected a small but direct impact from the snow cover.

- Geopotential height at 500 hPa ( $Z_{500}$ ) designates the elevation of the 500 hPa pressure level above mean sea level, and typically has values around 5,500 m. At this height, the pressure gradients and Coriolis force are typically in balance and winds are roughly parallel to  $Z_{500}$  isolines (see, for example, geostrophic winds in Wallace and Hobbs, 2006). Fields of  $Z_{500}$  very commonly serve as a proxy for forecasters of the general atmospheric flow structure and indeed synoptic pattern. So on the one hand one might expect a link with near-surface winds, but on the other the level is so far from the surface that it is unlikely to be a good predictor of local winds. This variable was partly included as a test of the veracity of our results. Even though on physical grounds we did not, overall, expect strong predictive skill from this variable, our results indicate an apparent influence on the inferred fields.

### 3.4 | Data padding

The training data obtained from MARS is defined on irregular grids where the number of grid nodes per latitude decreases with increasing latitude. As CNNs require the input data as multi-dimensional data arrays, the data need to be resampled on a regular grid structure. Since resampling using interpolation can smooth out and even remove relevant structures, the initial data are copied into rectangular 2D grids and padded appropriately. Therefore, the maximum number of longitudes for the latitude nearest to the equator is computed, and new points are padded to the remaining latitudes for each grid (cf. Figure 3). This approach preserves the spatial adjacency of grid nodes for a large proportion of the nodes, which is important to facilitate proper learning of spatial correlations. The true distance between grid nodes in world space is ignored, however, in the training process. The padded points are marked in a binary mask, which is passed to the objective function during network training to distinguish between valid and padded values in the loss computation.

0	0	0		4	4	4
1	1				5	5
2	2				6	6
3						7

**FIGURE 3** Example of padding and masking used to resample the initial (low-resolution) data from an irregular Gaussian grid to a Cartesian grid. Blue cells indicate the data points of the gridded wind field. The interior of the data domain is shown in light-blue, boundary points are drawn in dark-blue, and their values are represented by numbers. A regular grid is achieved by padding new data points to the grid (light-red cells) while replicating the corresponding boundary values

Padding is chosen based on the fact that CNNs do not take into account only neighborhood relations but also relative changes of neighboring values. Zero padding, which may cause steep gradients between neighboring values, is thus deemed unsuitable and replaced by replication padding using the values of the boundary grid points of the valid domain.

The initial low- and high-resolution data with respectively 1,918 and 20,416 grid points on irregular grids are mapped to regular grids of size  $36 \times 60$  and  $144 \times 180$  in latitude and longitude directions. This results in an increase in the number of grid points by a factor of  $4 \times 3$  between low-resolution and high-resolution grids, which reflects the actual difference in resolution between ERA5 and HRES simulations (see Figure 2).

### 3.5 | Data scaling

Before training, the padded data are standardized by subtracting sample mean and dividing by sample standard deviation. Standardization has proved useful in machine learning for improving the stability and convergence time of nonlinear optimization methods (e.g., Srivastava *et al.*, 2014; Ioffe and Szegedy, 2015). For time-dependent predictors, sample mean and standard deviation were computed node-wise from the snapshot statistics of the respective training datasets. Node-wise scaling is preferred over global domain scaling as spatial inhomogeneities are reduced, which we found to improve the downscaling results in our experiments. For static predictors, mean and standard deviation were computed from domain statistics. For sample standard deviations, we considered the unbiased ensemble estimate. Validation data are transformed accordingly before processing.

Standardization is performed also for the predictand variables. We found this useful due to strong differences in average wind speeds between coast or sea sites and mountain ranges. Further details are discussed in Section 5.3.

## 4 | NETWORK ARCHITECTURES

All of the models we use and compare in this work are constructed as parametric mappings of the form

$$y = f(x|\beta) \quad (1)$$

where  $y$  represents the array of high-resolution predictands,  $x$  denotes the array of predictor variables and  $\beta$  summarizes the model-specific parameters to be optimized during training. We use in particular CNNs, which repeatedly apply convolution kernels of fixed size to gridded input data at varying spatial positions to capture different types of features.

For the downscaling CNNs in our study, we consider input predictor arrays of shape  $c_X^{(LR)} \times s_{lat} \times s_{lon}$  or  $c_X^{(HR)} \times 4s_{lat} \times 3s_{lon}$ , for low-resolution or high-resolution predictors  $x^{(LR)}$  and  $x^{(HR)}$  respectively. Here,  $c_X^{(LR)}$  and  $c_X^{(HR)}$  indicate the number of low- and high-resolution predictor variables per grid node, and  $s_{lat}$  and  $s_{lon}$  denote the number of grid nodes of the low-resolution array grid in the latitude and longitude directions, as specified in Section 3. Note here that the values of  $s_{lat}$  and  $s_{lon}$  may equal the maximum values  $s_{lat} = 36$  and  $s_{lon} = 60$ , corresponding to running the model on the full domain inputs, but may also be set to smaller values as the convolution operations can adapt to varying input sizes by returning outputs of smaller size, accordingly. Choosing smaller values of  $s_{lat}$  and  $s_{lon}$  corresponds to running the models on limited sub-domains, which we use for data augmentation, as discussed in Section 5.2. Predictands  $y$  are assumed to be of shape  $c_Y \times 4s_{lat} \times 3s_{lon}$ , with  $c_Y$  indicating the number of predictand variables.

While  $c_Y = 2$  is fixed for all our models, corresponding to high-resolution wind components  $U$  and  $V$ ,  $c_X^{(LR)}$  and  $c_X^{(HR)}$  vary depending on the predictors supplied to the models, as detailed in Section 6. In particular, some of the models are provided with low-resolution predictors exclusively, whereas other model configurations are informed additionally with high-resolution topography predictors.

The (rectangular) filter kernels are parametrized per convolution layer as arrays of shape  $c_{in} \times c_{out} \times k_{lat} \times k_{lon}$ , with  $c_{in}$  and  $c_{out}$  denoting the numbers of input and output features of the layer, and  $k_{lat}$  and  $k_{lon}$  the spatial extent of the kernel filters in latitude and longitude. Due

to the size of the kernel, the number of elements in convolution output arrays differs from that of the input arrays. To compensate for this, suitable replication padding between successive convolution layers is employed to maintain the spatial shape of feature arrays constant throughout the series of convolutions.

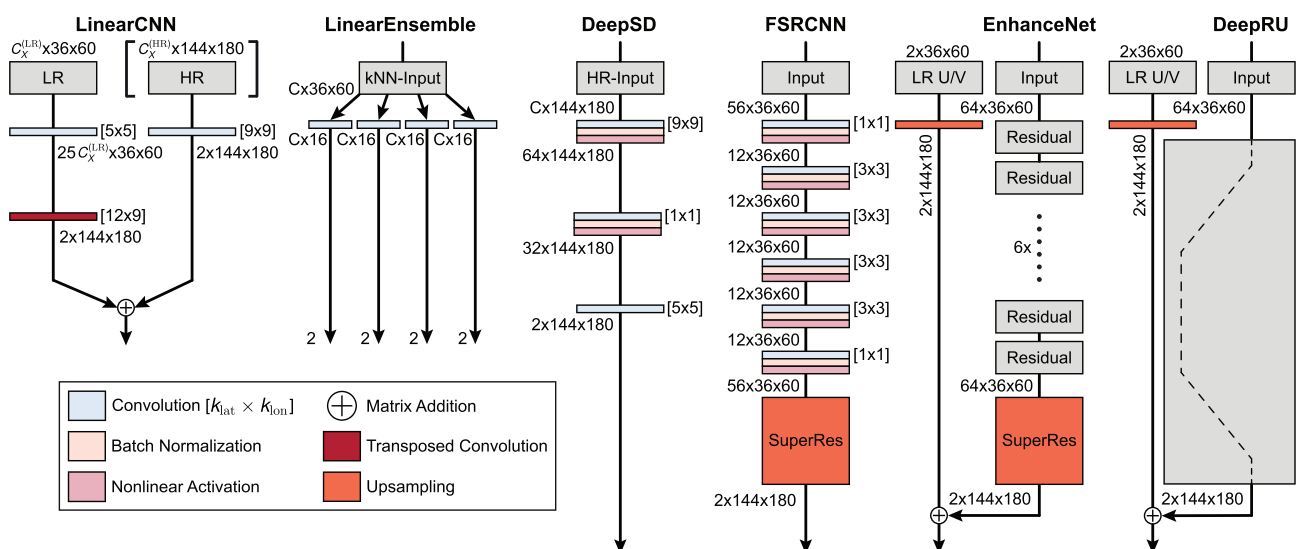
In the following, the details of the different model architectures used in our evaluation are described. A schematic summary of all models is provided in Figure 4.

#### 4.1 | Linear convolutional network model: LinearCNN

A simple way of mapping the low-resolution data to the high-resolution domain while exploiting the parameter sharing capabilities of CNNs is to learn local linear relationships between predictors and predictands via a linear convolutional model, that is, without nonlinear activation functions. For our experiments, we propose LinearCNN, an efficient two-layer CNN which is composed of two branches for processing low-resolution and high-resolution inputs separately.

The low-resolution branch is composed of a single standard convolution layer with kernel size  $(k_{lat}, k_{lon}) = (5, 5)$ , followed by a transposed convolution with kernel size  $(12, 9)$  and stride  $(4, 3)$ . Transpose convolutions can be understood as linear operations which are used to expand the spatial dimension of the input tensors using a linear kernel, which is applied pixel-wise to the inputs of the transpose convolution. A gain in resolution could also be

achieved by applying an interpolation scheme, but to let the network learn the proper transformation automatically the transpose convolution is preferred. Striding thereby refers to skipping pixels in the output domain between accumulating successive kernel evaluations and is applied to regulate the difference in resolution between input and output of the transpose convolution (e.g., Dumoulin and Visin, 2016). The architecture of the low-resolution branch of LinearCNN can be thought of as an encoder–decoder scheme. The standard convolution layer transforms the  $(5 \times 5)$ -pixel input patch into a multi-dimensional  $(1 \times 1)$ -pixel feature representation, whereas the transpose convolution decodes the features and expands the output to match the resolution of the target domain. Thereby, the dimension of the hidden feature representation can be chosen freely. Settings below  $25c_X^{(LR)}$ , with  $c_X^{(LR)}$  denoting the number of low-resolution predictor variables, correspond to a linear reduction of dimensionality before the decoding step. To maximize flexibility of the model, we choose  $25c_X^{(LR)}$  features, thus avoiding implicit constraints on the feature representation. By passing the decoding layer, a  $(1 \times 1)$ -pixel hidden feature vector is transformed into an output tensor of spatial shape  $12 \times 9$  in terms of high-resolution pixels. This output corresponds to a high-resolution estimate of the region, which marks the central  $(3 \times 3)$ -pixel sub-patch of the  $(5 \times 5)$ -pixel low-resolution input. Again, the parametrization of the transpose convolution does not constrain the rank of the linear mapping between predictors and predictands. When both convolution kernels are passed across the domain, the high-resolution estimates of



**FIGURE 4** Schematic of all downscaling models used in this paper. Input sizes of convolutional neural network (CNN) models refer to the final evaluation setting with full domain data. Training was conducted on smaller sub-patches of size  $c_X^{(LR)} \times 24 \times 36$  (low resolution) and  $c_X^{(HR)} \times 96 \times 108$  (high resolution), as detailed in Section 3

neighboring kernel evaluations overlap by 8 and 6 high-resolution pixels in latitude and longitude directions respectively, due to the selected stride values. Effectively, this results in an implicit averaging of predictions from neighboring predictor patches. This is useful to compensate for potential offsets between low-resolution and high-resolution coordinates in latitude and longitude, which may vary across the domain.

On the high-resolution branch, the predictors are fed into a single standard convolution layer with kernel size (9, 9). The outputs of this layer are directly added to those of the low-resolution transpose convolution. Empirically, we found that models with larger kernel sizes did not improve the performance.

## 4.2 | Simple nonlinear CNN: DeepSD

DeepSD is a simple nonlinear CNN architecture which has been proposed by Vandal *et al.* (2018) for downscaling climate change projections over extended spatial domains. The design of DeepSD builds upon the super-resolution CNN (SRCNN) by Dong *et al.* (2014)—one of the first CNN-based architectures for single-image super-resolution. SRCNN is composed of three convolution layers with rectified-linear activation functions in between, which are used to post-process the result of a bicubic interpolation of the low-resolution image data. Although Vandal *et al.* (2018) proposed composing DeepSD of several instances of stacked SRCNNs for better predictions, we found that for the magnification ratio of 3× in longitude and 4× in latitude a single stage of SRCNN already attains results on a par with those achieved by other SRCNN instances.

In the implementation of DeepSD we follow the design proposed by Dong *et al.* (2014) and Vandal *et al.* (2018). The first layer uses a large kernel size of (9, 9) to transform the input predictor fields into an abstract feature space representation with 64 features, followed by a nonlinear activation. The second layer applies a pixel-wise dimensionality reduction with a convolution of kernel size (1, 1) and 32 output features, and a second nonlinear activation. The final layer applies a convolution with kernel size (5, 5) to transform the features to the target resolution.

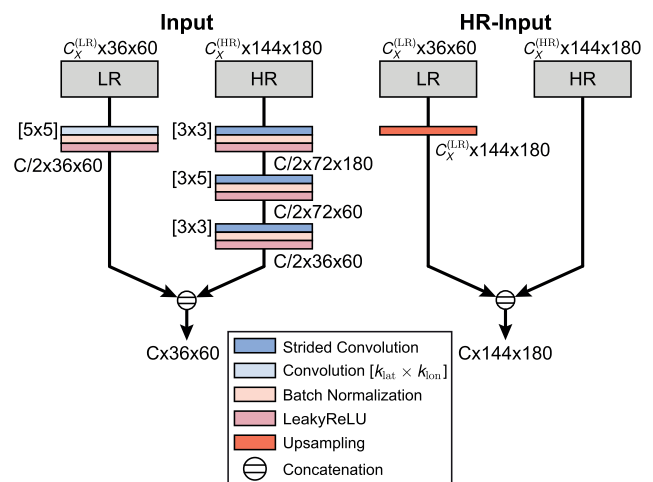
Vandal *et al.* (2018) further proposed to inform the model with high-resolution orography to learn the influence of the topography on the inferred climate variables. Hence, we include the high-resolution static orography predictors during training of all our DeepSD models. To match low-resolution and high-resolution predictors, the low-resolution predictors are first interpolated to high-resolution using a bicubic interpolation, and then

concatenated to the high-resolution predictors to create a combined input array. A schematic of the high-resolution input (HR-input) block is shown in Figure 5.

## 4.3 | Fast nonlinear CNN: FSRCNN

Beyond previously proposed downscaling models, we also took inspiration from ongoing work in computer vision on image super-resolution. With fast super-resolution CNN (FSRCNN) proposed by Dong *et al.* (2016a; 2016b), we include a direct successor of SRCNN in our comparison.

SRCNN has limitations in computational speed as it operates on a high-resolution interpolant of the original low-resolution image. This leads to an increased amount of floating point operations and requires larger convolution kernel sizes with a large number of trainable parameters to capture spatial features in high resolution. FSRCNN circumvents these problems by applying seven convolution layers to the low-resolution inputs directly and upsampling features to the final target resolution only at the very end. FSRCNN replaces convolution layers with large kernels, that is, (9, 9) or (5, 5), in SRCNN with a sequence of convolutions using smaller kernel sizes of (3, 3) and (1, 1). The smaller-sized convolutions, however, speed up the computation time by a factor similar to the magnification ratio in each dimension and are thus beneficial in terms of inference speed. Dong *et al.* (2016a; 2016b) also proposed an hourglass-shaped network architecture, where the highest number of feature channels is used for the outermost layers, while the channel size of the inner layers is reduced.



**FIGURE 5** Input blocks used in fast super-resolution convolutional neural network (FSRCNN), EnhanceNet, DeepRU (left) and DeepSD (right)

design pattern is supposed to avoid costly computations while maintaining prediction quality.

In our experiments, we slightly adapt the architecture of FSRCNN and split the model into three parts: an input processing stage for primary feature extraction, a feature processing stage and a super-resolution stage for successively increasing the resolution until the target resolution is reached.

The design of the input stage varies depending on the predictors in use. When employing low-resolution predictors exclusively, a single convolution layer of kernel size (5, 5) is used to transform the inputs into a set of 56 spatial feature fields, which coincides with the original design by Dong *et al.* (2016a; 2016b). For model configurations that employ both low-resolution and high-resolution predictors, a combined feature representation in the low-resolution spatial domain is created by applying the input block as depicted in Figure 5. We apply two independent convolution chains to low- and high-resolution predictors separately, and restrict the number of feature channels for both chains to  $c^{(LR)} = c^{(HR)} = 28$ . While on the low-resolution branch one single convolution with kernel size (5, 5) is used for feature extraction, the high-resolution branch consists of a sequence of strided convolutions with kernel sizes as indicated in Figure 5. This reduces the resolution of the features successively to low-resolution scale. The resulting features are concatenated with the previously computed low-resolution features and supplied to the feature processing stage.

The feature processing stage again reflects the original design choices by Dong *et al.* (2016a; 2016b). In an hourglass-like architecture, a convolution with a (1, 1) kernel is applied to reduce the number of features from 56 channels to 12, which is then followed by a sequence of four convolution layers with kernel size (3, 3), 12 output feature channels, batch normalization and nonlinear activation. The last convolution layer of the processing stage uses a (1, 1) kernel to return to the 56 feature channels.

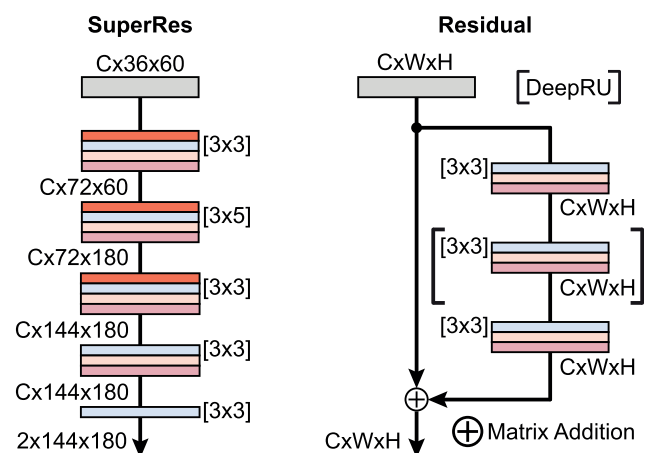
In the original FSRCNN, the resulting features are used as input for a single transpose convolution with a kernel size of (9, 9) for upsampling. In our experiments, however, we found that this very large transpose convolution can lead to slow training progress and can even prevent training from convergence. Furthermore, Odena *et al.* (2016) have shown that transpose convolutions can introduce checkerboard-like artifacts in the final prediction. To circumvent these problems, the extracted features are fed into a super-resolution block, as sketched in Figure 6, after the final batch normalization and nonlinear activation layer of the feature extraction stage. Hence, we avoid transpose convolutions in our work and,

instead, use bilinear upsampling first and apply conventional convolution afterwards to obtain an upsampled result (e.g., Dong *et al.*, 2016a; 2016b). In addition, we replace a single upsampling convolution with scaling factor (4, 3) by a sequence of three upsampling blocks with smaller scaling factors of (2, 1), (1, 3) and (2, 1) to obtain the final image in target resolution. The upsampling blocks are composed of bilinear interpolation, convolution layers with kernel size (3, 3), (3, 5) or (3, 3), batch normalization and a nonlinear activation function. Finally, upsampling is followed by an additional convolution layer with batch normalization and nonlinear activation, and a single output convolution without any activation function. Being a nonlinear model, all but the very last convolution layers in FSRCNN are followed by nonlinear activations, which are realized as parametric rectified linear units (PReLU), as proposed by Dong *et al.* (2016a; 2016b).

Note that, in the original FSRCNN architecture, batch normalization was not used. In our experiments, however, we found it beneficial to regularize the feature representations through batch normalization, since the increased depth of our FSRCNN variant may lead to instabilities in training due to, for example, internal covariate shifts (Ioffe and Szegedy, 2015). By applying batch normalization after each convolution, we could successfully stabilize the training process.

#### 4.4 | Deep nonlinear CNN: EnhanceNet

Previous work in deep learning (e.g., Timofte *et al.*, 2017, and references therein) has shown that increasing network depth can help improve prediction quality and can



**FIGURE 6** Super-resolution block (left) and residual block (right) for fast super-resolution convolutional neural network (FSRCNN), EnhanceNet and DeepRU

lead to network architectures which outperform shallow networks. However, deep networks can easily introduce instabilities in the optimization process, which is typically based on backpropagation of gradients. Specifically, training may become inefficient due to vanishing gradients (Glorot and Bengio, 2010), which originate from the accumulation of small parameter gradients in the chain-rule-based estimation of model parameter updates. The sequential algorithm for gradient estimation causes an exponential decay of parameter updates in early layers of the network, and prevents the parameters from changing significantly during training. While batch normalization may help to stabilize network training, vanishing gradients remain an intrinsic problem of deep neural network architectures.

An effective way to address this problem is the integration of so-called short-cut connections. The purpose of these connections is to pass output features of earlier layers directly to a later stage in the network, effectively skipping parameter dependences of intermediate model parts and circumventing the accumulation of small gradients. Two prominent examples are the skip connections used by Srivastava *et al.* (2015) and Ronneberger *et al.* (2015), as well as residual connections proposed by He *et al.* (2016). With skip connections, the output of a previous layer is concatenated with the result of an intermediate layer. An example is given in Figure 7, which is discussed in more detail in Section 4.5. Residual connections are similar to skip connections but, instead of being concatenated, the features before and after intermediate processing are added. This enables the model to learn mappings that are close to identity more directly.

As a deep CNN architecture with residual connections we selected EnhanceNet (Sajjadi *et al.*, 2017), which was originally proposed for image super-resolution. EnhanceNet is composed of an input stage for raw feature extraction, followed by a stack of 20 convolution layers for feature processing and a super-sampling stage (see Figure 6), similar to that of FSRCNN. Residual learning is incorporated into the architecture in two variants. On the one hand, convolutions for feature processing are subdivided into 10 blocks of two layers each, where each block is wrapped by a residual connection. A schematic representation of one of these residual blocks is shown in Figure 6. On the other hand, bicubic interpolation is used to interpolate the low-resolution wind field inputs to target resolution, yielding a baseline estimate for the high-resolution field, which is added to the model output.

For reasons of efficiency, the convolution layers of EnhanceNet use a kernel size of (3, 3). In our experiments, the number of feature channels is set to 64, which is equivalent to the parameters chosen in the original

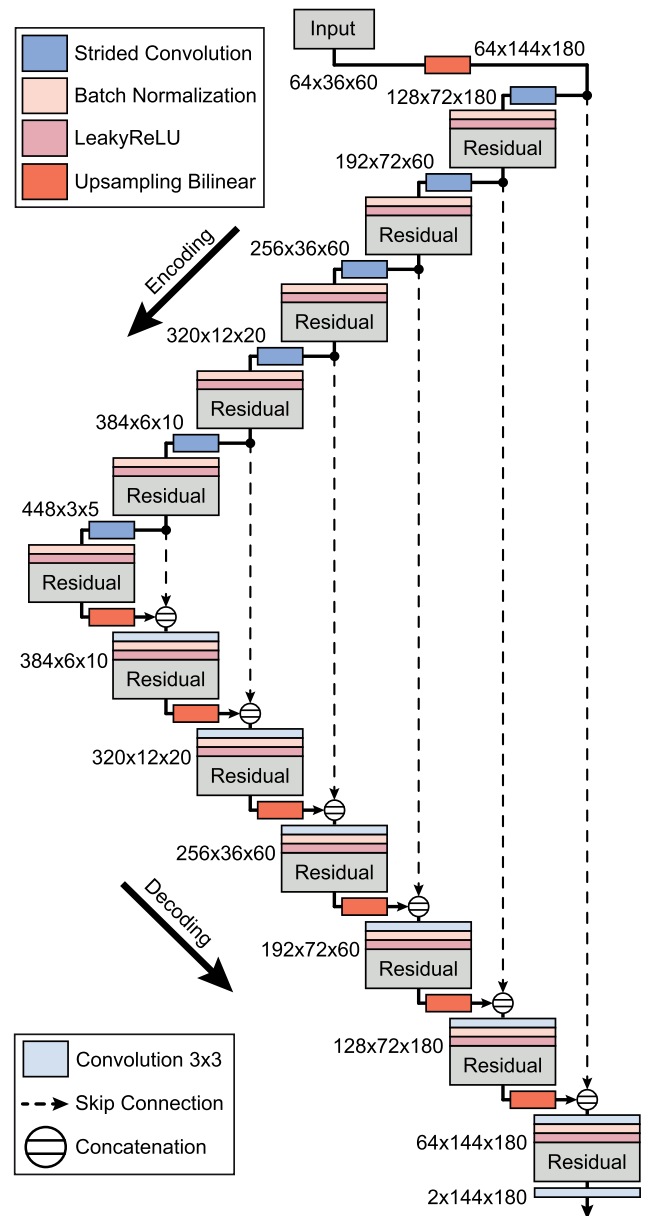


FIGURE 7 Schematic of the DeepRU architecture

paper by Sajjadi *et al.* (2017). The nonlinear activation functions for EnhanceNet are realized through rectified linear units. Similar to LinearCNN and FSRCNN, we consider network variants with varying settings of low-resolution dynamical predictors, as well as with and without high-resolution topography. Depending on the predictor configuration, either a single convolution layer with kernel size (3, 3) or the input block depicted in Figure 5 is used for primary feature extraction. Since the main focus of our study is on pixel-wise accuracy of the downscaling results, we refrain from using perceptual and adversarial losses that are typically used in super-resolution image tasks (Sajjadi *et al.*, 2017) and instead use pixel-wise losses as discussed in Section 5.3.

## 4.5 | DeepRU

Network architectures for super-resolution image generation have been optimized for natural images, which possess properties that are different from those of meteorological simulation results. For instance, natural images typically depict coherent objects, like cars or animals, with well-defined shapes and boundaries. In contrast, meteorological data contain different meteorological variables, which vary smoothly yet less coherently across the domain. Therefore, we expect that more skillful models can be obtained by tailoring model architectures explicitly to meteorological data.

For the present application, we argue that near-surface wind systems result from a complex interplay between a large-scale weather situation, that is, a continental-scale pressure distribution, and boundary-layer processes at finer horizontal scales. The correct treatment of physical processes at varying scales therefore appears as an important aspect in downscaling wind fields on extended spatial domains. This motivates the use of a model architecture that is not restricted to a single resolution scale for feature extraction, but uses different resolution stages to understand the data on multiple scales.

To account for this, we propose to use a U-Net architecture (Ronneberger *et al.*, 2015) with residual connections (He *et al.*, 2016) for downscaling, which we call deep residual U-Net (DeepRU). The U-Net architecture enables an efficient extraction of multi-scale features by design. It consists of two symmetric branches, which are connected by skip connections for simplified information transfer: an encoding branch, on which the data are encoded into an abstract reduced feature representation, and a decoding branch, on which the feature representations are then decoded to reconstruct wind fields at fine-scale target resolution. During the encoding stage, the number of grid points is successively reduced, at the same time increasing the number of feature channels per grid point. In this way, patterns of larger spatial extent can be extracted with small-size convolution kernels. During the decoding stage, the features are super-sampled to a finer scale while reducing the number of feature channels. The skip connections enable a direct information flow between encoding and decoding stages at equal resolutions. By concatenating features from the encoding branch with corresponding features on the decoding branch before further processing, details in the data that could get lost during the compression process can be preserved and localized precisely. In recent work, the U-Net architecture has also been employed for super-resolution tasks (e.g., Hu *et al.*, 2019; Lu and Chen, 2019).

The design of DeepRU is inspired by the results of prior work in image super-resolution (Yang *et al.*, 2019). Starting from the standard U-Net architecture (Ronneberger *et al.*, 2015), we conducted several tests with different U-Net variants to obtain the best model for downscaling. During our studies, we found that making the architecture deeper, that is, increasing the number of resolution levels, led to better training results. The maximum number of levels is limited by the input resolution, since during encoding the input can only be reduced to a tensor of spatial size  $1 \times 1$ . For downscaling, however, we found that a reduced tensor size of at least  $3 \times 5$  in lowest-resolution latent space led to more accurate predictions during patch training and more stable training progress.

While increasing the number of convolution layers with each encoding–decoding stage did not result in better prediction quality, an improvement could be observed when replacing standard convolutions with residual blocks (He *et al.*, 2016). When implementing residual connections across two and even three convolutions at each encoding–decoding stage, we have encountered noticeably improved prediction accuracy.

The reconstruction accuracy could be further improved by interpolating the primary input features after the input block to match the target scale before applying the U-Net model and using skip connections at both high- and low-resolution scale at each encoding–decoding stage. This option gave the most accurate downscaling results between a variety of alternatives that we have tried to process the input. Based on these gained insights, we propose the following architecture for DeepRU.

DeepRU is a six-stage U-Net architecture with both residual and skip connections at every resolution stage (see Figure 7). Similar to FSRCNN and EnhanceNet, we use the input blocks depicted in Figure 5 to transform the inputs to 64 primary low-resolution feature channels. We then super-sample the features using bilinear interpolation, to match the high-resolution grid of size  $144 \times 180$ . The high-resolution features are then fed into the adapted U-Net architecture. We use strided convolutions to downsample the features during encoding and bilinear interpolation with a successive convolution layer to increase the resolution again during decoding.

At each resolution stage, we apply batch normalization and leaky-ReLU activation before passing features to a residual block, as depicted in Figure 6. The residual blocks, originally proposed by He *et al.* (2016), have been slightly modified for the downscaling task. We find that extending the original residual block by another convolution layer before the addition operation leads to an increase in flexibility of the residuals, which translates to a better overall model performance.



We implemented skip connections so that a new combined input can be formed by concatenating the features from the encoding stage to the corresponding super-sampled features in the decoding stage. The combined input is then processed by a single convolution layer with batch normalization and leaky-ReLU activation to further reduce the number of feature channels. The reduced features are finally passed to an additional residual block. After the last residual block at the target resolution in the decoding stage, a convolution layer is added to output a set of features which are added to a bicubic interpolant of the low-resolution winds, resulting in the final wind field prediction.

#### 4.6 | Localized multi-linear regression model: LinearEnsemble

To enable a comparison of the CNN models with more classical approaches, we also consider a model that is based on standard multilinear regression instead of successive convolutions. Due to simplicity and interpretability, multilinear regression models are frequently used in downscaling and post-processing tasks (e.g., Eccel *et al.*, 2007; Fowler *et al.*, 2007; Gaitan *et al.*, 2014).

For multilinear regression models, Equation (1) can be rewritten in simplified form as

$$y = Wx + b \quad (2)$$

where  $W$  is a  $(c_Y d^{(HR)} \times c_X d^{(LR)})$ -shaped matrix of weight parameters capturing linear relationships between flattened predictor vectors  $x \in \mathbb{R}^{c_X d^{(LR)}}$  and flattened predictand vectors  $y \in \mathbb{R}^{c_Y d^{(HR)}}$ , and  $b \in \mathbb{R}^{c_Y d^{(HR)}}$  is a vector of offset parameters. Again,  $c_X$  and  $c_Y$  denote the number of predictor and predictand variables per grid node, and  $d^{(LR)}$  and  $d^{(HR)}$  are the numbers of nodes in the low-resolution and high-resolution domain. Due to the strong increase in the number of trainable parameters with  $\mathcal{O}(d^{(LR)} d^{(HR)})$  for increasing domain size, typical applications of multilinear downscaling models have been focused on local station data or small spatial domains with limited numbers of grid nodes.

For our comparison, we limit the number of trainable parameters to  $\mathcal{O}(k \cdot d^{(HR)})$ , for some user-defined constant  $k \leq d^{(LR)}$ . An ensemble of multilinear regression models is trained, where each model uses the  $k$ -nearest nodes from the low-resolution input to predict the wind components  $U$  and  $V$  at a single grid node of the high-resolution domain. This corresponds to an induced sparsity pattern on  $W$ , which allows at most  $k \cdot c_X \cdot c_Y \cdot d^{(HR)}$  entries of  $W$  to be non-zero.

In contrast to CNNs, we train only two different variants of the model ensemble. In a first step, we use only

the low-resolution wind components  $U$  and  $V$  to inform the model, resulting in a channel number of  $c_X = 2$ . In a second step, we also add the complementary low-resolution dynamic predictors BLH, FSR and Z500, resulting in a total of  $c_X = 5$  predictor channels. Static predictors are not included in the training process, as the resulting contributions in Equation (2) would be indifferent between samples and can thus be incorporated into the offset-vector  $b$  without loss of information. The  $k$ -nearest low-resolution grid nodes are determined based on the standard  $L_1$  distance (in latitude–longitude space) to the target node. We empirically determined that neighborhood sizes beyond  $k = 16$  did not improve the results significantly in our application.

## 5 | TRAINING METHODOLOGY

The time range of about 3 years that is covered by our data is comparatively short, when set in relation to time scales commonly used to define “climatology.” Moreover, temporal correlations between successive samples limit the number of independent examples of weather situations across the domain. This raises the need for efficient data splitting using cross-validation and employing suitable methods to increase the number of training samples. In the following, we shed light on the training methodology and loss functions used in our experiments, and provide details on the optimization process.

### 5.1 | Cross-validation

For all models, including LinearEnsemble, we employ cross-validation with three cycles of model training and validation. In each cycle, we exclude a subset of the data from training. As the data exhibit both short-term temporal correlations on time scales of up to a few days and variations due to seasonality, we decided to pick full consecutive years of data for validation. This minimizes information overlap between training and validation data due to systematic correlations at the beginning and end of validation intervals. Furthermore, it reduces impacts of seasonality on results by averaging model performance over the full seasonal cycle. The excluded validation epochs are chosen pair-wise disjoint and cover the time ranges from June 2016 to May 2017, June 2017 to May 2018 and June 2018 to May 2019, respectively. Each model was trained three times with varying random initializations of the regression parameters in each validation cycle. After convergence, the model with the smallest average validation loss was selected for further evaluation. The performance of the

overall model architecture was then assessed by combining the results of the best models of each of the three validation cycles.

## 5.2 | Patch training

To further increase diversity and variance of training samples, we perform CNN training on sub-patches of the full domain. This procedure limits the dimensionality of the model inputs, thus enforcing models to base their predictions on local information and reducing the chance of overfitting to statistical artifacts in the data. Specifically, fitting of potentially non-physical long-distance correlations is efficiently avoided.

From another perspective, patch training is advantageous due to an improved usage of static predictor information in comparison to full domain training. Static predictors remain invariant when training on the full domain and can thus be ignored by the network or be leveraged to establish a network operation mode of local pattern matching, instead of regression. In such a mode, models might learn to associate the invariant topography with preselected local patterns, learned by heart, instead of using the provided dynamic information to regress on.

Confirming our expectations, we found that patch-trained models yield lower training and validation losses compared to models trained on the entire domain. Experiments show that intermediate patch sizes yield the best training results. For very small patch sizes, we observe a decrease in prediction quality, which may be attributed to a loss of context information due to insufficient data supply. These findings may also be related to the concept of the minimum skillful scale of the underlying low-resolution simulation (Benestad *et al.*, 2008), that is, the smallest spatial domain size, for which the low-resolution data provide a sufficient amount of information for the downscaling model to generate skillful predictions.

In our experiments, low-resolution data were processed in patches of size  $24 \times 36$  and matched with the corresponding high-resolution patches of size  $96 \times 108$ . This was found to yield the most accurate full-grid predictions when applied to validation samples. The sub-patches for training were selected randomly for each predictor–predictand data pair and each training step, so that the induced randomness further decreases the chance of overfitting to the training input. Note, however, that patching was applied exclusively during training of the models. For validation and evaluation of model performance, predictions were computed based on the full domain.

## 5.3 | Loss functions

For measuring error magnitude between predictions and high-resolution targets, we consider different deviation measures, which put weight on distinct aspects of reconstruction accuracy. For optimization purposes we consider spatially averaged deviation scores, whereas for further evaluation we consider both average and local deviations.

Given that  $\vec{t}_i$  and  $\vec{y}_i$  represent the target wind and prediction wind vectors at node  $i$ , with  $1 \leq i \leq d^{(\text{HR})}$  indexing the nodes of the high-resolution grid, we consider in the first place the mean square error (MSE) with

$$\text{MSE}\left(\{\vec{t}_i\}, \{\vec{y}_i\}\right) = \left\langle \left| \vec{t}_i - \vec{y}_i \right|^2 \right\rangle_D$$

Here,  $\{\vec{t}_i\}$  and  $\{\vec{y}_i\}$  denote the sets of predictand and prediction vectors throughout the domain at a particular point in time,  $|\cdot|$  indicates the standard  $L_2$  vector norm and  $\langle \cdot \rangle_D$  indicates an average over the spatial domain. The main advantage of MSE is its invariance with respect to rotations of local vector directions, that is, predictand–prediction pairs which differ only by node-wise rotations of wind directions are assigned an identical deviation score.

However, a potential drawback of MSE is that local deviation scores scale quadratically with wind magnitude (the significance of this will ultimately depend on the application). In particular, small-angle deviations in areas of large wind speeds may contribute largely to the overall deviation score, whereas some strong directional deviations, such as opposite wind directions in areas of low wind speed, are hardly taken into account. This problem becomes particularly serious in certain scenarios where slow but strongly variable winds over mountainous areas are accompanied by increased wind speeds over the sea.

A solution to weaken the square dependence effect is to linearize MSE, resulting in the mean absolute error (MAE). Unfortunately, even MAE does not fully overcome the scaling issue and inherits the problems of MSE. Considering angular deviations instead, for instance as measured by cosine dissimilarity, does not provide an alternative either since angle-based deviation measures do not provide the model with information on differences in wind speed magnitude. A potential alternative would be to use a weighted average of the above-mentioned deviation metrics. However, we refrained from using such metrics as this would require an optimization of additional ad hoc hyper-parameters.

An effective solution is to use the standard MSE and reduce spatial inhomogeneity through node-wise standardization of the target predictands. The models then

learn to mimic a reduced representation of the non-standard predictands, which can easily be converted back to true scale through an easily invertible linear transformation. As stated in Section 3.5, sample mean and standard deviation are computed from the respective training dataset. For validation and evaluation purposes, we convert back to real-scale target predictands and predictions.

## 5.4 | Implementation and optimization

All models have been realized and evaluated in PyTorch (Paszke *et al.*, 2019). Optimization is performed using the ADAM optimizer (Kingma *et al.*, 2014) with an initial learning rate of  $10^{-3}$ , which is reduced by a factor of 0.1 whenever the validation loss in terms of MSE does not decay by more than a fraction of  $10^{-4}$  over a period of five training epochs. The process is continued until a minimum learning rate of  $10^{-6}$  is reached. To guarantee a proper convergence of the models, we train for 150 epochs in each of the three runs per cross-validation cycle, without early stopping. Saturation of training and validation losses was usually achieved after 50–60 epochs, and both training and validation losses showed only minor variations beyond. In particular, we did not observe tendencies of additional overfitting once the models converged.

## 5.5 | Regularization

During training, we employ weight decay with a rate of  $10^{-4}$  (Kingma and Welling, 2013). Additionally, nonlinear convolutional models use batch normalization (Ioffe and Szegedy, 2015) after each convolution operation, which we find to accelerate training convergence significantly. For DeepRU, we apply 2D dropout regularization (Srivastava *et al.*, 2014) with a dropout rate of 0.1 after each residual block; that is, succeeding each residual block a fraction of 0.1 of the respective output feature channels is selected randomly and set to zero. Although earlier studies reported performance issues when using batch normalization and dropout regularization in common (see for example Li *et al.*, 2019), we did not encounter any such negative effects.

## 6 | EVALUATION

To compare the different model architectures with respect to downscaling performance, we consider sample-wise deviations between target predictands and model predictions and investigate the extent to which the

predictions depend on particular predictors. To shed light on the importance of the choice of predictors, the CNN models are trained with four different predictor configurations, including low-resolution wind fields and orography only, providing supplementary high-resolution orography predictors or additional low-resolution dynamic predictors, or the full set of parameters. The predictor settings are detailed in Table 1 and indicated with letters (A) through (D).

Exceptions from this strategy arise for DeepSD and LinearEnsemble. In the case of DeepSD, we refrain from suppressing the use of high-resolution static predictors in order to stay close to the original implementation, which included high-resolution orography predictors by design. Therefore, for DeepSD, we only consider configurations (B) and (D). For LinearEnsemble we exclude static predictors in both low resolution and high resolution, as by design the model does not take advantage from static predictors (see Section 4.6); we therefore consider only configurations (A) and (C).

## 6.1 | Run-time performance and memory requirements

A general overview of the model performance with respect to the number of trainable parameters, memory consumption and computational time for yearly or daily predictions is provided in Table 2. The time measurements were conducted on the NVIDIA TITAN RTX GPU with 24 GB video memory.

At training time, data for all models except for LinearEnsemble were processed in batches of 30 to 200 samples, depending on the model complexity and memory requirements. During training, a significant amount of the memory consumption is caused by optimization computations which are significantly more complex for deeper model architectures. The measured training time spans the full training period until convergence of the respective model, including prediction time as well as time for loss computation and optimization. In the reference trainings, we considered all dynamic and static predictors at low and high resolution.

LinearEnsemble is exceptional here, as memory limitations arise from the need for rapidly accessible storage of the training data rather than from optimization computations. As the nearest-neighbor positions vary irregularly with spatial position, data selection for LinearEnsemble cannot be realized through efficient array-slicing operations, as is the case for CNNs. Nearest-neighbor indexing has to be performed for all linear models separately and was found to be too slow to be conducted at training time. As a result, data for the

**TABLE 1** Predictor configurations for model trainings with varying combinations of low-resolution (LR) and high-resolution (HR) predictors

Configuration	$c_X^{(LR)}$	$c_X^{(HR)}$	LR							HR	
			Wind		Dynamic			Static		Static	
			$U$	$V$	Z500	BLH	FSR	LSM	ALT	LSM	ALT
(A)	4	0	✓	✓	—	—	—	✓	✓	—	—
(B)	4	2	✓	✓	—	—	—	✓	✓	✓	✓
(C)	7	0	✓	✓	✓	✓	✓	✓	✓	—	—
(D)	7	2	✓	✓	✓	✓	✓	✓	✓	✓	✓

Notes:  $c_X^{(LR)}$  and  $c_X^{(HR)}$  denote the total number of low-resolution and high-resolution predictor fields supplied to the models.

Abbreviations: ALT, altitude; BLH, boundary layer height; FSR, forecast surface roughness; LSM, land–sea mask; Z500, geopotential height at 500 hPa .

**TABLE 2** Run-time performance statistics for LinearCNN, DeepSD, FSRCNN, EnhanceNet, DeepRU and LinearEnsemble

Model	TP (k)	MEM (MiB)	TR (hr)	PR (s)	TS (ms)
LinearCNN	68.9	0.3	0.7	5.4	0.6
DeepSD	50.6	0.2	0.9	5.8	0.7
FSRCNN	165.3	0.6	1.9	8.0	0.9
EnhanceNet	942.6	3.6	4.0	15.4	1.8
DeepRU	37,113.9	142.0	13.5	82.5	9.4
LinearEnsemble	3,307.4	12.6	25.8	11.8	1.4

Notes: For each model, the columns describe the total number of trainable parameters (TP) in k (thousands), individual memory consumption to store a model (MEM) in MiB, duration of an entire training procedure for a cross-validation run with 8,760 hourly data (TR), prediction time for all 8,760 inputs (PR) and the prediction time for one single time step (TS) in milliseconds.

Abbreviation: FSRCNN, fast super-resolution convolutional neural network.

LinearEnsemble had to be preselected and stored with high redundancy during training. For the full ensemble of 20,416 linear models with 16 nearest neighbors, the 3 year dataset, including all low-resolution dynamic predictors, required the allocation of roughly 137 GiB of memory, which is not feasible to be stored in RAM on a local machine with typically less than 32 GiB available. Hence, the data were outsourced to a separate HDF5 file and streamed from the hard drive during training, which delivers, by a large margin, the highest training time among all trained models. The training times for the remaining models scaled with model complexity, with the highest being for the most complex model—that is, DeepRU.

In contrast to the above, and for reasons of fair comparison, the computational time for model prediction is computed using a batch size of 220 for all networks; note that timings for loss computations and optimization are not included in the measurements. To compute the total time for model predictions, we make use of Python's timer

module to measure the plain time required by the model to perform downscaling on all input hours for 1 year, in our case 8,760 hr. As timings are often distorted due to hardware communication and process management, we conducted three measurement runs for all models and averaged the results to obtain the final total prediction time. The time for single hour predictions is represented by the ratio between the total computational time and the total number of inputs. In our study, we experienced that the measured time increased with the model complexity, with highest computational costs for DeepRU.

Regarding the number of trainable parameters, the deeper nonlinear solutions EnhanceNet and DeepRU exhibit a significantly higher number of convolutional layers in comparison to the remaining models and thus require more memory to store the trained parameters. Consequently, the general memory consumption scales with the model complexity (see MEM column in Table 2). Despite the higher consumption of memory for nonlinear models, in particular for DeepRU, we found that they achieved the best overall results in our experiments, which is further discussed in the following sections.

## 6.2 | Quantitative analysis

The statistics of spatially averaged MSE on the validation data are illustrated in Figure 8, confirming that both model architecture and predictor selection have a considerable effect on model performance. The weakest model is LinearCNN, showing the largest overall errors and profiting the least from supplementary predictor information. In particular, the use of high-resolution static predictors, which proved to be useful for all the nonlinear models, appears to have no effect on the performance of LinearCNN. The model appears unsuited to extracting useful correlations between low-resolution predictors and high-resolution wind fields. The reason for this is the

restrictive parametrization scheme, which is unsuitable for capturing random offsets and distortions between low- and high-resolution field variables caused by the data padding procedure (see Section 3.4). As the same linear kernels are shared across the entire domain, LinearCNN is forced to yield a most likely estimate, which, however, is found to be inaccurate for most of the grid nodes and poor regarding spatial detail.

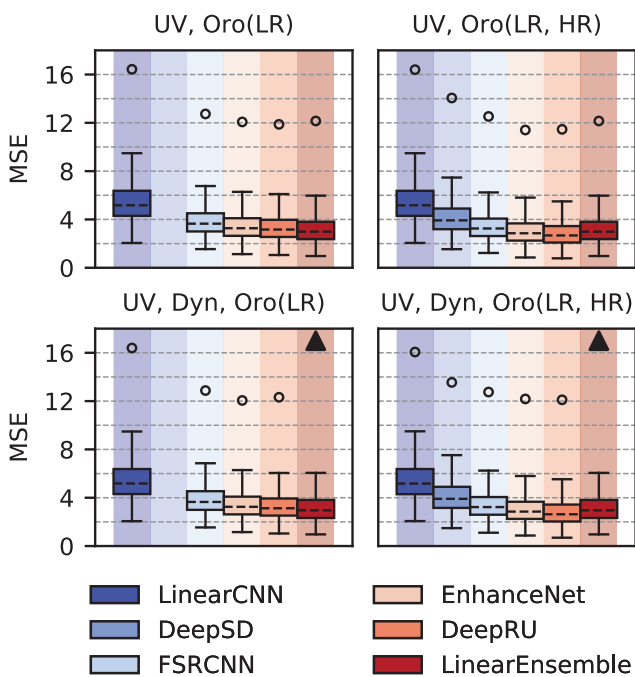
In contrast, LinearEnsemble takes advantage of the local parametrization and achieves considerably better results, comparable with or better than those of the nonlinear models DeepSD and FSRCNN. The gain in performance, however, comes at the expense of a higher tendency of the model to overfit on the training data. In particular, for model variants with a large number of predictors, either due to the use of additional dynamic predictors or larger environment size  $k$ , one observes severe overfitting. This is visible also in Figure 8, as the maximum reconstruction error of LinearEnsemble models with full predictor set (UV, Dyn, Oro(LR) and UV, Dyn, Oro(LR, HR)) exceeds the maximum error of even LinearCNN.  $L_2$  regularization did not improve

generalization performance but increased the reconstruction error on both training and test data. For the nonlinear models, in contrast, overfitting could be minimized through weight decay during optimization—having a similar effect as  $L_2$  regularization—and dropout regularization.

In agreement with earlier studies by Dong *et al.* (2016a; 2016b), FSRCNN achieves smaller down-scaling errors than DeepSD. The quality of the down-scaled wind fields, however, is slightly below that of the LinearEnsemble model for all predictor variants under consideration.

Nevertheless, prediction quality can be further improved by considering more complex models. EnhanceNet, which differs from FSRCNN by an increased number of convolution layers and the use of residual connections in combination with bicubic down-scaling as additive baseline estimate, is the first model to surpass the performance of LinearEnsemble. Notably, EnhanceNet achieves slightly worse results than LinearEnsemble when omitting the high-resolution orography predictors, but catches up after adding the high-resolution predictors. The same is true for DeepRU, which achieves another reduction of MSE.

Comparing DeepRU and LinearEnsemble directly, we find that DeepRU not only reduces the MSE but can also more effectively take advantage of additional predictors. Whereas LinearEnsemble responds with an increased tendency of overfitting, DeepRU achieves a reduction in deviation score when supplied with high-resolution static and low-resolution dynamic predictors. Specifically, model configuration (D) of DeepRU is the most accurate model in our comparison with an average MSE of around  $2.7 \text{ (m}\cdot\text{s}^{-1})^2$ .



**FIGURE 8** Comparison of validation losses for model variants with varying combinations of input predictors wind components (UV), orography variables altitude (ALT) and land–sea mask (LSM) in low and high resolution (Oro, LR/HR) and supplementary dynamic predictors boundary layer height (BLH), forecast surface roughness (FSR) and Z500 (Dyn). Circles indicate maximum deviation observed on the validation set; black triangles signal maximum reconstruction error beyond the scale of the plot

### 6.3 | Spatial distribution of prediction errors

To examine the spatial distribution of reconstruction errors, we consider additional angular and magnitude-specific deviation measures, which we average over the sample distribution instead of the spatial domain. Specifically, we consider cosine dissimilarity (CosDis)

$$\text{CosDis}(\vec{t}_i, \vec{y}_i) = \frac{1}{2} \left( 1 - \langle \cos(\vec{t}_i, \vec{y}_i) \rangle_x \right)$$

for angular deviations between target predictands and predictions. Systematic deviations in wind speed magnitude are measured in terms of the magnitude difference (MD)

$$\text{MD}(\vec{t}_i, \vec{y}_i) = \langle |\vec{t}_i| - |\vec{y}_i| \rangle_X$$

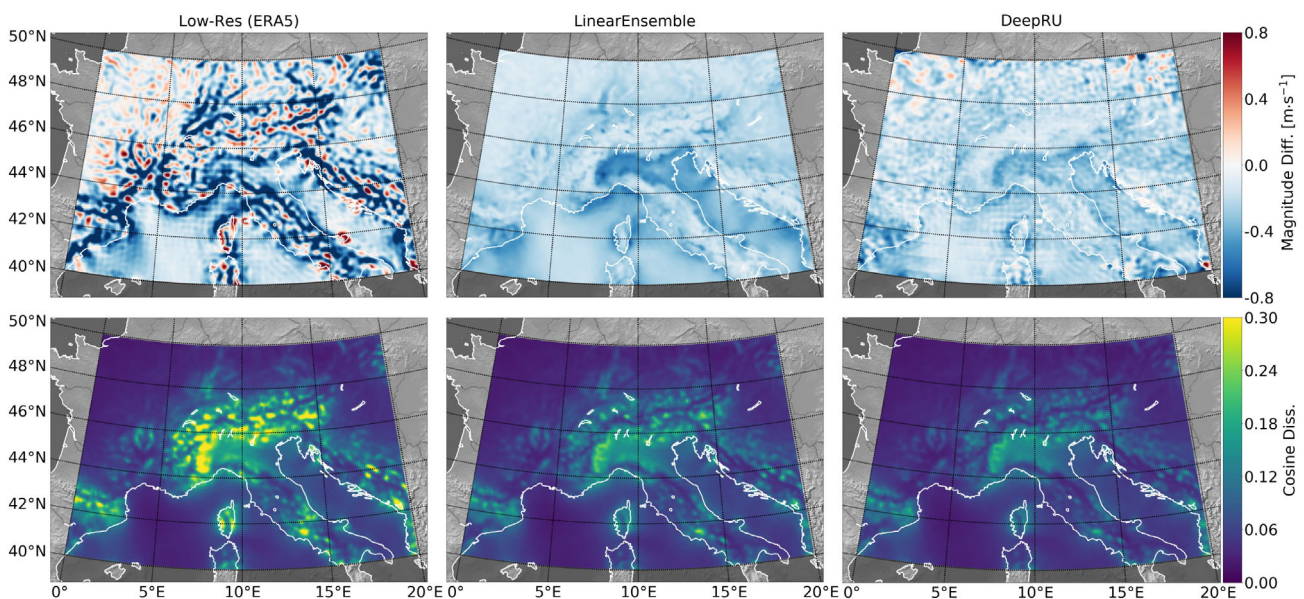
which provides a measure for how much the respective models overestimate or underestimate wind speed magnitudes. In both measures,  $\vec{t}_i$  and  $\vec{y}_i$  represent snapshots of target and prediction wind vectors at node  $i$ , and  $\langle \cdot \rangle_X$  indicates the sample average over the validation sets of the three cross-validation cycles, respectively.

Figure 9 shows the spatial distribution of magnitude difference and cosine dissimilarity for low-resolution forecasts interpolated bilinearly to the high-resolution grid, as well as outputs of the best-performing DeepRU and LinearEnsemble models relative to the high-resolution forecasts. Regarding the low-resolution simulation, velocities in specific regions near the coasts are not well captured and are mainly underestimated with magnitude shifts greater than  $1.0 \text{ m}\cdot\text{s}^{-1}$ . Angular deviations are more pronounced in mountainous areas. Typical values of cosine dissimilarity range between 0.25 and 0.30, which corresponds to average deviation angles of more than  $40^\circ$ . In the northern part of the Mediterranean Sea, the magnitude difference plot for the low-resolution simulation suggests checkerboard-like artifacts, which, however, are most likely due to a mismatch in spatial resolution and grid structure of low-resolution and high-resolution grids, as well as the use of bilinear interpolation for visualization purposes.

In contrast to the low-resolution simulation, LinearEnsemble tends to underestimate, on average, wind magnitudes at all local grid nodes. We expect that this

is mainly caused by an underestimation of extreme winds through LinearEnsemble, which is a common problem of statistical models that are optimized for minimizing MSE losses (e.g., Bishop, 2006). As expected, cosine deviations for LinearEnsemble are much lower than for the low-resolution simulations. However, in areas close to the mountains, LinearEnsemble fails to predict extreme shifts in both magnitude and direction properly, for example due to ridge lines.

DeepRU shows overall better performances with lowest cosine and magnitude differences. Prediction errors exhibit a spatially similar pattern to LinearEnsemble but with generally smaller amplitudes. Furthermore, DeepRU outperforms LinearEnsemble in capturing local variance in wind speed magnitude and directions. As a result, magnitude differences appear less uniform, with overestimation and underestimation in flat areas and near the boundaries, which are caused by imperfect information due to convolution padding. In the Mediterranean Sea, magnitude errors show large-scale wave-like patterns, which especially north of Corsica and east of Sardinia resemble ringing artifacts due to the Gibbs phenomenon (Gibbs, 1898). In turn this relates to the model's spectral representation of topography; issues arise in regions adjacent to where steep slopes meet flat land or sea. In fact the provided topographic height fields contain very similar patterns; sea altitudes look invalid.



**FIGURE 9** Mean magnitude difference (top row) and mean cosine deviations (bottom row) between target high-resolution forecast and low-resolution forecast simulation (left), prediction of LinearEnsemble (middle) and DeepRU (right). The average is taken over all three validation years

## 6.4 | Analysis of feature importance

For the model configuration which was trained on the full set of predictors (D), we also investigate the importance of particular predictors according to the method proposed by Breiman (2001). For this, we perturb the model inputs from the validation dataset by randomly shuffling single predictors, and then measure the change in the prediction error that is caused by the perturbation.

Let  $X = \{x_1, \dots, x_t, \dots, x_T\}$  be the (plain) validation dataset for the respective model run, with data samples  $x_t = (x_t^{(1)} \dots x_t^{(p)} \dots x_t^{(c_X)})$  containing the predictor variables  $x_t^{(p)} \in \mathbb{R}^{s_{\text{lon}} \times s_{\text{lat}}}$  for  $1 \leq p \leq c_X = c_X^{(\text{LR})} + c_X^{(\text{HR})}$ . Then, for every predictor  $p$  we generate a random permutation  $\Pi$  of the sample index set  $\{1, \dots, t, \dots, T\}$ , so that the feature- $p$ -perturbed dataset  $\tilde{X}^{(p)}$  contains samples of the form

$$\tilde{x}_t = (x_t^{(1)} \dots \Phi(x_{\Pi(t)}^{(p)}) \dots x_t^{(c_X)})$$

Here,  $\Phi(\cdot)$  denotes an additional shuffling operation in the spatial domain by decomposing the predictor data into equally sized sub-patches, rearranging the patches randomly and concatenating them again. In our experiments, we fix a patch size of  $6 \times 6$ . Results for different patch sizes are comparable, though. From the perturbed and non-perturbed predictions  $\tilde{y}_t^{(p)}$  and  $y_t$ , the relative change in prediction error is computed as

$$\rho_t^{(p)} = \frac{\langle \text{MSE}(\tilde{y}_t^{(p)}, y_t^*) \rangle_{\Pi, \Phi}}{\text{MSE}(y_t, y_t^*)}$$

where  $y_t^*$  denotes the ground-truth predictand and  $\langle \cdot \rangle_{\Pi, \Phi}$  denotes an average over 10 realizations of  $\Pi$  and  $\Phi$ . Large values of the change ratio  $\rho_t^{(p)}$  indicate a stronger impact of predictor  $p$  on downscaling accuracy, and thus higher importance of the predictor.

Figure 10 illustrates the sample statistics of  $\rho_t^{(p)}$  for the full set of predictors and all downscaling models. In good agreement with expectations, perturbations in the predictor wind components  $U$  and  $V$  have the largest effects on model performance for all architectures in our comparison, indicating that the models in fact use mainly the information on wind speed and direction for downscaling. The effect of perturbations in the wind components strengthens with increasing model complexity. Reasons for this may lie in the nonlinear structure of the more complex models, which could increase the sensitivity of the predictions to perturbations. Also, as shown in Figure 8, more complex models achieve smaller deviation scores when informed with unperturbed data. A similar

increase in prediction error in terms of absolute deviation score therefore yields a larger change ratio for more complex models. This implies that the change ratios  $\rho_t^{(p)}$  should be interpreted in a model-specific context.

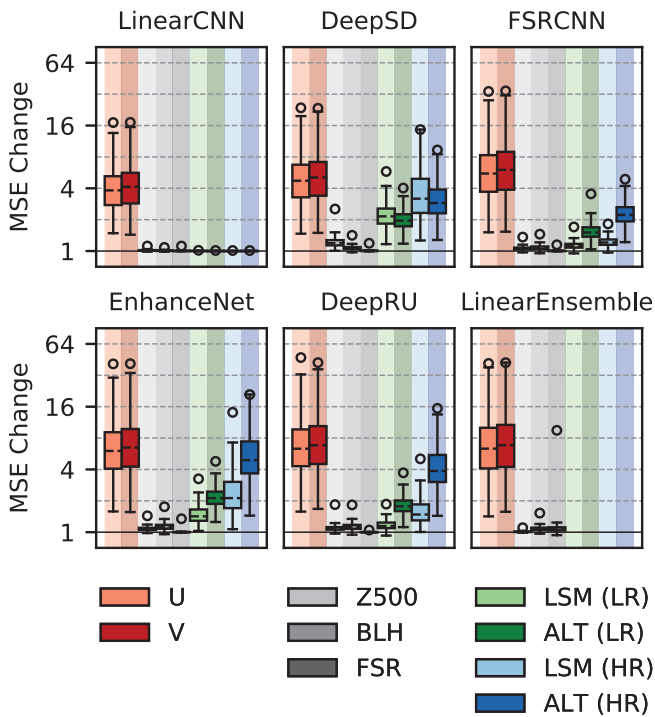
Assessing the relative importance of the remaining predictors, we find that least information is extracted from FSR, as perturbations in this predictor hardly affect any of the models. As FSR is provided on the same coarse grid resolution as the predictor winds, all the information it provides could already be encapsulated in the winds themselves, so that most models learn to ignore the redundant information. Interestingly, LinearEnsemble is the only model that fits correlations between FSR and high-resolution winds, which may be related to the overfitting problem of the model. Perturbations in BLH also have only a slight impact on prediction performance. This was quite a surprising result, given that this quantity varies considerably over time and given that wind speeds at 100 m can be closely related, especially when BLH values are small.

Z500 is leveraged mainly by the less complex models LinearCNN and DeepSD. Z500 provides information on large-scale weather patterns, and there is a known relationship between its gradients and 500 hPa geostrophic winds, which seems to be recognized most prominently by DeepSD. Nevertheless, direct links between Z500 and 100 m winds tend to be relatively weak, which explains its minor impact on the performance of other models.

## 6.5 | Analysis of reconstructed flow patterns

The quantitative analysis provides high-level abstract information on overall downscaling performance of the models, yet it does not convey detailed information on the ability of the models to reproduce the complex flow patterns that we see in the high-resolution simulation. To investigate this aspect in more detail, we select two example cases, which exhibit strong discrepancies between ERA5 and HRES forecasts, and compare the prediction skills of two different models for these examples. For conciseness, we limit the comparison to outputs of the best-performing nonlinear model DeepRU and the localized linear model LinearEnsemble.

To visualize wind vector fields, we use line integral convolution (LIC), introduced by Cabral and Leedom (1993). To generate a LIC visualization, a randomly sampled white-noise intensity image of user-defined resolution is convolved with a 1D smoothing kernel along streamlines in the vector field. Thus, while LIC generates high correlation between the intensities along the streamlines, different streamlines are emphasized by



**FIGURE 10** Relative change in mean square error (MSE) (sample-wise) for different models, when provided with perturbed predictor data. Circles indicate maximum values

low-intensity correlation between them. In addition, color mapping is used to encode additional parameters, such as the local vector field magnitude. In contrast to alternative visualizations, such as vector glyphs or streamline plots, LIC provides a global and dense view of the vector field and can avoid occlusion artifacts due to improper glyph size or sparse sub-domains due to improper streamline seeding. A disadvantage of LIC is that there is ambiguity about which of two opposite directions is represented.

The first example is given for lead time October 17, 2017, at 0900 UTC. This case represents a rather anticyclonic scenario with generally low wind speeds, as denoted by the surface charts in Figure 11. Figure 12 shows LIC visualizations of the underlying wind vector fields, obtained from low- and high-resolution forecast simulations. Color coding reflects total wind speed magnitude. Differences in flow patterns indicate that, especially in mountainous regions like the Alps, Apennines (Italy) and Dinaric Alps (Croatia), the low-resolution simulation fails to capture properly the local variability in wind direction and magnitude, which is present in the HRES simulation.

The results of LinearEnsemble and DeepRU are shown in Figures 12c and 12d, respectively. We have highlighted the most important visual differences between the two predictions with rectangles; specific cases are labeled with the letters A–C. In-detail views of

the streamlines for all highlighted cases are shown in Figures 13a–c, respectively. Quantitative differences to the HRES simulation are measured in terms of wind direction through local cosine dissimilarity and wind speed through local absolute relative error (ARE)

$$\text{ARE}(\vec{t}_i, \vec{y}_i) = \frac{|\vec{t}_i| - |\vec{y}_i|}{|\vec{t}_i|}$$

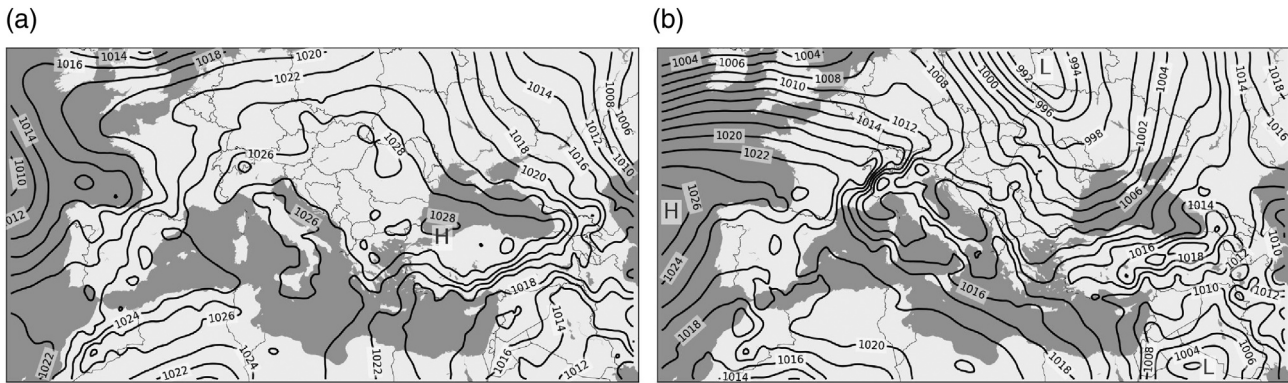
as well as local  $L_2$  deviation which combines both aspects. Results for the outputs of the low-resolution simulation and model predictions are depicted in Figure 14.

Based on the quantitative evaluation of all models in Section 6.2, it can be conjectured that both LinearEnsemble and DeepRU reconstruct meaningful down-scaling results, with DeepRU leading to overall better prediction quality in scenarios of high inhomogeneities. As seen, for example, in the cases A (Adriatic Sea) and B (Austrian Alps) in Figure 12, LinearEnsemble tends not to reconstruct the flow features when there is a pronounced mismatch in flow patterns between the low-resolution and high-resolution forecast simulations. DeepRU, in contrast, uses both local and global information about the orography, and presumably additional parameters, and is able to replicate the HRES wind fields better. In particular, over the Adriatic Sea (A) the winds are mainly northwesterly, tangential to the coast, and higher magnitudes are more pronounced. LinearEnsemble relies solely on local information in the low-resolution fields and is not able to reconstruct the ground truth faithfully.

In areas of complex surface topography, such as near the Austrian Alps (B), variations in wind speed and direction are usually more pronounced, as wind fields are highly influenced by surface interactions. Here, both models learn a reasonable mapping and are able to handle these cases quite well. According to cosine dissimilarity (Figure 14a), DeepRU performs slightly better than LinearEnsemble in terms of direction predictions. Also, DeepRU is able to replicate extreme transitions in magnitude occurring on small spatial scales better, which results in smaller relative and  $L_2$  errors (see Figure 14b,c).

A scenario with generally stronger and rather laminar flow, which exhibits some large differences in wind speed magnitude, is given in (C), where fine-scale mountains slow down winds in eastern France. Since fluctuations in wind direction are small in this area, both models exhibit small errors overall in wind direction. Nonetheless, LinearEnsemble is not really able to account for orography-mediated flow adjustments on small spatial scales, whilst DeepRU can more precisely predict deviations from





**FIGURE 11** Synoptic charts showing mean sea level pressure (hPa) for 0900 UTC October 17, 2017, and 0100 UTC March 19, 2017

laminar flow. This is also clearly demonstrated by the absolute relative errors in Figure 14b.

The second example is for March 19, 2017, 0100 UTC. Figure 15 depicts LIC plots of the wind fields for the simulations and predictions similar to Figure 12. As illustrated in Figure 11b, the weather pattern over our domain is mainly dominated by an Alpine lee cyclone, situated between Corsica and northwest Italy. Comparing low-resolution and high-resolution forecast simulations, major parts of the flow are rather laminar with high wind speeds up to  $18 \text{ m}\cdot\text{s}^{-1}$ . Contrary to the low-resolution simulation, HRES exhibits sharper changes in magnitude over mountain ridges and mountain edges, and exhibits higher distortions in wind directions over the sea. Two particular cases with differences between forecast simulations and model predictions are highlighted in Figure 15 and are labeled A and B.

In case A, the outputs of both the low-resolution simulation and the LinearEnsemble suggest a rather circular vortex pattern with moderate wind speeds over the Ligurian Sea, between the French Riviera and Corsica. The high-resolution simulation, in contrast, displays a distorted, more elongated flow pattern. DeepRU here elongates the flow around the vortex towards northern Italy and additionally enhances the southerly wind near the western coast of Corsica, which, in summary, better mirrors the predictions of HRES. Case B emphasizes the wind field above northern Italy, where the flow is more inhomogeneous since regions of high wind speeds are interleaved with topographically triggered vortex structures. Here, LinearEnsemble fails to predict as well as DeepRU the sharp magnitude changes seen in HRES along the mountain ridge of the Apennines and near to the three marked lakes.

## 7 | APPLICATIONS IN FORECASTING

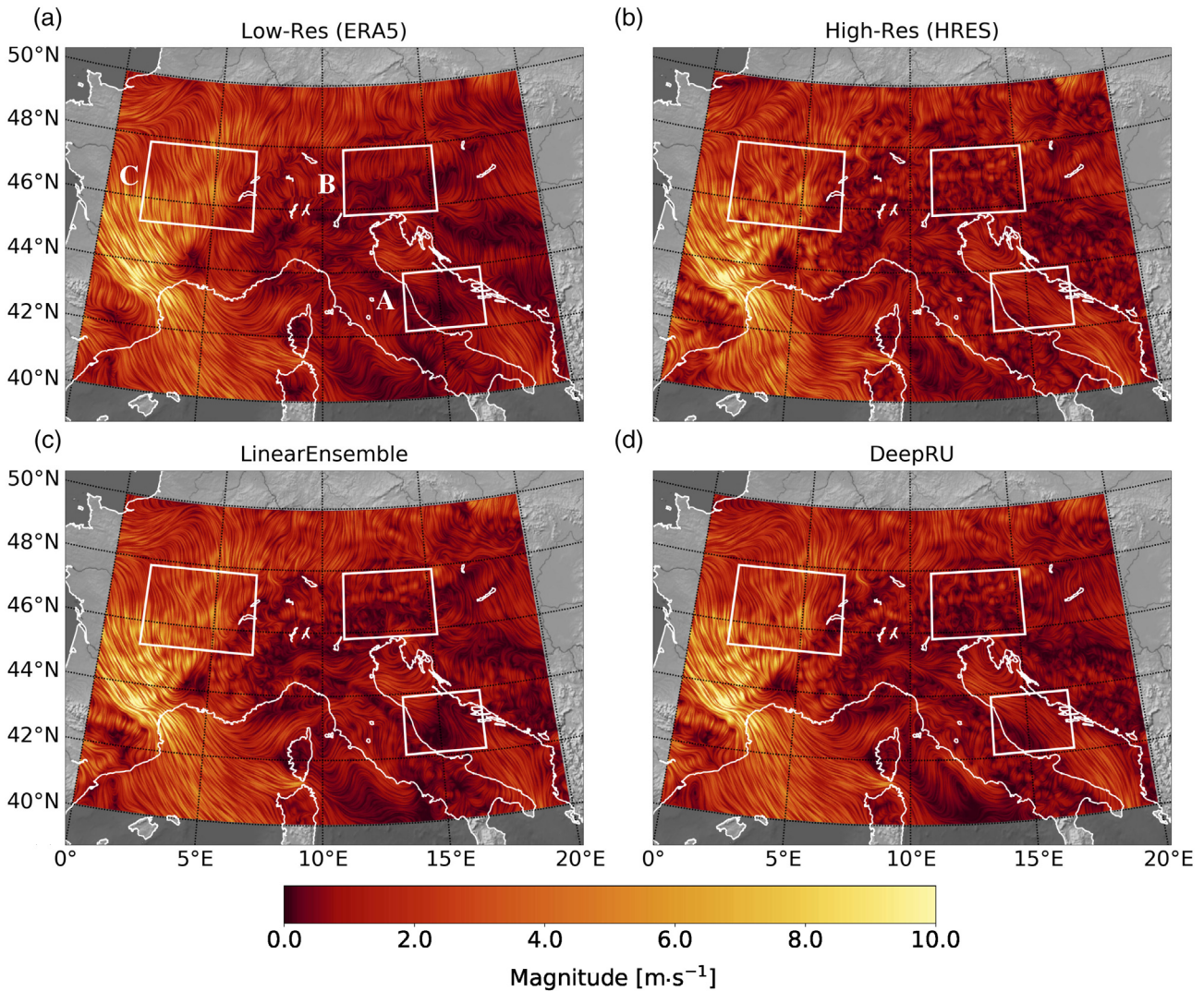
As our study sheds lights on the conceptual use of CNNs for downscaling of wind fields, it was not intended that the CNN

architectures proposed here would be used directly in operational forecasting. Indeed the spatial resolutions of our predictor and predictand datasets are not competitive relative to current operational configurations. In Europe, for example, operations nowadays use global models with spatial resolution  $\sim 10\text{--}20 \text{ km}$ , and for shorter leads up to, for example, day 3 use limited area models (LAMs) with resolution  $\sim 1\text{--}4 \text{ km}$ . Nonetheless, our results are sufficiently promising to provide a blueprint for future operational systems that successfully serve the needs of automated and forecaster-based predictions. So how might this work?

To realize this, we envisage first stepping down in scale to use predictor and predictand resolutions of  $\sim 5\text{--}10 \text{ km}$  and  $\sim 1\text{--}2 \text{ km}$  respectively. Regarding the predictors, international modeling centers such as ECMWF will upgrade their global ensembles to this resolution range in the next few years. Regarding the predictand, this is needed only for training and so need not be run operationally in real-time. So one could use, for example, a 1- or 2-year global reanalysis-style dataset, similar to that described by Dueben *et al.* (2020) but created with repeated observation-based initializations. This would deliver worldwide downscaling options, for any region the user selected. An alternative would be to use real-time LAM output for any region for which that was available.

Real-time CNN predictions realized via this route could be used in different ways. Where no LAM coverage exists, predictions could be delivered for short- and medium-range lead times. Where LAMs are available use would focus on the medium range, and if the same LAM were used for training this could nicely provide continuity across the LAM-global temporal boundary.

Another difficulty to address, at least in ECMWF output, is the apparently poor representation of  $10 \text{ m}$  winds over mountains—the reason we use  $100 \text{ m}$  winds in this study. This may improve in future, but if not the CNN approach is such that one could use  $100 \text{ m}$  winds as



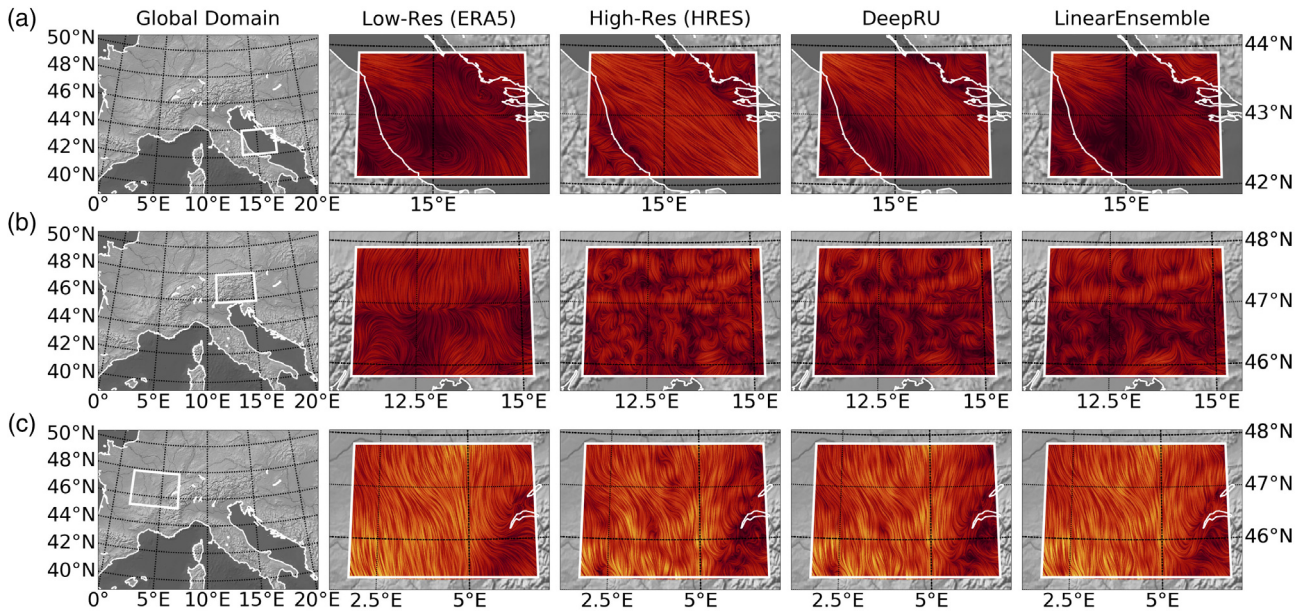
**FIGURE 12** Wind fields over Europe, as obtained from low-resolution and high-resolution short-range forecast simulations and model predictions for October 17, 2017, 0900 UTC. The top figures show the flow field for (a) the low-resolution and (b) the high-resolution simulation and highlight differences between the two predictions, (c) depicts the predictions of the localized linear model, LinearEnsemble, whilst (d) represents the wind flow predicted by DeepRU. These line integral convolution (LIC) images show the current motion of particle flow produced from the wind field products. The LIC field is colored according to local wind magnitude in  $\text{m}\cdot\text{s}^{-1}$ . Regions with strong differences between predictions are marked by rectangles A, B and C. Errors of LinearEnsemble are  $\text{MSE} = 1.33 (\text{m}\cdot\text{s}^{-1})^2$ ,  $\text{CosDis} = 0.15$ , and of DeepRU are  $\text{MSE} = 0.88 (\text{m}\cdot\text{s}^{-1})^2$ ,  $\text{CosDis} = 0.097$

predictor and 10 m winds as target, if the latter were better represented at 1–2 km resolution—which there is some evidence for, at least for LAMs (Hewson, 2019, Figure 7).

There are numerous application areas that need better, locally refined wind speed predictions. Renewable energy is clearly one. Others include local pollutant dispersal, coastal and open water shipping, rig operations, leisure activities such as sailing, aviation, the construction industry and warnings in general. Applications for which mean speed predictions are important across the full speed range, such as renewables, will potentially benefit most. For applications with a focus on extremes

more investigation will be needed; the training period may not be sufficient. Predictions may be systematically too weak, or become unstable. For very hazardous but less rare gap-flow phenomena we can be more optimistic, however. Here we expect the CNN predictions to deliver major benefits for users compared to raw model output.

Society requires not only predictions of mean wind speeds, but also forecasts of gusts, particularly extreme gusts, because of the dangers posed to life and infrastructure. Gusts have not been directly explored in this study. One might be able to convert mean speeds into reasonable gust forecasts using empirically defined gust-to-



**FIGURE 13** Example flow patterns on 0900 UTC October 17, 2017, as obtained from low-resolution and high-resolution short-range forecast simulations, and predictions of LinearEnsemble and DeepRU, visualized as line integral convolution (LIC) plots. The location of the regions within the data domain is marked on a global map on the left for each case. (a) The flow field outputs in a region between Italy and Croatia over the Adriatic Sea, (b) the flow over the Austrian Alps with low-speed winds and large directional variations, and (c) the wind flow of areas near central France

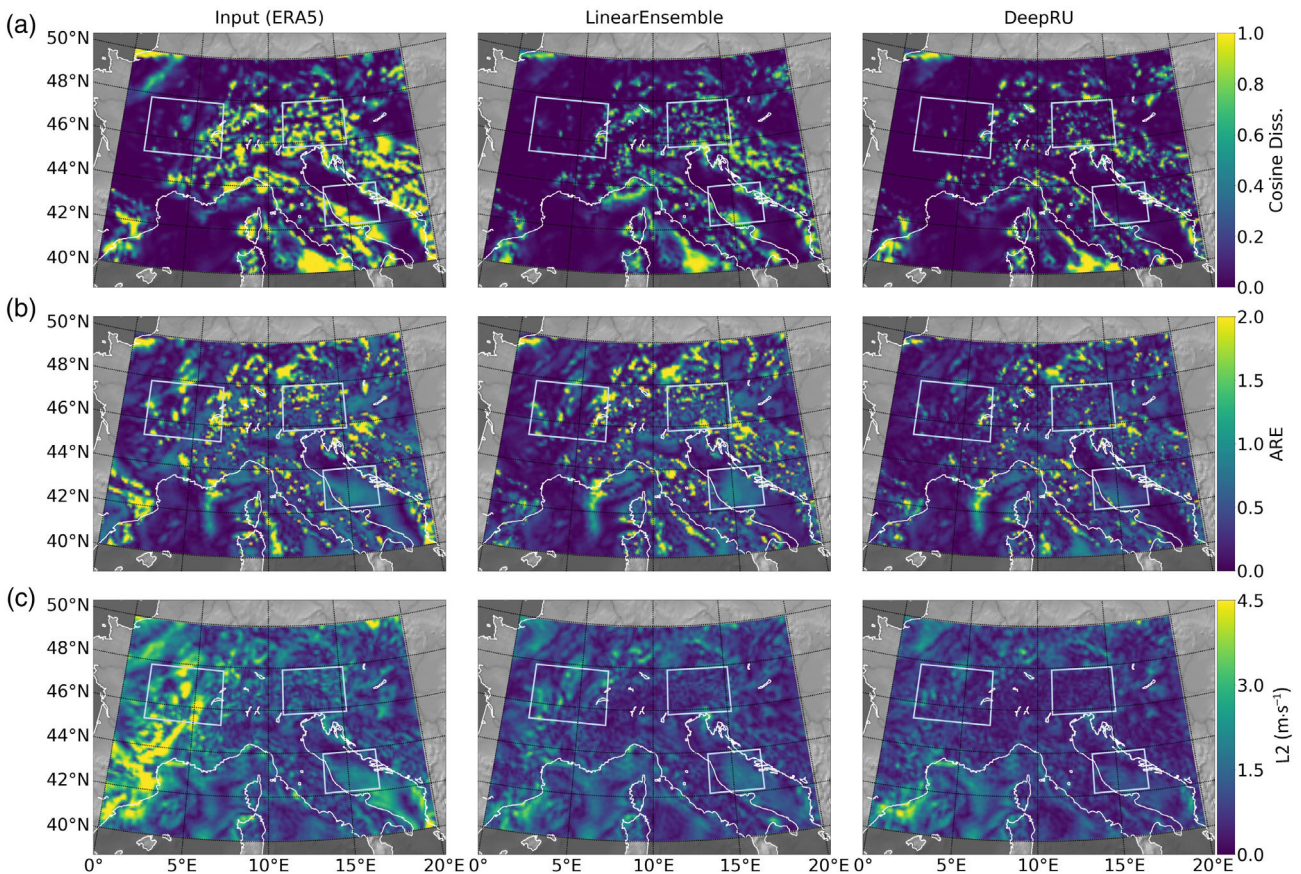
mean relationships (see for example Ashcroft, 1994), developed for different land surface types, although for cyclone-related gusts, which tend to be the major wind-related hazard in the vicinity of storm tracks (e.g., in northern and western Europe), caution is needed. Low-level stability, and destabilization mechanisms, as outlined in Hewson and Neu (2015), are of paramount importance for determining the strengths of phenomena such as the cold jet, warm jet and sting jet (see also Browning, 2004). In that context it is curious that the BLH parameter used in our study, which relates directly to stability, did not add much predictive value for the CNNs. Our use of a region that is relatively remote from storm tracks may explain this.

It is important to reiterate that airflow, and thus winds, can be very scale dependent. On meter scales speeds around city buildings vary dramatically, whilst on a lake the behavior of a yacht can be influenced by clumps of bushes nearby. Indeed scale dependence is more acute than it is for other parameters, such as rainfall and temperature. Thus model resolution increases bring with them more and more application areas for forecasts, particularly for regions that are topographically and/or physically complex. In turn this brings sustainability, whereby the method outlined in this paper, and variants of it, can find utility for the foreseeable future as numerical weather prediction models continue to evolve.

## 8 | DISCUSSION AND OUTLOOK

Driven by fast developments in computer science, applications of data-driven machine learning methods in a meteorological context are attracting increasing interest. In the current study, we have investigated the use of CNNs for learning-based downscaling of wind fields. However, the sheer volume of potential design choices which could impact model performance tends to preclude a complete understanding of reasons for the performance of particular model architectures. Therefore, we have selected a set of design patterns based on the experience of what model types have worked well on similar tasks. Our proposed final architecture marks the preliminary endpoint of an iterative process of repeated model training, evaluation and architectural refinement, and was found to achieve the most promising performance in our application. It is clear, on the other hand, that even with only a limited range of design patterns the computational cost of training a large number of CNNs rules out a complete and direct comparison of model architectures. Thus, given the ever-increasing number of studies in data science and machine learning, it can be expected that alternative architectures can be found that achieve similar or superior downscaling accuracy, ideally with reduced model complexity.

Our study has shown that the prediction accuracy of a linear ensemble model is higher than what can be



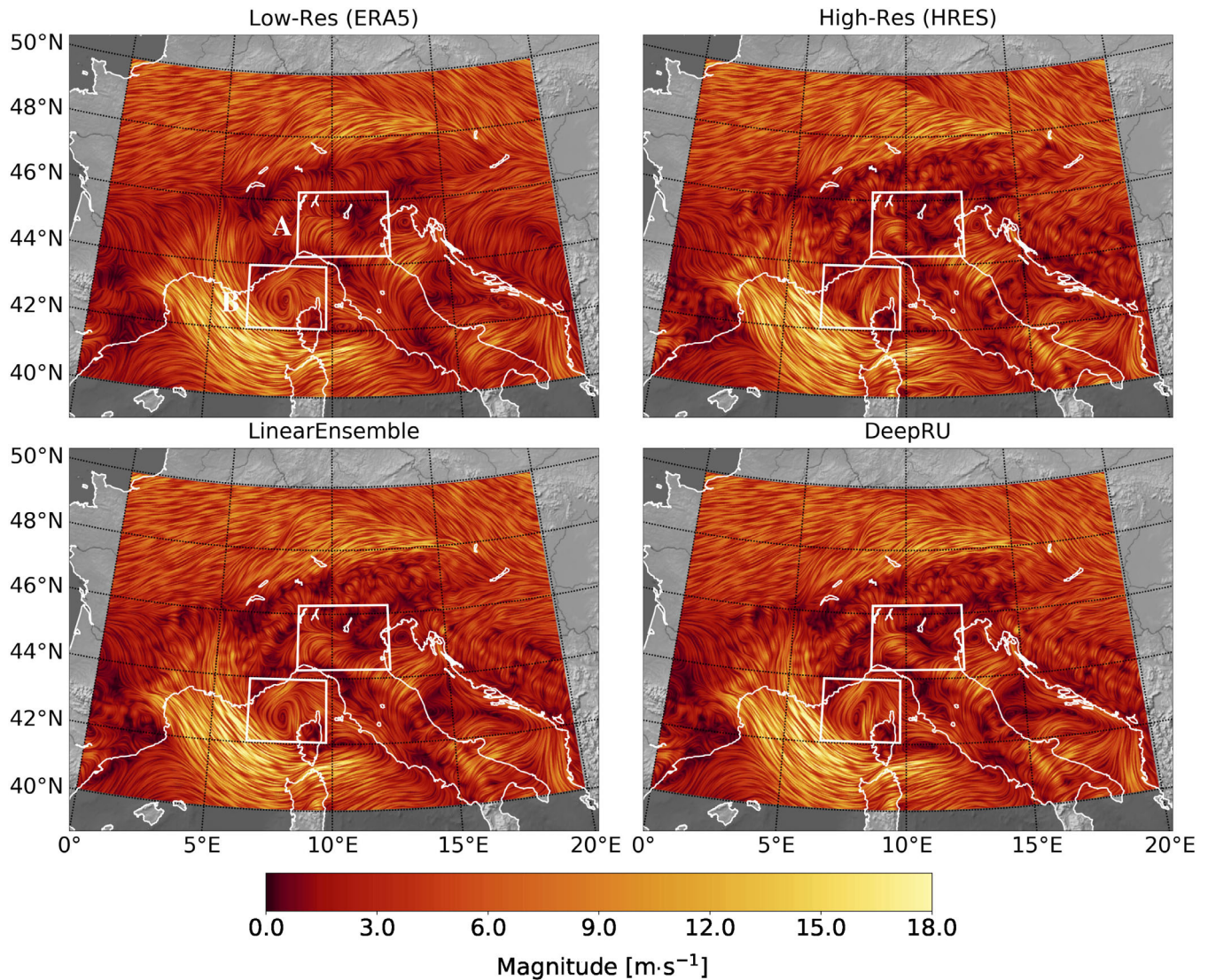
**FIGURE 14** Visualization of spatial deviations of the low-resolution simulation, LinearEnsemble, and DeepRU wind predictions compared with the output of the high-resolution simulation shown in Figure 12. Here, the deviations are (a) cosine dissimilarity, (b) absolute relative error and (c)  $L_2$  norm

achieved with shallow nonlinear CNN architectures. In particular, for simplistic nonlinear models with only a few convolution layers, it seems that the nonlinearity even hinders performance. We attribute this to distortion of the wind field information by the nonlinear activations on its way through the network, which prevents the model from benefitting from simple mapping schemes, such as for example interpolation kernels. Thus the use of overly simplistic and shallow nonlinear models may be one reason why earlier studies found hardly any additional value in applying neural-network-based machine learning methods (e.g., Eccel *et al.*, 2007; Vandal *et al.*, 2019).

Deeper nonlinear CNNs, on the other hand, are able to compete with the prediction quality of the linear ensemble model and even show superior results when incorporating an increasing number of predictors and high-resolution topographic information. In particular, we identified EnhanceNet, previously proposed for single-image super-resolution, as a deep CNN that achieves this. As seen in Figure 9, EnhanceNet exhibits a clear increase in prediction quality with additional

parameters while LinearEnsemble is unable to make use of this information and tends to overfit on the training data, finally with an overall slightly inferior prediction performance. EnhanceNet thus appears more flexible and minimizes the need for incorporating prior knowledge and manual selection of suitable predictor variables. Instead one can select *candidate* predictor variables and refine those later based on an analysis such as is shown in Figure 10.

With DeepRU, we propose a novel deep residual U-Net architecture, which outperforms both the linear model and EnhanceNet in terms of reconstruction accuracy. The major advantage of DeepRU lies in its ability to process features at different spatial scales. This is particularly useful for downscaling of wind fields, where local wind systems have to be consistent with large-scale flow patterns. Although we still observe some deviations between high-resolution model predictions and native high-resolution forecast simulations, we are confident that CNNs can provide promising downscaling results and add more value to downscaling than linear models at a reasonable computational cost.



**FIGURE 15** Wind fields over Europe, as obtained from low-resolution and high-resolution short-range forecast simulations and model predictions for March 19, 2017, 0100 UTC, similar to Figure 12. Color coding indicates the local wind velocity

Our study suggests that deep CNN approaches are particularly effective for downscaling with high magnification ratios on large spatial domains. In this setting, the use of classical models becomes computationally inefficient, and linear link functions between predictor variables and predictands become insufficient to account for non-trivial variability in the local flow, for example due to pronounced flow distortion around obstacles. We found that deep CNNs are better suited to replicating this variance, especially in mountainous areas or over the sea near to coasts, and expect that the same holds true also at finer spatial scales.

Important aspects that need to be further examined in the future are model verification and generalizability. In our study, we have trained CNNs on downscaling tasks using wind fields over a particular spatial domain, that is, the predictive skills of the resulting networks are

specific to this concrete setting. Low-level winds were selected as a variable assumed to be particularly appropriate for this type of methodology, because their structure in the vicinity of coasts and complex topography is very much determined by those physical features together with the broader scale flow patterns delivered by ERA5. Downscaling of some other climate variables will require different modeling approaches, because physically the problem can be very different. Each variable, and suitability thereto, must be considered individually. For accumulated precipitation, for example, the range of possible outcomes at high resolution, for a given low-resolution representation, might be limited for one type of precipitation (e.g., orographically enhanced) but considerable for another (e.g., convective) and therefore precipitation downscaling lends itself to a completely different and innately probabilistic approach

(e.g., Hewson and Pilloso, 2020). Even so, there will be climate variables other than low-level winds for which, given suitable predictor variables, the model architectures proposed in this paper can serve as flexible feature extractors and yield skillful downscaling results.

An alternative notion of generalizability refers to applying readily trained network models to predictor data which formally depict the same climate variables as the data used during training but deviate from the training set with respect to certain properties, such as geographic reference domain or applied simulation model. In such cases, we expect a poorer performance. Specifically, we have applied our networks to wind field data over a region in North America, covering parts of the west coast of the northern United States and Canada, as well as the Rocky Mountains and parts of the interior plains. Although the data were generated with the same simulation procedures as for the original training data over Europe, we observed a drop in performance of about 70% in MAE and 90% in cosine dissimilarity. When training directly on the data from North America, however, similar reconstruction quality as reported in this paper could be achieved also on the other domain. Our findings indicate that the generalizability of our CNN-based downscaling approaches should be assessed carefully. One possible workaround for future applications could be accepting the lack of portability of the models and training many different networks, each of which is specialized and validated within its particular scope. Additionally, though, it seems promising to examine further how networks can learn to model concurrently the relationships occurring between meteorological variables over a variety of different domains and data sources. Our results suggest that the apparent lack of generalizability is not due to insufficient flexibility of the models, which is in line with earlier work on generalizability of deep learning models (e.g., Zhang *et al.*, 2016). Specifically, our models can be taught to achieve high reconstruction scores over both domains, North America and Europe, when data from both regions are seen during training. The main focus should thus be on increasing the data efficiency of the models to facilitate generalization, for example by incorporating prior physical knowledge concerning recurring atmospheric processes into model design or training regularization.

What we have neglected so far in this paper is the temporal dimension of the data, which can probably be used to understand the model predictions better and further to improve their performance. In preliminary research, we have assessed how temporal correlations are reflected in the model's predictions and found that temporal correlation between model predictions and target wind fields yields information complementary to that

conveyed by MSE measurements. In particular, we found that, according to temporal correlation, our models exhibit highest uncertainty over mountains while MSE deviation is largest over the sea. In the present experimental setting, however, the role of the temporal dimension is more similar to that of a sample index, instead of a temporal coordinate, which parametrizes the time evolution of physical processes. Specifically, training of our proposed models has focused on purely spatial correlations on a single-time-step level and temporal coherence between predictions has not been enforced. Consequently, it would be interesting in the future to design neural network models which consider the temporal correlation of wind vector fields across multiple time steps and analyze the models in terms of predictability. This would require the definition of a suitable and interpretative temporal correlation measure for vector-valued inputs which, in our opinion, appears to be a non-trivial task. For instance, we have found that the temporal average of the scalar product between mean-centered predictor and predictand wind vectors, as a standard correlation measure, strongly resembles cosine deviations, which we attribute to the strong relationship between the scalar product and the definition of the cosine deviation. Another option would be to examine local coordinate-wise temporal correlations between the scalar wind components, which, however, would require the selection of reference directions for computing these correlations. The best candidates for these may not be known beforehand and presumably depend on the local surface topography. Beyond coordinate-wise correlations, full correlation matrices might be necessary to examine existing cross-correlations between wind components in a complete and principled way.

Furthermore, including temporal information also into the process of model building (e.g., using long short-term memory (Hochreiter and Schmidhuber, 1997), gated recurrent units (Cho *et al.*, 2014) or related temporal neural network building blocks) or model training (e.g., using optimization objectives, which enforce temporal coherence) could be an interesting direction for future research. To be convincing to an end-user, one ultimately wants the time-series coherence in predictions for given sites to be comparable to time-series coherence in the training data, and therefore devoid of odd jumps except those that are physically realistic—for example due to passage of a front. The current time-independent approach is good in that it might help preserve frontal passage wind-shifts at points, but on the other hand this may possibly be at the expense of other unexplainable temporal shifts in wind velocity.

Another important question for future research, which directly follows on from these ideas, is how to account for

the spherical domain geometry in CNN-based downscaling. While data padding was found to be well suited for reshaping irregular grids on domains of up to a few thousand kilometers of horizontal extent, increasing domain size even further may lead to distortion artifacts due to disregard of the spherical geometry of the Earth's surface. The same is true for interpolation-based resampling methods, where the horizontal spacing of the sampling points varies with latitude, limiting data resolution close to the equator and enforcing data redundancy closer to the poles. Furthermore, an inappropriate treatment of domain geometries might become a serious problem, especially for models which are supposed to work on multiple domains. The use of more appropriate convolutional model architectures, like spherical CNNs for unstructured grids (e.g., Jiang *et al.*, 2019) or geometric deep learning approaches in general (e.g., Bronstein *et al.*, 2017), may help to overcome such limitations, thus increasing physical plausibility and data efficiency of the models.

From the exciting perspective of real-time application, one would ideally want to step down in scale and apply the results of this proof-of-concept study in a finer resolution setting. We envisage that operational real-time forecast runs—single deterministic and/or ensemble—could be downscaled in real-time to 1–2 km, over any preselected domains, for customer applications. This could be activated on a central cloud-type platform or locally by customers to meet their own needs. Given the small number of low-resolution predictors, data transfer requirements for the second option would be minimal, compared to say the task of transferring 4D (full-atmosphere) fields for many variables.

At such very high target resolutions, particularly if a high multiplier were used, the correct treatment of ambiguity in the data becomes increasingly important, since the same coarse-scale flow pattern may correspond to multiple fine-scale realizations. Similar to stochastic weather generators, generative CNN models like variational auto-encoders (Kingma and Welling, 2013) or convolutional generative adversarial networks (e.g., Goodfellow *et al.*, 2014; Radford *et al.*, 2015) may provide promising approaches for building flexible models for ensemble-based probabilistic downscaling. Moreover, if the low-resolution feed were based on ensemble data itself, one could then generate a super-ensemble (i.e., ensemble of ensembles) to provide the final smooth-format probabilistic output for users.

## 9 | CONCLUSION

In this study, we have analyzed convolutional neural networks (CNNs) for downscaling of wind fields on

extended spatial domains. By going from a simple linear CNN to deeper and more elaborate nonlinear models, we have investigated how the network complexity affects downscaling performance. We have further compared the performance of different CNNs to that of an ensemble of localized linear regression models.

We have shown that deeper and more complex network models are able to discover skillful mappings by exploiting nonlinear correlations for modeling the relationship between low- and high-resolution fields. Specifically, we found that all nonlinear models in our study take advantage of additional high-resolution static predictor data, such as information on local orography. In comparison, the use of three pre-defined low-resolution dynamic predictors gave only minor improvements.

Building upon the results of our study, we have envisioned a number of possible further research directions, like inclusion of temporal information into the training process, or examination of generative neural network models for probabilistic downscaling. We firmly believe that the demonstrated performance of CNNs for downscaling tasks should motivate further research towards the use of such architectures for predictive tasks.

## ACKNOWLEDGEMENTS


This research has been done within the subprojects B5 and A7 of the Transregional Collaborative Research Center SFB/TRR 165 Waves to Weather funded by the German Research Foundation (DFG). We thank all reviewers for their constructive criticism and valuable comments. Open access funding enabled and organized by Projekt DEAL.

## ORCID

Kevin Höhle  <https://orcid.org/0000-0002-4483-8388>

Michael Kern  <https://orcid.org/0000-0002-8060-3367>

Timothy Hewson  <https://orcid.org/0000-0002-3266-8828>

Rüdiger Westermann  <https://orcid.org/0000-0002-3394-0731>

## REFERENCES

- Ashcroft, J. (1994) The relationship between the gust ratio, terrain roughness, gust duration and the hourly mean wind speed. *Journal of Wind Engineering and Industrial Aerodynamics*, 53(3), 331–355. [https://doi.org/10.1016/0167-6105\(94\)90090-6](https://doi.org/10.1016/0167-6105(94)90090-6).
- Baño-Medina, J., Manzanas, R. and Gutiérrez, J.M. (2019) Configuration and intercomparison of deep learning neural models for statistical downscaling. *Geoscientific Model Development Discussions*, 13(4), 2109–2124. ISSN 1991-959X.
- Belušić, D., Hrastinski, M., Večenaj, Ž. and Grisogono, B. (2013) Wind regimes associated with a mountain gap at the northeastern Adriatic coast. *Journal of Applied Meteorology and Climatology*, 52(9), 2089–2105. ISSN 1558-8424.

- Benestad, R.E., Chen, D. and Hanssen-Bauer, I. (2008) *Empirical-Statistical Downscaling*. World Scientific Publishing Company. ISBN 978-981-281-912-3.
- Bishop, C. (2006) *Pattern Recognition and Machine Learning*, 1st edition. New York, New York: Springer. ISBN 9780387310732.
- Breiman, L. (2001) *Machine Learning*, 45(1), 5–32. <http://dx.doi.org/10.1023/a:1010933404324>.
- Bronstein, M.M., Bruna, J., LeCun, Y., Szlam, A. and Vandergheynst, P. (2017) Geometric deep learning: going beyond Euclidean data. *IEEE Signal Processing Magazine*, 34(4), 18–42.
- Browning, K.A. (2004) The sting at the end of the tail: damaging winds associated with extratropical cyclones. *Quarterly Journal of the Royal Meteorological Society*, 130(597), 375–399. ISSN 00359009. <https://doi.org/10.1256/qj.02.143>.
- Buzzi, M., Guidicelli, M. and Liniger, M.A. (2019) Nowcasting wind using machine learning from the stations to the grid.
- Cabral, B. and Leedom, L.C. (1993) Imaging vector fields using line integral convolution. *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '93*, pp. 263–270, New York, New York, ACM Press. ISBN 0897916018.
- Chandler, R.E. (2005) On the use of generalized linear models for interpreting climate variability. *Environmetrics*, 16(7), 699–715. ISSN 1180-4009.
- Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y. (2014) Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.
- Dong, C., Loy, C.C., He, K. and Tang, X. (2014) Learning a deep convolutional network for image super-resolution. In: Fleet, D., Pajdla, T., Schiele, B. and Tuytelaars, T. (Eds.) *Computer Vision – ECCV*. Cham, Switzerland: Springer International Publishing, pp. 184–199. ISBN 978-3-319-10593-2.
- Dong, C., Loy, C.C., He, K. and Tang, X. (2016a) Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2), 295–307.
- Dong, C., Loy, C.C. and Tang, X. (2016b) Accelerating the super-resolution convolutional neural network. In: Leibe, B., Matas, J., Sebe, N. and Welling, M. (Eds.) *Computer Vision – ECCV*. Cham, Switzerland: Springer International Publishing, pp. 391–407. ISBN 978-3-319-46475-6.
- Dueben, P. D., Wedi, N., Saarinen, S. and Zeman, C. (2020) Global Simulations of the Atmosphere at 1.45 km Grid-Spacing with the Integrated Forecasting System. *Journal of the Meteorological Society of Japan. Ser. II*, 98(3), 551–572. <http://dx.doi.org/10.2151/jmsj.2020-016>.
- Dumoulin, V. and Visin, F. (2016) A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*.
- Eccel, E., Ghielmi, L., Granitto, P., Barbiero, R., Grazzini, F. and Cesari, D. (2007) Prediction of minimum temperatures in an Alpine region by linear and non-linear post-processing of meteorological models. *Nonlinear Processes in Geophysics*, 14(3), 211–222. ISSN 1607-7946.
- ECMWF. (2017) IFS documentation CY46r1, Part VII: ECMWF wave model.
- Fowler, H.J., Blenkinsop, S. and Tebaldi, C. (2007) Linking climate change modelling to impacts studies: recent advances in downscaling techniques for hydrological modelling. *International Journal of Climatology*, 27(12), 1547–1578. ISSN 08998418.
- Gaitan, C.F., Hsieh, W.W., Cannon, A.J. and Gachon, P. (2014) Evaluation of linear and non-linear downscaling methods in terms of daily variability and climate indices: surface temperature in southern Ontario and Quebec, Canada. *Atmosphere-Ocean*, 52(3), 211–221. ISSN 0705-5900.
- Gibbs, J.W. (1898) Fourier's series. *Nature*, 59(1522), 200–200. ISSN 0028-0836.
- Glorot, X. and Bengio, Y. (2010) Understanding the difficulty of training deep feedforward neural networks. In: Teh, Y.W. and Titterton, M. (Eds.) *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 9, 249–256. JMLR Workshop and Conference Proceedings.
- Goodfellow, I., Bengio, Y. & Courville, A. (2016) *Deep Learning*. Cambridge: MIT Press. ISBN 9780262035613.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. (2014) Generative adversarial nets. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D. and Weinberger, K.Q. (Eds.) *Advances in Neural Information Processing Systems 27*. Curran Associates, Inc, pp. 2672–2680.
- Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S. and Lew, M.S. (2016) Deep learning for visual understanding: a review. *Neurocomputing*, 187, 27–48. ISSN 09252312.
- Guo, L., Ye, S., Han, J., Zheng, H., Gao, H., Chen, D.Z., Wang, J. and Wang, C. (2020) Ssr-vfd: spatial super-resolution for vector field data analysis and visualization. *2020 IEEE Pacific Visualization Symposium (PacificVis)*, 71–80.
- Gutiérrez, J.M., Maraun, D., Widmann, M., Huth, R., Hertig, E., Benestad, R., Roessler, O., Wibig, J., Wilcke, R., Kotlarski, S., San, D., Martín, S., Herrera, J., Bedia, A., Casanueva, R., Manzananas, M., Iturbide, M., Vrac, M., Dubrovsky, J., Ribalaygua, J., Pórtoles, O., Rätty, J., Räisänen, B., Hingray, D., Raynaud, M.J., Casado, P., Ramos, T., Zerenner, M., Turco, T., Bosshard, P., Štěpánek, J., Bartholy, R.P., Keller, D.E., Fischer, A.M., Cardoso, R.M., Soares, P.M.M., Czernecki, B. and Pagé, C. (2019) An intercomparison of a large ensemble of statistical downscaling methods over Europe: results from the VALUE perfect predictor cross-validation experiment. *International Journal of Climatology*, 39(9), 3750–3785. ISSN 08998418.
- Han, J. and Wang, C. (2020) Tsr-tvd: temporal super-resolution for time-varying data analysis and visualization. *IEEE Transactions on Visualization and Computer Graphics*, 26(1), 205–215.
- Han, J., Tao, J., Zheng, H., Guo, H., Chen, D.Z. and Wang, C. (2019) Flow field reduction via reconstructing vector data from 3-d streamlines using deep learning. *IEEE Computer Graphics and Applications*, 39(4), 54–67.
- He, K., Zhang, X., Ren, S. and Sun, J. (2016) Deep residual learning for image recognition. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.
- Hersbach, H., Bell, B., Berrisford, P., Hirahara, S., Horányi, A., Muñoz-Sabater, J., Nicolas, J., Peubey, C., Radu, R., Schepers, D., Simmons, A., Soci, C., Abdalla, S., Abellan, X., Balsamo, G., Bechtold, P., Biavati, G., Bidlot, J., Bonavita, M., Chiara, G., Dahlgren, P., Dee, D., Diamantakis, M., Dragani, R., Flemming, J., Forbes, R., Fuentes, M., Geer, A., Haimberger, L., Healy, S., Hogan, R. J., Hólm, E., Janisková, M., Keeley, S., Laloyaux, P., Lopez, P., Lupu, C., Radnoti, G., Rosnay, P., Rozum, I., Vamborg, F., Villaume, S.



- and Thépaut, J.-N. (2020) The ERA5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730), 1999–2049. <http://dx.doi.org/10.1002/qj.3803>.
- Hewitson, B.C. and Crane, R.G. (1996) Climate downscaling: techniques and application. *Climate Research*, 7(2), 85–95. ISSN 0936-577X.
- Hewson, T. (2019) Use and verification of ECMWF products in member and co-operating states (2018). *ECMWF Technical Memorandum*, 840 <https://doi.org/10.21957/jgz6nh0uc>.
- Hewson, T. D. and Neu, U. (2015) Cyclones, windstorms and the IMILAST project. *Tellus A: Dynamic Meteorology and Oceanography*, 67(1), 27128. <http://dx.doi.org/10.3402/tellusa.v67.27128>.
- Hewson, T.D. and Pillosu, F.M. (2020) A new low-cost technique improves weather forecasts across the world. arXiv preprint arXiv:2003.14397
- Hochreiter, S. and Schmidhuber, J. (1997) Long short-term memory. *Neural Computation*, 9(8), 1735–1780.
- Höhlein, K. and Kern, M.. (2020) A comparative study of convolutional neural network models for wind field downscaling: code for CNN experiments, September. Available at: <https://doi.org/10.5281/zenodo.4021023>.
- Holtstlag, A.A.M., Svensson, G., Baas, P., Basu, S., Beare, B., Beljaars, A.C.M., Bosveld, F.C., Cuxart, J., Lindvall, J., Steeneveld, G.J., Tjernström, M. and Van De Wiel, B.J.H. (2013) Stable atmospheric boundary layers and diurnal cycles: challenges for weather and climate models. *Bulletin of the American Meteorological Society*, 94(11), 1691–1706. ISSN 00030007.
- Hu, X., Naiel, M.A., Wong, A., Lamm, M. and Fieguth, P. (2019) RUNet: a robust UNet architecture for image super-resolution. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.
- Huang, H.Y., Capps, S.B., Huang, S.C. and Hall, A. (2015) Downscaling near-surface wind over complex terrain using a physically-based statistical modeling approach. *Climate Dynamics*, 44, 0 529–0 542. ISSN 14320894.
- Ioffe, S. and Szegedy, C. (2015). Batch normalization: accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.
- Jiang, C.M., Huang, J., Kashinath, K., Prabhat, P.M. and Niessner, M. (2019) Spherical CNNs on unstructured grids, *International Conference on Learning Representations*. 1–16. <https://openreview.net/forum?id=Bkl-43C9FQ>.
- Kingma, D.P. and Welling, M. (2013) Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.
- Kingma, D.P., Mohamed, S., Rezende, D.J. and Welling, M. (2014) Semi-supervised learning with deep generative models. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D. and Weinberger, K.Q. (Eds.) *Advances in Neural Information Processing Systems 27*. Curran Associates, Inc., pp. 3581–3589.
- Kipf, T.N. and Welling, M. (2016) Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.
- Kotlarski, S., Keuler, K., Christensen, O.B., Colette, A., Déqué, M., Gobiet, A., Goergen, K., Jacob, D., Lüthi, D., Van Meijgaard, E., Nikulin, G., Schär, C., Teichmann, C., Vautard, R., Warrach-Sagi, K. and Wulfmeyer, V. (2014) Regional climate modeling on European scales: a joint standard evaluation of the EURO-CORDEX RCM ensemble. *Geoscientific Model Development*, 7 (4), 1297–1333. ISSN 19919603.
- LeCun, Y., Bengio, Y. and Hinton, G. (2015) Deep learning. *Nature*, 521(7553), 436–444. ISSN 0028-0836.
- Lee, C.M., Askari, F., Book, J., Carniel, S., Cushman-Roisin, B., Dorman, C., Doyle, J., Flament, P., Harris, C. K., Jones, B.H., Kuzmic, M., Martin, P., Ogston, A., Orlic, M., Perkins, H., Poulain, P.-M., Pullen, J., Russo, A., Sherwood, C., Signell, R.P. and Thaler, D. (2005) Northern Adriatic response to a wintertime bora wind event. *Eos, Transactions American Geophysical Union*, 86(16), 157–165. ISSN 0096-3941.
- Li, X., Chen, S., Hu, X. and Yang, J. (2019) Understanding the disharmony between dropout and batch normalization by variance shift. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Lu, Z. and Chen, Y. (2019) Single image super resolution based on a modified u-net with mixed gradient loss. *arXiv preprint arXiv:1911.09428*.
- Maass, C. (2019) Mars user documentation. Available at: <https://confluence.ecmwf.int/display/UDOC/MARS+user+documentation> [Accessed 19th May 2020].
- Mao, Y. and Monahan, A. (2018) Linear and nonlinear regression prediction of surface wind components. *Climate Dynamics*, 51, 03291–03309. ISSN 0930-7575.
- Maraun, D., Widmann, M., Gutiérrez, J.M., Kotlarski, S., Chandler, R.E., Hertig, E., Wibig, J., Huth, R. and Wilcke, R.A. I. (2015) VALUE: a framework to validate downscaling approaches for climate change studies. *Earth's Future*, 3(1), 1–14. ISSN 23284277.
- Maraun, D., Widmann, M. and Gutiérrez, J.M. (2019) Statistical downscaling skill under present climate conditions: a synthesis of the VALUE perfect predictor experiment. *International Journal of Climatology*, 39(9), 3692–3703. ISSN 08998418.
- Mass, C.F., Ovens, D., Westrick, K. and Colle, B.A. (2002) Does increasing horizontal resolution produce more skillful forecasts? *Bulletin of the American Meteorological Society*, 83(3), 407–430 ISSN 0003–0007.
- McQueen, J.T., Draxler, R.R. and Rolph, G.D. (1995) Influence of grid size and terrain resolution on wind field predictions from an operational mesoscale model. *Journal of Applied Meteorology*, 34(10), 2166–2181 ISSN 0894-8763.
- Michelangeli, P.-A., Vrac, M. and Loukos, H. (2009) Probabilistic downscaling approaches: application to wind cumulative distribution functions. *Geophysical Research Letters*, 36(11) ISSN 0094-8276.
- Odena, A., Dumoulin, V. and Olah, C. (2016) Deconvolution and checkerboard artifacts. *Distill*, (), <https://doi.org/10.23915/distill.00003>.
- Pan, B., Hsu, K., AghaKouchak, A. and Sorooshian, S. (2019) Improving precipitation estimation using convolutional neural network. *Water Resources Research*, 55(3), 2301–2321 ISSN 0043-1397.
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J. and Pytorch, S.C. (2019) An imperative style, high-performance deep learning

- library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. and Garnett, R. (Eds.) *Advances in Neural Information Processing Systems 32*. Curran Associates: Inc., pp. 8026–8037.
- Pryor, S.C. (2005) Empirical downscaling of wind speed probability distributions. *Journal of Geophysical Research*, 110(D19) ISSN 0148-0227. <https://doi.org/10.1029/2005JD005899>.
- Radford, A., Metz, L. and Chintala, S. (2015) Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.
- Radić, V. and Clarke, G.K.C. (2011) Evaluation of IPCC models' performance in simulating late-twentieth-century climatologies and weather patterns over North America. *Journal of Climate*, 24(20), 5257–5274 ISSN 0894-8755.
- Räisänen, J., Hansson, U., Ullerstig, A., Döscher, R., Graham, L. P., Jones, C., Meier, H.E.M., Samuelsson, P. and Willén, U. (2004) European climate in the late twenty-first century: regional simulations with two driving global models and two forcing scenarios. *Climate Dynamics*, 22(0), 13–31 ISSN 0930-7575.
- Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N. and Prabhat. (2019) Deep learning and process understanding for data-driven earth system science. *Nature*, 566(7743), 195–204 ISSN 14764687.
- Ronneberger, O., Fischer, P. and Brox, T. (2015) U-net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M. and Frangi, A.F. (Eds.) *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Cham: Springer International Publishing, pp. 234–241 ISBN 978-3-319-24574-4.
- Rummukainen, M. (1997) *Methods for Statistical Downscaling of GCM Simulations*. RMK: Report Meteorology and Climatology, SMHI.
- Rummukainen, M. (2010) State-of-the-art with regional climate models. *Wiley Interdisciplinary Reviews: Climate Change*, 1(1), 82–96.
- Sajjadi, M.S.M., Scholkopf, B. and Enhancenet, M.H. (2017) Single image super-resolution through automated texture synthesis. *IEEE International Conference on Computer Vision (ICCV)*.
- Salvador, R., Calbó, J. and Millán, M.M. (1999) Horizontal grid size selection and its influence on mesoscale model simulations. *Journal of Applied Meteorology*, 38(9), 1311–1329. ISSN 0894-8763.
- Shen, C. (2018) A transdisciplinary review of deep learning research and its relevance for water resources scientists. *Water Resources Research*, 54(11), 8558–8593. ISSN 19447973.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. (2014) Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1), 1929–1958. ISSN 1532-4435.
- Srivastava, R.K., Greff, K. and Schmidhuber, J. (2015) Highway networks. *arXiv preprint arXiv:1505.00387*.
- Stull, R. (2017) *Practical Meteorology—an Algebra-Based Survey of Atmospheric Science*. Vancouver: BC Campus. ISBN 9780888652836.
- Timofte, R., Agustsson, E., Van Gool, L., Yang, M.-H. and Ntire, L. Z. (2017) Challenge on single image super-resolution: methods and results. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2017.
- Vandal, T., Kodra, E., Ganguly, S., Michaelis, A., Nemani, R. and Ganguly, A.R. (2018) Generating high resolution climate change projections through single image super-resolution: an abridged version. *International Joint Conferences on Artificial Intelligence Organization*. Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, 7, 5389–5393. <https://doi.org/10.24963/ijcai.2018/759>
- Vandal, T., Kodra, E. and Ganguly, A.R. (2019) Intercomparison of machine learning methods for statistical downscaling: the case of daily and extreme precipitation. *Theoretical and Applied Climatology*, 137, 0557–0570. ISSN 0177-798X.
- Vannitsem, S., Bremnes, J.B., Demaeyer, J., Evans, G.R., Flowerdew, J., Hemri, S., Lerch, S., Roberts, N., Theis, S., Atencia, A., Ben Bouallègue, Z.B., Bhend, J., Dabernig, M., De Cruz, L., Hieta, L., Mestre, O., Moret, L., Plenković, I. O., Schmeits, M., Taillardat, M., Van den Bergh, J., Van Schaeybroeck, B., Whan, K. and Ylhaisi, J. (2020) Statistical postprocessing for weather forecasts—review, challenges and avenues in a big data world. *arXiv preprint arXiv:2004.06582*.
- Wallace, J.M. and Hobbs, P.V. (2006) *Atmospheric Science: an Introductory Survey*. San Diego: Academic Press.
- Weiss, S., Chu, M., Thurey, N. and Westermann, R. (2019) Volumetric isosurface rendering with deep learning-based super-resolution. *IEEE Transactions on Visualization and Computer Graphics (Early Access)*. <https://doi.org/10.1109/TVCG.2019.2956697>.
- Wilby, R.L. and Wigley, T.M.L. (1997) Downscaling general circulation model output: a review of methods and limitations. *Progress in Physical Geography: Earth and Environment*, 21(4), 530–548. ISSN 0309-1333.
- Wilks, D.S. (2010) Use of stochastic weather generators for precipitation downscaling. *Wiley Interdisciplinary Reviews: Climate Change*, 1(6), 898–907. ISSN 17577780.
- Wilks, D.S. (2012) Stochastic weather generators for climate-change downscaling, part II: multivariable and spatially coherent multisite downscaling. *Wiley Interdisciplinary Reviews: Climate Change*, 3(3), 267–278. ISSN 17577780.
- Wood, A.W., Leung, L.R., Sridhar, V. and Lettenmaier, D.P. (2004) Hydrologic implications of dynamical and statistical approaches to downscaling climate model outputs. *Climatic Change*, 62, 0189–0216. ISSN 01650009.
- Xue, Y., Janjic, Z., Dudhia, J., Vasic, R. and Sales, F.D. (2014) A review on regional dynamical downscaling in intraseasonal to seasonal simulation/prediction and major factors that affect downscaling ability. *Atmospheric Research*, 147-148, 68–85. ISSN 01698095.
- Yang, W., Zhang, X., Tian, Y., Wang, W., Xue, J. and Liao, Q. (2019) Deep learning for single image super-resolution: a brief review. *IEEE Transactions on Multimedia*, 21(12), 3106–3121.
- Zhang, C., Bengio, S., Hardt, M., Recht, B. and Vinyals, O. (2016) Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*.
- Zhou, Z., Hou, Y., Wang, Q., Chen, G., Lu, J., Tao, Y., and Lin, H. (2017) Volume upscaling with convolutional neural

networks. *Proceedings of the Computer Graphics International Conference*, CGI '17, New York, NY. Association for Computing Machinery. ISBN 9781450352284: <https://doi.org/10.1145/3095140.3095178>.

Zorita, E. and von Storch, H. (1999) The analog method as a simple statistical downscaling technique: comparison with more complicated methods. *Journal of Climate*, 12(8), 2474–2489. ISSN 0894-8755.

**How to cite this article:** Höhle K, Kern M, Hewson T, Westermann R. A comparative study of convolutional neural network models for wind field downscaling. *Meteorol Appl.* 2020;27:e1961. <https://doi.org/10.1002/met.1961>



# Attribution 4.0 International Creative Commons

## Deed – reformatted for display in this thesis

### You are free to:

1. **Share** — copy and redistribute the material in any medium or format for any purpose, even commercially.
2. **Adapt** — remix, transform, and build upon the material for any purpose, even commercially.
3. The licensor cannot revoke these freedoms as long as you follow the license terms.

### Under the following terms:

1. **Attribution** — You must give appropriate credit, provide a link to the license, and indicate if changes were made . You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
2. **No additional restrictions** — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation.

No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material.

### Notice

This deed highlights only some of the key features and terms of the actual license. It is not a license and has no legal value. You should carefully review all of the terms and conditions of the actual license before using the licensed material.

Creative Commons is not a law firm and does not provide legal services. Distributing, displaying, or linking to this deed or the license that it summarizes does not create a lawyer-client or any other relationship.

### Deed Source / Canonical URL

<https://creativecommons.org/licenses/by/4.0/>



# Postprocessing of Ensemble Weather Forecasts Using Permutation-Invariant Neural Networks

KEVIN HÖHLEIN<sup>a</sup>, BENEDIKT SCHULZ<sup>b</sup>, RÜDIGER WESTERMANN<sup>a</sup>, AND SEBASTIAN LERCH<sup>b,c</sup>

<sup>a</sup> *Technical University of Munich, Munich, Germany*

<sup>b</sup> *Karlsruhe Institute of Technology, Karlsruhe, Germany*

<sup>c</sup> *Heidelberg Institute for Theoretical Studies, Heidelberg, Germany*

(Manuscript received 18 August 2023, in final form 31 October 2023, accepted 20 November 2023)

**ABSTRACT:** Statistical postprocessing is used to translate ensembles of raw numerical weather forecasts into reliable probabilistic forecast distributions. In this study, we examine the use of permutation-invariant neural networks for this task. In contrast to previous approaches, which often operate on ensemble summary statistics and dismiss details of the ensemble distribution, we propose networks that treat forecast ensembles as a set of unordered member forecasts and learn link functions that are by design invariant to permutations of the member ordering. We evaluate the quality of the obtained forecast distributions in terms of calibration and sharpness and compare the models against classical and neural network-based benchmark methods. In case studies addressing the postprocessing of surface temperature and wind gust forecasts, we demonstrate state-of-the-art prediction quality. To deepen the understanding of the learned inference process, we further propose a permutation-based importance analysis for ensemble-valued predictors, which highlights specific aspects of the ensemble forecast that are considered important by the trained postprocessing models. Our results suggest that most of the relevant information is contained in a few ensemble-internal degrees of freedom, which may impact the design of future ensemble forecasting and postprocessing systems.

**KEYWORDS:** Neural networks; Ensembles; Probability forecasts/models/distribution; Model evaluation/performance; Postprocessing; Model interpretation and visualization

## 1. Introduction

Operational weather forecasting relies on numerical weather prediction (NWP) models. Since such models are subject to multiple sources of uncertainty, such as uncertainty in the initial conditions or model parameterizations, a quantification of the forecast uncertainty is indispensable. To achieve this, NWP models generate a set of deterministic forecasts, so-called ensemble forecasts, based on different initial conditions and variations of the underlying physical models. Since these forecasts are subject to systematic errors such as biases and dispersion errors, statistical postprocessing is used to enhance their reliability (see, e.g., Vannitsem et al. 2018). Recently, machine learning (ML) approaches for statistical postprocessing have shown superior performance over classical methods. For instance, Rasp and Lerch (2018) propose a distribution regression network (DRN) that predicts the parameters of a temperature forecast distribution from a suitable family of parametric distributions. In subsequent work, Schulz and Lerch (2022b) found that shallow multilayer perceptrons (MLPs) with forecast distributions of different flexibility achieve state-of-the-art results in postprocessing wind gust ensemble forecasts.

An ensemble forecast consists of multiple separate member forecasts, which are generated by repeatedly running NWP

simulations with different model parameterizations and initial conditions. Typically, the configurations of different runs are sampled randomly from an underlying distribution of plausible simulation conditions, obtained, for example, from uncertainty-aware data assimilation. The member forecasts can then be seen as identically distributed and interchangeable random samples from a distribution of possible future weather states. In this setting, statistical postprocessing of ensemble forecasts can be phrased as a prediction task on unordered predictor vectors and requires solutions that are tailored to match the predictor format. Specifically, member interchangeability demands that the predictions of a well-designed postprocessing system should not be affected by permutations, that is, shuffling, of the ensemble members. Systems that satisfy this requirement are called *permutation invariant*. Established postprocessing methods rely on basic summary statistics of the raw ensemble forecast to inform the estimation of the postprocessed distribution and are thus permutation invariant by design. However, especially in large ensembles, the details of the distribution may carry valuable information for postprocessing, and a more elaborate treatment of the inner structure of the raw forecast ensembles may help to improve forecast accuracy for example in ambiguous forecast situations, where summary-based methods fail to evaluate the likelihood of different weather patterns accurately.

While studies have started to explore how specialized model architectures can help to improve postprocessing only recently (Bremnes 2020; Mlakar et al. 2023; Ben-Bouallegue et al. 2023), ML provides a variety of further approaches to enforcing permutation invariance in data-driven learning. Motivated by the success of permutation-invariant neural network (NN) architectures in representation learning, anomaly detection or

Supplemental information related to this paper is available at the Journals Online website: <https://doi.org/10.1175/AIES-D-23-0070.s1>.

Corresponding author: Kevin Höhlelein, kevin.hoehlein@tum.de

set classification (e.g., Ravanbakhsh et al. 2016; Zaheer et al. 2017; Lee et al. 2019; Sannai et al. 2019; Zhang et al. 2019), where the models profit from the ability to extract concise feature representations from unordered data, permutation-invariant NNs appear as promising candidates for improving ensemble postprocessing.

### Contribution

In this study, we investigate the capabilities of different permutation-invariant NN architectures for univariate postprocessing of station predictions. We evaluate the proposed models on two exemplary stationwise postprocessing tasks with different characteristics. The ensemble-based network models are compared to classical methods and basic NNs, which operate only on ensemble summary statistics but are trained under identical predictor conditions otherwise. We further assess how much of the predictive information is carried within the details of the ensemble distribution, and how much of the model skill arises from other factors. To shed light on the sources of model skill, we propose an ensemble-oriented feature importance analysis and study the effect of ensemble-internal degrees of freedom using conditional feature permutation.

## 2. Related work

### a. Statistical postprocessing of ensemble forecasts

One of the most popular methods for statistical postprocessing of ensemble forecasts is ensemble model output statistics (EMOS; Gneiting et al. 2005), which performs a distributional regression based on a suitable family of parametric distributions and summary statistics of the ensemble. Due to its simplicity, EMOS has been applied to a wide range of weather variables including temperature (Gneiting et al. 2005), wind gusts (Pantillon et al. 2018), precipitation (Scheuerer 2014), and solar radiation (Schulz et al. 2021). Following the simple statistical EMOS approach, the success of ML methods (Taillardat et al. 2016; Messner et al. 2017), which are able to incorporate additional information and learn more complex patterns, have motivated the use of modern NN methods. First NN-based approaches are DRN (Rasp and Lerch 2018) as an extension of the EMOS framework, and the Bernstein quantile network (BQNs; Bremnes 2020) that provides a more flexible forecast distribution. In Schulz and Lerch (2022b), NN-based approaches were adapted toward the prediction of wind gusts and outperformed classical methods. Recently, research has shifted toward the use of more sophisticated network architectures. Examples include convolutional NNs that incorporate spatial NWP output fields (Scheuerer et al. 2020; Grönquist et al. 2021; Veldkamp et al. 2021; Horat and Lerch 2023), and generative models to produce multivariate forecast distributions (Dai and Hemri 2021; Chen et al. 2022).

Only recently, Mlakar et al. (2023) have proposed NN models that explicitly admit the use of ensemble-structured predictors by employing a dynamic attention mechanism. The resulting models perform best in the benchmark study of Demaeyer et al. (2023). Mlakar et al. (2023) address postprocessing with similar

methods as this work, but do not focus explicitly on comparing different network design patterns for inference based on ensemble-valued predictors. In orthogonal work, Finn (2021) and Ben-Bouallegue et al. (2023) apply transformer-based NNs to ensemble postprocessing. In contrast to this study, both approaches focus on gridded predictor data, thus relying on different network architectures, and postprocess ensembles in a member-by-member fashion, whereas this work concentrates on distributional regression.

For a general review of statistical postprocessing of weather forecasts, we refer to Vannitsem et al. (2018), a review of recent developments and challenges can be found in Vannitsem et al. (2021) and Haupt et al. (2021).

### b. Neural network architectures for regression on set-structured data

From an ML perspective, postprocessing of ensemble forecasts can be phrased as a regression task on set-structured predictors. Multiple studies have demonstrated that dedicated permutation-invariant NN architectures can help to improve prediction quality and generalization in diverse learning problems (e.g., Vinyals et al. 2015; Lyle et al. 2020), thus motivating the exploration of permutation-invariant NNs also for ensemble postprocessing. Early works on permutation-invariant layers for NNs (Ravanbakhsh et al. 2016) and pooling-based permutation-invariant NNs (Edwards and Storkey 2016) were followed by the more comprehensive framework *DeepSets* (Zaheer et al. 2017), which encompasses some of the most common design patterns for ML inference on set-structured predictors. Due to its generality, *DeepSets* is selected as one of the representative learning approaches in this study and is further discussed in section 4a. Soelch et al. (2019) highlight that architectural improvements, such as the use of more expressive pooling functions, may enhance model performance, which we consider in the design of the model architectures for postprocessing.

An alternative approach to permutation-invariant inference has been proposed by Lee et al. (2019), who use (multihead) attention functions (Vaswani et al. 2017) for permutation-invariant inference on set-valued data. Attention-based NNs, also known as transformers, have proven powerful in a variety of computer vision tasks (e.g., Khan et al. 2022) as well as postprocessing (Finn 2021; Ben-Bouallegue et al. 2023, see their section 2a) and are thus considered as a second paradigm for building permutation-invariant NNs.

### c. Machine learning explainability and feature importance

ML explainability has attracted substantial interest throughout the last decade (for recent surveys, see, e.g., Guidotti et al. 2018; Linardatos et al. 2021; Burkart and Huber 2021) and is increasingly adopted in the Earth-system sciences (e.g., Reichstein et al. 2019; Höhle et al. 2020; Labe and Barnes 2021; Farokhmanesh et al. 2023) to gain understanding on the reasoning mechanisms behind ML-based inference approaches. The most relevant approaches for this work are based on permutation feature importance (PFI; Breiman 2001), which aims to assess the (relative) importance of different predictors for inference.



In PFI, relevance scores are assigned to the predictors based on the accuracy loss after permuting the predictor values within the test dataset and have been applied in the postprocessing before (e.g., Rasp and Lerch 2018; Schulz and Lerch 2022b) with a focus on scalar-valued predictors. In this work, we propose a conditional PFI measure for ensemble-valued predictors, which allows attributing importance values to different aspects of the ensemble-internal variability. Conditional perturbation measures have been considered in earlier works (e.g., Strobl et al. 2008; Molnar et al. 2024), where the conditioning is used to evaluate the importance of specific predictors in the context of the remaining predictors. By contrast, our approach addresses specifically the distribution characteristics of the ensemble-valued predictors encountered in postprocessing.

### 3. Benchmark methods and forecast distributions

#### a. Assessing predictive performance

We evaluate probabilistic forecasts based on the paradigm of Gneiting et al. (2007), that is, a forecast should maximize sharpness subject to calibration. Both sharpness and calibration can be assessed quantitatively using proper scoring rules (Gneiting and Raftery 2007). A popular choice is the continuous ranked probability score (CRPS; Matheson and Winkler 1976):

$$\text{CRPS}(F, y) = \int_{-\infty}^{\infty} [F(z) - \mathbb{1}\{y \leq z\}]^2 dz,$$

wherein  $y \in \mathbb{R}$  is the observed value,  $F$  is the cumulative distribution function (CDF) of the forecast distribution, and  $\mathbb{1}$  is the indicator function. The CRPS can be computed analytically for a wide range of distributions including the truncated logistic distribution and probabilistic forecasts in ensemble form (Jordan et al. 2019).

In addition to the CRPS, we assess calibration based on the empirical coverage of prediction intervals (PIs) derived from the forecast distribution, and sharpness on the corresponding length. Under the assumption of calibration, the observed coverage of a PI should match the nominal level, and a forecast is sharper the smaller the length of the PI. In line with Schulz and Lerch (2022b), we choose the PI level based on the size of the underlying ensemble. For an ensemble of size  $M \in \mathbb{N}$ , this gives rise to a PI with nominal level  $(M - 1)/(M + 1)$ .

Further, we qualitatively assess calibration based on (unified) probability integral transform (PIT) histograms (Gneiting and Katzfuss 2014; Vogel et al. 2018). While a flat histogram indicates that the forecasts are calibrated, systematic deviations indicate miscalibration. For more details on the evaluation of probabilistic forecasts, we refer to Gneiting and Katzfuss (2014).

#### b. Distributional regression with parametric forecast distributions (EMOS, DRN)

In this study, we consider postprocessing of the ensemble forecast for a real-valued random variable  $Y$  as a distributional regression task on ensemble-structured predictors. We focus on the case of stationwise forecasts, which are given as

prediction vectors  $\mathbf{x} \in \mathbb{R}^p$ , each comprising the predictions of  $p$  scalar-valued meteorological variables, such as surface temperature or 10-m wind speed at a station site. Typically, one of the forecast variables corresponds directly to the target variable  $Y$  and is thus termed the primary prediction. An  $M$ -member ensemble forecast  $X = \{\mathbf{x}_1, \dots, \mathbf{x}_M\} \subset \mathbb{R}^p$  is composed of  $M$  such prediction vectors, which represent samples from the predicted distribution of future weather states. The space of  $M$ -member ensemble forecasts will be denoted as  $[\mathbb{R}^p]_M$ .

Within the (parametric) distributional regression framework, the parameter vector  $\boldsymbol{\theta}$  of a parametric distribution  $\mathcal{F}_{\boldsymbol{\theta}}$  is linked to the predictors via a function that is estimated by minimizing a proper scoring rule. The underlying model can be written as

$$\begin{aligned} Y|X &\sim \mathcal{F}_{\boldsymbol{\theta}}, \\ \boldsymbol{\theta} &= g(X) \in \Theta, \end{aligned} \tag{1}$$

where  $g: [\mathbb{R}^p]_M \rightarrow \Theta$  is called the link function and  $\Theta \subseteq \mathbb{R}^D$  denotes the  $D$ -dimensional parameter space corresponding to  $\mathcal{F}_{\boldsymbol{\theta}}$ .

For EMOS,  $g$  is defined as a generalized affine-linear function of ensemble summary statistics, such as ensemble mean or standard deviation, and provides only limited flexibility for distribution estimation. DRN (Rasp and Lerch 2018; Schulz and Lerch 2022b), in contrast, admits the data-driven estimation of arbitrary link functions using NNs, thus increasing the learning ability. The forecast distribution as well as the underlying proper scoring rule used for optimization are two implementation choices.

#### c. Flexible distribution estimator (BQN)

Distributional regression methods based on a parametric forecast distribution are robust but lack flexibility as they are bound to the parametric distribution family of choice. Typical choices of forecast distributions include the normal (Gneiting et al. 2005; Rasp and Lerch 2018), logistic (Schulz and Lerch 2022b) or generalized extreme value distribution (Lerch and Thorarinsdottir 2013; Scheuerer 2014). They all lack the ability to express multimodalities that are required, for example, when different weather patterns occur. Hence, methods that do not rely on parametric assumptions have been proposed in the postprocessing literature. Examples are the direct adjustment of the ensemble members (van Schaeybroeck and Vannitsem 2015) or quantile regression forests (Taillardat et al. 2016). BQN (Bremnes 2020) models the forecast distribution as a quantile function, which is represented as a linear combination of Bernstein (basis-) polynomials of degree  $d \in \mathbb{N}$  with variable mixing coefficients  $\boldsymbol{\alpha} = (\alpha_0, \dots, \alpha_d) \in \mathbb{R}^{d+1}$ , such that  $\alpha_0 \leq \dots \leq \alpha_d$ . The inference network is designed to output parameters  $\boldsymbol{\theta}$  that parameterize the mixing coefficients, that is,  $\boldsymbol{\alpha} = \boldsymbol{\alpha}(\boldsymbol{\theta})$ . In contrast to DRN, this formulation offers increased flexibility for modeling multimodality, while requiring hard upper and lower bounds for the values of the forecast variable. For BQN models, the optimization is guided by an average of quantile scores (Koenker and Bassett 1978), which can be seen as a discrete approximation of the CRPS (Gneiting and Ranjan 2011). For the evaluation of BQN forecasts, we generate an ensemble

TABLE 1. Predictor utilization by postprocessing methods. Methods used in this study are indicated by “ours.” Abbreviations: permutation-invariant (perm.-inv.); standard deviation (SD), station embedding (SE); station predictors and embedding (SP + SE).

Predictors	Ensemble-valued		Scalar-valued	
	Method	Primary prediction	Auxiliary predictions	Spatial
EMOS (Schulz and Lerch 2022b, ours)	Mean + SD	—	Different models per station and month	
BQN (Bremnes 2020)	Ensemble (sorted)	—	SE	—
BQN (Schulz and Lerch 2022b)	Ensemble (sorted)	Mean	SP + SE	Day of year
BQN (ours)	Mean + SD	Mean	SP + SE	Day of year
DRN (Schulz and Lerch 2022b, ours)	Mean + SD	Mean	SP + SE	Day of year
Perm.-inv. DRN + BQN (ours)	Ensemble (perm.-inv.)	Ensemble (perm.-inv.)	SP + SE	Day of year

of equidistant quantiles analogous to Schulz and Lerch (2022b). In the original implementation, the link function of BQN is specified as an NN, which receives a sorted sequence of univariate ensemble predictors as its input (Bremnes 2020). Schulz and Lerch (2022b) augment this approach by using ensemble-valued predictors for the predictor of the target variable and ensemble means for additional auxiliary predictor variables. Despite admitting permutation-invariant inference, both model variants are constrained to processing ensembles of fixed size. To alleviate this limitation, we avoid the sorting operation in this work and inform BQN models analogous to DRN using ensemble summary statistics. A comparison of both variants is conducted in the online supplemental materials, demonstrating the equivalence of the approaches.

#### d. Use of auxiliary predictors

In addition to the predictions of the target variable, most algorithms use auxiliary information to improve the prediction performance (see Table 1). We distinguish between ensemble-valued and scalar-valued predictors, where ensemble-valued predictors vary between different members and scalar-valued predictors do not. In the ensemble-valued case, we differentiate the primary prediction from auxiliary predictions of other meteorological variables. For either of these, postprocessing models can have access to the full set of ensemble values or only to summary statistics.

Scalar-valued predictors refer to contextual information, such as station-specific coordinates and orography details (cf. Table 1, station predictors), as well as to temporal information, such as the day of the year. We consider only models that are trained on predictions for specific initialization and lead times, such that information about the diurnal cycle is not required. While most approaches include the scalar predictors explicitly as features in the regression process, EMOS takes advantage of categorical location and time information implicitly by fitting independent models for each station and month (Schulz and Lerch 2022b). BQN- and DRN-type models are trained separately for each lead time but employ a learned station embedding (Rasp and Lerch 2018; Schulz and Lerch 2022b) to share the same model between different station sites. Notably, the permutation-invariant models (cf. Table 1, permutation-invariant) considered in this study have access to the

richest predictor pool. A complete list of model inputs on the parameter level can be found in Tables A1–A3 in appendix A.

## 4. Permutation-invariant neural network architectures

From the variety of permutation-invariant model architectures, we select two representative approaches, *set pooling architectures* and *set transformers*, which we adapt for distributional regression. Compared with the benchmark methods of section 3, the proposed networks replace the summary-based ensemble processing while the parameterization of the forecast distributions remains unchanged. A schematic comparison of both permutation-invariant architectures is shown in Fig. 1.

### a. Set pooling architectures

Set pooling architectures (Zaheer et al. 2017), also known as DeepSets, achieve permutation invariance via extraction and permutation-invariant summarization of learned latent features. The features are obtained by applying an encoder MLP to all ensemble members separately, followed by a permutation-invariant pooling function and a decoder MLP, which outputs the distribution parameters  $\theta$ . Due to the division into encoding, pooling, and decoding, we will thus use the names *set pooling* and *encoder–decoder* (ED) architecture synonymously.

In experiments, we considered different variants of ensemble summarization based on average and extremum pooling, as well as adaptive pooling functions based on an attention mechanism (Lee et al. 2019; Soelch et al. 2019), discussed below. Overall, we find that the pooling mechanism is of minor importance. Detailed comparisons are thus deferred to the supplemental materials and all subsequent experiments use attention-based pooling consistently.

### b. Set transformer

Set transformers (Lee et al. 2019) are NNs, that model interactions between set members via self-attention. *Attention* is a form of nonlinear activation function, in which the relevance of the inputs is determined via a matching of input-specific key and query vectors. *Multihead attention* allows the model to attend to multiple key patterns in parallel (Vaswani et al. 2017).

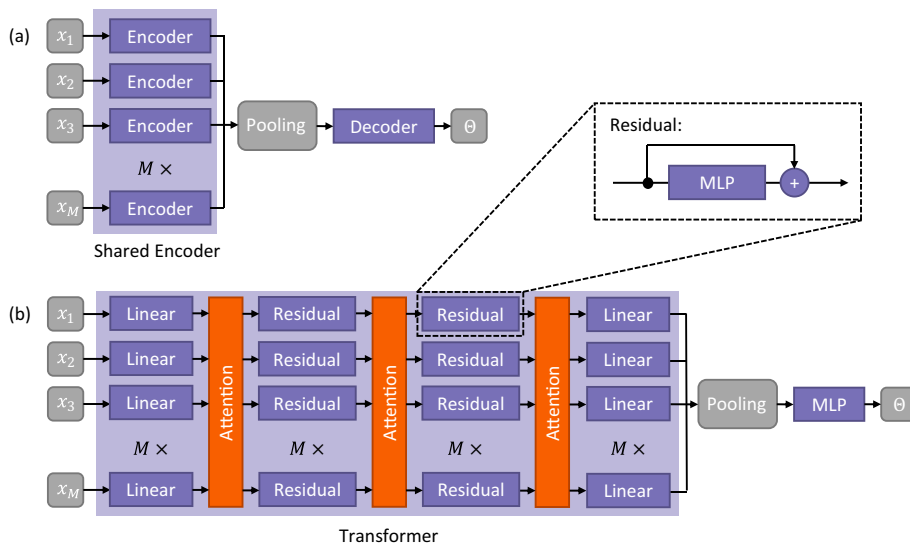


FIG. 1. (a) Set pooling architecture, consisting of encoder and decoder MLPs, and (b) set transformer, featuring attention blocks and intermediate MLPs with residual connections. While the encoder–decoder architecture admits interactions between members only inside the pooling step, the set transformer admits information transfer between the members in each attention step.

Lee et al. (2019) combine multihead attention with a memberwise NN to build a permutation-invariant set-attention block, from which a set transformer is constructed by stacking multiple instances. Set transformers apply straightforwardly to ensemble data and can exploit all aspects of the available ensemble dataset by allowing for information exchange between ensemble members early in the inference process. We construct a set transformer by using three set-attention blocks with 8 attention heads (Vaswani et al. 2017; Lee et al. 2019). Each block comprises a separate MLP with two hidden layers. Additionally, the first set-attention block is preceded by a linear layer to align the channel number of the ensemble input with the hidden dimension of the set-attention blocks. To construct vector-valued predictions from set-valued inputs, Lee et al. (2019) propose attention-based pooling, in which the output query vectors are implemented as learnable parameters. After pooling, the final prediction  $\theta$  is obtained by applying another two-layer MLP.

**5. Data**

We evaluate the performance of the proposed models in two postprocessing tasks using the datasets described in Table 2. An overview of the predictor and target variables is provided in appendix A.

*a. Wind gust prediction in Germany*

In the first case study, we employ our methods for station-wise postprocessing of wind gust forecasts using a dataset that has previously been used in Pantillon et al. (2018) and Schulz and Lerch (2022b). The ensemble forecasts are based on the COSMO ensemble prediction system (COSMO-DE; Baldauf et al. 2011) and consist of 20 members forecasts, simulated with a horizontal resolution of 2.8 km. The forecasts are initialized at 0000 UTC, and we consider the lead times 6, 12, and 18 h. Other than wind gusts, the dataset comprises ensemble forecasts of several meteorological variables, such as temperature, pressure, precipitation, and radiation. An overview of all predictors is shown in Table A1 (appendix A). The

TABLE 2. Overview of the data used in the postprocessing applications described in section 5.

Dataset	Wind gust forecasts	EUPPbench (re)forecasts
Underlying NWP model	COSMO-DE-EPS	ECMWF-IFS
Initialization time	0000 UTC	0000 UTC
Ensemble size $M$	20	Reforecasts: 11 Forecasts: 51
Predicted ensemble forecast quantities $p$	61	28
Region	Germany	Central Europe
Stations	175	117
Lead times considered in h	6, 12, 18	24, 72, 120
Training samples	315 000	374 000
Test samples	63 000	Reforecasts: 97 000 Forecasts: 85 000

predictions are verified against observations measured at 175 stations of the German Weather Service [Deutscher Wetterdienst (DWD)]. Forecasts for the individual weather stations are obtained from the closest grid point. The time period of the forecast and observation data starts on 9 December 2010 and ends on 31 December 2016. The models use the data from 2010 to 2015 for model estimation, using 2010–14 as training and 2015 as validation period. The forecasts are then verified in 2016. As in Schulz and Lerch (2022b), each lead time is processed separately.

As detailed in Schulz and Lerch (2022b), a minor caveat is caused by a nontrivial substructure of the forecast ensembles. The 20-member ensembles constitute a conglomerate of four subensembles, which are generated with slightly different model configurations. While this formally violates the assumption of statistical interchangeability of the members, the subensembles are sufficiently similar to justify the application of permutation-invariant models.

For the benchmark methods EMOS and DRN, we use the exact same forecasts as in Schulz and Lerch (2022b), both estimating the parameters of a truncated logistic distribution by minimizing the CRPS, see their section 3 for details. BQN is adapted as described in section 3 and Table 1.

#### *b. Temperature forecasts from the EUPPBench dataset*

In a second example, we postprocess ensemble forecasts of surface temperature using a subset of the EUPPBench postprocessing benchmark dataset (Demaeyer et al. 2023). EUPPBench provides paired forecast and observation data from two sets of samples. The first part consists of 20 years of reforecast data (1997–2016) from the Integrated Forecasting System (IFS) of the ECMWF with 11 ensemble members. Mimicking typical operational approaches, the reforecast dataset is used as training data, complemented by additional two years (2017 and 2018) of 51-member forecasts as test data. EUPPBench comprises sample data from multiple European countries—Austria, Belgium, France, Germany, and the Netherlands—which are publicly accessible via the CliMetLab API (ECMWF 2013). Additional data for Switzerland can be requested from the Swiss Weather Service but is not used in this study. EUPPBench constitutes a comprehensive dataset of samples over a long time period. In contrast to the wind gust forecasts, the EUPPBench ensemble members are exchangeable, so that permutation-invariant model architectures are optimally suited.

Deviating from the EUPPBench convention, models are tested on the 51-member forecasts, and the last 4 years of the reforecast dataset are considered as an independent test set of 11-member forecast samples. This allows us to assess the generalization capabilities of the ensemble-based postprocessing models on data equivalent to the training data, as well as on data with larger ensemble sizes. Furthermore, we use the full set of available surface- and pressure-level predictor variables, whereas the original EUPPBench task is restricted to using only surface temperature data. While this design choice hinders the direct comparison of the evaluation metrics in this paper with the original EUPPBench models, it enables a

more comprehensive assessment of the relative benefits of using summary-based versus ensemble predictors. An overview of the predictors can be found in Table A2 (appendix A). From the pool of available forecast lead times, we select 24, 72, and 120 h for a closer analysis.

Unlike previous postprocessing applications for temperature (e.g., Gneiting et al. 2005; Rasp and Lerch 2018), we employ a zero-truncated logistic distribution as parametric forecast distribution for DRN, instead of a zero-truncated normal, as preliminary tests showed a slightly superior predictive performance of the logistic distribution pattern (see supplemental material for details). The zero-censoring arises from the use of the Kelvin scale for measuring temperatures and allows the use of the same model configuration for both temperature and wind gust predictions. In particular, the EMOS and DRN benchmark approaches are identical for both datasets.

## 6. Performance evaluation

For each of the postprocessing methods, we generated a pool of 20 networks in each forecast scenario. To ensure a fair comparison to the benchmark methods, we follow the approach from Schulz and Lerch (2022a,b), who build an ensemble of 10 networks and combine the forecasts via quantile aggregation. Hence, we draw 10 members from the pool and repeat this procedure 50 times to quantify the uncertainty of sampling from the general pool. For all model variants and resamples, we select those configurations as the final forecast that yield the lowest CRPS on the validation set. Details on hyperparameter settings are listed in appendix B and tuning procedures are discussed in the supplemental material. For both datasets, we compute the average CRPS, PI length, and PI coverage for the different forecast lead times based on the respective test datasets. The average is calculated over the resamples of the aggregated network ensembles. In what follows, we refer to pooling-based encoder–decoder (ED) models and set transformers (ST), and suffixes DRN and BQN indicate the parameterization of the forecast distribution. The model categories DRN and BQN without additional prefixes refer to the benchmark models based on summary statistics.

#### *a. Wind gust forecasts*

Table 3 shows the quantitative evaluation for lead times 6, 12, and 18 h. All permutation-invariant model architectures perform similarly to the DRN and BQN benchmarks and outperform both the EPS and conventional postprocessing via EMOS, thus achieving state-of-the-art performance for all lead times. Further, the PI lengths and coverages are similar to those of the benchmark methods with the same forecast distribution, indicating that the ensemble-based models achieve approximately the same level of sharpness as the benchmark networks while being well calibrated. Note that the underlying distribution type should be taken into account when comparing the sharpness of different postprocessing models based on the PI length, as the DRN and BQN forecast distributions exhibit different tail behavior, which affects the PI lengths for different nominal levels (see supplemental materials for details). A noticeable difference

TABLE 3. Mean CRPS ( $m s^{-1}$ ), PI length ( $m s^{-1}$ ), and PI coverage (%) of the postprocessing methods for the different lead times of the wind gust data (20-member ensemble, year 2016). Recall that the nominal level of the PIs is approximately 90.48%. The best-performing models (w.r.t. CRPS) are marked in bold.

Lead time	6 h			12 h			18 h		
	Method	CRPS	PI length	PI coverage	CRPS	PI length	PI coverage	CRPS	PI length
EPS	1.31	2.37	43.18	1.26	3.31	56.32	1.32	3.80	59.78
EMOS	0.88	5.58	92.83	0.97	6.01	91.92	1.04	6.43	92.46
BQN	<b>0.79</b>	4.60	90.23	<b>0.85</b>	4.90	89.65	<b>0.95</b>	5.56	90.70
DRN	<b>0.79</b>	4.75	91.43	<b>0.85</b>	5.11	91.08	<b>0.95</b>	5.68	91.78
ED-BQN	0.80	4.56	89.83	0.86	4.92	89.56	<b>0.95</b>	5.55	90.55
ED-DRN	<b>0.79</b>	4.70	91.17	0.86	5.15	91.13	<b>0.95</b>	5.76	92.07
ST-BQN	0.80	4.67	90.20	0.87	5.01	89.94	0.96	5.61	90.70
ST-DRN	0.80	4.77	91.34	0.86	5.17	91.13	0.96	5.83	92.24

between the network classes is that the ED models result in sharper PIs than the ST models. This coincides with the empirical PI coverages of the methods in that wider PIs typically result in a higher coverage. Figure 2 shows the PIT histograms of the postprocessed forecasts. While differences are seen between DRN-type and BQN-type models, all DRN-type and all BQN-type models show very similar patterns. While all models are well calibrated, DRN-type models reveal limitations in the resolution of gusts in the lower segment of the distribution. BQN-type models all yield very uniform calibration histograms.

*b. EUPPBench surface temperature reforecasts*

As shown in Table 4, both ED and ST models show significant advantages compared to the EPS and EMOS in terms of CRPS and PI length for the EUPPBench dataset. Differences between the network variants arise mainly due to the use of

different forecast distribution types. Note that the lead times of the wind gust dataset are in the short range with a maximum of 18 h, whereas the lead times considered in the EUPPBench dataset range from one to five days. Hence, the differences between the lead times in the effects of postprocessing are more pronounced. For example, for a lead time of 120 h, the improvement of the network-based postprocessing methods over the conventional EMOS approach is much smaller than for shorter lead times. In particular, ST models perform the best for lead time 24 h and all newly proposed models result in the smallest CRPS for lead time 120 h. In terms of the PI length and coverage, we find that the ED and ST models tend to generate slightly sharper predictions. A more detailed discussion of the differences in the PI lengths due to the choice of the underlying distribution is provided in the supplemental material. The PIT histograms in Fig. 2 show that the BQN models struggle to set accurate upper and lower

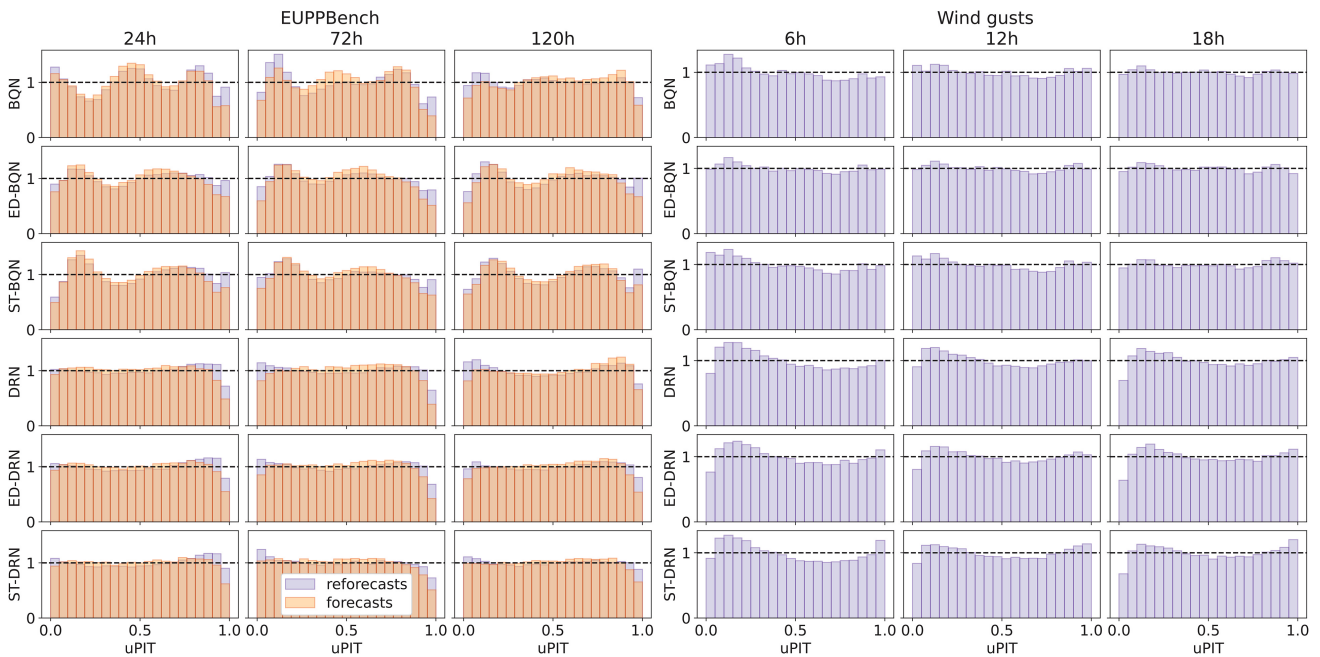


FIG. 2. PIT histograms of the postprocessing models for EUPPBench (left) 11-member reforecast and 51-member forecast ensembles and (right) 20-member wind gust forecasts.

TABLE 4. Mean CRPS (K), PI length (K), and PI coverage (%) of the postprocessing methods for the different lead times for the EUPPBench reforecast data (11-member ensemble, years 2013–16). Recall that the nominal level of the PIs is approximately 83.33%. The best-performing models (w.r.t. CRPS) are marked in bold.

Lead time	24 h			72 h			120 h		
	Method	CRPS	PI length	PI coverage	CRPS	PI length	PI coverage	CRPS	PI length
EPS	1.21	1.81	39.85	1.28	3.29	56.62	1.54	4.89	63.44
EMOS	0.82	3.85	82.56	0.96	4.72	83.96	1.25	6.05	83.12
BQN	0.67	3.32	84.73	0.87	4.44	86.08	1.20	6.24	86.09
DRN	0.67	3.28	84.16	<b>0.86</b>	4.27	84.58	<b>1.19</b>	5.70	83.09
ED-BQN	0.67	3.29	84.57	0.87	4.45	86.05	<b>1.19</b>	6.03	85.45
ED-DRN	0.67	3.19	83.39	0.87	4.25	84.11	<b>1.19</b>	5.64	82.60
ST-BQN	<b>0.66</b>	3.16	84.01	0.87	4.31	84.85	<b>1.19</b>	6.07	85.15
ST-DRN	<b>0.66</b>	3.06	82.67	0.87	4.18	83.44	<b>1.19</b>	5.77	83.17

bounds for the predicted distribution, whereas DRN distributions do not show such issues. Instead, they face the problem that the tail is too heavy. Overall, all postprocessing methods result in calibrated forecasts, while the DRN forecasts appear slightly better calibrated than the BQN forecasts, yielding PIT histograms with a wavelike structure.

### c. Generalization to 51-member forecast ensembles

As before, postprocessing outperforms the EPS forecasts and results in calibrated and accurate forecasts (cf. Table 5 and Fig. 2). Notably, all models have been trained purely on 11-member reforecasts and are not fine-tuned to the 51-member forecast ensembles. The CRPS scores are similar with almost identical values for all models, except EMOS, for all lead times. The ST models again perform the best for the shortest lead time. For the DRN forecasts, we find that the ensemble-based networks tend to reduce the PI length, as it is smaller for all cases except for lead time 120 h. The corresponding PI coverages are closely connected to the length of the PIs and indicate that the PIs are too large for most postprocessing models, as the observed coverages are above the nominal level.

The calibration of the methods is not as good as in the other case studies, as indicated by the PIT histograms in Fig. 2, which may be a consequence of the large learning rate used in

training the models (cf. supplemental materials). All BQN forecasts have problems in the tails, where the lower and upper bound are too extreme, such that insufficiently many observations fall into the outer bins. DRN yields similar results as for the reforecast data with too heavy-tailed forecast distribution, as indicated by the least frequent last bin. The differences between the methods themselves are again minor. Still, all postprocessing methods generate reasonably well-calibrated forecasts. Overall, the ensemble-based models result in state-of-the-art performance for generalization on 51-member forecasts or offer advantages over the summary-based benchmark methods.

## 7. Analysis of predictor importance

We analyze how the different model types distill relevant information out of the ensemble predictors. For this, we propose an ensemble-oriented PFI analysis to assess which distribution properties of the ensemble-valued predictors have the most effect on the final prediction. In its original form, PFI (e.g., Breiman 2001; Rasp and Lerch 2018; Schulz and Lerch 2022b) is used to assign relevance scores to scalar-valued predictors by randomly shuffling the values of a single predictor across the dataset. While the idea of shuffling predictor

TABLE 5. Mean CRPS (K), PI length (K), and PI coverage (%) of the postprocessing methods for the different lead times for EUPPBench forecast data (51-member ensemble, years 2017–18). Recall that the nominal level of the PIs is approximately 96.15%. The best-performing models (w.r.t. CRPS) are marked in bold.

Lead time	24 h			72 h			120 h		
	Method	CRPS	PI length	PI coverage	CRPS	PI length	PI coverage	CRPS	PI length
EPS	1.21	2.65	57.54	1.18	4.71	74.78	1.38	7.14	83.26
EMOS	0.79	6.31	96.26	0.90	7.74	97.49	1.16	9.92	97.47
BQN	0.64	4.32	94.13	<b>0.80</b>	6.52	97.23	1.13	9.18	97.58
DRN	0.64	5.48	97.92	<b>0.80</b>	7.21	98.37	1.13	9.58	98.28
ED-BQN	0.64	4.74	96.30	0.81	6.49	97.42	<b>1.12</b>	8.81	97.15
ED-DRN	0.64	5.31	97.62	0.81	7.09	98.19	<b>1.12</b>	9.61	97.90
ST-BQN	<b>0.62</b>	4.61	95.96	<b>0.80</b>	6.18	96.31	1.13	8.68	96.05
ST-DRN	<b>0.62</b>	5.10	97.40	0.81	6.88	97.55	1.13	9.43	97.11

samples translates identically from scalar-valued to ensemble-valued predictors, ensemble predictors possess internal degrees of freedom (DOFs), such as ensemble mean and ensemble range, which may affect the prediction differently. In addition to ensemble-internal DOFs, the perturbed predictor ensemble is embedded in the context of the remaining multivariate ensemble predictors, such that covariances, copulas or the rank order of the ensemble members may carry information. To account for such effects, we introduce a conditional permutation strategy that singles out the effects of different ensemble properties.

*a. Importance of the ensemble information*

Following the notation of section 3, let  $g: [\mathbb{R}^p]_M \rightarrow \Theta$  denote a postprocessing system that translates a raw ensemble forecast  $X = \{\mathbf{x}_1, \dots, \mathbf{x}_M\} \in [\mathbb{R}^p]_M$  into a postprocessed distribution descriptor  $\theta \in \Theta$ , and let for each member forecast  $\mathbf{x}_m$  the forecast value of the  $i$ th predictor be  $x_m^{(i)} \in \mathbb{R}$  (for  $i = 1, \dots, p$ ). Given a test dataset consisting of known raw forecast-observation pairs, as well as a (negatively oriented) accuracy measure  $\bar{S}$ , such as the expected CRPS, we write  $\bar{S}[g]$  to denote the accuracy score of  $g$  on the test data. In this notation, the relative PFI, as used in Schulz and Lerch (2022b), can be written as

$$\Delta_0(P) := \frac{\bar{S}[g \circ P] - \bar{S}[g]}{\bar{S}[g]}, \quad (2)$$

wherein  $P$  indicates a perturbation operator that alters parts of the predictor data, and the  $\circ$  symbol denotes function composition. For the classical PFI, we denote the permutation operator as  $\Pi_\pi^{(i)}$ , which shuffles the  $i$ th predictor channel of the raw ensembles according to a permutation  $\pi$  of the dataset.

For ensemble-valued predictors, we consider two generalizations of this operator. We refer to these as the fully random permutation  $\Pi_\pi^{(i)}$  and the rank-aware random permutation  $\tilde{\Pi}_\pi^{(i)}$ . The former acts as a direct analog of the scalar-valued permutation case, that is, given a dataset  $\mathcal{D} := \{(X(t), y(t)) : t = 1, \dots, T\} \subseteq [\mathbb{R}^p]_M \times \mathbb{R}$  of forecast-observation pairs, it replaces for all  $m = 1, \dots, M$  the values  $x_m^{(i)}$  of the ensemble  $X(t)$  with arbitrary values  $x_{m'}^{(i)}$ ,  $1 \leq m' \leq M$ , from the ensemble  $X(\pi(t))$ , without replacement. Thus, it destroys all information of the original ensemble. The latter ranks the member values  $x_m^{(i)}$  in  $X(t)$  and replaces them with values  $x_{m'}^{(i)}$  from  $X(\pi(t))$ , where  $m'$  are chosen such that all members are used exactly once and the perturbed ensemble possesses the same ranking order as the original one. It thus preserves the ordering of the perturbed predictors in the context of the remaining predictors. In practice, we note that the differences in feature importance for both variants are very minor, such that we select only the rank-aware variant for further analysis.

To probe the importance of ensemble-internal DOFs, we consider additional perturbation operators, which rely on conditional shuffling of the ensemble predictors. For this, let  $s: [\mathbb{R}]_M \rightarrow \mathbb{R}$  be a summary function, which translates an ensemble of scalar predictor values into a real-valued summary statistic, such as ensemble mean or standard deviation. Then an  $s$ -conditional shuffling operator  $\Pi_{\{\pi_b\}_s}^{(i)}$  is defined as follows. For all raw predictions  $X(t)$  in the dataset, the predictor ensemble for the  $i$ th predictor,  $X^{(i)}(t) = \{\mathbf{x}_m^{(i)} : \mathbf{x}_m \in X(t)\}$ , is extracted and summary statistics  $s(X^{(i)}(t))$  are computed. The

observed summary statistics are ranked from 1 to  $T$  and the corresponding ensembles  $X(t)$  are distributed into  $B \in \mathbb{N}$  evenly spaced bins, according to these ranks. For each bin  $b$ ,  $0 \leq b < B$ , a permutation  $\pi_b$  is sampled randomly and the values of the  $i$ th predictor are shuffled binwise according to these permutations. For suitably sized bins, the shuffling preserves information about  $s$  and erases information about other DOFs, thereby ensuring that each of the bins contains an approximately equal number of samples, independent of the details of the predictor distribution. Empirically,  $B = 100$  bins yielded a good balance between information preservation and randomization. Results for larger and smaller bin sizes were qualitatively similar. Note that for predictors in which certain values appear with large multiplicities, such as zero in censored variables like precipitation, the ranking is computed on the unique values of the summary statistics. This enforces a small amount of variation even in bins with degenerate values. In analogy to the rank-aware (unconditional) shuffling, the rank-aware  $s$ -conditional shuffling is denoted as  $\Pi_{\{\pi_b\}_s}^{(i)}$ . For the conditional PFI analysis, we suggest the computation of importance ratios,

$$\chi(P|R) := \frac{\bar{S}[g \circ P] - \bar{S}[g]}{\bar{S}[g \circ R] - \bar{S}[g]}, \quad (3)$$

which measure the fraction of skill restored (or destroyed) by applying a shuffling operation  $P$  instead of a reference operation  $R$ . The ratios of interest are  $\chi(\tilde{\Pi}_{\{\pi_b\}_s}^{(i)}, \tilde{\Pi}_\pi^{(i)})$ , which measure how much of the prediction skill deficit due to randomized shuffling of predictor  $i$  is restored by preserving information about the summary statistic  $s$ . In absence of sampling errors due to finite data,  $\chi(\tilde{\Pi}_{\{\pi_b\}_s}^{(i)}, \tilde{\Pi}_\pi^{(i)})$  yields values between 0 and 1, with 0 indicating uninformative summary statistics, and 1 suggesting that knowledge of  $s$  is sufficient to restore the original model skill entirely. Empirically, we find that the theoretical bounds are preserved well for predictors with sufficiently large PFI.

*b. Results*

We compute PFI scores  $\Delta_0(\Pi_\pi^{(i)})$  for all ensemble predictors and model variants. Figure 3 depicts a selection of the PFI scores of the most important ensemble-valued predictors in both tasks. A figure with all ensemble-valued predictors is shown in the supplemental materials. Scalar-valued predictors (cf. section 3d for the terminology) are omitted to simplify comparisons with the conditional importance measures. The bar charts show the median of ratios obtained from 20 separate model runs, which have been evaluated independently, and the error bars indicate the interquartile range (IQR).

The accuracy of the wind gust models is dominated by VMAX-10M and supplemented by additional predictors with lower importance. Temperature-like predictors obtain similar or higher scores than, for example, winds at the 850- and 950-hPa pressure levels. Note that for each lead time, the importance highlights different temperature predictors, which may be attributed to the diurnal cycle. Similar arguments can explain the increasing importance of short-wavelength radiation balance at the surface (ASOB-S) with increasing lead time. In a direct comparison of the model variants, we find that the differences between BQN-type and DRN-type models are very



FIG. 3. (top) Permutation feature importance for summary-based networks and (bottom) permutation-invariant models for EUPPBench and wind gust postprocessing. Bar heights indicate the median of an ensemble of 20 separate models, the error bars depict the IQR. Predictors named “ens” in the top panels correspond to the primary predictors t2m and VMAX-10M, respectively. The suffix “sd” indicates the ensemble standard deviation of the predictor.

minor. However, ED-type models attribute higher importance to the most relevant predictors (VMAX-10M, T1000, T-2M), whereas ST-type models distribute the importance more evenly and use more diverse predictor information.

In the EUPPBench case, the models focus mainly on temperature-like predictors as well as surface radiation balances. Notably, for the summary-based models, mn2t6 and mx2t6 tend to be more important than the primary predictor t2m up to lead time 72 h. Since the diurnal cycle does not cause variations between the lead times here, differences in the predictor utilization must be due to the increasing uncertainty at longer lead times. The ensemble-based models rely relatively more strongly on the t2m predictor for the shorter

lead time, whereas for longer lead times, the information utilization is more diverse. Qualitative differences between ED- and ST-type models are observed with respect to the humidity-related predictors tcw and tcwv. Only ST models recognize the value in these predictors, which may explain in parts the different generalization properties of ED and ST models on the EUPPBench reforecast and forecast datasets.

Figures 4 and 5 investigate the importance of ensemble-internal DOFs of selected ensemble predictors for the permutation-invariant model architectures. For both datasets, we choose a set of representative high-importance predictors and display the DOF importance for the ensemble-based models. Corresponding figures for the remaining predictors are shown in the



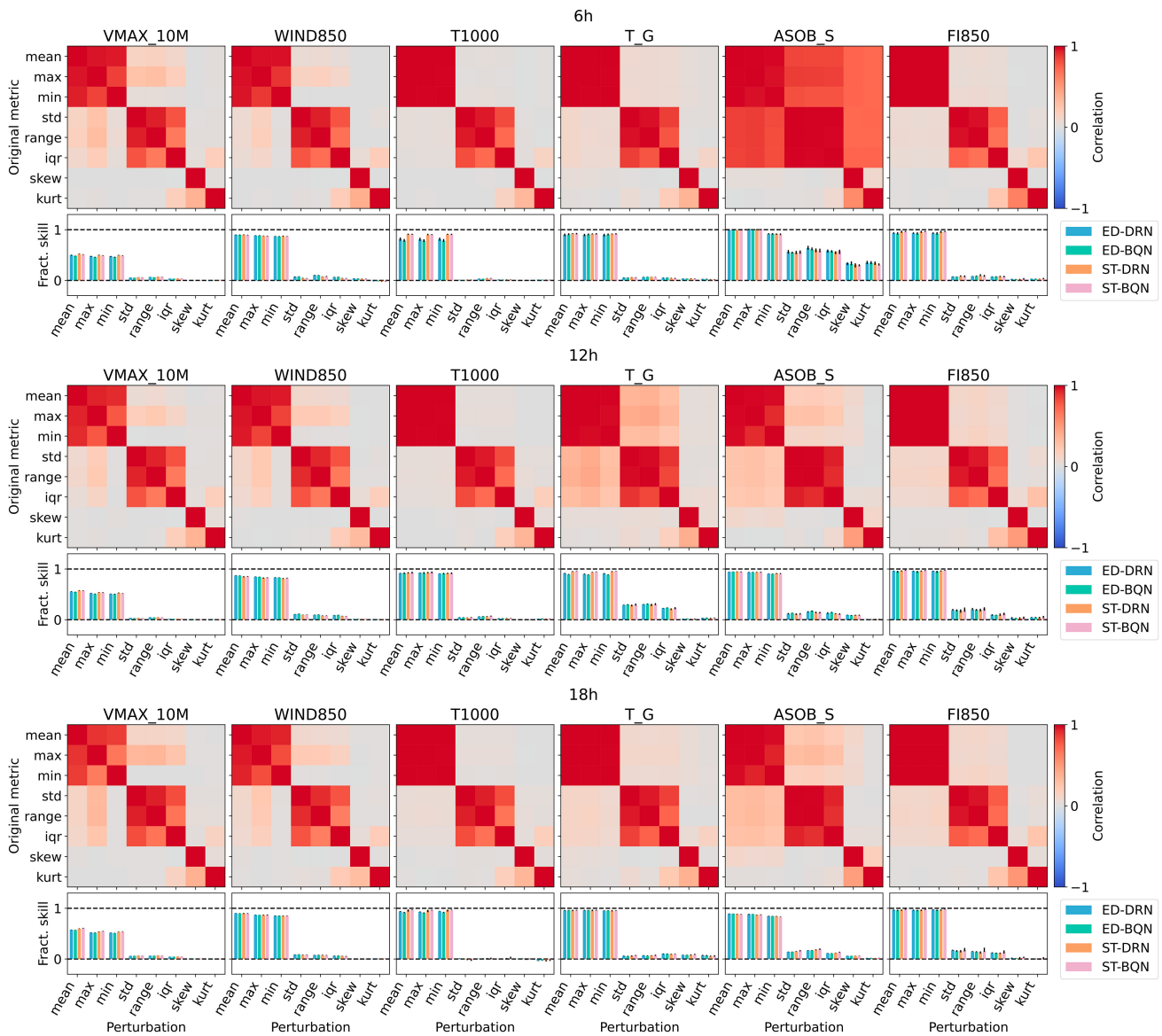


FIG. 4. Importance of ensemble-internal DOFs for wind gust postprocessing. Bar charts show importance ratios  $\chi(\tilde{\Pi}_{\{\pi_b\}_s}^{(i)}, \tilde{\Pi}_{\pi}^{(i)})$  for selected summary statistics  $s$ , and heat maps display the Spearman rank correlation between the summary statistics computed on the original dataset and the same statistics after conditional shuffling with respect to the different summary statistics. Bar heights indicate the median of an ensemble of 20 separate models, and the error bars depict the IQR.

supplemental materials. For all predictors and lead times, we compute importance ratios  $\chi(\tilde{\Pi}_{\{\pi_b\}_s}^{(i)}, \tilde{\Pi}_{\pi}^{(i)})$  for a selection of commonly used ensemble summary statistics. Specifically, we consider the ensemble mean as a proxy for the location of the distribution, ensemble maximum and minimum to assess the impact of extreme values, standard deviation, IQR, and full range (difference between maximum and minimum) to quantify the scale of the distribution, as well as skewness and kurtosis as higher-order summary statistics. Due to the pairwise similarity of some of the measures, it is to be expected that conditional shuffling with respect to one of the measures preserves information about others. To assess the information overlap between shuffling patterns with different reference statistics, Spearman rank correlations are computed between

the shuffled statistics and the original statistics. The resulting correlation matrices illustrate how accurately the rank order for one statistic is preserved if the data is conditionally shuffled with respect to another. Rank correlations are chosen to minimize the effect of the marginal distribution of the respective statistics values, since these may vary considerably between different predictors and summary statistics. The results are depicted as heat maps in Figs. 4 and 5.

For wind gust postprocessing (Fig. 4), the importance ratios suggest in many cases that most of the predictor information can be restored by conditioning the shuffling procedure on the ensemble mean. This is the case for T-1000, T-G, and FI850. The interaction plots suggest that the mean conditioning preserves information about extrema to a high degree,

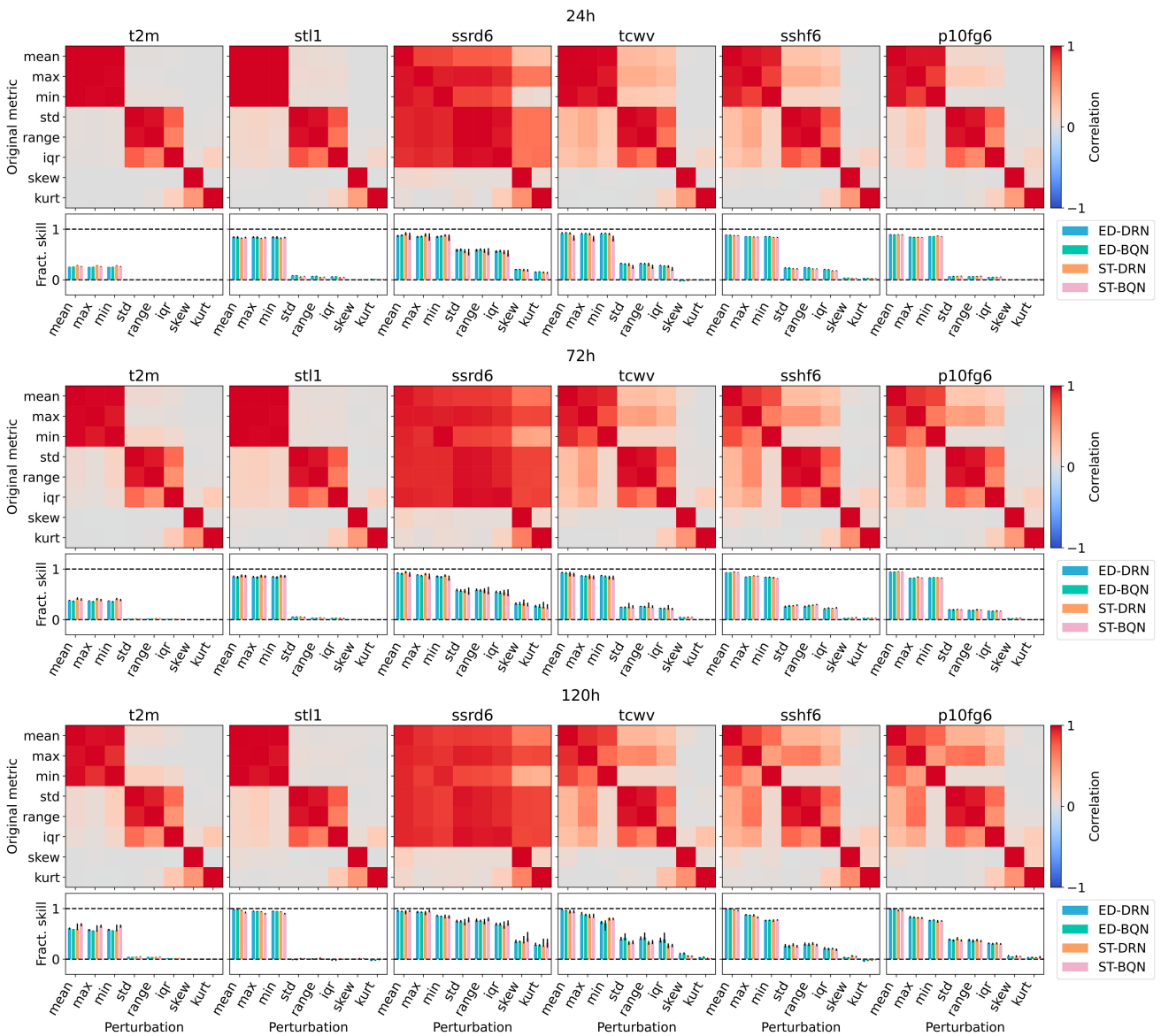


FIG. 5. As in Fig. 4, but for temperature postprocessing.

whereas ensemble range and higher-order statistic information are randomized. These findings are supported by observations in [Schulz and Lerch \(2022b\)](#), who note that omitting the standard deviation of the auxiliary ensemble predictors helps to improve the quality of the network predictions. Larger importance ratios of the scale-related and higher-order DOFs are observed, for example, for T-G at lead time 12 h and ASOB-S at lead time 6 h. However, these cases coincide with increased correlations between the respective perturbation patterns and the location-preserving perturbations, which show fractional skill ratios close to unity. This may be seen as an indication that the relevance of the remaining DOFs must in part be attributed to information overlap with the location-related DOFs. Note here that the strong information overlap between locationlike and scalelike metrics for ASOB-S predictors at 6-h lead time is again an artifact due to the diurnal cycle. At 6-h lead time, a substantial fraction of the ASOB-S

predictor ensembles falls to zero mean and no variance due to the lack of solar irradiation, which impacts the correlation values as well as the effectiveness of perturbations. WIND850 is a corner case, in which the mean conditioning restores substantial amounts of the model skill but fails to restore the unperturbed performance completely. This indicates that, while the ensemble mean is an important predictor, the remaining DOFs deliver complementary information that modulates the interpretation of the mean value. VMAX-10M, being the primary predictor, constitutes the only example for which no single predictor is sufficient to restore the unperturbed model skill, thus indicating that both ED- and ST-type models learn to attend to the details of the ensemble distribution.

In surface temperature postprocessing, t2m is the primary predictor and displays similar characteristics as VMAX-10M in the wind-gust study. The mean-conditional shuffling of t2m tends to become more effective in restoring the model skill

with increasing lead time. This may be due to the decreasing reliability of the EPS-based predictor ensembles with increasing lead time. Similar patterns are observed also in the remaining predictors. While the model skill cannot be restored with mean-only conditioning for 24-h lead time, the mean appears to become more informative for longer lead times. The radiation parameter `ssrd6` sticks out visually with high correlations between location-related predictors, which occurs due to the same reasons as for the `ASOB-S` parameter discussed before.

## 8. Discussion and conclusions

We have introduced permutation-invariant NN architectures for postprocessing ensemble forecasts by selecting two exemplary model families and adapting them to postprocessing. In two case studies, using datasets for wind gusts and surface temperature postprocessing, we evaluated the model performance and compared the permutation-invariant models against benchmark models from prior work. Our results show that permutation-invariant postprocessing networks achieve state-of-the-art performance in both applications. All permutation-invariant architectures outperform both the raw ensemble forecast and conventional postprocessing via EMOS by a large margin, but systematic differences between the (more complex) permutation-invariant models and existing NN-based solutions are very minor and can mostly be attributed to differences in the distribution parameterization. Qualitatively similar results were observed for extreme events in both case studies but are not shown explicitly in the interest of brevity.

Based on a subsequent assessment of the permutation importance of ensemble-internal DOFs, we have seen that for many auxiliary ensemble predictors, preserving information about the ensemble mean is sufficient to maintain almost the complete information about the postprocessing target, while more detailed information is required about the primary predictors. These findings are consistent with prior work and are more comprehensive due to the larger variety of summary statistics considered in the analysis.

A striking advantage of the permutation-invariant models lies in the generality of the approach, that is, the models possess the flexibility of attending to the important features in the predictor ensembles and the capability of identifying those during training (as shown in our feature analysis). As the added flexibility comes with a surplus of computational complexity, the benefits of the respective methods should be weighed carefully. In operational settings, it may be reasonable to consider permutation-invariant models, as proposed here, as a tool for identifying relevant aspects of the input data. The gained knowledge can then be used for data reduction and to train reduced models with a more favorable accuracy–complexity trade-off.

Despite this, the apparent similarity between the performance of the ensemble-based and summary-based models

remains baffling and requires further clarification. Supposing capable ensemble predictions, it seems reasonable, from a meteorological perspective, to expect that postprocessing models that operate on the entire ensemble can learn more complex patterns and relationships than models that operate on simple summary statistics. The lack of substantial improvements, as seen in this study, admits different explanations. One possibility would be that the available datasets are insufficient to establish statistically relevant connections between higher-order ensemble-internal patterns and the predicted variables. Problems could arise, for example, due to insufficient sample counts of the overall datasets or due to ensemble sizes being too low to provide reliable representations of the forecast distribution. Yet another reason could lie in the fact that the generation mechanisms underlying the NWP ensemble forecasts fail to achieve meaningful representations of such higher-order distribution information, which would raise follow-up questions regarding the design of future ensemble prediction systems. Given the impact and potential implications of the latter alternative, future work should examine the information content of raw ensemble predictions in more detail. The proposed permutation-invariant model architectures may help to achieve this, for example, by conducting postprocessing experiments with dynamical toy systems that are cheap to simulate and simple to understand, such that large datasets can be generated and evidence for both hypotheses can be distinguished.

*Acknowledgments.* This research was funded by the sub-projects B5 and C5 of the Transregional Collaborative Research Center SFB/TRR 165 “Waves to Weather” ([www.wavestoweather.de](http://www.wavestoweather.de)) funded by the German Research Foundation (DFG). Sebastian Lerch gratefully acknowledges support by the Vector Stiftung through the Young Investigator Group “Artificial Intelligence for Probabilistic Weather Forecasting.” We thank two anonymous reviewers for their constructive comments.

*Data availability statement.* The case study on surface temperature postprocessing is based on the EUPPBench dataset, which is publicly available. See [Demaeyer et al. \(2023\)](#) for details. The wind gust dataset is proprietary but can be obtained from the DWD for research purposes. Code with implementations of all methods is publicly available ([Höhlein 2023](#)).

## APPENDIX A

### Description of Predictors

The descriptions of the ensemble-valued predictor variables used in both case studies are shown in [Tables A1](#) and [A2](#) for wind gust and surface temperature postprocessing, respectively. The predictors listed in [Table A3](#) are not ensemble-valued and are used equally in both case studies.

TABLE A1. Description of meteorological parameters for wind-gust postprocessing (cf. [Schulz and Lerch 2022b](#)). Target variable: wind speed of gust (observations). Primary predictor: VMAX-10 m (ensemble forecast).

Short name	Units	Full name	Levels
VMAX	$\text{m s}^{-1}$	Max wind, i.e., wind gusts	10 m
$U$	$\text{m s}^{-1}$	$U$ component of wind	10 m, 1000 hPa, 950 hPa, 850 hPa, 700 hPa, 500 hPa
$V$	$\text{m s}^{-1}$	$V$ component of wind	10 m, 1000 hPa, 950 hPa, 850 hPa, 700 hPa, 500 hPa
WIND	$\text{m s}^{-1}$	Wind speed, derived from $U$ and $V$ via $\sqrt{U^2 + V^2}$	10 m, 1000 hPa, 950 hPa, 850 hPa, 700 hPa, 500 hPa
OMEGA	$\text{Pa s}^{-1}$	Vertical velocity (pressure)	1000, 950, 850, 700, 500 hPa
$T$	K	Temperature	Ground-level, 2 m, 1000 hPa, 950 hPa, 850 hPa, 700 hPa, 500 hPa
T-D	K	Dewpoint temperature	2 m
RELHUM	%	Relative humidity	1000, 950, 850, 700, 500 hPa
TOT-PREC	$\text{kg m}^{-2}$	Total precipitation (accumulated)	—
RAIN-GSP	$\text{kg m}^{-2}$	Large-scale rain (accumulated)	—
SNOW-GSP	$\text{kg m}^{-2}$	Large-scale snowfall—water equivalent (accumulated)	—
W-SNOW	$\text{kg m}^{-2}$	Snow depth water equivalent	—
W-SO	$\text{kg m}^{-2}$	Column integrated soil moisture	Multilayers: 1, 2, 6, 18, 54
CLC	%	Cloud cover	T: total; low (L): soil to 800 hPa; middle (M): 800 to 400 hPa; high (H): 400 to 0 hPa
HBAS-SC	m	Cloud base above mean sea level, shallow connection	—
HTOP-SC	m	Cloud top above mean sea level, shallow connection	—
ASOB-S	$\text{W m}^{-2}$	Net shortwave radiation flux	Surface
ATHB-S	$\text{W m}^{-2}$	Net longwave radiation flux (m)	Surface
ALB-RAD	%	Albedo (in shortwave)	—
PMSL	Pa	Pressure reduced to mean sea level	—
FI	$\text{m}^2 \text{s}^{-2}$	Geopotential	1000, 950, 850, 700, 500 hPa

TABLE A2. Description of meteorological parameters for surface temperature postprocessing (EUPPBench, cf. [Demaeyer et al. 2023](#)). Target variable: t2m (observations). Primary predictor: t2m (ensemble forecast).

Short name	Units	Full name	Levels
$t$	K	Temperature	2 m, 850 hPa
mx2t6	K	Max temperature (6 h preceding)	2 m
mn2t6	K	Min temperature (6 h preceding)	2 m
$z$	$\text{m}^2 \text{s}^{-2}$	Geopotential	500 hPa
$u$	$\text{m s}^{-1}$	$U$ component of wind	10 m, 100 m, 700 hPa
$v$	$\text{m s}^{-1}$	$V$ component of wind	10 m, 100 m, 700 hPa
p10fg6	$\text{m s}^{-1}$	Max wind gust in the last 6 h	10 m
$q$	$\text{kg kg}^{-1}$	Specific humidity	700 hPa
$r$	%	Relative humidity	850 hPa
cape	$\text{J kg}^{-1}$	Convective available potential energy	—
cin	$\text{J kg}^{-1}$	Convective inhibition	—
tp6	m	Total precipitation (6 h preceding)	—
cp6	m	Convective precipitation (6 h preceding)	—
tcw	$\text{kg m}^{-2}$	Total column water	—
tcwv	$\text{kg m}^{-2}$	Total column water vapor	—
tcc	0–1	Total cloud cover	—
vis	m	Visibility	—
sshf6	$\text{J m}^{-2}$	Surface sensible heat flux (6 h preceding)	—
slhf6	$\text{J m}^{-2}$	Surface latent heat flux (6 h preceding)	—
ssr6	$\text{J m}^{-2}$	Surface net shortwave (solar) radiation (6 h preceding)	—
ssrd6	$\text{J m}^{-2}$	Surface net shortwave (solar) radiation downward (6 h preceding)	—
str6	$\text{J m}^{-2}$	Surface net longwave (thermal) radiation (6 h preceding)	—
strd6	$\text{J m}^{-2}$	Surface net longwave (thermal) radiation downward (6 h preceding)	—
swv	$\text{m}^3 \text{m}^{-3}$	Volumetric soil water	11: 0–7 cm
sd	m	Snow depth—water equivalent	—
st	K	Soil temperature	11: 0–7 cm

TABLE A3. Auxiliary predictors for both datasets (cf. [Schulz and Lerch 2022b](#)).

Predictor	Type	Description
yday	Temporal	Cosine transformed day of the year
lat	Spatial	Latitude of the station
lon	Spatial	Longitude of the station
alt	Spatial	Altitude of the station
orog	Spatial	Difference of station altitude and model surface height of nearest grid point
loc-bias	Spatial	Mean bias of ensemble forecasts, computed from the training data
loc-cover	Spatial	Mean coverage of ensemble forecasts, computed from the training data

## APPENDIX B

## Model Hyperparameters

Table B1 displays the hyperparameter settings for all model configurations used in the experiments. For details about the hyperparameter tuning process, we refer to the supplemental material.

TABLE B1. Hyperparameters for model experiments.

Method	Parameter	Settings					
		Wind gusts			EUPPBench		
		6 h	12 h	18 h	24 h	72 h	120 h
Common	Optimizer	Adam					
	Batch size	64 for DRN and BQN, 128 else					
	Hidden layers	2 hidden layers per MLP, 3 transformer blocks					
	Dimension of station embedding	10					
	Activation (at output layer)	Softplus (softplus)					
	DRN/EMOS distribution family	Truncated logistic					
	Training epochs	150			250		
DRN	Channels (first layer/second layer)	64/32			64/32		
	Learning rate	$5 \times 10^{-4}$		$3 \times 10^{-3}$	$3 \times 10^{-3}$	$4 \times 10^{-4}$	
	Patience	10			24		
BQN	Channels (first layer/second layer)	48/24			48/24		
	Polynomial degree	12			12		
	Learning rate	$5 \times 10^{-4}$		$4 \times 10^{-3}$	$4 \times 10^{-3}$	$3 \times 10^{-4}$	
ED-DRN	Channels (encoder)	64			64		
	Channels (decoder)	64			48		
	Dropout (encoder)	0.02			0.05		
ED-BQN	Dropout (decoder)	0.00			0.05	0.05	0.10
	Learning rate	$4 \times 10^{-4}$		$2 \times 10^{-4}$	$2 \times 10^{-4}$	$1 \times 10^{-4}$	
	Patience	24			20		
	Channels (encoder)	64			64		
	Channels (decoder)	64			48		
ST-DRN	Dropout (encoder)	0.00			0.05	0.10	0.10
	Dropout (decoder)	0.00			0.05	0.10	0.10
	Polynomial degree	12			8		
	Learning rate	$2 \times 10^{-4}$		$5 \times 10^{-4}$	$2 \times 10^{-4}$	$2 \times 10^{-4}$	$2 \times 10^{-4}$
	Patience	24			20		
ST-BQN	Channels (transformer)	64			64		
	Channels (decoder)	64			48		
	Dropout (transformer)	0.02			0.00		
	Dropout (decoder)	0.05			0.00		
	Learning rate	$2 \times 10^{-4}$	$1 \times 10^{-4}$	$5 \times 10^{-4}$	$2 \times 10^{-4}$	$1 \times 10^{-4}$	$5 \times 10^{-4}$
ST-BQN	Patience	24			24		
	Channels (transformer)	64			48		
	Channels (decoder)	64			48		
	Dropout (transformer)	0.10			0.20	0.25	0.20
	Dropout (decoder)	0.00			0.05	0.10	0.15
	Polynomial degree	12			8		
	Learning rate	$1 \times 10^{-4}$				$5 \times 10^{-5}$	
	Patience	24			24	15	20

## REFERENCES

- Baldauf, M., A. Seifert, J. Förstner, D. Majewski, M. Raschendorfer, and T. Reinhardt, 2011: Operational convective-scale numerical weather prediction with the COSMO model: Description and sensitivities. *Mon. Wea. Rev.*, **139**, 3887–3905, <https://doi.org/10.1175/MWR-D-10-05013.1>.
- Ben-Bouallegue, Z., J. A. Weyn, M. C. A. Clare, J. Dramsch, P. Dueben, and M. Chantry, 2023: Improving medium-range ensemble weather forecasts with hierarchical ensemble transformers.

- arXiv, 2303.17195v3, <https://doi.org/10.48550/arXiv.2303.17195>.
- Breiman, L., 2001: Random forests. *Mach. Learn.*, **45**, 5–32, <https://doi.org/10.1023/A:1010933404324>.
- Bremnes, J. B., 2020: Ensemble postprocessing using quantile function regression based on neural networks and Bernstein polynomials. *Mon. Wea. Rev.*, **148**, 403–414, <https://doi.org/10.1175/MWR-D-19-0227.1>.
- Burkart, N., and M. F. Huber, 2021: A survey on the explainability of supervised machine learning. *J. Artif. Intell. Res.*, **70**, 245–317, <https://doi.org/10.1613/jair.1.12228>.
- Chen, J., T. Janke, F. Steinke, and S. Lerch, 2022: Generative machine learning methods for multivariate ensemble post-processing. arXiv, 2211.01345v1, <https://doi.org/10.48550/arXiv.2211.01345>.
- Dai, Y., and S. Hemri, 2021: Spatially coherent postprocessing of cloud cover ensemble forecasts. *Mon. Wea. Rev.*, **149**, 3923–3937, <https://doi.org/10.1175/MWR-D-21-0046.1>.
- Demaeyer, J., and Coauthors, 2023: The EUPPBench postprocessing benchmark dataset v1.0. *Earth Syst. Sci. Data*, **15**, 2635–2653, <https://doi.org/10.5194/essd-15-2635-2023>.
- ECMWF, 2013: CliMetLab. GitHub, <https://github.com/ecmwf/climetlab>.
- Edwards, H., and A. Storkey, 2016: Towards a neural statistician. arXiv, 1606.02185v2, <https://doi.org/10.48550/arXiv.1606.02185>.
- Farokhmanesh, F., K. Höhle, and R. Westermann, 2023: Deep learning-based parameter transfer in meteorological data. *Artif. Intell. Earth Syst.*, **2**, e220024, <https://doi.org/10.1175/AIES-D-22-0024.1>.
- Finn, T. S., 2021: Self-attentive ensemble transformer: Representing ensemble interactions in neural networks for earth system models. arXiv, 2106.13924v2, <https://doi.org/10.48550/arXiv.2106.13924>.
- Gneiting, T., and A. E. Raftery, 2007: Strictly proper scoring rules, prediction, and estimation. *J. Amer. Stat. Assoc.*, **102**, 359–378, <https://doi.org/10.1198/01621450600001437>.
- , and R. Ranjan, 2011: Comparing density forecasts using threshold- and quantile-weighted scoring rules. *J. Bus. Econ. Stat.*, **29**, 411–422, <https://doi.org/10.1198/jbes.2010.08110>.
- , and M. Katzfuss, 2014: Probabilistic forecasting. *Annu. Rev. Stat. Appl.*, **1**, 125–151, <https://doi.org/10.1146/annurev-statistics-062713-085831>.
- , A. E. Raftery, A. H. Westveld III, and T. Goldman, 2005: Calibrated probabilistic forecasting using ensemble model output statistics and minimum CRPS estimation. *Mon. Wea. Rev.*, **133**, 1098–1118, <https://doi.org/10.1175/MWR2904.1>.
- , F. Balabdaoui, and A. E. Raftery, 2007: Probabilistic forecasts, calibration and sharpness. *J. Roy. Stat. Soc.*, **69B**, 243–268, <https://doi.org/10.1111/j.1467-9868.2007.00587.x>.
- 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. *Philos. Trans. Roy. Soc.*, **A379**, 20200092, <https://doi.org/10.1098/rsta.2020.0092>.
- Guidotti, R., A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, 2018: A survey of methods for explaining black box models. *ACM Comput. Surv.*, **51** (5), 1–42, <https://doi.org/10.1145/3236009>.
- 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. *Philos. Trans. Roy. Soc.*, **A379**, 20200091, <https://doi.org/10.1098/rsta.2020.0091>.
- Höhlein, K., 2023: Code “postprocessing of ensemble weather forecasts using permutation-invariant neural networks.” Zenodo, <https://doi.org/10.5281/zenodo.8329345>.
- , M. Kern, T. Hewson, and R. Westermann, 2020: A comparative study of convolutional neural network models for wind field downscaling. *Meteor. Appl.*, **27**, e1961, <https://doi.org/10.1002/met.1961>.
- Horat, N., and S. Lerch, 2023: Deep learning for post-processing global probabilistic forecasts on sub-seasonal time scales. arXiv, 2306.15956v1, <https://doi.org/10.48550/arXiv.2306.15956>.
- Jordan, A., F. Krüger, and S. Lerch, 2019: Evaluating probabilistic forecasts with scoringRules. *J. Stat. Software*, **90** (12), 1–37, <https://doi.org/10.18637/jss.v090.i12>.
- Khan, S., M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, 2022: Transformers in vision: A survey. *ACM Comput. Surv.*, **54** (10s), 1–41, <https://doi.org/10.1145/3505244>.
- Koenker, R., and G. Bassett Jr., 1978: Regression quantiles. *Econometrica*, **46**, 33–50, <https://doi.org/10.2307/1913643>.
- Labe, Z. M., and E. A. Barnes, 2021: Detecting climate signals using explainable AI with single-forcing large ensembles. *J. Adv. Model. Earth Syst.*, **13**, e2021MS002464, <https://doi.org/10.1029/2021MS002464>.
- Lee, J., Y. Lee, J. Kim, A. Kosiorok, S. Choi, and Y. W. Teh, 2019: Set transformer: A framework for attention-based permutation-invariant neural networks. *Proc. 36th Int. Conf. on Machine Learning*, Long Beach, CA, PMLR, 3744–3753.
- Lerch, S., and T. L. Thorarindottir, 2013: Comparison of non-homogeneous regression models for probabilistic wind speed forecasting. *Tellus*, **65A**, 21206, <https://doi.org/10.3402/tellusa.v65i0.21206>.
- Linaratos, P., V. Papastefanopoulos, and S. Kotsiantis, 2021: Explainable AI: A review of machine learning interpretability methods. *Entropy*, **23**, 18, <https://doi.org/10.3390/e23010018>.
- Lyle, C., M. van der Wilk, M. Kwiatkowska, Y. Gal, and B. Bloem-Reddy, 2020: On the benefits of invariance in neural networks. arXiv, 2005.00178v1, <https://doi.org/10.48550/arXiv.2005.00178>.
- Matheson, J. E., and R. L. Winkler, 1976: Scoring rules for continuous probability distributions. *Manage. Sci.*, **22**, 1087–1096, <https://doi.org/10.1287/mnsc.22.10.1087>.
- Messner, J. W., G. J. Mayr, and A. Zeileis, 2017: Nonhomogeneous boosting for predictor selection in ensemble postprocessing. *Mon. Wea. Rev.*, **145**, 137–147, <https://doi.org/10.1175/MWR-D-16-0088.1>.
- Mlakar, P., J. Merše, and J. F. Pucer, 2023: Ensemble weather forecast post-processing with a flexible probabilistic neural network approach. arXiv, 2303.17610v3, <https://doi.org/10.48550/arXiv.2303.17610>.
- Molnar, C., G. König, B. Bischl, and G. Casalicchio, 2024: Model-agnostic feature importance and effects with dependent features: A conditional subgroup approach. *Data Min. Knowl. Discovery*, <https://doi.org/10.1007/s10618-022-00901-9>, in press.
- Pantillon, F., S. Lerch, P. Knippertz, and U. Corsmeier, 2018: Forecasting wind gusts in winter storms using a calibrated convection-permitting ensemble. *Quart. J. Roy. Meteor. Soc.*, **144**, 1864–1881, <https://doi.org/10.1002/qj.3380>.
- Rasp, S., and S. Lerch, 2018: Neural networks for postprocessing ensemble weather forecasts. *Mon. Wea. Rev.*, **146**, 3885–3900, <https://doi.org/10.1175/MWR-D-18-0187.1>.
- Ravanbakhsh, S., J. Schneider, and B. Poczos, 2016: Deep learning with sets and point clouds. arXiv, 1611.04500v3, <https://doi.org/10.48550/arXiv.1611.04500>.

- Reichstein, M., G. Camps-Valls, B. Stevens, M. Jung, J. Denzler, N. Carvallhais, and Prabhat, 2019: Deep learning and process understanding for data-driven earth system science. *Nature*, **566**, 195–204, <https://doi.org/10.1038/s41586-019-0912-1>.
- Sannai, A., Y. Takai, and M. Cordonnier, 2019: Universal approximations of permutation invariant/equivariant functions by deep neural networks. arXiv, 1903.01939v3, <https://doi.org/10.48550/arXiv.1903.01939>.
- Scheuerer, M., 2014: Probabilistic quantitative precipitation forecasting using ensemble model output statistics. *Quart. J. Roy. Meteor. Soc.*, **140**, 1086–1096, <https://doi.org/10.1002/qj.2183>.
- , M. B. Switanek, R. P. Worsnop, and T. M. Hamill, 2020: Using artificial neural networks for generating probabilistic subseasonal precipitation forecasts over California. *Mon. Wea. Rev.*, **148**, 3489–3506, <https://doi.org/10.1175/MWR-D-20-0096.1>.
- Schulz, B., and S. Lerch, 2022a: Aggregating distribution forecasts from deep ensembles. arXiv, 2204.02291v1, <https://doi.org/10.48550/arXiv.2204.02291>.
- , and —, 2022b: Machine learning methods for postprocessing ensemble forecasts of wind gusts: A systematic comparison. *Mon. Wea. Rev.*, **150**, 235–257, <https://doi.org/10.1175/MWR-D-21-0150.1>.
- , M. E. Ayari, S. Lerch, and S. Baran, 2021: Post-processing numerical weather prediction ensembles for probabilistic solar irradiance forecasting. *Sol. Energy*, **220**, 1016–1031, <https://doi.org/10.1016/j.solener.2021.03.023>.
- Soelch, M., A. Akhundov, P. van der Smagt, and J. Bayer, 2019: On deep set learning and the choice of aggregations. *Artificial Neural Networks and Machine Learning—ICANN 2019: Theoretical Neural Computation*, I. Tetko et al., Eds., Lecture Notes in Computer Science, Vol. 11727, Springer, 444–457, [https://doi.org/10.1007/978-3-030-30487-4\\_35](https://doi.org/10.1007/978-3-030-30487-4_35).
- Strobl, C., A.-L. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis, 2008: Conditional variable importance for random forests. *BMC Bioinf.*, **9**, 307, <https://doi.org/10.1186/1471-2105-9-307>.
- Taillardat, M., O. Mestre, M. Zamo, and P. Naveau, 2016: Calibrated ensemble forecasts using quantile regression forests and ensemble model output statistics. *Mon. Wea. Rev.*, **144**, 2375–2393, <https://doi.org/10.1175/MWR-D-15-0260.1>.
- Vannitsem, S., D. S. Wilks, and J. W. Messner, 2018: *Statistical Postprocessing of Ensemble Forecasts*. Elsevier, 347 pp., <https://doi.org/10.1016/C2016-0-03244-8>.
- , and Coauthors, 2021: Statistical postprocessing for weather forecasts: Review, challenges, and avenues in a big data world. *Bull. Amer. Meteor. Soc.*, **102**, E681–E699, <https://doi.org/10.1175/BAMS-D-19-0308.1>.
- van Schaeybroeck, B., and S. Vannitsem, 2015: Ensemble post-processing using member-by-member approaches: Theoretical aspects. *Quart. J. Roy. Meteor. Soc.*, **141**, 807–818, <https://doi.org/10.1002/qj.2397>.
- Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, 2017: Attention is all you need. *Proc. 31st Int. Conf. on Neural Information Processing Systems*, Long Beach, CA, Association for Computing Machinery, 6000–6010, <https://dl.acm.org/doi/10.5555/3295222.3295349>.
- Veldkamp, S., K. Whan, S. Dirksen, and M. Schmeits, 2021: Statistical postprocessing of wind speed forecasts using convolutional neural networks. *Mon. Wea. Rev.*, **149**, 1141–1152, <https://doi.org/10.1175/MWR-D-20-0219.1>.
- Vinyals, O., S. Bengio, and M. Kudlur, 2015: Order matters: Sequence to sequence for sets. arXiv, 1511.06391v4, <https://doi.org/10.48550/arXiv.1511.06391>.
- Vogel, P., P. Knippertz, A. H. Fink, A. Schlueter, and T. Gneiting, 2018: Skill of global raw and postprocessed ensemble predictions of rainfall over northern tropical Africa. *Wea. Forecasting*, **33**, 369–388, <https://doi.org/10.1175/WAF-D-17-0127.1>.
- Zaheer, M., S. Kottur, S. Ravanbakhsh, B. Póczos, R. Salakhutdinov, and A. J. Smola, 2017: Deep sets. *Proc. 31st Int. Conf. on Neural Information Processing Systems*, Long Beach, CA, Association for Computing Machinery, 3394–3404, <https://dl.acm.org/doi/10.5555/3294996.3295098>.
- Zhang, Y., J. Hare, and A. Prügél-Bennett, 2019: FSPool: Learning set representations with featurewise sort pooling. arXiv, 1906.02795v4, <https://doi.org/10.48550/arXiv.1906.02795>.





# AMS

American Meteorological Society

## Supplemental Material

*Artificial Intelligence for the Earth Systems*

Postprocessing of Ensemble Weather Forecasts Using Permutation-Invariant Neural Networks

<https://doi.org/10.1175/AIES-D-23-0070.1>

© Copyright 2024 American Meteorological Society (AMS)

For permission to reuse any portion of this work, please contact [permissions@ametsoc.org](mailto:permissions@ametsoc.org). Any use of material in this work that is determined to be “fair use” under Section 107 of the U.S. Copyright Act (17 USC §107) or that satisfies the conditions specified in Section 108 of the U.S. Copyright Act (17 USC §108) does not require AMS’s permission. Republication, systematic reproduction, posting in electronic form, such as on a website or in a searchable database, or other uses of this material, except as exempted by the above statement, requires written permission or a license from AMS. All AMS journals and monograph publications are registered with the Copyright Clearance Center (<https://www.copyright.com>). Additional details are provided in the AMS Copyright Policy statement, available on the AMS website (<https://www.ametsoc.org/PUBSCopyrightPolicy>).

# Postprocessing of Ensemble Weather Forecasts Using Permutation-invariant Neural Networks: Supplementary Material

KEVIN HÖHLEIN,<sup>a</sup> BENEDIKT SCHULZ,<sup>b</sup> RÜDIGER WESTERMANN,<sup>a</sup> AND SEBASTIAN LERCH<sup>b,c</sup>

<sup>a</sup> *Technical University of Munich*

<sup>b</sup> *Karlsruhe Institute of Technology*

<sup>c</sup> *Heidelberg Institute for Theoretical Studies*

## 1. Hyperparameter selection

Appropriate tuning of the training- and architecture-related hyperparameters of the respective model classes is essential to achieve a fair comparison. In what follows, we detail the hyperparameter settings chosen for the respective model classes, as well as the methods that led to the decision. To find good sets of hyperparameters for all model variants, we follow the suggestions by Godbole et al. (2023) and conduct a multi-phase parameter search, consisting of randomized parameter space exploration, followed by automated tuning of the hyperparameters using Bayesian optimization, and ablations to avoid excessive complexity of the models.

### a. Model classes

As shown in Fig. 1, we impose a hierarchical classification on the model types for parameter tuning. On the highest level, we distinguish ensemble-based from summary-based models, which is the most severe differentiation, since models of both groups are trained on data with different information content. On the second level, we group the models by the architecture. This is relevant mainly for ensemble-based models, where encoder-decoder models process information differently from transformers. Following Schulz and Lerch (2022), all summary-based models use simple MLPs. The third level distinguishes with respect to posterior parameterization, i.e., DRN-type vs. BQN-type output parameterization. Note that all DRN models parameterize a truncated logistic distribution, both for wind-gust and temperature postprocessing. The fourth level is relevant only for ensemble-based encoder-decoder models and separates the models by the merging strategy used between member-wise encoder and decoder.

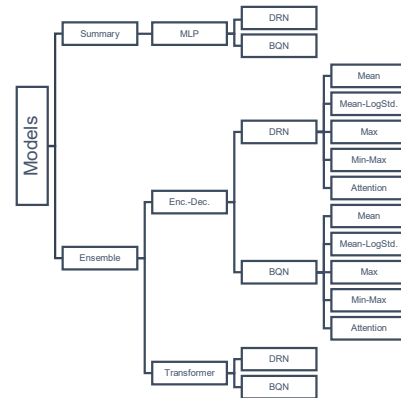


FIG. 1. Model hierarchy for parameter tuning. Models are grouped by 1) input type, 2) architecture, 3) posterior parameterization and 4) merging strategy (encoder-decoder models only).

### b. Initial exploration

For each model class, we execute a number of initial trial runs to gain an understanding of how different parameters affect the model performance. Performance was assessed by comparing average losses on the training and validation parts of the respective datasets. We found that the batch size for training can be chosen flexibly, as long as the remaining parameters are tuned accordingly. We saw that similar sets of training-related hyperparameters (learning rate, patience for early stopping, dropout rates) lead to different results 1) when predicting different lead times and 2) when changing architecture-related hyperparameters (number of layers, channels per layer) of the respective model classes. Especially encoder-decoder and transformer models with larger MLP components appeared to profit from dropout-based regularization. We also found that the dimension of the station embedding affects the performance of ensemble models. Starting from a default value of 10 (cf. Schulz and Lerch 2022), trials showed that smaller values improve the quality of single-model predic-

Corresponding author: Kevin Höhle, kevin.hoehlein@tum.de

tions but penalize the accuracy of the ensemble prediction, while larger values lead to overall reduced performance. The station embedding was therefore not considered in the subsequent parameter search. Similarly, the model depth was excluded early on as a hyper parameter since there was no indication that increasing the model depth to more than three layers (three attention blocks for set transformers) leads to better results.

### c. Bayesian parameter search

Based on the findings of the parameter space exploration, we designed Bayesian optimization experiments for the ensemble-based model classes. The complete set of hyperparameters is split into training-related (cf. Tab. 1) hyperparameters and architecture-related hyperparameters (cf. Tab. 2). Bayesian search is used for the former, grid search was found more reliable for the latter. During Bayesian tuning, we use the Bayesian optimization functionality of Ax (<https://github.com/facebook/Ax>) and optimize for a multi-objective, consisting of validation loss and training time. Due to time constraints, we train only one model per parameter configuration, i.e., ensembling multiple models is not considered. Once an optimal setting was identified, a full ensemble of 20 models is trained and the best configuration of architecture-related parameters wrt. validation scores is selected. To save additional time in tuning encoder-decoder models, we choose to tune parameters only for the attention-based merger, which was found to yield the best results in the early exploration phase. Other merger configurations are considered in ablation studies. The respective parameter selections and ranges for parameter search are summarized in Tabs. 1 and 2. Hyperparameters are tuned separately for different datasets and lead times, since models for larger lead times were found to require stronger regularization. The final hyperparameters are shown in Table B1 in Appendix B of the main paper.

### d. Ablations

To evaluate the effect of various changes in model architecture, we conduct ablation experiments, in which we disable specific aspects of the training or replace them by other mechanisms. We conduct the following ablations:

**No dropout:** For ensemble-based models, the Bayesian parameter search indicated that randomized dropout during training can improve model performance. To evaluate the veracity of this finding for the full ensemble model, we retrain an ensemble of 20 models with optimal Bayesian hyperparameters but with dropout disabled, and evaluate the performance of 10-member average models. The results show that dropout slightly worsens CRPS as well as PI length

TABLE 1. Search space of training-related hyperparameters.

Model class	Parameter	Range
DRN	learning rate	$[10^{-5}, 10^{-2}]$
	patience	[12, 24]
BQN	learning rate	$[10^{-5}, 10^{-2}]$
	patience	[12, 24]
ED-DRN	learning rate	$[10^{-5}, 10^{-2}]$
	patience	[12, 24]
	encoder dropout rate	$[5 \times 10^{-3}, 0.5]$
	decoder dropout rate	$[5 \times 10^{-3}, 0.5]$
ED-BQN	learning rate	$[10^{-5}, 10^{-3}]$
	patience	[12, 24]
	encoder dropout rate	$[5 \times 10^{-3}, 0.5]$
	decoder dropout rate	$[5 \times 10^{-3}, 0.5]$
ST-DRN	learning rate	$[10^{-5}, 10^{-3}]$
	patience	[8, 24]
	transformer dropout rate	$[5 \times 10^{-3}, 0.5]$
	decoder dropout rate	$[5 \times 10^{-3}, 0.5]$
ST-BQN	learning rate	$[10^{-5}, 10^{-3}]$
	patience	[8, 24]
	transformer dropout rate	$[5 \times 10^{-3}, 0.5]$
	decoder dropout rate	$[5 \times 10^{-3}, 0.5]$

TABLE 2. Selection of architecture-related hyperparameters.

Model class	Parameter	Values
DRN	channels (first layer)	32, 48, 64
	polynomial degree	8, 12, 16
ED-DRN	channels (encoder)	48, 64
	channels (decoder)	48, 64
ED-BQN	channels (encoder)	48, 64
	channels (decoder)	48, 64
	polynomial degree	8, 12, 16
ST-DRN	channels (transformer)	48, 64
	channels (decoder)	48, 64
ST-BQN	channels (transformer)	48, 64
	channels (decoder)	48, 64
	polynomial degree	8, 12, 16

for DRN-type ED models. For the final comparison, dropout is therefore disabled in this class.

**1D vs. 2D yearday embedding:** In contrast to the original DRN and BQN Schulz and Lerch (2022), ensemble-based models in this study use a 2D embedding of the day of year information through sine in cosine modes, which was found profitable during the initial exploration. We evaluate the effect of using

both variants but do not find significant differences upon closer inspection.

**Spherical vs. plain lat-lon embedding:** In contrast to the original DRN and BQN Schulz et al. (2021), ensemble-based models in this study use a 3D embedding of station positions in terms of spherical coordinates. The 3D embedding offers a more accurate representation of the spherical geometry of the Earth, which may have an advantage when considering stations all around the globe. We ablate this design choice and supply the models with plain latitude- and longitude coordinates instead (whitened and normalized). For the data considered in this study, the location embedding is found to be of minor importance. We attribute this to the fact that for both datasets all stations are located in Europe, such that potential distortions due to coordinate projections are sufficiently small to not affect model performance.

**Choice of the merger:** We compare the model performance of ensemble-based encoder-decoder models with different merging algorithms. Attention-based merging is overall favorable for both DRN- and BQN-type models. For DRN, the results for long lead times suggest that the training outcomes suffer from instability and don't always converge to good local optima. BQNs are less prone to this behaviour.

## 2. Additional results

### a. Comparison of PI length for DRN and BQN

Here, we briefly comment on the differences arising from the underlying distribution type, which agree for both data sets considered in this work. Plotting the PI length on the nominal level for different choices thereof in Fig. 2, we find that the choice of the forecast distribution defines the magnitude of the PI length. Up to a nominal level of around 90%, the PI lengths of the DRN and BQN forecasts both increase linearly at the same length, but for higher levels, they start to deviate as the PI lengths of the DRN models increase exponentially, while that of the BQN models still increase linearly. This can be explained by the fact that the BQN forecast distribution has compact support defined by the coefficients, whereas the distributions of DRN models are heavy-tailed with support on the positive real line. However, a detailed comparison of these two approaches for modeling tails of distributions in terms of predictive accuracy and appropriateness is not discussed in the following.

The data sets we consider include ensembles of three different sizes with ensemble ranges that correspond to PIs at the 83.33%, 90.48%, and 96.15% levels. Comparing the BQN and DRN forecasts for these different levels, we find that for 11-member EUPPBench reforecasts the DRN

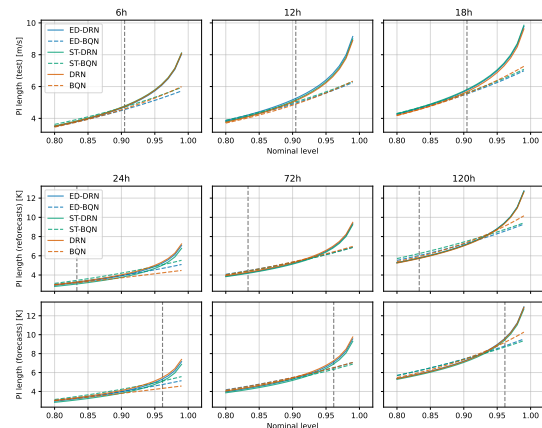


FIG. 2. PI length as a function of the nominal level for 11-member reforecast and 51-member forecast ensembles. PI lengths are computed based on averaged predictions of 10-member model ensembles. The shown values are averaged over 50 such ensemble models. The vertical dashed lines indicate the nominal level of the underlying ensemble.

models result in sharper forecasts, while the BQN models generate sharper forecasts for the 51-member ensemble forecasts. In the case of the wind gust data, the 20-member ensemble corresponds to a nominal level close to the intersection of the PI length curves, hence the differences between the distribution types are less pronounced than for the EUPPBench data.

Fig. 3 shows the relative deviation of the PI lengths of the permutation-invariant models from the mean-based models. In general, we cannot conclude that either of the approaches produces sharper forecasts, in general. Comparing the behavior over the nominal levels, we find that differences in the PI lengths are constant for the DRN models, while we see a trend in the differences for the BQN models, namely, either a monotonic increase or a decrease. This means that the permutation-invariant BQN models lead to different behavior in the tails of the distribution. In the case of wind-gust postprocessing, we see a decrease in the deviation, which corresponds to a lighter tail. For both temperature datasets, we see an increase in deviation for 24h lead time, i.e., a heavier tail, whereas for 120h lead time a decrease is seen as in the wind-gust case. At lead time 72h, the difference remains roughly constant with values close to zero on reforecast data, indicating approximately equal weighting of the tails. On the forecast dataset, a slight negative trend is observed. Further, the confidence bands, corresponding to the 95% interval of the mean, show that the tendency is fixed for each of the nine different forecast cases.

### b. Variability over neural network resamples

Comparing the differences between the permutation-invariant model classes for the wind gust data, we find

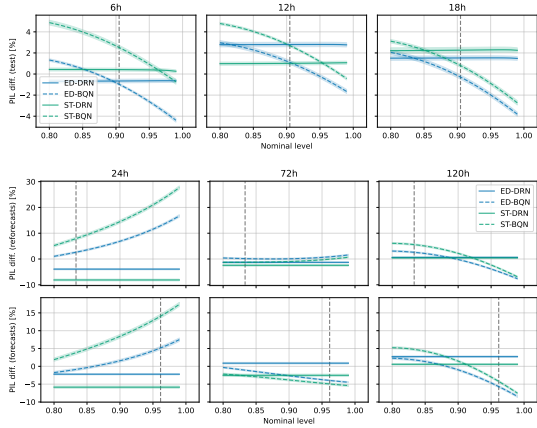


FIG. 3. Relative deviation of PI length from the mean-based model as a function of the nominal level for 11-member reforecast and 51-member forecast ensembles. PI lengths are computed based on averaged predictions of 10-member model ensembles. The shown values are averaged over 50 such ensemble models. PI lengths for DRN and BQN are considered as the baseline for the respective permutation-invariant models. The vertical dashed lines indicate the nominal level of the underlying ensemble.

only minor differences. Fig. 4 explores these differences in more detail. Shown are boxplots of the CRPS and PI length of the permutation-invariant network classes distinguished by the employed ensemble merging algorithm. For all lead times and network classes, the same pattern is observed. Mergers based on extreme values such as the minimum or maximum result in a worse CRPS and wider PIs than those based on calculations of the mean and attention models. The same pattern as for the extreme value models is observed for the set transformer models, that is, a worse CRPS and wider PIs. With respect to the benchmark models, we find that the CRPS is larger for almost all cases considered. In terms of the PI length, we find that the mergers based on the mean mostly result in slightly smaller PIs. The overall differences observed are however only on a small scale. Attention-based merging is employed for all experiments in the main text.

#### c. DRN for EUPP: Truncated logistic vs. truncated normal posterior

To maintain close similarity between the wind gust and the EUPPBench case study, we construct DRN models that parameterize a truncated logistic posterior distribution, analogous to the case of wind gust postprocessing. Note that temperature observations  $T_{\text{obs}}$  are recorded in Kelvin, such that  $T_{\text{obs}} > 0$  holds, which justifies the truncated logistic distribution as a valid choice for temperature postprocessing. However, a more common choice would be the (truncated) normal distribution (e.g., Gneiting et al. 2005; Rasp and Lerch 2018). To ensure a fair comparison,

we validate here that the design choice of using the truncated logistic posterior does not affect the postprocessing capabilities of DRN negatively.

Tab. 3 displays the prediction metrics as obtained for DRN models with truncated logistic (DRN-TL) and normal posterior (DRN-TN). Note that DRN-TN has undergone the same hyperparameter search as DRN-TL models and that the model selection was based on validation scores, not on the test scores that are shown in Tab. 3. DRN-TN models are identical in architecture to the DRN-TL counterparts, including the softmax constraint for both the location and the scale parameters, but are trained to optimize an analytical expression of the CRPS for a normal distribution Gneiting et al. (2005), given the training data. For evaluation, we compute the CRPS based on a zero-truncated normal distribution. The distinction is made to avoid numerical instabilities that are caused by the truncation terms during training. Due to the magnitude of the temperature observations and the expected value range of the fitted distributions,  $T_{\text{obs}} \sim 300\text{K}$ ,  $\Delta T_{\text{obs}} \sim \pm 30\text{K}$ , however, this does not have a large effect on the final outcome. As seen in Tab. 3, we find that DRN-TL and DRN-TN score roughly identically in terms of CRPS. Due to the heavier tails, the PI length of DRN-TL models is slightly larger, and the coverage probabilities are met slightly more accurately for DRN-TN. Fig. 5 displays calibration histograms for both model variants. We note that both models overestimate high-temperature extremes slightly at lead times 24h and 72h on reforecast data, and underestimate low-temperature extremes at lead times 72h and 120h. Similar findings apply to the case of forecast data. Despite minor differences, both forecasts appear overall well-calibrated and we do not see reasons to expect qualitatively different results in our study when exchanging the truncated logistic posterior with the truncated normal.

#### d. BQN: Full ensemble vs. summary statistics as predictors

To enable a more direct comparison between DRN- and BQN-type models, we deviate from prior work and train MLP-based models with BQN posterior using the mean and the standard deviation of the primary predictor ensemble (t2m for EUPPBench, VMAX-10M for wind gusts). This differs from the work by Bremnes (2020) and Schulz and Lerch (2022), who use the full ensemble in sorted order as input to the BQN models. Here we validate this design decision and show that both approaches yield qualitatively similar results. The case of EUPPBench is of particular interest here, since ensemble-based models have to cope with different ensemble sizes in the reforecast and forecast test cases.

But first, we compare the forecasts for the wind gust data set. Table 4 shows the evaluation metrics for the different lead times, where we observe almost identical

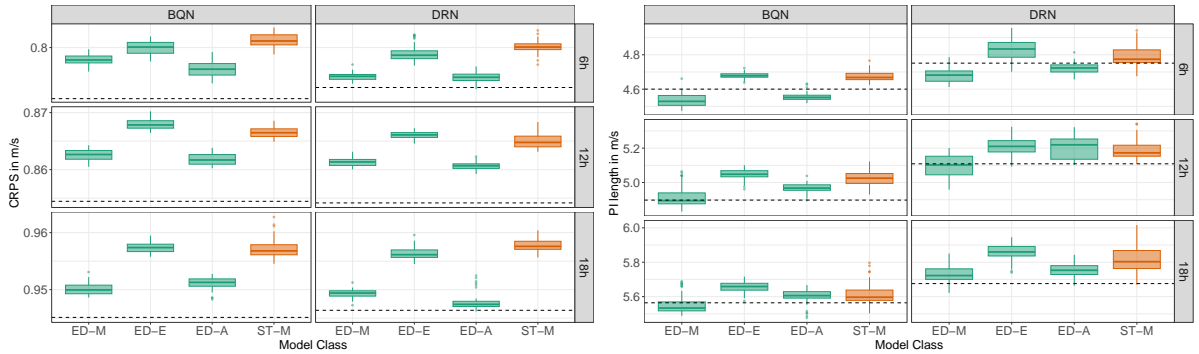


FIG. 4. Boxplot of the CRPS (left) and PI length (right) in m/s for the different lead times and model classes, the dashed lines display the value of the corresponding benchmark. The last letter refers to merger, where M comprises the mergers associated with the mean (mean, mean-logstd, weighted-mean, weighted-mean-logstd), E the extremes (max, min-max), and A the attention models. The boxes are calculated based on the ensemble resamples, where for each repetition we picked that configuration within the model class that yields the smallest CRPS on the validation set.

Lead Time		24h			72h			120h		
Dataset	Method	CRPS	PI length	PI cov.	CRPS	PI length	PI cov.	CRPS	PI length	PI cov.
Reforecasts	DRN-TN	0.67	3.27	83.87	0.86	4.14	83.41	1.20	5.93	84.17
	DRN-TL	0.67	3.28	84.16	0.86	4.27	84.58	1.19	5.70	83.09
Forecasts	DRN-TN	0.64	5.01	96.91	0.81	6.58	97.35	1.14	8.62	96.83
	DRN-TL	0.64	5.48	97.92	0.80	7.21	98.37	1.13	9.58	98.28

TABLE 3. Test scores for DRN architectures with truncated normal (TN) and truncated logistic (TL) posterior on EUPPBench data for different lead times. PI length and coverage are computed for a significance level corresponding to an 11-member ensemble ( $\sim 83.33\%$ ) for reforecasts and a 51-member ensemble ( $\sim 96.15\%$ ) for forecasts.

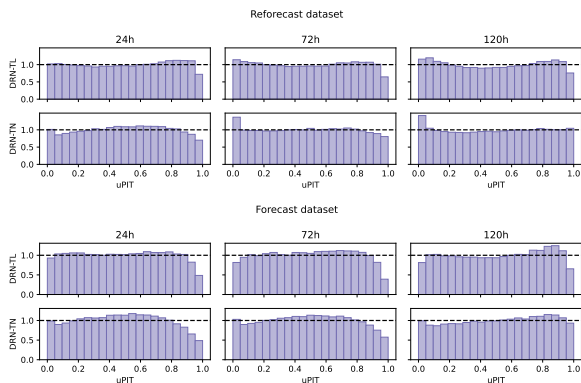


FIG. 5. Calibration of postprocessing models on 11-member EUPPBench reforecast ensembles (top) and 51-member forecast ensembles (bottom).

results. The negligible differences seen only in the PI lengths and coverages might also be a result of the different realizations of the underlying network ensembles trained. For the wind gust data, we conclude that there are no differences in the predictive performance between the two variants.

Tab. 4 shows the model scores on EUPPBench reforecast and forecast datasets, comparing a BQN model informed with summary statistics (mean and standard deviation, BQN-Sum) and the full ensemble model (BQN-Ens). Both models adhere to the hyperparameter configuration listed in Appendix 1. Most notably, BQN-Ens comes with a slightly larger PI length on reforecast data at 24h lead time, while yielding the same CRPS as BQN-Sum. According to PI coverage, BQN-Sum matches the theoretical value of 83.33% more accurately. For the remaining lead times, the differences are negligible. To make BQN-Ens applicable to the forecast dataset, which comprises more members per ensemble, we distinguish randomized subsampling (BQN-Ens-R, 11 out of 51 members, sampled without replacement) and quantile-based subsampling (BQN-Ens-Q). For the latter, we sort the 51-member ensemble in ascending order and pick the members with rank  $(51 + 1)/(11 + 1) * i$ , for  $i = 1, \dots, 11$ , as the predictor ensemble. BQN-Sum yields marginally sharper forecasts at 24h and 120h lead time. The differences between the subsampling variants are negligible. Fig. 6 displays calibration histograms for all model variants. All histograms exhibit a wave-like structure, but appear otherwise very similar, despite slight differences in the placement of the distribution peaks. The

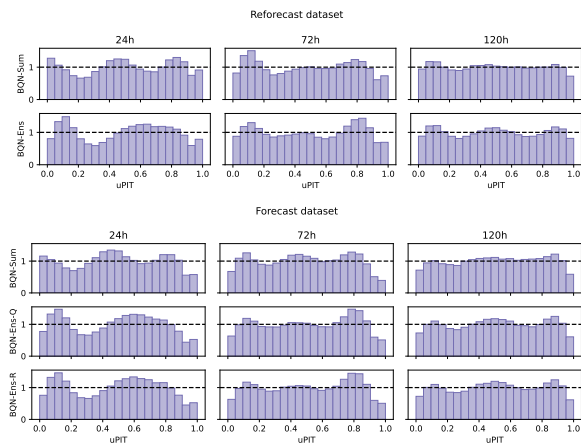


FIG. 6. Calibration of postprocessing models on 11-member EUPP-Bench reforecast ensembles (top) and 51-member forecast ensembles (bottom).

120h case achieves the most uniform calibration, overall. Again, hardly any differences are seen between the subsampling variants. We conclude that it is well justified to replace the full ensemble with the summary-based predictors for BQN models

### 3. Additional figures

Here we include figures, which are obtained using the methods in the main paper but are too exhaustive to include in the main text. We add an overview of the complete set of permutation feature importance values, shown in Fig. 7. We also provide illustrations of the ensemble-oriented permutation feature importance for all variables and lead times. The data for wind gust postprocessing are shown in Figs. 8 to 16, the data for the EUPPBench case are shown in Figs. 17 to 20. Note that for some predictors the bar charts show large variations. The reason for these behaviors is extreme outliers, which distort the statistics. However, such extreme cases are only observed for parameters of limited permutation importance (cf. Fig. 3 in the main text).

### References

- Bremnes, J. B., 2020: Ensemble postprocessing using quantile function regression based on neural networks and Bernstein polynomials. *Monthly Weather Review*, **148** (1), 403–414, <https://doi.org/10.1175/mwr-d-19-0227.1>.
- 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** (5), 1098–1118, <https://doi.org/10.1175/MWR2904.1>.
- Godbole, V., G. E. Dahl, J. Gilmer, C. J. Shallue, and Z. Nado, 2023: Deep learning tuning playbook. URL [http://github.com/google-research/tuning\\_playbook](http://github.com/google-research/tuning_playbook), version 1.0.

Rasp, S., and S. Lerch, 2018: Neural networks for postprocessing ensemble weather forecasts. *Monthly Weather Review*, **146** (11), 3885–3900, <https://doi.org/10.1175/MWR-D-18-0187.1>.

Schulz, B., M. E. Ayari, S. Lerch, and S. Baran, 2021: Post-processing numerical weather prediction ensembles for probabilistic solar irradiance forecasting. *Solar Energy*, **220**, 1016–1031, <https://doi.org/10.1016/j.solener.2021.03.023>.

Schulz, B., and S. Lerch, 2022: Machine learning methods for post-processing ensemble forecasts of wind gusts: A systematic comparison. *Monthly Weather Review*, **150** (1), 235–257, <https://doi.org/10.1175/mwr-d-21-0150.1>.

Lead Time		6h resp. 24h			12h resp. 72h			18h resp. 120h		
Dataset	Method	CRPS	PI length	PI cov.	CRPS	PI length	PI cov.	CRPS	PI length	PI cov.
Wind gusts	BQN-Sum	0.79	4.60	90.23	0.85	4.90	89.65	0.95	5.56	90.70
	BQN-Ens	0.79	4.61	90.20	0.85	4.94	89.91	0.95	5.56	90.65
Reforecasts	BQN-Sum	0.68	3.19	82.90	0.87	4.43	85.87	1.19	5.91	84.75
	BQN-Ens	0.68	3.37	84.92	0.87	4.43	85.93	1.19	5.90	84.58
Forecasts	BQN-Sum	0.64	4.32	94.13	0.80	6.52	97.23	1.13	9.18	97.58
	BQN-Ens-Q	0.65	4.97	96.36	0.80	6.50	97.11	1.13	9.31	97.65
	BQN-Ens-R	0.65	5.00	96.39	0.81	6.56	97.11	1.13	9.30	97.65

TABLE 4. Test scores for BQN architectures with predictors based on the full ensemble (BQN-Ens) of primary predictors and predictors based on summary statistics (mean and standard deviation, BQN-Sum). PI length and coverage are computed for a significance level corresponding to a 20-member ensemble ( $\sim 90.48\%$ ) for the wind gust data, an 11-member ensemble ( $\sim 83.33\%$ ) for the EUPPBench reforecasts, and a 51-member ensemble ( $\sim 96.15\%$ ) for the EUPPBench forecasts. In the case of the EUPPBench forecast data with ensemble-valued predictors, the full 51-member ensemble is subsampled randomly (BQN-Ens-R) or based on quantiles (BQN-Ens-Q) to match the 11-member training dataset.



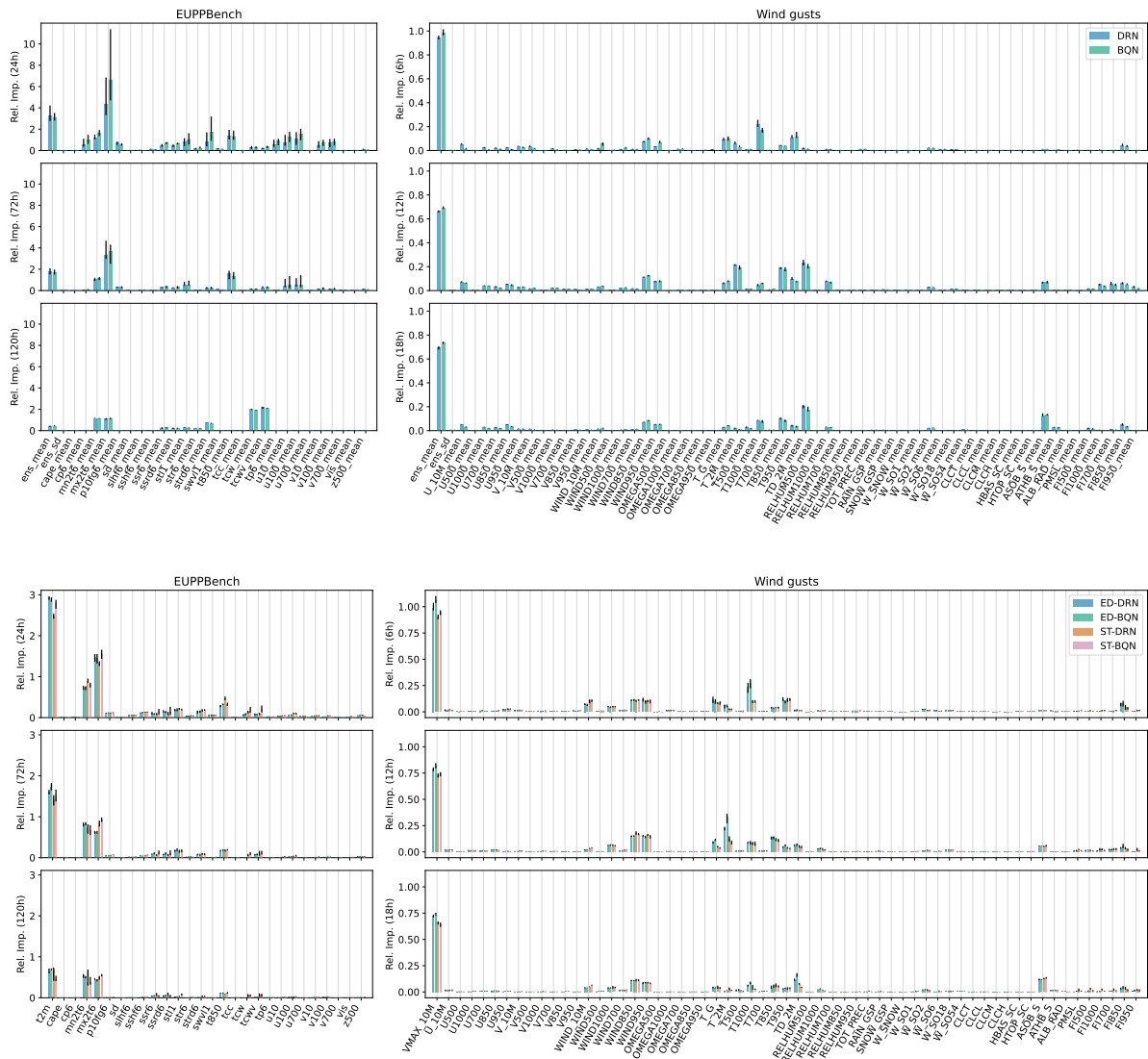


Fig. 7. Permutation feature importance for summary-based networks (top) and permutation-invariant models (bottom) for EUPPBench and wind gust postprocessing. Predictors named ens in the top figure correspond to the primary predictors t2m and VMAX-10M, respectively. The suffix sd indicates the ensemble standard deviation of the predictor. Same as Fig. 3 in the main text.

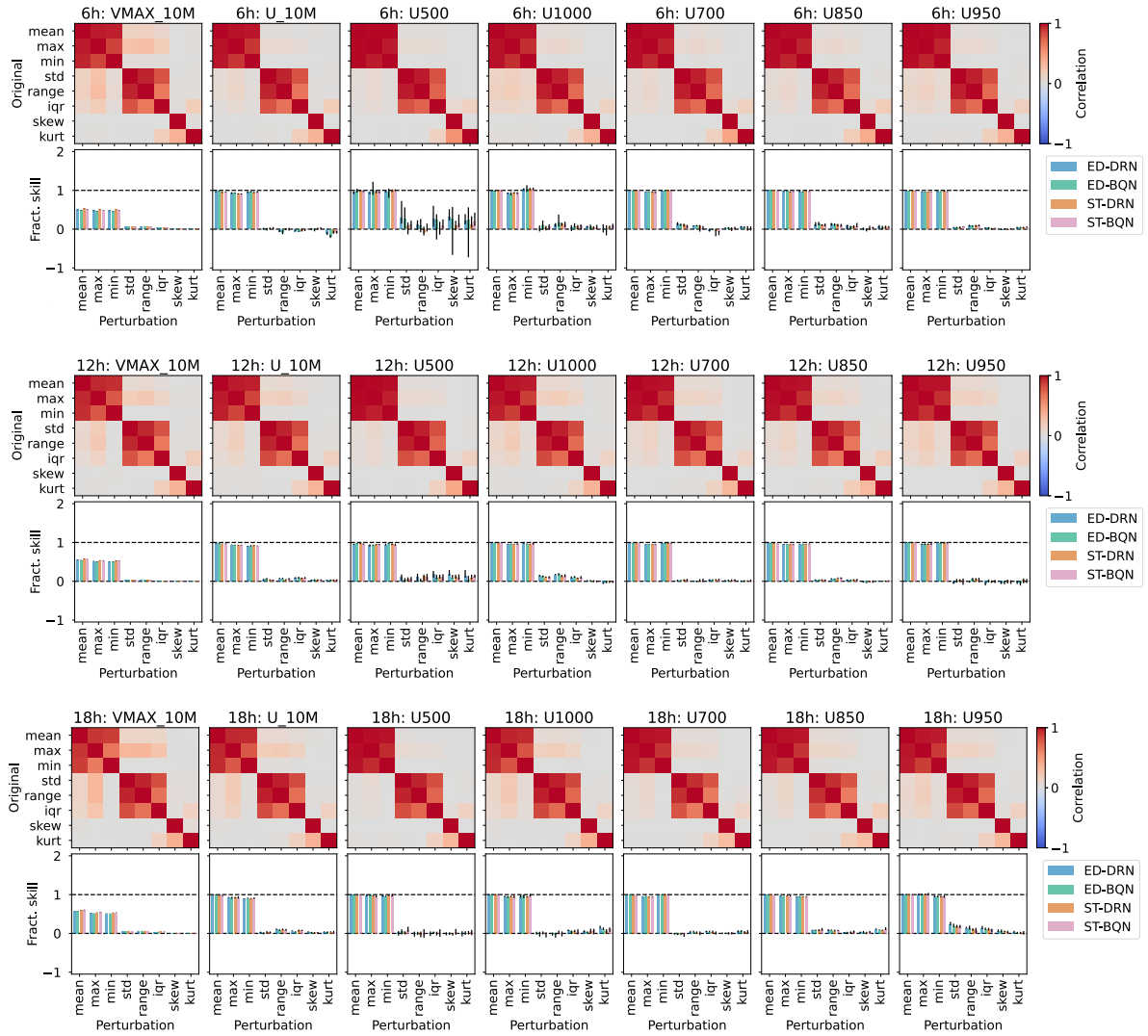


FIG. 8. Importance of ensemble-internal DOFs for wind-gust postprocessing (predictor batch 1). Same as Fig. 4 in the main text.

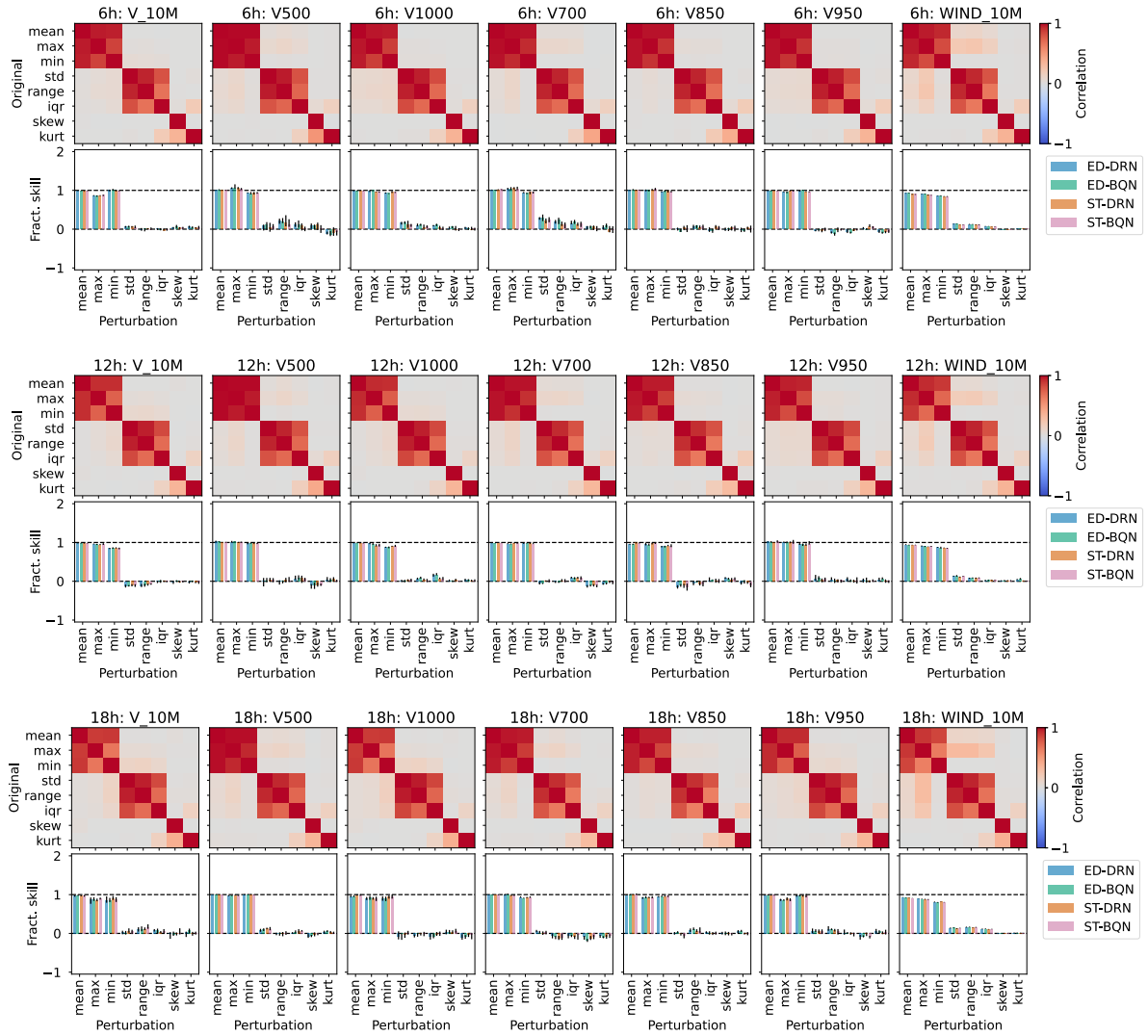


Fig. 9. Importance of ensemble-internal DOFs for wind-gust postprocessing (predictor batch 2). Same as Fig. 4 in the main text.

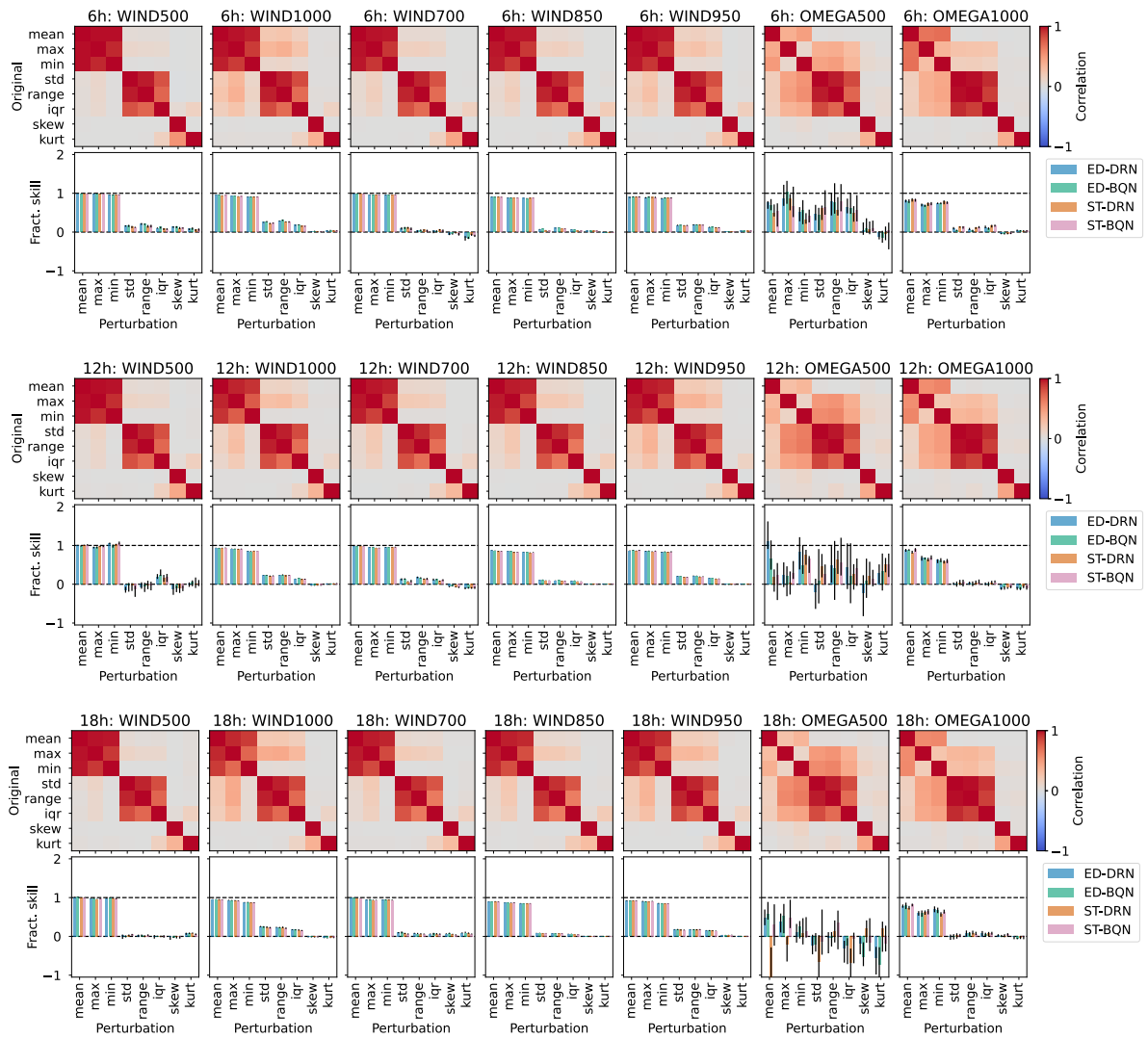


FIG. 10. Importance of ensemble-internal DOFs for wind-gust postprocessing (predictor batch 3). Same as Fig. 4 in the main text.

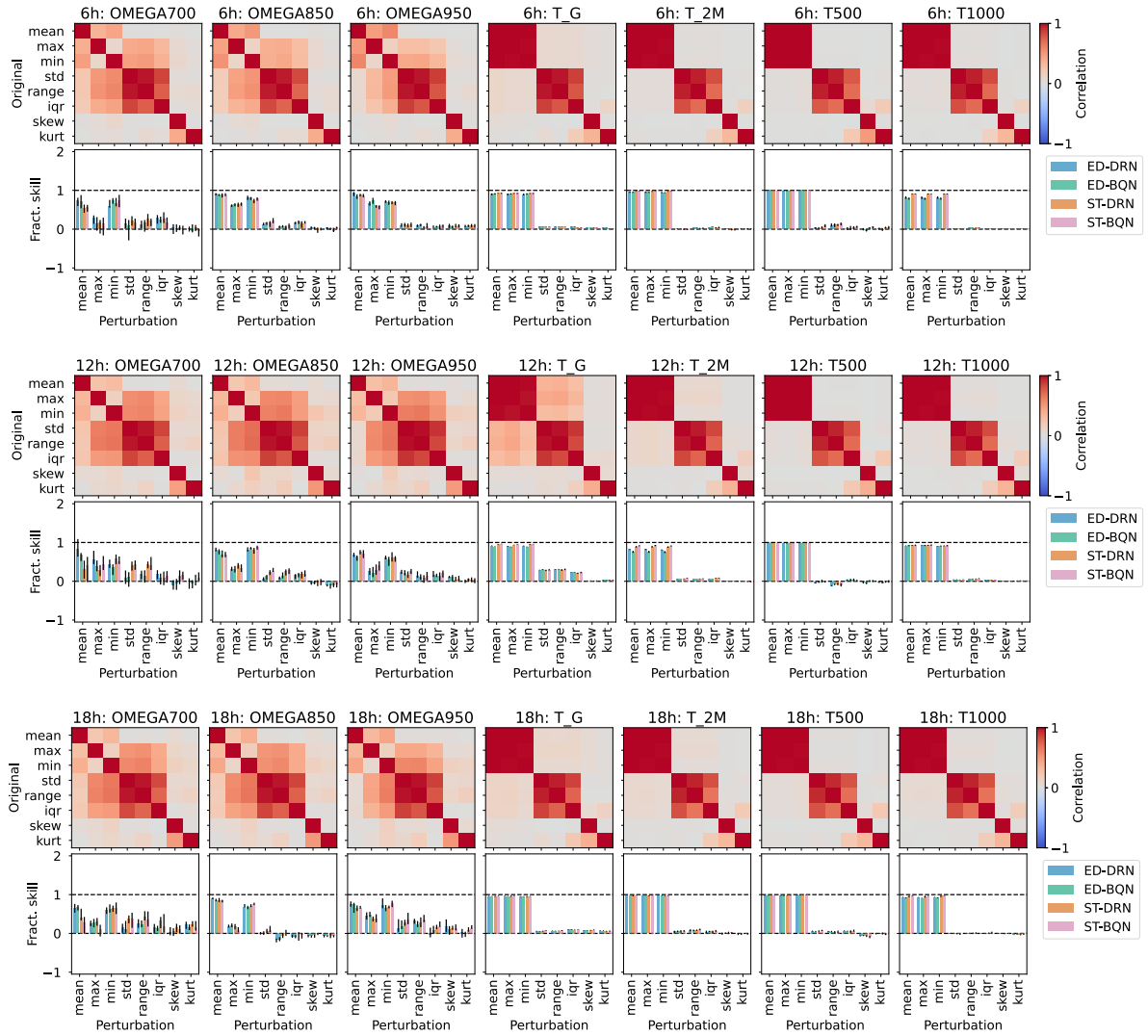


FIG. 11. Importance of ensemble-internal DOFs for wind-gust postprocessing (predictor batch 4). Same as Fig. 4 in the main text.

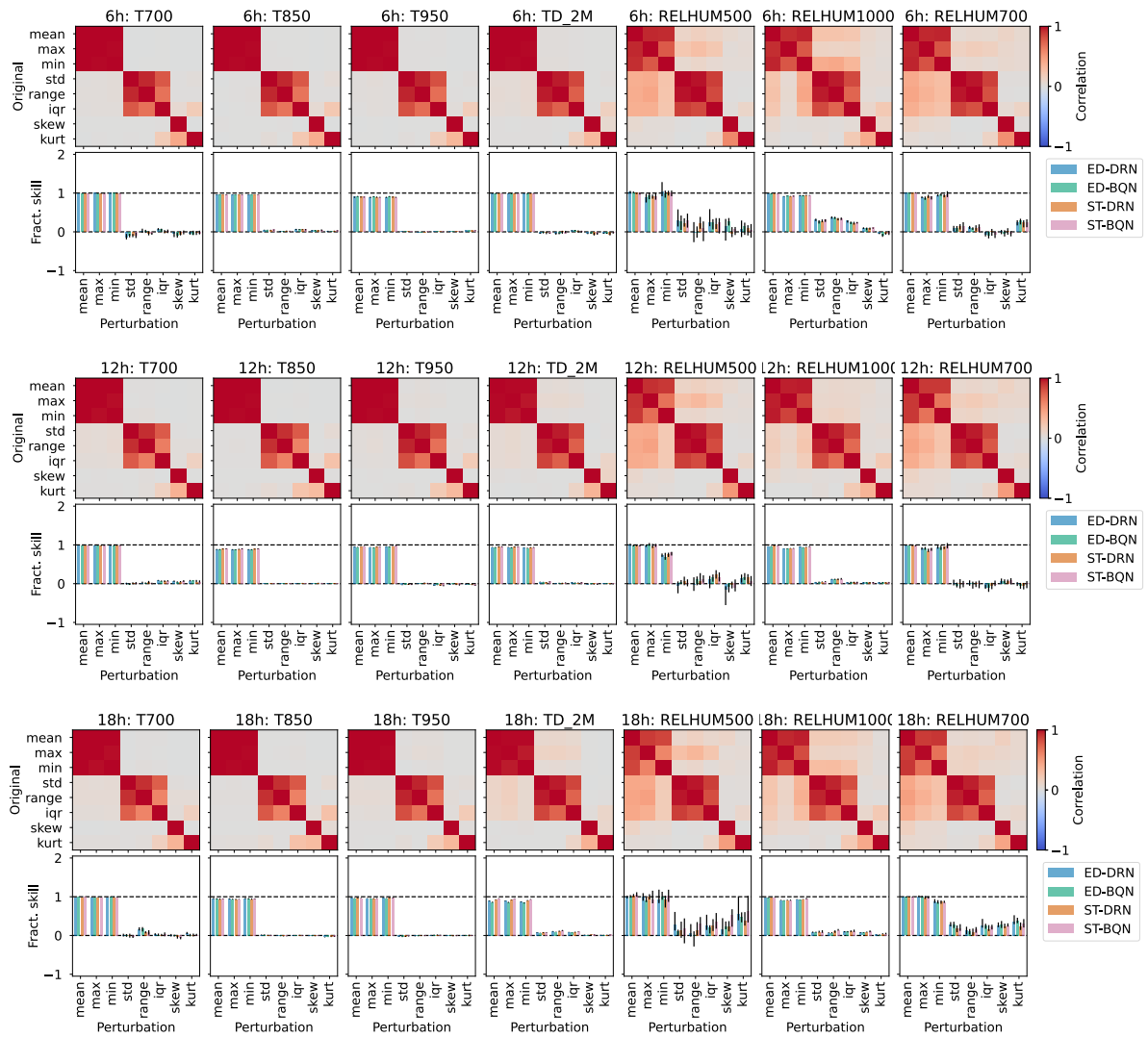


FIG. 12. Importance of ensemble-internal DOFs for wind-gust postprocessing (predictor batch 5). Same as Fig. 4 in the main text.

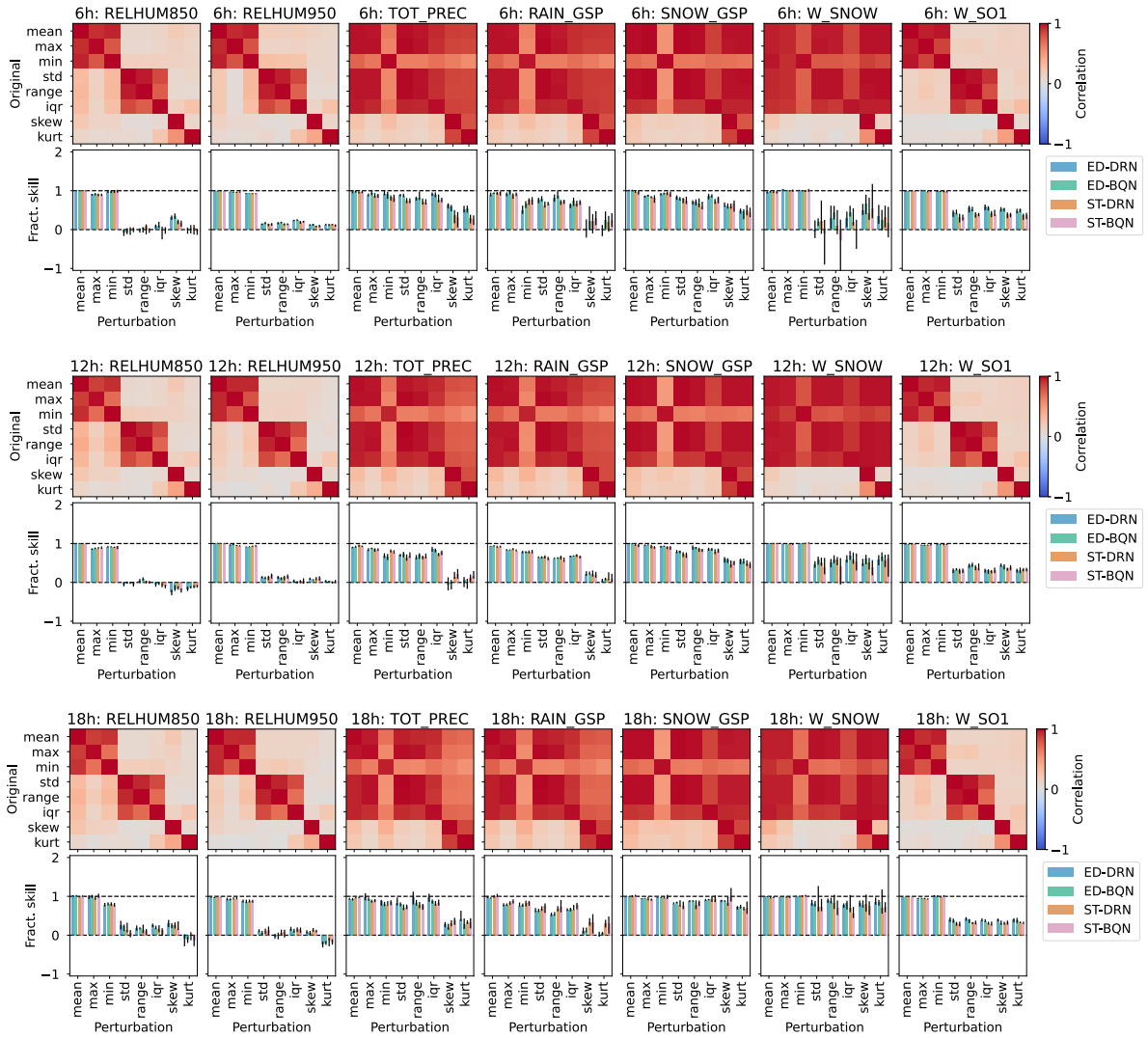


FIG. 13. Importance of ensemble-internal DOFs for wind-gust postprocessing (predictor batch 6). Same as Fig. 4 in the main text.

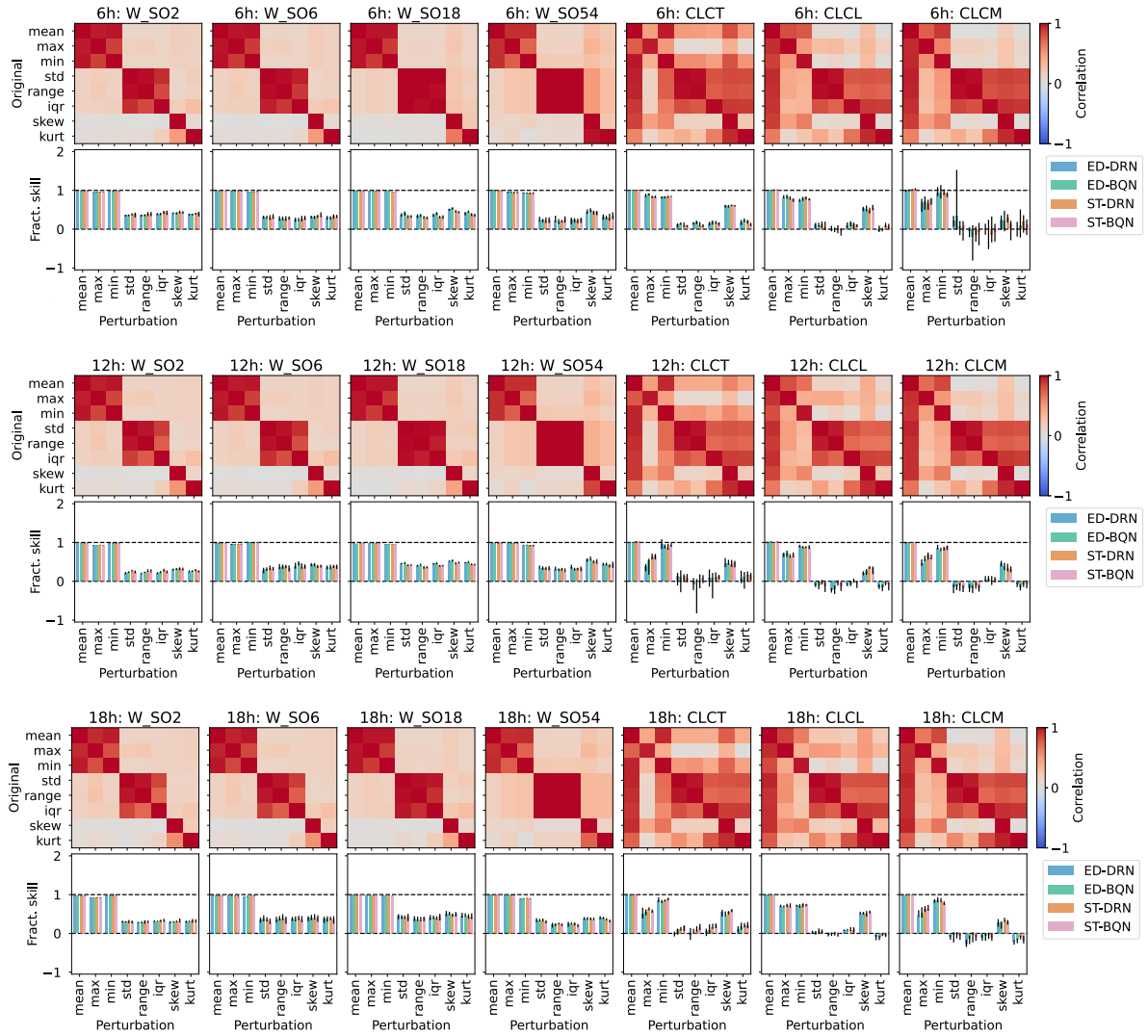


FIG. 14. Importance of ensemble-internal DOFs for wind-gust postprocessing (predictor batch 7). Same as Fig. 4 in the main text.



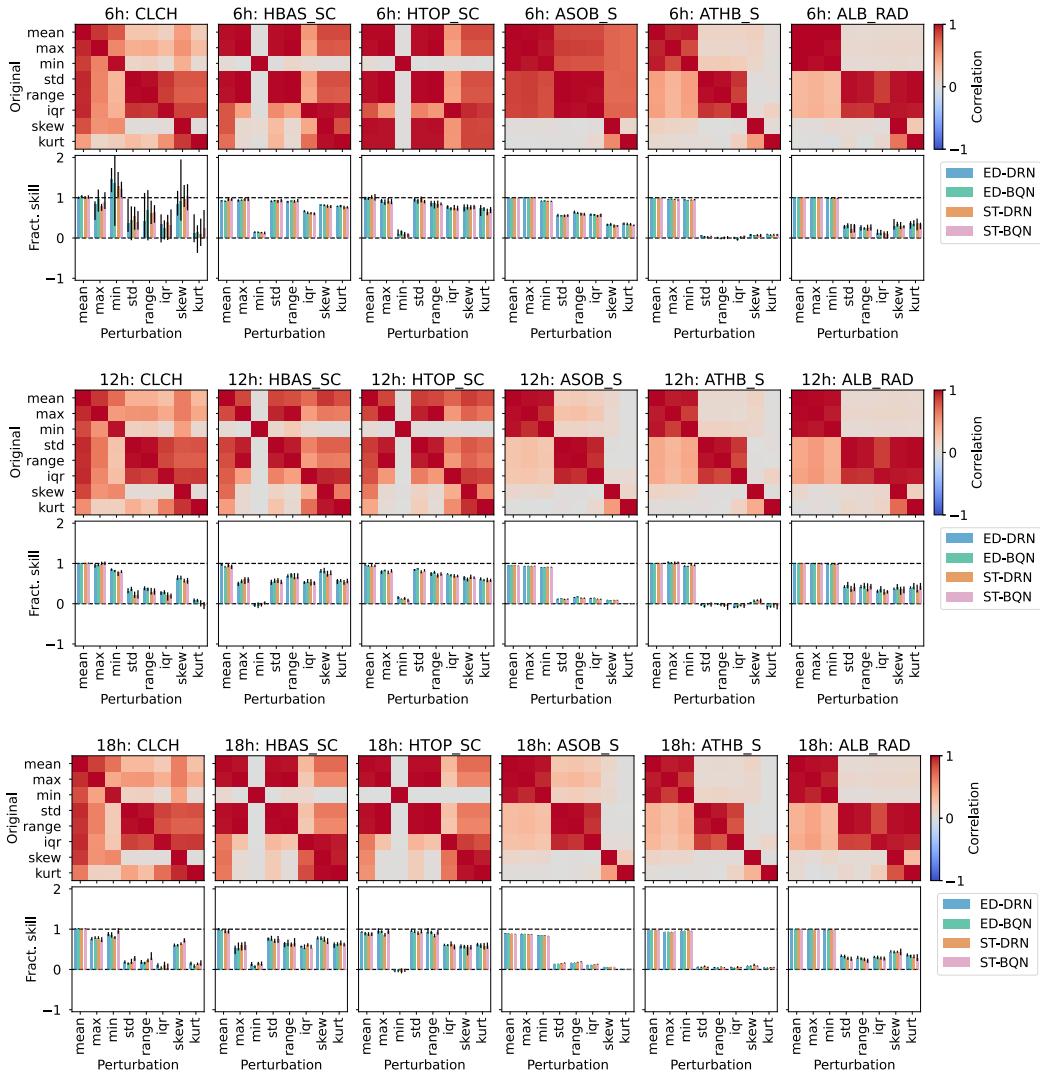


FIG. 15. Importance of ensemble-internal DOFs for wind-gust postprocessing (predictor batch 8). Same as Fig. 4 in the main text.

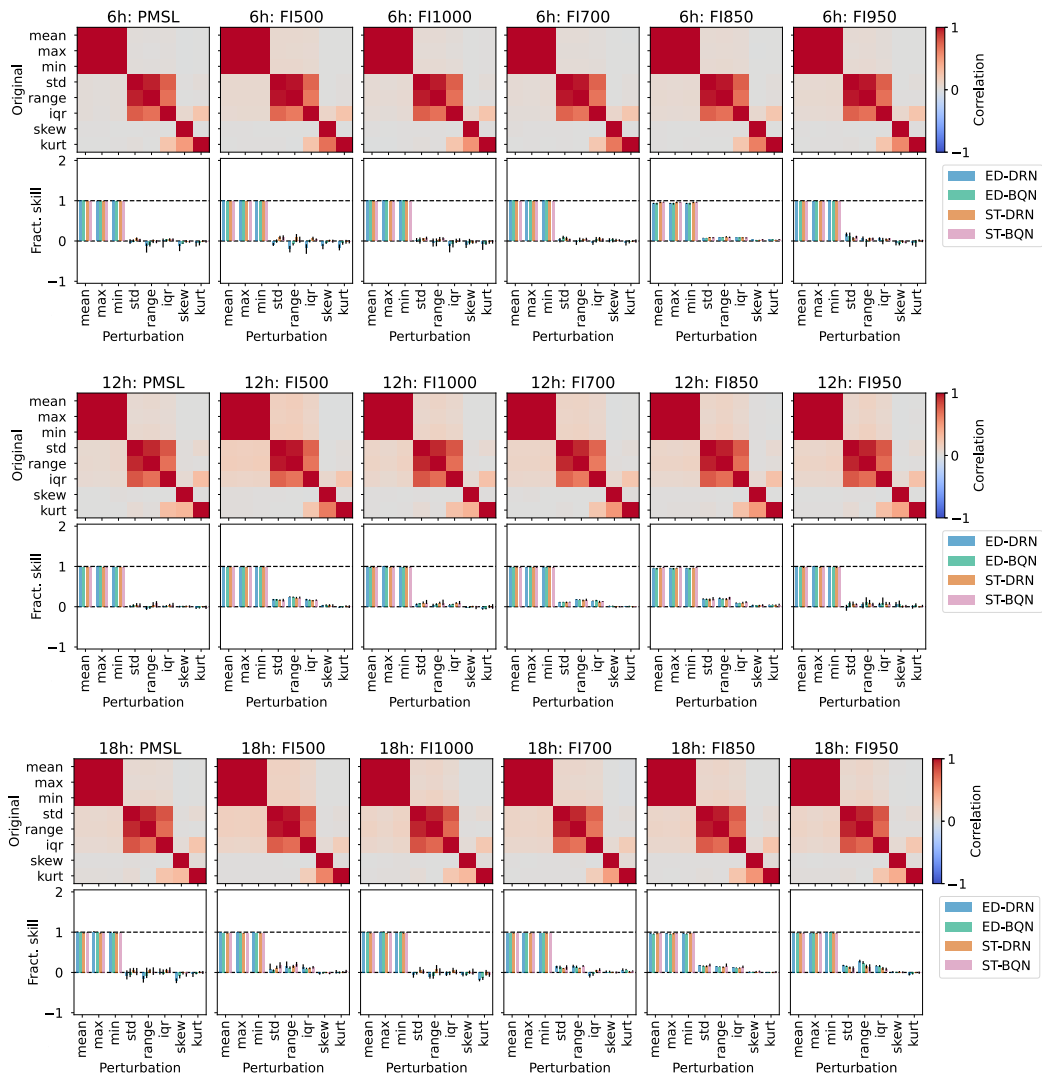


Fig. 16. Importance of ensemble-internal DOFs for wind-gust postprocessing (predictor batch 9). Same as Fig. 4 in the main text.

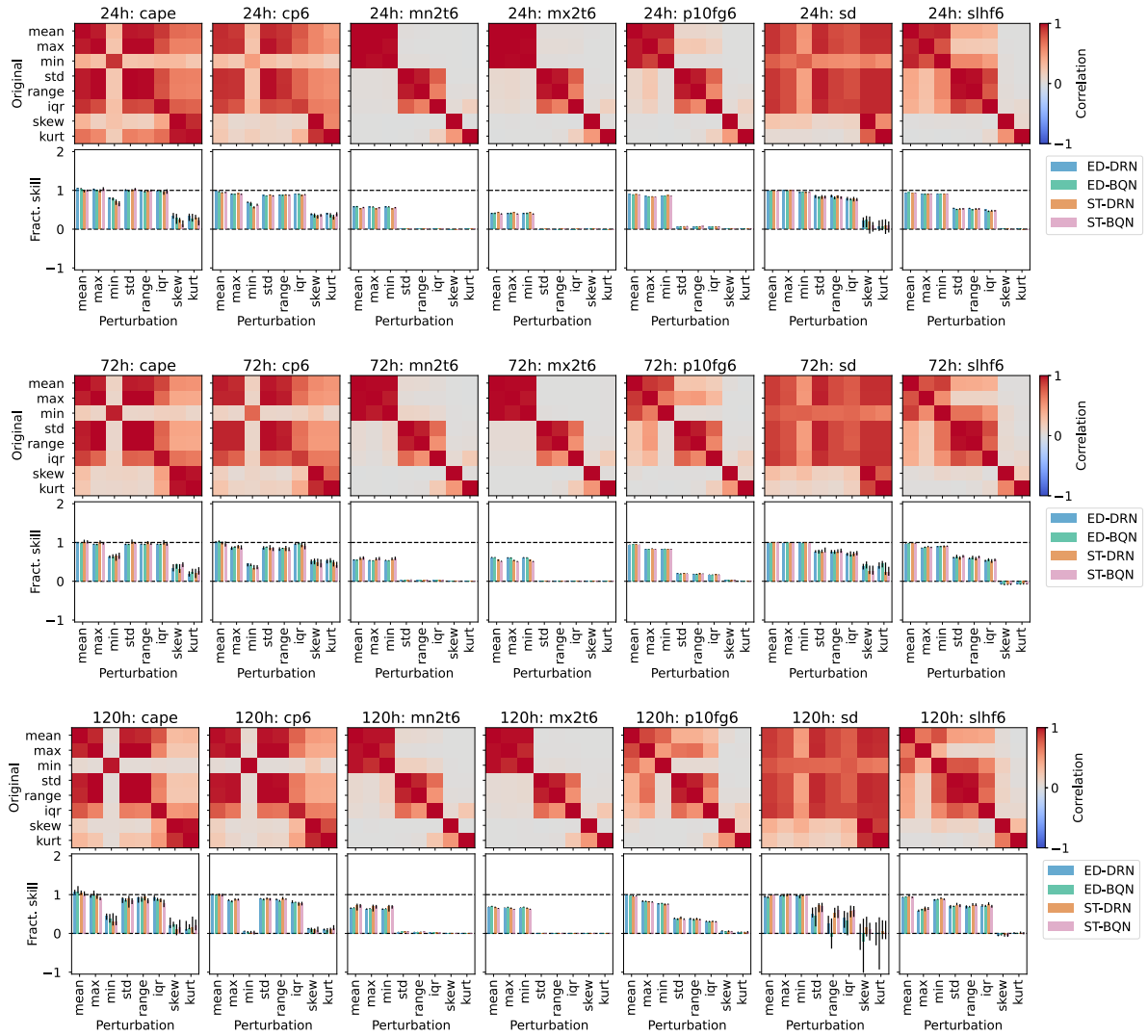


FIG. 17. Importance of ensemble-internal DOFs for temperature postprocessing (predictor batch 1). Same as Fig. 5 in the main text.

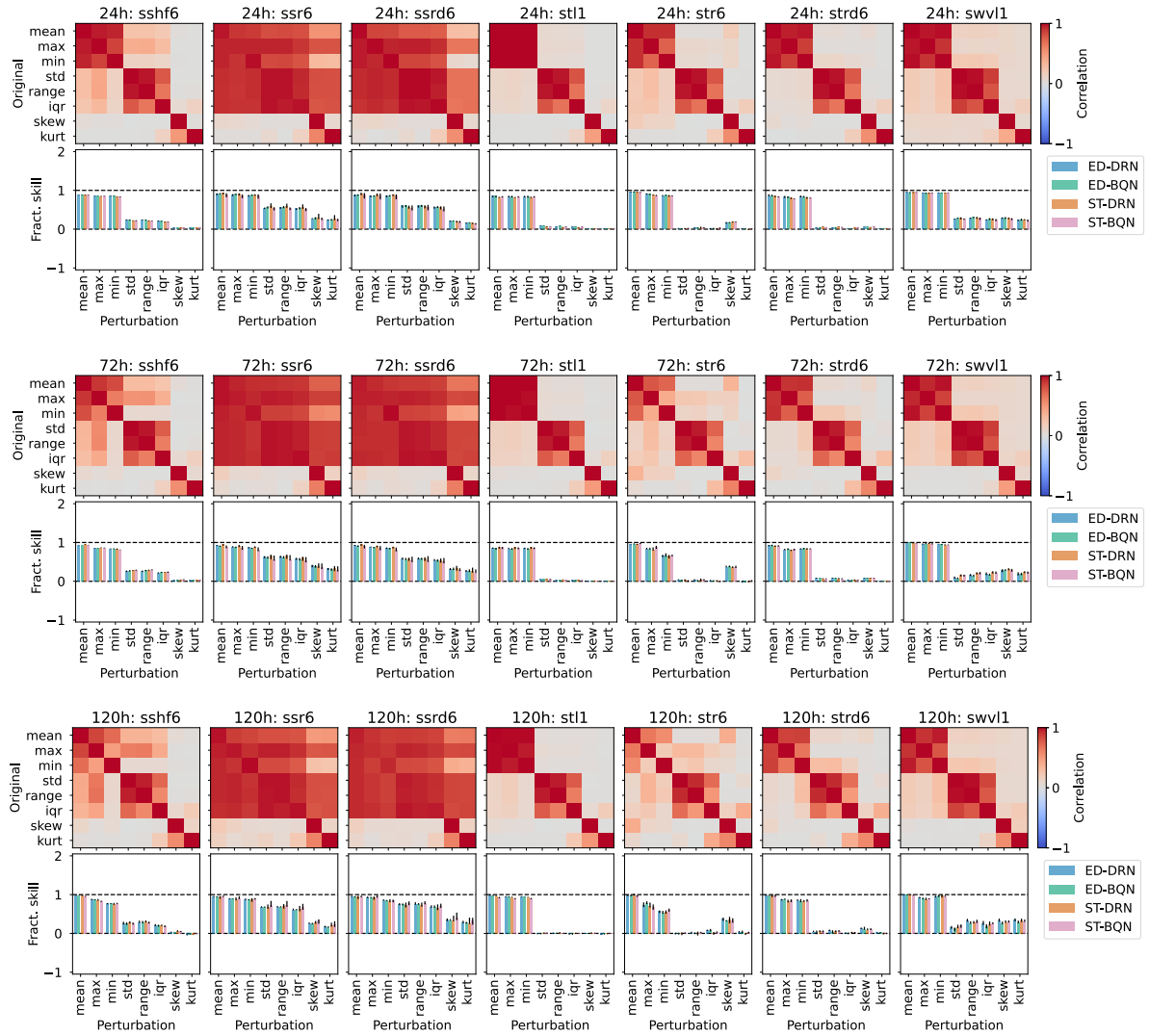


FIG. 18. Importance of ensemble-internal DOFs for temperature postprocessing (predictor batch 2). Same as Fig. 5 in the main text.

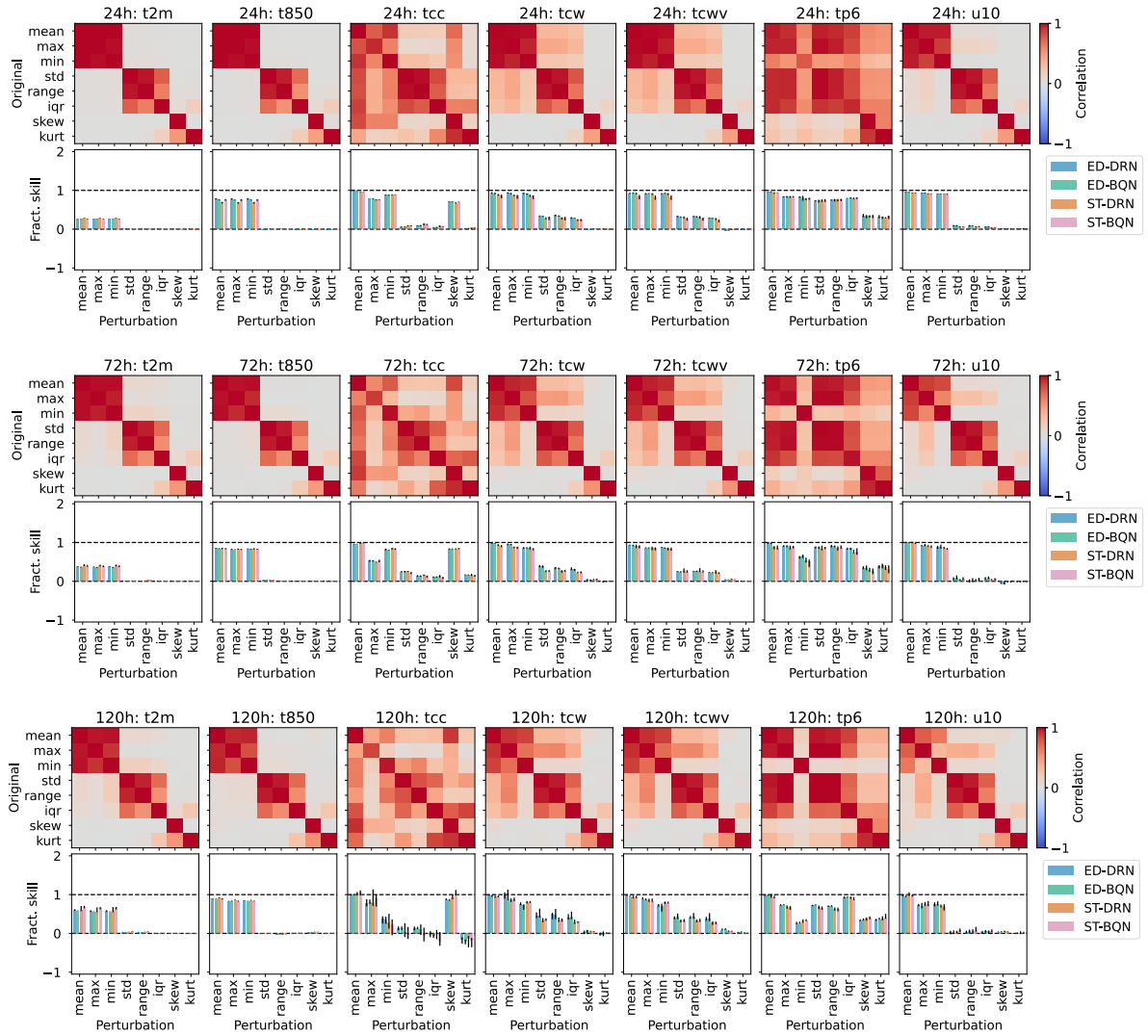


FIG. 19. Importance of ensemble-internal DOFs for temperature postprocessing (predictor batch 3). Same as Fig. 5 in the main text.

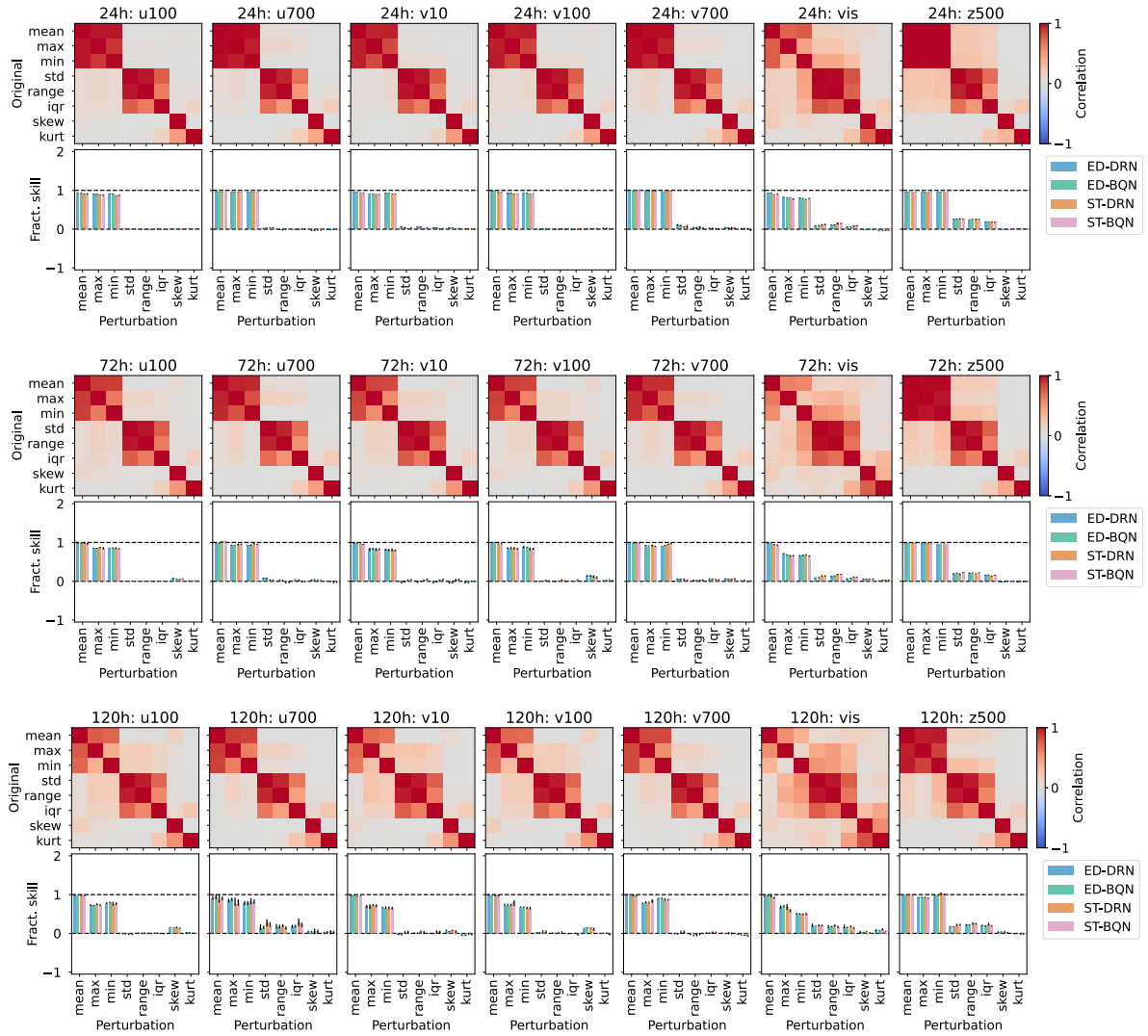


FIG. 20. Importance of ensemble-internal DOFs for temperature postprocessing (predictor batch 4). Same as Fig. 5 in the main text.

# Permission to Reuse

## Granted by the American Meteorological Society

E-Mail sent to kevin.hoehlein@tum.de on 23 August, 2024:

Dear Mx. Höhlelein,

Thank you for your email. This signed message constitutes permission to use the material requested below.

You may include your 2024 AIES article in your Technical University of Munich thesis "Data-Driven Modeling and Analysis of Numerical Weather Predictions," with the following conditions:

1. Include the complete bibliographic citation of the original source.
2. Include the following statement with that citation: **© American Meteorological Society. Used with permission.**

If you have any questions or need additional information, please feel free to contact me.

Best,

[Signature]

**Ms. Erin Gumbel, she/her/hers**

Senior Peer Review Support Associate

Senior Permissions Specialist

egumbel@ametsoc.org




617-226-3926

Hand-written signature removed for display in this thesis.





# Topographic Visualization of Near-surface Temperatures for Improved Lapse Rate Estimation

Kevin Höhle , Timothy Hewson , and Rüdiger Westermann 

**Abstract**—Numerical model forecasts of near-surface temperatures are prone to error. This is because terrain can exert a strong influence on temperature that is not captured in numerical weather models due to spatial resolution limitations. To account for the terrain height difference between the forecast model and reality, temperatures are commonly corrected using a vertical adjustment based on a fixed lapse rate. This, however, ignores the fact that true lapse rates vary from 1.2 K temperature drop per 100 m of ascent to more than 10 K temperature rise over the same vertical distance. In this work, we develop topographic visualization techniques to assess the resulting uncertainties in near-surface temperatures and reveal relationships between those uncertainties, features in the resolved and unresolved topography, and the temperature distribution in the near-surface atmosphere. Our techniques highlight common limitations of the current lapse rate scheme and hint at their topographic dependencies in the context of the prevailing weather conditions. Together with scientists working in postprocessing and downscaling of numerical model output, we use these findings to develop an improved lapse rate scheme. This model adapts to both the topography and the current weather situation. We examine the quality and physical consistency of the new estimates by comparing them with station observations around the world and by including visual representations of radiation-slope interactions.

**Index Terms**—Topographic Visualization, Surface Temperature, Spatio-temporal Data.

## 1 INTRODUCTION

One of the most important parameters for weather forecast users is temperature. Ordinarily, and by convention, this means "2 m temperature" – i.e., measured  $\sim 2$  m above ground. Numerical weather prediction models are generally good at forecasting 2 m temperatures over flat terrain but can struggle elsewhere. This is because 2 m temperature depends strongly on altitude and because (away from plains) the altitude of a selected site does not generally equal the altitude of the corresponding numerical model region. Numerical models have a finite spatial resolution, and within a grid box, all terrain is implicitly at the same height. For instance, the global model of the European Centre for Medium-range Weather Forecasts (ECMWF) operates on grids with a box size of approximately 9 km by 9 km, which is far too coarse to resolve topographic details. Figure 1 illustrates this by comparing terrain representations of the same geographical region at different resolutions. Shown is the terrain around Mont Blanc, as an example, at 9 km and 1 km resolution. Topographic extremes are smoothed out significantly or dismissed entirely in the coarser representation.

*Downscaling methods* are needed to correct the model outputs and remove the low-res bias. An introduction to the goals and principles of basic downscaling methods can be found, e.g., in Wilby and Wigley [55]. To correct near-surface temperatures for terrain altitude mismatches, a common approach is modeling the local lapse rate – i.e., the rate of temperature change with height – and using the terrain height difference as a multiplier for this to estimate the required correction. Simple correction schemes apply a fixed lapse rate such that, e.g., temperatures always drop, going upwards, by 0.65 K per 100 m, or 6.5 K/km, as specified in the standard Atmosphere, defined by the International Civil Aviation Organization (ICAO) [22].

In practice, however, lapse rates vary a lot in different situations. In sunny weather, near-surface air tends to be markedly warmer than air higher up. The temperature drop rate then reaches around 10 K/km before the air masses become unstable to convection. Indeed, values of 12 K/km can be reached temporarily during strong insolation.

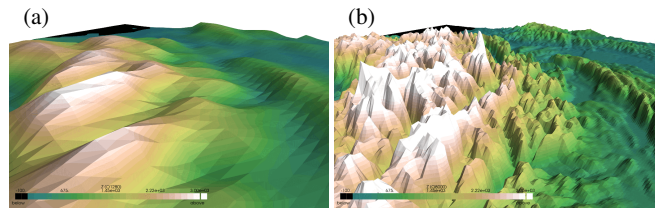


Fig. 1: Comparison of the orography around Mont Blanc at different resolutions. (a) Average grid spacing 9 km, as used in the ECMWF medium range model; (b) Average spacing 1 km.

In other scenarios, the drop rate can fall below 5 K/km and even reverse its sign. Weather situations where the temperature rises with increasing altitude are called inversions. In calm and clear weather conditions, inversions can lead to positive temperature gradients of, say, 100 K/km across tens or hundreds of meters. Diurnal variations in lapse rate are also commonplace, especially in light wind situations with predominantly clear skies. Figure 2 illustrates temperature profiles associated with regular and more unusual weather conditions.

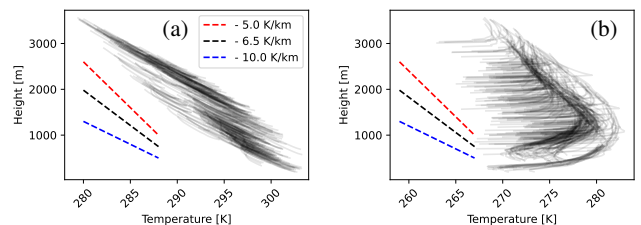


Fig. 2: Vertical temperature profiles in the bulk atmosphere above selected grid points in the domain of Figure 1. Each line represents the temperature profile over one grid point. Dashed lines are shown for reference. (a) Temperature profiles on a summer afternoon (July 23, 2021, 1400 UTC); the profiles are dominated by regular negative temperature gradients. (b) Inversion situation on a winter morning (December 19, 2021, 0600 UTC); at the lower end of the profiles, the temperature increases with altitude, indicating an inversion situation close to the earth's surface. Towards the upper end of the profiles – i.e., away from the surface – the regular temperature stratification is resumed.

- Kevin Höhle is with Technical University of Munich. E-mail: kevin.hoehlein@tum.de.
- Timothy Hewson is with the European center for medium-range weather forecasting (ECMWF). E-mail: Timothy.Hewson@ecmwf.int.
- Rüdiger Westermann is with Technical University of Munich. E-mail: westermann@tum.de.

In this study, we set out to improve upon the fixed lapse rate assumption by using 3D topographic visualizations to assist in understanding forecast-observation mismatches and to evaluate and optimize an alternative adjustment approach based on dynamically varying lapse rates. Through analysis of the immediate impact of adjustments to lapse rate parameters on model temperature output, one can clarify how, when, and where there is a high sensitivity, obtain enhanced process understanding, and eventually improve the predictions. On the one hand, this requires a methodology to effectively locate sensitive behavior precisely in space, but on the other, there is a need to contextualize relative to the geospatial frame and simulated atmospheric processes. To achieve both, we present topographic visualizations to assess the sensitive behavior of estimated 2 m temperature to lapse rate definitions and reveal relationships between sensitive regions, features in low- and high-res orography, and the vertical temperature distribution in the near-surface atmosphere.

## Contribution

We propose an interactive visualization workflow to

- picture how downscaled 2 m temperature varies with height,
- identify common limitations of the current lapse rate scheme, including its topographic dependencies in the context of the prevailing weather conditions,
- compare low-res and station site near-surface temperatures and visualize temperature distribution in the near-surface atmosphere,
- assess the downscaling accuracy by comparing with observations.

We use the proposed workflow for analyzing near-surface temperature, downscaled from model data, i.e., orography and hourly temperature fields, at (e.g.) 9 km spatial resolution to 1 km. Our work is motivated by the following analysis questions relevant to different user groups in meteorology:

- To what degree do 2 m temperature values corrected with a fixed lapse rate assumption make physical sense in different topographic/meteorological settings? (Q1)
- When fixed-lapse-rate-based correction fails, how do the resulting errors relate to temperature distribution in the near-surface atmosphere? (Q2)
- What is the impact of changing the lapse rate formula, and how well can such alternatives correct the 2 m temperature? (Q3)
- What is the accuracy of 2 m temperature fields, corrected in different ways, relative to independent surface station observations, scattered across the terrain and expected to be imperfect due to measurement and metadata errors in different classes? (Q4)

Q1 to Q4 are relevant to scientists working in postprocessing and downscaling of numerical model output, as well as forecasters and specialist forecast users where the emphasis shifts more to real-world scenarios such as, e.g., operating a ski resort. The code for the project is publicly available [21]

## 2 BACKGROUND AND RELATED WORKS

Our dynamic lapse rate scheme was motivated by work by Sheridan et al. [46, 47]. These studies focused on small regions of the British Isles, using data from a limited area model as input. Here, we expand to use a global domain instead. In Sheridan et al. [46], the input data spatial resolution ( $L$ ) = 4 km, whilst in our case  $L$  = 9 km. Our data is sourced from the operational global medium-range weather forecasts produced by the ECMWF, where one author is based. This step down in resolution presents some different challenges, although the overarching scientific hypothesis we use is similar.

Regarding output data, Sheridan et al. [46] present values at a set of measurement sites in a region of England; here, our aim is instead to expand substantially by devising methods that are scalable to very-high-resolution *global* grids (1 km in this study, see section 3), whilst still using high-density observations for verification. Sheridan et al. [47] describe some complex refinements to the earlier study, such as the

representation of cold air pooling in valleys. In this study, those are not explicitly used, partly for simplicity and partly because topographic characteristics worldwide are much more diverse than in the UK. From the application perspective, our main aim is to improve upon the standard lapse rate assumption to deliver better forecasts whilst keeping the new method explainable and intelligible for users.

### 2.1 Downscaling surface temperatures

Given low-res model outputs and higher-res orography fields, common operational downscaling schemes use horizontal interpolation (or "nearest neighbor") in combination with a vertical correction according to a constant lapse rate linked to the ICAO standard atmosphere [22]. Especially in regions with complex orography, this simplistic approach leads to physically implausible or inaccurate predictions, which must undergo further postprocessing to produce useful predictions (cf., e.g., Fiddes and Gruber [12]).

More elaborate downscaling and interpolation approaches have been developed (e.g., [13, 19, 27, 28]) but often come with a significant compute footprint (as in full dynamical downscaling – e.g., two of the three methods in [25]) or data requirements and are therefore difficult to deploy to global-scale applications. Several studies rely on observations (rather than model data) as input and derive a range of complex techniques for handling those [13, 19, 28]. Whilst these engender more vertical lapse rate complexity and potentially greater accuracy, the techniques are considered too involved for global application. Also, the needed high-density observations are missing in most parts of the world. Numerical methods for downscaling meteorological variables to sub-grid resolution use orography-related predictors at high spatial resolution [11], downscale temperatures by interpolating pressure-level data [12], or quantile mapping is applied in postprocessing to compensate for station-wise statistical biases [10]. The authors of [13] suggest fitting nonlinear vertical temperature profiles with compact parametric forms emulating the vertical variation in temperature. The approach uses a two-step procedure by first estimating a background temperature field on coarse spatial resolution, which is then superimposed with the vertical variation. So-called optimal interpolation methods have been applied to derive high-res temperature maps from high-res observation networks in the Alps region [52] and Norway [28]. The authors of [27] propose to reduce elevation-related biases in reanalysis datasets using an elevation correction method with internal lapse rates derived from different reanalysis pressure levels.

### 2.2 Meteorological map visualisation

Analyzing the spatio-temporal distribution and relations between atmospheric variables, measured and numerically simulated, is at the core of meteorological data visualization. Central to this task is the use of visualizations that can simultaneously provide views of topographic information to reveal geospatial information, such as station locations and terrain as spatial frames of reference [2], and the physical variables on the terrain and in the surrounding atmosphere.

The process of meteorological map-making was discussed by Monmonier [33], and the review by Stephens et al. [49] focuses on probabilistic information communication in atmospheric sciences. In several follow-up summary reports, the tools and techniques in climate and weather research have been reviewed [31, 34]. More recently, Rautenhaus et al. [40], Aftal et al. [1], and Roeber et al. [42] have provided overviews of atmospheric data visualization, including taxonomies of techniques, discussions of differences between operational use and research, as well as specific approaches in climate science.

Advice on the generation of meteorological maps to enable effective human comprehension of the displayed data is given in the book by Hoffmann et al. [20]. In operational settings, 2D surface maps using color coding of temperature in combination with contour lines of surface pressure are still most often used. These maps are augmented by glyphs to indicate station data and linked to domain-specific diagrams. Pressure level charts often visualize the 500 hPa level via 2D maps to represent atmospheric flow at the mid-troposphere. In this context, especially the effectiveness of visual attributes such as color has been studied [48, 51]. Improved readability and better communication of

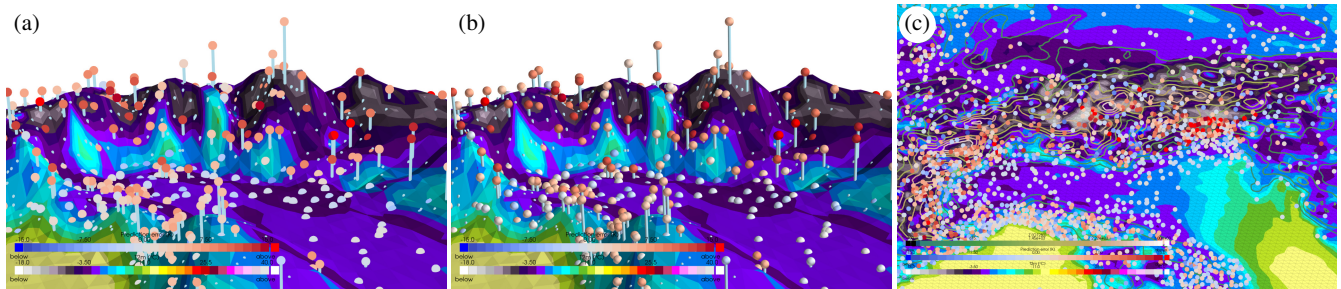


Fig. 3: Low-res orography with near-surface temperature encoded in color. Station locations are shown as spheres, with their color displaying temperature differences. Vertical lines below station sites provide a reference for locating the stations in the 2D domain. Dots on the terrain suggest the presence of stations below the surface at this location. The shading of the spheres can be toggled off (a) or on (b) to enhance the readability of the color scale. A 2D map-like view (c) is obtained by toggling a bird view with parallel projection along the elevation axis.

quantitative meteorological variables have been reported with the perceptual linear hue-chroma-luminance color space. An evaluation of maps and additional climate-specific visualization was pursued by Dasgupta et al. [6], who provide a list of design guidelines for color and visual saliency. Duebel et al. [7] discuss the visualization of geospatial data on 2D height fields, i.e., terrain fields, and provide means to visually communicate simultaneously the terrain field and data, including data-associated uncertainty.

Driven by the use of numerical ensemble simulations in atmospheric science, the visualization of uncertainty has become a major research area in recent years. Prevalent to many of the existing ensemble visualization techniques is the question of visually conveying the ensemble spread of atmospheric variables from different numerical simulations. Guidance on the mapping of uncertain variables was provided by Kaye et al. [24] and Retchless and Brewer [41], for instance, to combine color and pattern for visualizing climate change parameters with uncertainty. The survey by MacEachron [29] focuses explicitly on uncertainty in geospatial science and cartography. Griethe et al. [14] categorize uncertainty visualization into intrinsic and extrinsic techniques, depending on whether existing graphical representations are modified to convey the uncertainty or additional graphical primitives are added. Several summaries shed light on the sources and models of uncertainty [3, 23, 38, 53], including categorizations of uncertainty visualization techniques depending on whether stochastic uncertainty models or ensembles are used. Representative examples of atmospheric data visualizations, to name just a few, are statistical summaries [17, 37], spaghetti plots [43], contour box plots [54] and streamline variability plots [9]. Most similar to our method for visualizing confidence information across the terrain are visualizations of the effect of uncertainty on the position and structure of isosurfaces, e.g., by using surface displacements [15] and confidence surfaces [35, 36, 56].

### 3 DATASETS

The dataset for our study comprises global temperature prediction data generated by the medium-range prediction system at the ECMWF. Data are available on a (cubic octahedral) reduced Gaussian grid [30] (O1280) with global coverage and average grid spacing of 9 km. Model predictions are retrieved for 2 m temperatures and volumetric temperatures for the 20 lowest model levels. As the model operates on terrain-following hybrid levels, volumetric visualizations require the precomputation of the local geometric altitude of the respective model levels. While the accurate elevation levels usually depend on the pressure and humidity distribution of the weather situation, the altitude of the lowest model levels is only marginally affected by such variations. Therefore, we approximate the model levels using a standard atmosphere assumption. This procedure does not affect the quantitative evaluation of the proposed methodology, as model-level data is not used here. We have verified that the difference between approximate and physical model levels is imperceptible for the visualizations. Hourly data are available for the time period from April 1, 2021 to March 31, 2022, resulting in a total size of the dataset of approx. 2 TB. For

quantitative analysis, we use the full dataset. For visualizations, case studies are selected based on meteorological prior knowledge (see subsection 5.1). As a test case for a region with complex topographic structure, we select a geographic region between  $43^\circ$  and  $49^\circ$  latitude, as well as  $4^\circ$  and  $18^\circ$  longitude. The region is located in central Europe and covers the Alps mountain range and parts of northern Italy.

In addition to the model elevation field, we use a high-res orography dataset with 1 km average grid spacing (O8000) provided by the ECMWF and composited together from the following sources: SRTM30 for 60S to 60N [8]; GLOBE for the north of 60N [16], RAMP2 for the south of 60S [26], BPRC for Greenland [5], IS 50V for Iceland [45]. A high-resolution land-sea mask is computed by downsampling a watermask at 100m resolution [32] to the O8000 grid.

Global near-surface temperature observations are retrieved from the HDOBS database of the ECMWF, comprising 86 Mio. records of surface temperature, station location, and station elevation from more than 16000 weather stations worldwide (see [39]). Missing values and faulty observations reduce the number of valid records to 65 Mio. observations at 14500 station sites. Data records cover the time between April 1, 2021, and March 31, 2022, and are generally available multiple times a day, with frequency depending on each station's schedule.

## 4 METHODS

The topographic visualization workflow by which we address the requirements from meteorology features three different visualization options: the terrain map, including station data, the atmosphere layer, and the elevation variability plot. Additionally, control panels facilitate data selection and interaction with data processing and display. The result of applying our improved lapse rate scheme for temperature correction can be compared directly to station data. The tool is implemented in Python, using the visualization library PyVista [50], providing Python bindings to the visualization toolkit (VTK) [44]. The graphical user interface is based on Python bindings of Qt5.

### 4.1 Terrain map

The terrain map panel displays the 3D terrain field augmented by intrinsic and extrinsic visual encodings [14] of additional information like the temperature distribution, orography, or land cover. It provides an interactive environment allowing one to switch between low- and high-res terrain, showing differences in height between them and showing differences between the low-res surface temperature and ground truth station data. Via zooming, this interaction facilitates analysis at both global and regional scales.

Figure 3 shows visualizations of the low-res terrain map with color coding of temperature. For temperature fields, meteorology users demand color maps that are compatible with the ones used in operational forecasting. Specifically, temperature maps should clearly distinguish temperatures with a resolution of around 2 K and follow a predefined listed color map. For lapse rate-related color maps, the user prefers color schemes that respect physical prior knowledge and allow for a

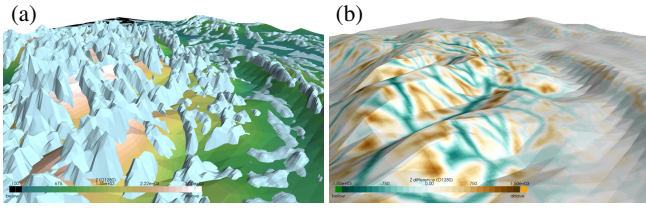


Fig. 4: Visualizing two terrains simultaneously. (a) High-res orography occludes low-res geometry. (b) Color coding elevation difference on the low-res domain helps to display positive and negative offsets equally.

simple comparison against the default lapse rate of  $-6.5$  K/km. We address this by employing diverging color maps with configurable opacity functions (see subsection 5.1).

Temperature is encoded intrinsically by mapping the respective fields to color. We employ an extrinsic encoding on spheres embedded into the terrain to visualize stations and temperature measurements or differences. We provide two different visualization options:

I) Spheres can be colored with a constant color indicating temperature or difference but without applying shading. This gives the most unobscured display of temperature values, yet it makes comparison to near-surface temperature on the terrain – whose colors are modulated by surface shading – difficult (see Figure 3 (a)). Constant sphere coloring, however, is advantageous if a 2D map view is generated by looking from above the terrain using an orthographic projection and showing only unmodulated surface colors (see Figure 3 (c)).

II) Spheres can be shaded according to the selected lighting conditions to let spheres stand out less in the visualization (see Figure 3 (b)). We use the 3-light illumination model provided by PyVista, which simulates multiple lights to realize shading without letting certain sphere parts become too dark. In either case, we use additional lines to emphasize how much above or below the terrain a station is located. Note here that while accurate station positions are available, the terrain – regardless of whether it is the low- or high-res version – never represents orography perfectly. Since some stations are located below the terrain, the user can use transparency for the terrain surface to let these stations shine through, change the camera position, or invert the station offset to show stations below the surface on the opposite side. The line color is chosen to stand out against the white background and diverge from the colors in the temperature color map toward the extreme temperatures. Optionally, the user can switch to an alternative lighting mode, which simulates parallel sunlight. This is useful for investigating factors that are potentially impacting temperature anomalies (see subsection 5.1).

To show the height differences between the low- and high-res terrain, both geometries can be visualized simultaneously, with the common perceptual problems arising from such a visualization like occlusions and clutter (see Figure 4 (a)). Note that in our use case, the low-res terrain field is needed to show the relation between simulated temperatures and either measured temperatures at stations or differences between measured and corrected temperatures at stations. The high-res field is needed to show the relation between station temperature mismatches and high-res orographic features. Thus, the domain expert usually does not use the option to show both terrains in one single view. Overall, a direct depiction of the low-res terrain can be preferable due to its simpler structure. Then, to indicate the high-res orography, height differences can be encoded via color; this is shown in Figure 4 (b).

## 4.2 Atmosphere layer

The user can visualize atmospheric variables in a volumetric layer over the terrain via the atmosphere layer. This functionality was deemed important by domain experts because it shows the relationship between lapse rate, orographic features, and the 3D temperature profiles, which are predicted by the numerical weather model.

To enable such visualizations, the terrain-following model grid on which a variable is given is loaded and can be shown as a colored wireframe (Figure 5 (a)). Then, the gridded data can be rendered via

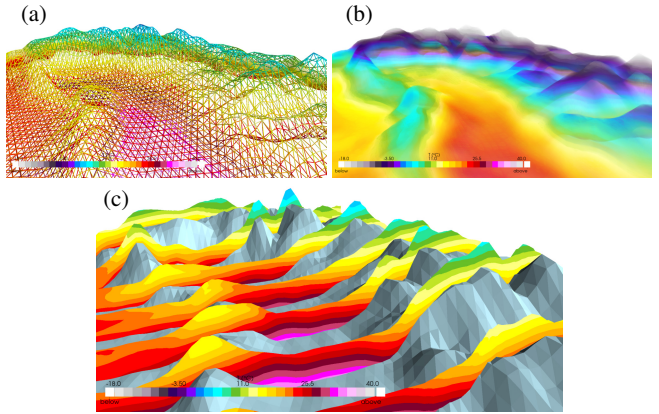


Fig. 5: (a) The 3D terrain-following model grid as wireframe. (b) Direct volume rendering of the 3D temperature field. (c) Slicing the 3D temperature volume.

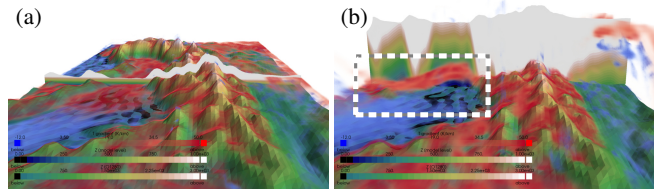


Fig. 6: Direct volume rendering of vertical temperature gradient field, including a slice plane encoding true elevation via color. (a) No offset scaling. (b) With offset scaling. While the atmosphere layer and slice are scaled, the terrain remains unchanged. Here, the red density feature in the left-hand valley (see box) shows no connection to the valley bottom, in contrast to the red areas on top of the mountains. Colors on the reference slice indicate the red feature at around 250m altitude.

volume ray-casting, with or without the terrain (Figure 5 (b)). While volume visualization can provide a rough overview of the temperature profiles of bulk atmosphere, it hinders a fine granular analysis due to the typical attenuation and blending effects inherent to volume rendering.

To enable a more unoccluded view of the values of a gridded variable, we offer the possibility to select 2D slices oriented parallel to the 2D domain over which the terrain heightfield is given or aligned vertically along the latitudinal or longitudinal direction. Slicing is shown in Figure 5 (c). Importantly, since the slices can be moved along their respective orthogonal direction, they can be positioned to capture certain orographic features. From the color coding of temperature on the 2D slices, the temperature distribution in the near-surface atmosphere can be revealed effectively.

We provide visualizations of the vertical temperature gradient to further shed light on the local weather situation, e.g., to indicate the strength of inversions or cooling/heating effects over ground and water. Figure 6 shows a volume rendering of this temperature gradient field overlaid on the terrain. The elevation of the sliced terrain is color-coded onto a vertical slice. Since the atmosphere layer is very narrow, the gradient distribution cannot be perceived well. To mitigate this problem, we enable offset rescaling, i.e., vertical scaling of only the extrinsic parts of the visualization, such as the volumetric grid and the slice, relative to the terrain altitude.

## 4.3 Elevation summary plots

Both the visualization of stations over the terrain and vertical slices through the 3D temperature field can be seen as a vertical reference indicator relative to the model orography. We provide another type of visualization in a reference frame that is defined by an elevation quantile

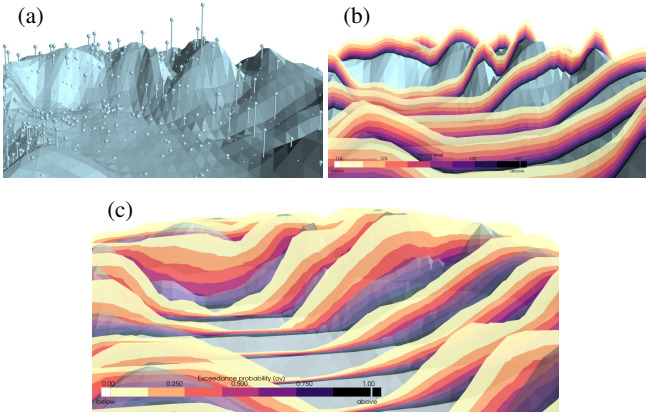


Fig. 7: Overview of a vertical reference frame relative to the model orography. (a) station sites, (b) terrain-following model levels, and (c) elevation summary quantile levels with a summary radius of 60 km. The vertical coordinate of the volumetric grids is shown as color code on vertical slices through the volume. The elevation volumes stretch on both sides of the terrain, whereas the model levels are located exclusively on the upper side.

coordinate. We call these *elevation summary plots* (see Figure 7).

Elevation summary plots share similarity with confidence surfaces indicating quantiles with respect to a mean surface. Such approaches typically build upon the presence of a stochastic uncertainty model or a set of ensembles from which the required statistics can be computed. In our scenario, we derive such plots for either the low- or the high-res terrain to assess respectively the elevation statistics of the lapse rate algorithm or to examine the expected sub-grid variability below the resolution of the low-res terrain.

Firstly, the user sets a search radius in units of kilometer to be considered in the statistics computation, as well as the number of quantiles to compute. For each vertex of the low-res terrain, all vertices from the so-called summary grid (the high-res or the low-res terrain surface) that are closer than the specified radius are retrieved. Note here that the distance between vertices is considered in the 2D domain over which the terrain is defined. From the elevation values of all retrieved vertices, the minimum and maximum elevation, as well as uniformly spaced quantiles of the elevation distribution according to the user-set value are computed. The quantiles are interpreted as z-coordinates of a new volumetric grid (different from the model levels), on which scalar quantities can be computed and displayed using volume visualization. In Figure 7 (c), for example, the vertical extend of the volume slices is determined by the range of minimum and maximum surface elevation in a radius of 60 km around the reference point.

Quantile computation is repeated for the area covering the interquartile range from 25% to 75%, yielding another volumetric mesh. On this mesh, special quantiles can be shown independently as surface meshes, e.g., the median surface and iso-layers of the 25% and 75% quantiles (IQR bounds). By analogy with statistical box-and-whisker plots, two additional surfaces are defined through the local elevation values:

$$\begin{aligned} z_{\text{upper}} &= z(75\%) + f \cdot \text{IQR}, \\ z_{\text{lower}} &= z(25\%) - f \cdot \text{IQR}, \\ \text{IQR} &= z(75\%) - z(25\%) \end{aligned} \quad (1)$$

The upper and lower whisker heights are then given as the largest, respectively, smallest elevation sample that falls inside the range between  $z_{\text{upper}}$  and  $z_{\text{lower}}$ . Figure 8 illustrates the components of the elevation summary. In Figure 8 (a), the median field is shown together with the summarized grid. In (b), the elevation summary is augmented by surfaces that visualize the IQR as well as the whisker levels. In (c), we show slices through the volume that is spanned by the minimum

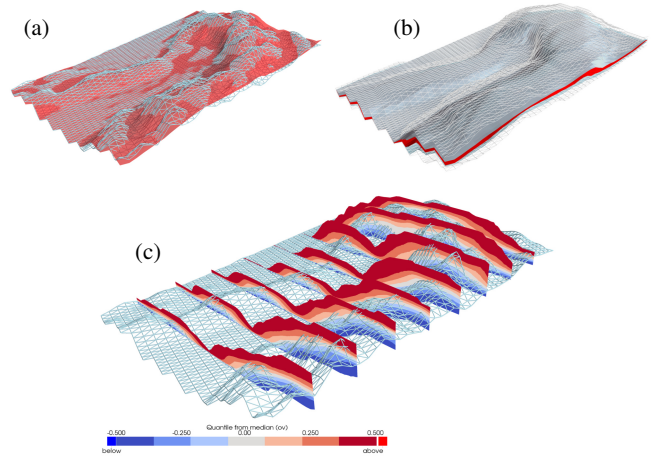


Fig. 8: Elevation summary plots for the O1280 model grid with a summary radius of 60 km. (a) model grid (wireframe) with the median surface, (b) elevation boxplot with the median surface (red), IQR bounds (blue), and whiskers, and (c) model grid (wireframe) with slices through the elevation summary volume. Color on the slices indicates quantile isolevels.

and maximum elevation levels of the local environments and use the slices to display statistical information. All summary elements can be rendered jointly with stations or high-res orography, to obtain a notion of outliers in terms of local orography. For instance, the surface summary can be rendered jointly with the prediction error encoded on station sites to find stations that have high errors due to deviation.

The provided visualization methods can be used to assess the quality of the current lapse rate scheme (see subsection 5.1) by showing the error between the corrected temperatures at stations and the temperature that was measured at these sites. As we will show in our case study, the constant lapse rate scheme introduces significant deviations to the measured temperatures. Thus, in the following we suggest an alternative lapse rate scheme that results in improved temperature corrections.

#### 4.4 Improved lapse rate scheme

Instead of a fixed global lapse rate, we propose to compute a local lapse rate from the current model data. The method starts with estimating the local lapse rate at the vertices (gridpoints) of the low-res model grid. To do this, the domain expert sets a radius that corresponds to a "reasonable" scale in terms of local weather conditions, typically a value between 40 and 60 kilometers. From all vertices (excluding non-land vertices) within the region indicated by the set radius, all 2 m temperatures and elevation values are collected. Through a linear model that fits predicted temperature as a function of elevation, we obtain a local weight coefficient relating elevation to temperature. This coefficient is used as the local lapse rate estimate. To avoid spatial discontinuities as a consequence of using a hard cutoff radius, a Gaussian weighting scheme is applied, which assigns a higher weight to the samples close to the reference location. Validity of the local linear model is assessed via the coefficient of determination, also called  $R^2$  score. To guarantee proper convergence of the estimator, lapse rates are estimated only for grid vertices, which have at least 20 non-sea grid vertices within their radial neighborhood. For other sites, the scheme reverts to the default lapse rate.

To circumvent the potentially harmful effects of extreme lapse rate estimates, min-max clamping is applied, in which the upper and lower bounds depend on  $R^2$  of the local linear model. We provide an interface to parametrize upper and lower bounds  $b_{\text{up/low}}$  as a ramp function:

$$b = \begin{cases} c_{\text{lower}} & \text{if } R^2 < r_{\text{lower}}, \\ c_{\text{upper}} & \text{if } R^2 \geq r_{\text{upper}}, \\ c_{\text{lower}} + \frac{c_{\text{upper}} - c_{\text{lower}}}{r_{\text{upper}} - r_{\text{lower}}} (R^2 - r_{\text{lower}}) & \text{else.} \end{cases} \quad (2)$$

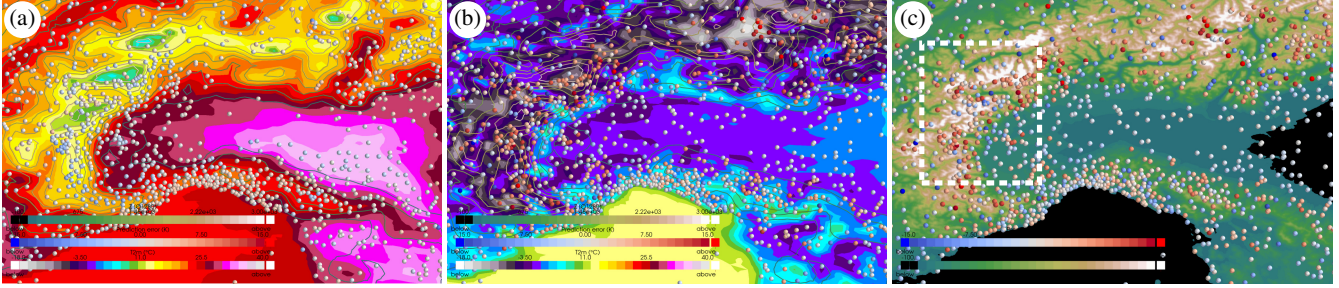


Fig. 9: Overview of prediction errors and model surface temperatures for (a) summer and (b) winter case study, and (c) shows the winter prediction errors in the context of the model orography. The box indicates the focus region of figures 10 and 11.

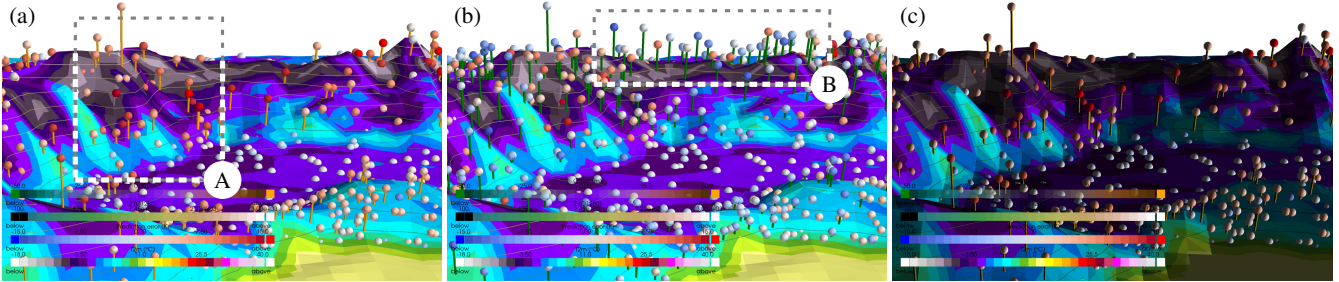


Fig. 10: Northward overview of prediction errors and model surface temperature for the winter case, (a) in regular view on the mountain stations, (b) with inverse station offset to display valley stations, and (c) with solar lighting, highlighting vertical structures in the model temperature field.

Default values for the parameters are selected empirically (see subsection 5.3) and are set to  $(c_{\text{lower}}, c_{\text{upper}}, r_{\text{lower}}, r_{\text{upper}}) = (20 \text{ K/km}, 50 \text{ K/km}, 0, 1)$  for the upper lapse rate bound, and  $(-6.5 \text{ K/km}, -11 \text{ K/km}, 0.75, 0.95)$  for the lower bound. Based on the estimated lapse rate, a corrected prediction for each location of interest (grid vertex or station site) is then obtained as  $T_{\text{site}} = T_{\text{model}} + \gamma(z_{\text{site}} - z_{\text{model}})$ , wherein  $T_{\text{model}}$ ,  $z_{\text{model}}$ , and  $\gamma$  are 2 m temperature model prediction, model grid elevation, the local lapse rate estimate at the closest vertex in the model grid, and  $z_{\text{site}}$  is the station altitude.

## 5 USE CASE

### 5.1 Visual analysis of near-surface temperature correction

To obtain an impression of the limitations of the constant lapse rate approach (**Q1 + Q2**), two timestamps are selected, for which model predictions are visualized jointly with station observations. The cases are indicative, over central Europe, of the two ends of the lapse rate spectrum. That is a meteorologically less stable situation, from a summer afternoon, July 12, 2021, 1500 UTC, and a meteorologically more stable situation from an anticyclonic winter morning, December 19, 2021, 0600 UTC. Both featured light near-surface winds, as stronger winds tend to promote more standard lapse rates via turbulent mixing. Furthermore, due to the low-level inversion / high stability, the winter case had proved especially challenging for operational model 2 m temperature forecasts and reanalysis (Figure 3 in [4]).

Figure 9 (a) and (b) display top-down map views for both cases. A constant-altitude projection is used to enable the display of stations above and below the terrain at the same time. The color of station indicators encodes the prediction errors of the standard lapse rate scheme at the station site. Orography is displayed as isocontours to reveal relations between temperature distribution and terrain properties. Station indicators and terrain isocontours are shaded as volumetric spheres and tubes, respectively, to improve their perception in front of the bright model temperature map in the background. Temperature isocontours are visible as color boundaries in the temperature color map.

In the summer case, temperature isocontours adhere closely to the contour lines of the orography field, both in orientation and spacing. This indicates that the constant lapse rate assumption is a good approxi-

mation of the weather situation. Correspondingly, the prediction errors, encoded in sphere color, are generally no more than a few Kelvin.

In the winter case, the prediction errors are much larger, especially in the mountainous areas. This suggests that the constant lapse rate scheme does not yield skillful predictions here. The deviation of the temperature conditions from the idealized model is also confirmed by the relation of the temperature and orography isocontours. In parts, the isocontours intersect orthogonally, indicating that temperatures do not change with altitude at all. To examine the relation between orography and prediction errors in more detail, the model temperature map is replaced with a map representation of the high-res orography in Figure 9 (c). The view reveals even more clearly that the magnitude of prediction errors correlates with the mountainous character of the terrain. Additionally, the user may recognize that stations with large prediction errors are located mainly in places with right-facing mountain slopes (i.e., on eastern slopes). An area where this is particularly apparent is highlighted with a white box in Figure 9 (c) and examined in more detail in a 3D view in Figure 10.

In Figure 10 the surface color is used to encode model 2 m temperature. Station sites are shown on top of tube lines, which are colored according to the elevation difference between station and model terrain. Stations with an elevation difference above 50 m are classified as convex (mountain) stations, whereas stations more than 50 m below the model orography are classified as concave (valley) stations. Figure 10 (a) displays station sites above the model orography, and suggests that convex stations exhibit large positive discrepancies versus the model predictions (see box A). The offset inversion feature is then used to switch to the view of Figure 10 (b), in which convex stations change positions with concave stations. In contrast to the convex stations, the view now suggests that valley stations on the backside of the mountain ridge (see box B) have a tendency to show lower values than predicted.

Considering the timestamp of the weather situation, 0600 UTC, corresponding to 0700 CET – the local time in the area of interest – the temperature anomalies may be caused by the formation of cold air pools in mountain valleys for various reasons: e.g. radiative cooling overnight, katabatic drainage into the valleys, mountains casting shadows onto the valleys by day. The latter aspect can be investigated by switching to solar lighting mode, which simulates a light source coming from the

direction of the solar irradiation. This option is seen in [Figure 10 \(c\)](#), indicating that this may indeed contribute to higher temperature measurements, as the sunlight is coming from the south-eastern direction.

[Figure 11 \(a\)](#) and [\(b\)](#) illustrate the distribution of model-level temperatures for the summer and the winter case using vertical slices through the temperature volume. The lighting of the slices has been turned off, and the density of elevation isocontours has been increased to achieve a visual contrast between the appearance of the terrain surface and the slice surfaces. For the user, this improves the possibility of accurately detecting the intersection between terrain and slice surfaces. Slicing is generally preferred over volume ray-casting due to the better run-time performance of the visualization, which enhances interactivity.

In the summer case, [Figure 11 \(a\)](#), the isolines of the volumetric temperature are mostly flat, supporting the validity of a constant lapse rate scheme again. The isolines curve slightly upwards at the intersection between slices and terrain (see box A1). This may be interpreted as a sign of warm air masses rising on the terrain surface as superadiabats form due to ground heating by solar irradiation. Further clarity on this could be obtained using additional information on near-surface air movement but it is not of primary importance for the lapse rates and is, therefore, beyond the scope of this work.

In the winter case, [Figure 11 \(b\)](#), the isolines near the surface indicate a significant temperature gradient between the lowest model level and the 2 m temperature. The gradient is manifested in a strong curvature of the isolines close to the surface (see box A2). To examine the vertical temperatures in more detail, [Figure 11 \(c\)](#) displays the same scene as [Figure 11 \(b\)](#), but with slice color encoding the vertical temperature gradient. The color mapping uses a dedicated diverging color scale, which respects the physical limits of plausible temperature gradients. The color scale is centered and  $-6.5$  K/km, corresponding to the value of the default lapse rate of the temperature correction model, and is capped at  $-12$  K/km due to the physical instability of air masses beyond that limit (10 K/km is the true standard for instability, using 12 K/km allows for some superadiabatic behavior at the 2 m level). A V-shaped opacity function is applied to minimize occlusions, which has a user-configured minimum opacity at the color scale center and higher opacity towards the extremes. By default, the visualization applies a linear slope for the opacity increase. However, to prune larger parts of the volume or increase the visualization's density, the user can switch to a polynomial opacity increase with exponents  $> 1$  or  $< 1$ . [Figure 11 \(c\)](#) shows clearly the complexity of the volumetric temperature field. Close to the surface, temperature gradients of more than 50 K/km are observed. Inside the valley (boxes B1 and B2), multiple air layers with alternating gradient signs are stacked on top of each other, clearly invalidating the assumption of a constant lapse rate. Only at higher altitudes do the temperature gradients revert towards the regular value of  $-6.5$  K/km.

## 5.2 Visual analysis of the adaptive lapse rates

To evaluate the quality and tune parameters of the proposed adaptive lapse rate scheme ([Q3](#)), we support the visual analysis of lapse rate scores and the associated clamping metrics  $R^2$  score. For this, the user selects the required hyperparameter in the graphical user interface, obtains handles to view the lapse rate estimates, and applies the clamping as a postprocessing step.

Visualizing the data is challenging since multiple aspects of the local model and terrain data determine the lapse rate. The primary input variables are the orography and the land-sea mask, which determines whether a grid vertex is used as a valid sample location. The threshold for evaluating the land-sea mask can be set interactively, but a value of 0.5 is a good default as this is implicit in the model formulation.

The radius parameter is more critical and is explored in [Figure 12](#). The figures display the range of elevation values used in computing the lapse rate estimate. Notably, the range size changes discontinuously, especially for radii 60 km and 90 km, when grid vertices with extreme elevation enter or fall out of the radius neighborhood (circle patterns in [Figure 12 \(b\)](#) and [\(c\)](#)). Using a hard cutoff radius also leads to a discontinuity in the lapse rate estimates and motivates the use of a distance-based Gaussian weighting scheme. This way, more weight

is put on vertices closer to the reference location during the lapse rate estimation. Empirically, a cutoff radius of 60 km in combination with a Gaussian weight scale of 30 km yields the best balance of stability of the estimates and feature resolution.

[Figure 13](#) displays lapse rate estimates for the summer case using these parameters. It appears that the estimator yields extreme lapse rate estimates. In [Figure 13 \(a\)](#), the estimates are displayed in the land-sea mask and orography context. It is seen that many of the extreme cases arise in the transition region between land and sea, where lapse rate estimation tends to be difficult, and adaptive lapse rate estimation using temperature and elevation samples is ill-defined due to the lack of orographic variation. Therefore, the scheme reverts to the default lapse rate for sea-site locations. One should also note that "extreme" lapse rates diagnosed in topographically almost-flat areas are inconsequential for 2 m temperature reconstruction, as the elevation difference multipliers are so small.

In [Figure 13 \(b\)](#), the volumetric display includes additional information about the lapse rate reliability metric  $R^2$ . Values of  $R^2$  are shown as isocontours on the terrain and are color-coded by their respective value. For additional context, an elevation range summary is added as vertical slices, indicating the elevation values range seen by the lapse rate estimator (cf. [Figure 12](#)). It can be seen that large values of  $R^2$  may occur both in regions with large elevation ranges and in areas with minimal elevation ranges. Especially in the latter regions, the resulting lapse rate estimates exhibit extremes, which appear unreasonably large and are more likely a numerical sampling artifact rather than evidence in favor of a positive or negative lapse rate different from the default. Like sea-site locations, such stations are handled by reverting to the default lapse rate.

Based on the user interface, the clamping of the lapse rates can be optimized. [Figure 14](#) displays maps with the clamped lapse rate encoded in the surface color and the raw lapse rate in the color of the mesh parameters. A difference in overall shade can be perceived, indicating more negative values for the summer case (blue shades) and more positive lapse rates (red shades) for the winter case.

## 5.3 Quality of the lapse rate estimator

Despite the possibility to visualize predictions and lapse rates, decisions on the parameterization of the lapse rate scheme have to be put on a statistical footing ([Q4](#)). The observation dataset is used for this purpose as follows. The observations are first grouped by the station that generated the data. Then, from the pool of ca. 14500 available stations, 20% are selected for parameter tuning, and the remaining are kept for testing. The station groups are sorted by the amount of available observations and split into consecutive groups of 5 stations each. From each group, one station is selected randomly, and the observations of this station are moved to the training dataset. This procedure yields training and test datasets with a similar average number of observations per station and minimizes information overlap between training and test data. Note that the fraction of training data is chosen low compared to other statistical estimation tasks. This is done because the low parameter count of the models justifies a reduction of training data. More data is available for verification.

Parameter tuning involves mainly the threshold settings of the clamping functions. For this, lapse rates and  $R^2$  scores are computed for all available observations and are divided into ten groups according to the value of  $R^2$ . For each group, a grid search is performed to identify the set of fixed upper and lower bounds on the lapse rate, for which the prediction accuracy is optimized. The parameterization of the clamping functions [Equation 2](#) is obtained by comparing and balancing bound parameters of all  $R^2$  groups.

For quantitative testing, lapse rates,  $R^2$  scores, and predictions are computed for observations from the evaluation dataset. The stations are grouped into ten bins according to the value of  $R^2$  and are classified further concerning their elevation difference against the model terrain. As in [subsection 5.1](#), stations more than 50 m below the model terrain are classified as concave (valley) stations, whereas stations more than 50 m above the terrain are called concave (mountain) sites. Stations in between are called neutral.

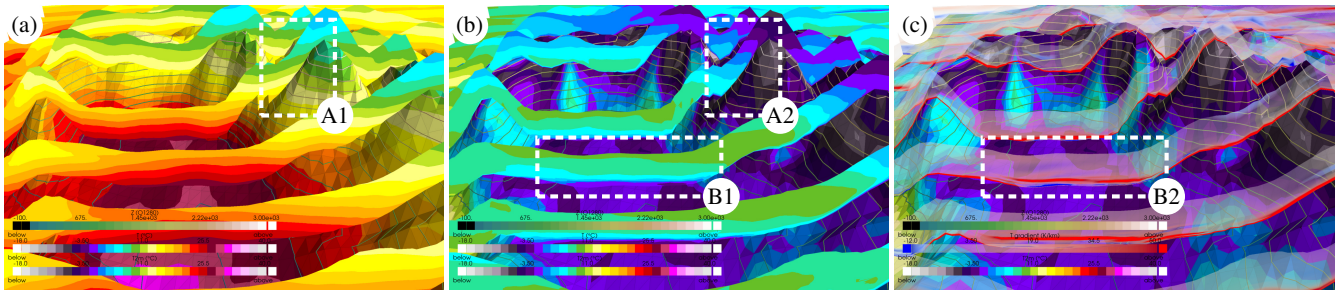


Fig. 11: Westward detail view of the temperature distribution on vertical slices through the model level volume, (a) using model level temperatures for the summer case, (b) using model level temperatures for the winter case, and (c) using vertical temperature gradients for the winter case.

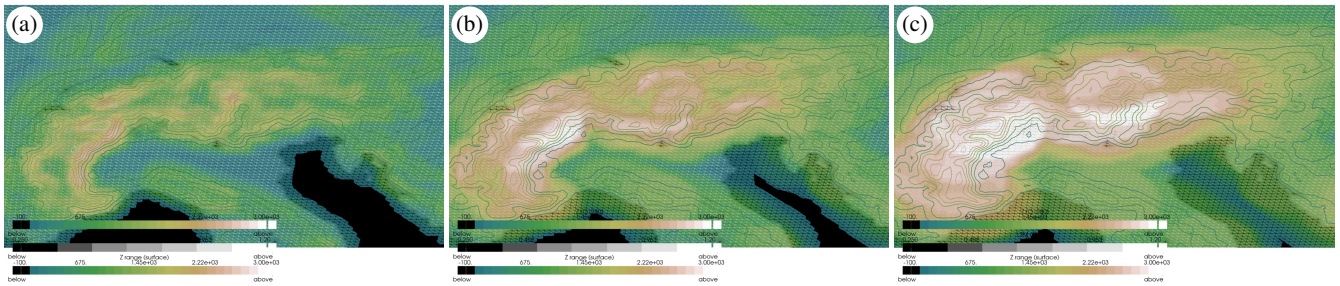


Fig. 12: Elevation range (surface color) and land-sea mask (grid lines) as seen by the lapse rate algorithm with radius settings (a) 30 km, (b) 60 km and (c) 90 km. Orography isocontours are shown for orientation.

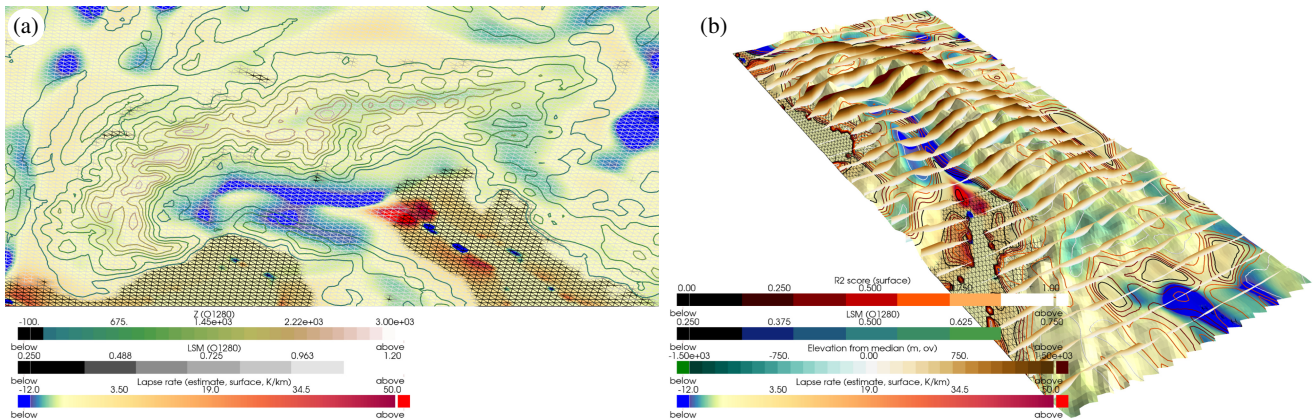


Fig. 13: Visualisation of summer-case lapse rates (a) in terrain context with orography contours and land-sea mask on grid lines, and (b) as a 3D view with clamping metrics terrain range and  $R^2$  score displayed on elevation summary slices and isocontours, respectively, as well as land-sea mask on grid lines.

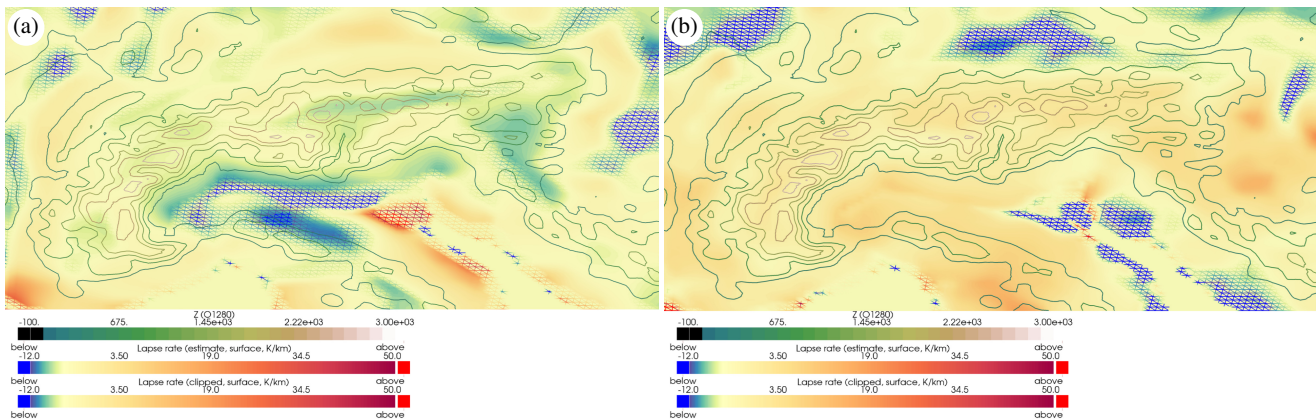


Fig. 14: Visualisation of lapse rates after clamping for (a) the summer case and (b) the winter case. Isocontours indicate the orography for orientation.



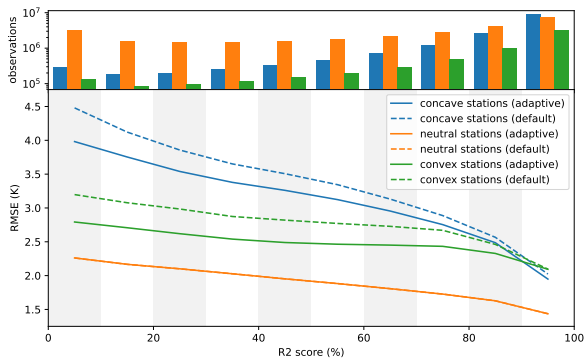


Fig. 15: RMSE prediction error of the adaptive lapse rate scheme compared to the default scheme. Observations are binned according to the observed  $R^2$  score. Histograms indicate the number of observations falling in each bin for valley-site, mountain-site, and neutral stations. In difficult local weather conditions (low  $R^2$ ), predictions are improved by 10-20% for both valley and mountain stations.

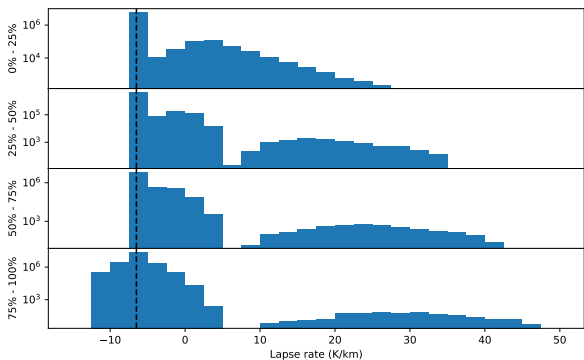


Fig. 16: Histograms of adaptive lapse rates after clamping for different ranges of the observed  $R^2$  score. Ranges are given on the vertical axes. The black dashed line indicates the default value of  $-6.5$  K/km.

Figure 15 shows the predictions' root mean squared error (RMSE) against the temperature observations. For both valley and mountain stations, the adaptive lapse rate scheme improves the prediction accuracy by between 10% and 20%. The histograms indicate the amount of observations falling into each bin during the evaluation.

Figure 16 visualizes the distribution of estimated lapse rates for different levels of the observed  $R^2$  score. The lapse rates are shown after clamping, which explains why no lapse rates with values below  $-6.5$  K/km are observed for the score groups below 75%. However, the mode of the distribution persists even in the most determined category. This is reassuring since the models often suggest physically sensible lapse rates. It is seen that with increasing determination coefficient, the lapse rates develop an increasingly bimodal distribution with growing variance. No lapse rates are observed between 5 and 10 K/km among the most determined models. The histograms demonstrate that the lapse rate scheme identifies weather situations in which the local lapse rates deviate from the default.

## 6 DISCUSSION AND CONCLUSION

We have applied existing and developed new topographic visualization techniques to assess the quality of lapse rate schemes in the context of low- and high-res orography. Through these techniques, relations between lapse rate quality, orographic features, and seasonal conditions could be revealed. The visualizations have helped to spot specific relationships that have been considered in developing an improved adaptive lapse rate scheme.

Figure 15 shows that our adaptive lapse rate scheme, on average, improves 2 m temperature RMSE in concave and convex locations. Such improvements are all the more striking given the relatively high frequency of cases where the dynamic lapse rate is similar to the standard value and errors are the same. Concave sites exhibit larger errors than concave in both lapse rate schemes, perhaps because of the particular challenge of cold air pooling (which attracted an additional postprocessing step in [47]). Such errors fall into the class of situation-dependant systematic model biases, which can be an Achilles heel for attempts to reduce downscaling errors [46]. Though not included directly in this study, a further global postprocessing step could be included in a future operational incarnation of our dynamical lapse rate adjustment. This is, for instance, to use the ECMWF postprocessing framework ecPoint [18] to apply, in tandem, situation-dependant grid-scale bias correction to raw model 2 m temperatures (Figure 3 (a), (b), (d) in [4] is an example).

An interesting question for future work is the role of outliers in the local environments. In the present study, lapse rate estimation is performed with linear regression models. Despite achieving considerable improvements in prediction skill, Figures 13 and 14 show that there exist weather situations in which the lapse rate scheme suggests extreme lapse rates beyond physical plausibility. The issues become more visible the smaller the environment radius is selected. Clamping has been identified as a countermeasure but may lead to information loss at the cost of reduced prediction accuracy. We believe that robust regression methods (which are more stable against outlying temperature and elevation samples than standard linear regression) may help to improve the accuracy further. As such models have their own intricacies and limitations, this aspect is left for future work.

A statistically robust extension of the proposed scheme would also be suitable for downscaling other meteorological variables. While we have focused on 2 m temperature in this study, rainfall is an equally important variable for forecast users in certain applications. Precipitation, too, can have a strong but variable topographic dependence. Therefore, a possible extension of this work would be investigating a similar dynamical lapse-rate scheme for rainfall downscaling, wherein "lapse rate" would reference rainfall rate or rainfall totals.

User motivation for the improved lapse rate scheme derives from the most-used graphical product out of the many produced operationally by ECMWF, namely meteograms. These display the range of possible forecast outcomes in the upcoming 10-15 days for a handful of key surface weather parameters, including 2 m temperature. The user selects a site, and then, via the fixed lapse rate assumption, adjustments are automatically made to deliver the meteogram 2 m temperatures for them. The uncertainty range bounds are derived from the use of multiple (ensemble) forecast realizations. Due to the computational simplicity of the adaptive approach, the novel scheme is perfectly portable to forecast ensembles. We envisage that an operational implementation of our approach would replace the fixed lapse rate assumption in this processing chain with a situation-dependent variable lapse rate.

The used visualizations show that while 3D representations often help and are even necessary to convey the relevant information, at the same time, they can hinder effective information communication due to occlusions and visual clutter when overloaded with too many additional, yet functional, visual mappings. Our analysis shows that 2D maps are still indispensable for topographic data visualization, especially in operational use. In the future, we intend to develop dedicated 2D map views to effectively convey the many aspects that need to be considered in operational forecast products, and we will equip them with linked 3D views to support specific investigations and/or representations.

Given that numerical models now represent the vertical structure of the atmosphere in great detail, one could imagine that extraction from a model profile would suffice for higher elevations. For relatively isolated peaks, this can work, but within the terrain, near-surface temperatures can be decoupled from the free atmosphere due, for example, to the sometimes strong influence of surface fluxes (see [47]). The 3D visualizations developed in this work will be helpful in investigating such aspects since they allow for a comprehensive view of different atmospheric conditions and their interplay with orography.

## ACKNOWLEDGMENTS

The authors acknowledge funding by the Munich Center for Machine Learning (MCML) initiated by the Federal Ministry of Education and Research and the State of Bavaria.

## REFERENCES

- [1] S. Afzal, M. Hittawe, S. Ghani, T. Jamil, O. Knio, M. Hadwiger, and I. Hoteit. The state of the art in visual analysis approaches for ocean and atmospheric datasets. *Computer Graphics Forum*, 38(3):881–907, 2019. 2
- [2] N. Andrienko and G. Andrienko. *Exploratory analysis of spatial and temporal data: a systematic approach*. Springer Science & Business Media, 2006. 2
- [3] G.-P. Bonneau, H.-C. Hege, C. Johnson, M. Oliveira, K. Potter, and P. Rheingans. *Overview and State-of-the-Art of Uncertainty Visualization*, vol. 37. 09 2014. 3
- [4] M. Bottazzi, L. Rodríguez-Muñoz, B. Chiavarini, C. Caroli, G. Trotta, C. Dellacasa, G. F. Marras, M. Montanari, M. Santini, M. Mancini, A. D’Anca, P. Mercogliano, M. Raffa, G. Villani, F. Tomei, N. Loglisci, E. Gascón, T. Hewson, G. Chillemi, R. Valentini, D. Gianelle, E. Mas-sarenti, M. Forconi, L. Mazzoni, and G. Scipione. High performance computing to support land, climate, and user-oriented services: The high-lander data portal. *Meteorological Applications*, 31(2):e2166, 2024. 6, 9
- [5] B. B. P. R. Center. Greenland 1km dem, 2003. [https://research.byrd.osu.edu/rs1/greenland\\_data/dem/index.html](https://research.byrd.osu.edu/rs1/greenland_data/dem/index.html). 3
- [6] A. Dasgupta, J. Poco, Y. Wei, R. Cook, E. Bertini, and C. T. Silva. Bridging theory with practice: An exploratory study of visualization use and design for climate model comparison. *IEEE Transactions on Visualization & Computer Graphics*, 21(09):996–1014, sep 2015. 3
- [7] S. Duebel, M. Röhlig, C. Tominski, and H. Schumann. Visualizing 3d terrain, geo-spatial data, and uncertainty. *Informatics*, 4(1), 2017. 3
- [8] T. G. Farr, P. A. Rosen, E. Caro, R. Crippen, R. Duren, S. Hensley, M. Kobrick, M. Paller, E. Rodriguez, L. Roth, et al. The shuttle radar topography mission. *Reviews of geophysics*, 45(2), 2007. 3
- [9] F. Ferstl, K. Bürger, and R. Westermann. Streamline variability plots for characterizing the uncertainty in vector field ensembles. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):767–776, Jan 2016. 3
- [10] J. Fiddes, K. Aalstad, and M. Lehning. Topoclim: rapid topography-based downscaling of regional climate model output in complex terrain v1. 1. *Geoscientific Model Development*, 15(4):1753–1768, 2022. 2
- [11] J. Fiddes and S. Gruber. Toposub: a tool for efficient large area numerical modelling in complex topography at sub-grid scales. *Geoscientific Model Development*, 5(5):1245–1257, 2012. 2
- [12] J. Fiddes and S. Gruber. Toposcale v. 1.0: downscaling gridded climate data in complex terrain. *Geoscientific Model Development*, 7(1):387–405, 2014. 2
- [13] C. Frei. Interpolation of temperature in a mountainous region using nonlinear profiles and non-euclidean distances. *International Journal of Climatology*, 34(5):1585–1605, 2014. 2
- [14] H. Griethe and H. Schumann. The visualization of uncertain data: Methods and problems. In *SimVis*, pp. 143–156, 2006. 3
- [15] G. Grigoryan and P. Rheingans. Point-based probabilistic surfaces to show surface uncertainty. *IEEE Transactions on Visualization and Computer Graphics*, 10:564–573, 2004. 3
- [16] D. A. Hastings and P. K. Dunbar. Global land one-kilometer base elevation (globe). 1999. 3
- [17] S. Hazarika, S. Dutta, H. Shen, and J. Chen. Codda: A flexible copula-based distribution driven analysis framework for large-scale multivariate data. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):1214–1224, Jan 2019. 3
- [18] T. Hewson and F. Pillosu. A low-cost post-processing technique improves weather forecasts around the world. *Commun Earth Environ*, 2:132, 2021. 9
- [19] J. Hiebl and C. Frei. Daily temperature grids for austria since 1961—concept, creation and applicability. *Theoretical and applied climatology*, 124:161–178, 2016. 2
- [20] R. R. Hoffman, D. S. LaDue, H. M. Mogil, P. J. Roebber, and J. G. Trafton. *Minding the Weather: How Expert Forecasters Think*. The MIT Press, 08 2017. 2
- [21] K. Höhle. Code for paper draft: "Topographic Visualization of Near-Surface Temperatures for Improved Lapse Rate Estimation", June 2024. doi: 10.5281/zenodo.11550983 2
- [22] ICAO. *Manual of the ICAO Standard Atmosphere: extended to 80 kilometres (262 500 feet)*, vol. 7488. International Civil Aviation Organization, 1993. 1, 2
- [23] A. Kamal, P. Dhakal, A. Y. Javaid, V. K. Devabhaktuni, D. Kaur, J. Zaiantz, and R. P. Marinier. Recent advances and challenges in uncertainty visualization: a survey. *Journal of Visualization*, 24:861 – 890, 2021. 3
- [24] N. Kaye, A. Hartley, and D. Hemming. Mapping the climate: guidance on appropriate techniques to map climate variables and their uncertainty. *Geoscientific Model Development*, 5, 02 2012. 3
- [25] B. Kruyt, R. Mott, J. Fiddes, F. Gerber, V. Sharma, and D. Reynolds. A downscaling intercomparison study: The representation of slope-and ridge-scale processes in models of different complexity. *Frontiers in Earth Science*, 10:789332, 2022. 2
- [26] H. Liu, K. C. Jezek, B. Li, and Z. Zhao. Radarsat antarctic mapping project digital elevation model, version 2, 2015. 3
- [27] H. Luo, F. Ge, K. Yang, S. Zhu, T. Peng, W. Cai, X. Liu, and W. Tang. Assessment of ecmwf reanalysis data in complex terrain: Can the cera-20c and era-interim data sets replicate the variation in surface air temperatures over sichuan, china? *International Journal of Climatology*, 39(15):5619–5634, 2019. 2
- [28] C. Lussana, O. Tveito, and F. Uboldi. Three-dimensional spatial interpolation of 2 m temperature over norway. *Quarterly Journal of the Royal Meteorological Society*, 144(711):344–364, 2018. 2
- [29] A. M. MacEachren, A. Robinson, S. Hopper, S. Gardner, M. Robert, M. Gahegan, and E. Hetzler. Visualizing geospatial information uncertainty: What we know and what we need to know. *Cartography and Geographic Information Science*, 32(3):139–160, 2005. 3
- [30] S. Malardel, N. Wedi, W. Deconinck, M. Diamantakis, C. Kühnlein, G. Mozdzyński, M. Hamrud, and P. Smolarkiewicz. A new grid for the ifs. *ECMWF newsletter*, 146(23-28):321, 2016. 3
- [31] D. Middleton, T. Scheitlin, and B. Wilhelmson. *Visualization in Weather and Climate Research*, pp. 845–XCXVII. 12 2005. 2
- [32] K. Mikelsons, M. Wang, X.-L. Wang, and L. Jiang. Global land mask for satellite ocean color remote sensing. *Remote Sensing of Environment*, 257:112356, 2021. 3
- [33] M. Monmonier. Air apparent: How meteorologists learned to map, predict, and dramatize weather. *Measurement Science and Technology*, 12(3):353, mar 2001. 2
- [34] T. Nocke, T. STERZEL, M. Böttinger, and M. Wrobel. Visualization of climate and climate change data: An overview. in *Ehlers et al. (Eds.) Digital Earth Summit on Geoinformatics 2008: Tools for Global Change Research (ISDE’08)*, Wichmann, Heidelberg, pp. 226-232, 2008, 01 2008. 2
- [35] T. Pfaffelmoser, M. Reitingner, and R. Westermann. Visualizing the positional and geometrical variability of isosurfaces in uncertain scalar fields. *Computer Graphics Forum*, 30(3):951–960, 2011. 3
- [36] K. Pothkow and H.-C. Hege. Positional uncertainty of isocontours: Condition analysis and probabilistic measures. *IEEE Transactions on Visualization and Computer Graphics*, 17(10):1393–1406, 2010. 3
- [37] K. Potter, J. Kniss, R. Riesenfeld, and C. R. Johnson. Visualizing summary statistics and uncertainty. *Computer Graphics Forum*, 29(3):823–832, 2010. 3
- [38] K. Potter, P. Rosen, and C. R. Johnson. From quantification to visualization: A taxonomy of uncertainty visualization approaches. In A. M. Dienstfrey and R. F. Boisvert, eds., *Uncertainty Quantification in Scientific Computing*, pp. 226–249. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. 3
- [39] C. Prates, V. Firat, U. Modigliani, B. Ingleby, M. Dahoui, C. Zanna, E. Kuşçu, and T. Haiden. Increased use of surface observations. *ECMWF Newsletter*, 176:16–18, 2023. 3
- [40] M. Rautenhaus, M. Böttinger, S. Siemen, R. Hoffman, R. M. Kirby, M. Mirzargar, N. R’ober, and R. Westermann. Visualization in meteorology - a survey of techniques and tools for data analysis tasks. *IEEE Transactions on Visualization and Computer Graphics*, PP(99), 2017. 2
- [41] D. P. Retchless and C. A. Brewer. Guidance for representing uncertainty on global temperature change maps. *International Journal of Climatology*, 36(3):1143–1159, 2016. 3
- [42] N. Röber, M. Böttinger, and B. Stevens. Visualization of climate science simulation data. *IEEE Computer Graphics and Applications*, 41(1):42–48, 2021. 2

- [43] J. Sanyal, S. Zhang, J. Dyer, A. Mercer, P. Amburn, and R. Moorhead. Noodles: A tool for visualization of numerical weather model ensemble uncertainty. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1421–1430, 2010. 3
- [44] W. Schroeder, K. M. Martin, and W. E. Lorensen. *The visualization toolkit an object-oriented approach to 3D graphics*. Prentice-Hall, Inc., 1998. 3
- [45] I. I. M. Service) and N. N. L. S. of Iceland). Is 50 v. 3
- [46] P. Sheridan, S. Smith, A. Brown, and S. Vosper. A simple height-based correction for temperature downscaling in complex terrain. *Meteorological Applications*, 17(3):329–339, 2010. 2, 9
- [47] P. Sheridan, S. Vosper, and S. Smith. A physically based algorithm for downscaling temperature in complex terrain. *J. Appl. Meteor. Climatol.*, 57:1907–1929, 2018. 2, 9
- [48] R. Stauffer, G. J. Mayr, M. Dabernig, and A. Zeileis. Somewhere over the rainbow: How to make effective use of colors in meteorological visualizations. *Bulletin of the American Meteorological Society*, 96(2):203–216, 2015. 2
- [49] E. M. Stephens, T. L. Edwards, and D. Demeritt. Communicating probabilistic information from climate model ensembles—lessons from numerical weather prediction. *Wiley interdisciplinary reviews: climate change*, 3(5):409–426, 2012. 2
- [50] C. Sullivan and A. Kaszynski. Pyvista: 3d plotting and mesh analysis through a streamlined interface for the visualization toolkit (vtk). *Journal of Open Source Software*, 4(37):1450, 2019. 3
- [51] A. J. Teuling, R. Stöckli, and S. I. Seneviratne. Bivariate colour maps for visualizing climate data. *International Journal of Climatology*, 31(9):1408–1412, 2011. 2
- [52] F. Uboldi, C. Lussana, and M. Salvati. Three-dimensional spatial interpolation of surface meteorological observations from high-resolution local networks. *Meteorological Applications: A journal of forecasting, practical applications, training techniques and modelling*, 15(3):331–345, 2008. 2
- [53] J. Wang, S. Hazarika, C. Li, and H. Shen. Visualization and visual analysis of ensemble data: A survey. *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2018. 3
- [54] R. T. Whitaker, M. Mirzargar, and R. M. Kirby. Contour boxplots: A method for characterizing uncertainty in feature sets from simulation ensembles. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2713–2722, Dec 2013. 3
- [55] R. L. Wilby and T. M. Wigley. Downscaling general circulation model output: a review of methods and limitations. *Progress in physical geography*, 21(4):530–548, 1997. 1
- [56] B. Zehner, N. Watanabe, and O. Kolditz. Visualization of gridded scalar data with uncertainty in geosciences. *Computers & Geosciences*, 36:1268–1275, 10 2010. 3



# Attribution 4.0 International Creative Commons

## Deed – reformatted for display in this thesis

### You are free to:

1. **Share** — copy and redistribute the material in any medium or format for any purpose, even commercially.
2. **Adapt** — remix, transform, and build upon the material for any purpose, even commercially.
3. The licensor cannot revoke these freedoms as long as you follow the license terms.

### Under the following terms:

1. **Attribution** — You must give appropriate credit, provide a link to the license, and indicate if changes were made . You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
2. **No additional restrictions** — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation.

No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material.

### Notice

This deed highlights only some of the key features and terms of the actual license. It is not a license and has no legal value. You should carefully review all of the terms and conditions of the actual license before using the licensed material.

Creative Commons is not a law firm and does not provide legal services. Distributing, displaying, or linking to this deed or the license that it summarizes does not create a lawyer-client or any other relationship.

### Deed Source / Canonical URL

<https://creativecommons.org/licenses/by/4.0/>



# Evaluation of Volume Representation Networks for Meteorological Ensemble Compression

K. Höhle<sup>1</sup>, S. Weiss<sup>1</sup>, T. Necker<sup>2</sup>, M. Weissmann<sup>2</sup>, T. Miyoshi<sup>3</sup>, and R. Westermann<sup>1</sup>

<sup>1</sup>Technical University of Munich, Department of Informatics, Germany

<sup>2</sup>University of Vienna, Department of Meteorology and Geophysics, Austria

<sup>3</sup>RIKEN Center for Computational Science, Kobe, Japan

---

## Abstract

Recent studies have shown that volume scene representation networks constitute powerful means to transform 3D scalar fields into extremely compact representations, from which the initial field samples can be randomly accessed. In this work, we evaluate the capabilities of such networks to compress meteorological ensemble data, which are comprised of many separate weather forecast simulations. We analyze whether these networks can effectively exploit similarities between the ensemble members, and how alternative classical compression approaches perform in comparison. Since meteorological ensembles contain different physical parameters with various statistical characteristics and variations on multiple scales of magnitude, we analyze the impact of data normalization schemes on learning quality. Along with an evaluation of the trade-offs between reconstruction quality and network model parameterization, we compare compression ratios and reconstruction quality for different model architectures and alternative compression schemes.

## CCS Concepts

• **Computing methodologies** → **Learning latent representations**; • **Applied computing** → **Earth and atmospheric sciences**;

---

## 1. Introduction

Meteorological ensemble data comprise multiple weather forecast simulations, which can differ in initial conditions, numerical approximations or even physical model assumptions, and are used to assess uncertainties of the forecast outcome. Over the last decade, researchers have continually pushed ensemble sizes to larger scales, while, at the same time, extending spatial domain size, resolution and time horizon. Thus, meteorological ensembles can become extremely large. Ensembles are produced daily by weather centers and require large amounts of secondary disk space for backup.

Due to the sheer volume of meteorological ensemble data, any attempt to analyse such datasets is intrinsically difficult. In the scenario we consider, the ensemble dataset comprises 1000 runs of a high-resolution numerical atmospheric dynamics model [NGW\*20], thus pushing the data volume to 60GB of memory for only a single time step. This makes it impossible to keep the data entirely on recent GPUs and fosters the need for effective compression schemes for multi-dimensional arrays of floating-point data. Yet, besides targeted strategies for reducing I/O bandwidth and storage requirements by converting such ensembles into compact data representations, random access to the data is mandatory to avoid decoding the entire ensemble for analysis tasks.

While lossless compression schemes allow for bit-wise accurate reconstruction of the original data, they typically achieve up

to only 2x compression or less [SCH\*14]. Lossy data compression schemes, such as ZFP [Lin14], SZ [DC16], or TThresh [BRLP19], in contrast, offer higher compression ratios of 100x or more, at the cost of introducing noticeable reconstruction errors. For most downstream analysis tasks, however, a certain error level is acceptable, such that lossy compression becomes a suitable tool for memory reduction [BHM\*16, CDL\*19].

As an alternative to classical lossy compressors for multi-dimensional scalar fields, compression schemes based on fully-connected neural networks have been proposed recently. Volume scene representation networks (V-SRNs) have been introduced by Lu *et al.* [LJLB21], and were further improved and accelerated by Weiss *et al.* [WHW21] (fV-SRN). V-SRNs are an extension of scene representation networks (SRNs), which were first developed for representing opaque surface models [MON\*19, CZ19, PFS\*19]. Besides offering the ability to directly reconstruct single samples from the compressed representation, V-SRNs are capable of exploiting non-local coherence in the data [CLI\*20]. This makes V-SRNs a promising tool for compressing meteorological ensemble data, in which coherence and correlation are often observed between multiple parameter fields of the same simulation run or between different members of the same ensemble, but are more difficult to exploit for compression than, e.g., auto-correlations in space and time.

**Contribution** In this work, we evaluate the potential of V-SRNs for learning compact representations of ensembles of volumetric multi-parameter fields. We compare two different model architectures, which allow for efficient parameter sharing between multiple parameter fields and ensemble members. We demonstrate that this results in compression rates that are higher or on par with those achieved by classical compression schemes, which have been adapted to exploit redundancy between different ensemble members. We do not focus on data with temporal variability explicitly, but our methods generalize straight-forwardly also to ensembles of time-variate multi-parameter fields.

We propose and analyze binary model architectures, leveraging a combination of low-resolution grids of trainable spatial latent features and small neural networks to read out the features and serve as non-linear interpolation functions. Different combinations of grids and networks are evaluated to identify trade-offs between model parameterization, reconstruction accuracy, and compression rate. Our analyses are tightly coupled to a case study using meteorological simulation data from a convective-scale ensemble by Necker *et al.* [NGW\*20]. Based on this ensemble dataset, we discuss general methodological aspects, such as model design and training procedures, and highlight the importance of data-related aspects, such as the impact of data normalization. The code for the project is publically available at [HW22].

## 2. Related Work

**Scene representation networks** The concept of scene representation networks (SRNs) was concurrently introduced by Mescheder *et al.* [MON\*19], Chen and Zhang [CZ19] and Park *et al.* [PFS\*19], who present the idea of encoding an opaque, uncolored surface model as an implicit function that is implemented as a fully-connected neural network. The authors use feature vectors to encode object specific information and enable reusing models for different objects. The idea of trainable latent features was developed further by Chabra *et al.* [CLI\*20], who replace the single feature vector by a feature grid to improve reconstruction accuracy. Multiple studies explore improvements and extensions of this idea. Martel *et al.* [MLL\*21] use an adaptive data structure that is refined during training to allocate more resources in areas of larger errors. A fixed multi-resolution grid is used by Takikawa *et al.* [TLY\*21] and later extended with spatial hashing by Müller *et al.* [MESK22], together with an efficient network implementation [MRNK21]. For a more comprehensive review of SRN-related literature, we refer to the overview articles by Hoang *et al.* [HSB\*20] and Tewari *et al.* [TFT\*20]. The works by Lu *et al.* [LJLB21] and Weiss *et al.* [WHW21] extend SRNs for volumetric data compression. The latter contributes in particular a fast network evaluation method to speed up training and decompression. Mishra *et al.* [MHBB22] leverage fully-connected neural networks for interpolating scientific data. We build upon and extend these works by focusing explicitly on the multi-parameter and ensemble compression capabilities of V-SRNs.

**Lossy volume compression schemes** Prior work in the area of lossy compression schemes can be categorized into three classes of algorithms. Transform coding-based schemes [YL95, LCA08]

**Table 1:** List of available simulation parameters.

Name (Short name)	Unit	Value range
Temperature (tk)	Kelvin	[200, 300]
3D wind speed (u, v, w)	ms <sup>-1</sup>	[-40, 40]
relative humidity (rh)	%	[0, 100]
water vapor mixing ratio (qv)	1	[0, 0.02]
mixing ratio of hydrometers (qhydro)	1	[0, 0.01]
geopotential height (z)	m	[200, 20000]
radar reflectivity (dbz)	dBZ	[-30, 40]

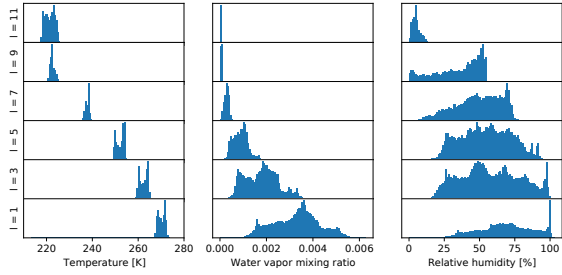
employ the discrete cosine or wavelet transformation to transform the data into a basis in which only few coefficients are relevant, while many others can be removed. Quantization schemes represent contiguous data blocks by a single index or a sparse combination of learned representative values [SW03, FM07, GIGM12, GG16]. One instance of this class of compression algorithms is the SZ algorithm [DC16, ZDL\*20] using lossy curve fittings. Tensor decomposition schemes decompose the data directly using, e.g., a singular value decomposition. As one instance of such schemes, TThresh [BRLP19] can achieve extremely high compression ratios of 1000x or more. In interactive scenarios, mostly transform coding-based schemes are applied brick-wise, in which case high compression ratios are traded in on fast GPU-based decompression, see e.g. [DMG20, MAG19]. Focusing on applied scientific data compression, various studies have evaluated the applicability and performance of lossless and lossy data compression algorithms on atmospheric datasets [HWK\*13, BHM\*16, DCG19, KRK\*21], and Dueben *et al.* [DLB19] discuss methods for efficient storage of weather forecast ensembles. Baker *et al.* [BPH22] have introduced a data-based similarity measure, termed DSSIM, for evaluating the quality loss in scientific data after lossy compression.

## 3. Data

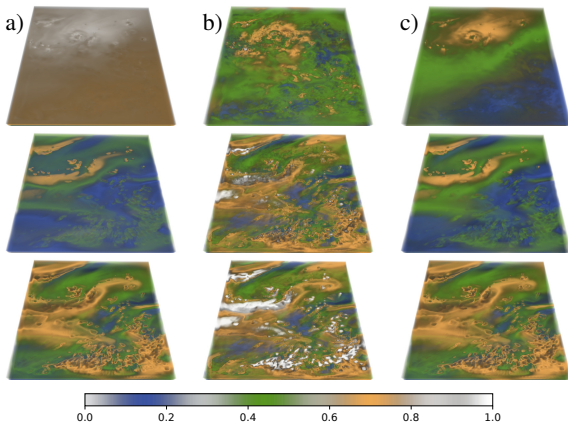
We evaluate the compression capabilities of V-SRNs on a multi-parameter ensemble dataset, which was generated to study correlation patterns in atmospheric dynamics [NGW\*20]. The dataset comprises 1000 runs of an atmospheric dynamics model over a rectangular domain in central Europe. Nine prognostic atmospheric parameters are stored at regular time steps of one hour, on a rectangular grid with 352 × 250 nodes and 20 levels in height. Due to the presence of mountain ranges and topography in the simulated domain, large parts of the data in lower levels are missing due to grid cells lying below the earth surface. For simplicity, we omit grid levels with missing values and restrict the dataset to the 12 top-most levels, which are free of missing values. A list of the available parameters is given in Tab. 1. The fields possess different physical interpretations and differ in value range and statistical distribution. As shown in Fig. 1, the distribution of field values varies not only between different parameters, but also between height levels of the same field, which complicates the learning task for deep learning-based compression algorithms.

**Data normalization** To facilitate model optimization, we examine the effect of different normalization methods, which rescale all





**Figure 1:** Value distribution marginalized over different height levels for parameters  $tk$ ,  $rh$  and  $qv$ . Distributions of the same parameter may differ with respect to value range or variability.



**Figure 2:** Influence of the three variants of interval rescaling on the data parameters  $tk$  (top row),  $qv$  (middle row) and  $rh$  (bottom row): a) global min-max, b) local min-max, c) level-wise min-max.

parameters to a value range of  $[0, 1]$ . In the context of data compression, data rescaling has been discussed by Dueben *et al.* [DLB19] and was found to improve compression efficiency. We compare three alternative variants of min-max normalization (see Fig. 2), which reflect a trade-off between expressiveness of the rescaling and storage space required for keeping the meta information:

- Global min-max rescaling: minimum and maximum values are computed over the whole domain, all ensemble members, and all time steps. Minimum and maximum values can be stored as one floating point number each.
- Local min-max rescaling: minimum and maximum values are computed for each grid location separately from the statistics of all ensemble members and all time steps. Minimum and maximum values are stored as a full grid of floating point numbers.
- Level min-max rescaling: minimum and maximum values are computed separately for each height-level in the data. Minimum and maximum values are stored as one-dimensional arrays of floating point values.

#### 4. Model design

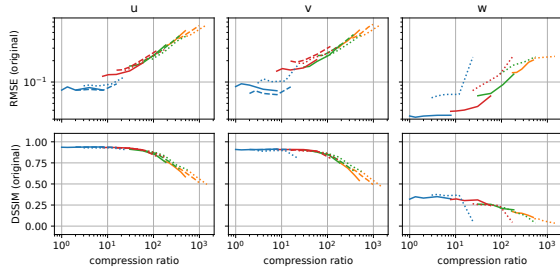
(V-)SRNs, in their basic form, are fully-connected neural networks that define a parametric mapping from 3D position coordinates to the  $d$ -dimensional data domain [MON\*19, CZ19, PFS\*19]. To enable sharing of model parameters between different ensemble members, we consider generalized V-SRN mappings, which receive information about the member identity as an additional input.

**Encoding of the spatial coordinates** For our experiments, we assume that position coordinates are normalized to have values in  $[0, 1]^3$ . The analyzed V-SRN architectures can be subdivided into three modules: a constant input encoding, a low-resolution grid of trainable feature vectors, and a compact fully-connected auto-decoder. For the input encoding, we use Fourier features [MST\*20], which map the position coordinates,  $\mathbf{p} = (p_x, p_y, p_z) \in [0, 1]^3$ , to wave-like features

$$\mathbf{f}_{ij} = (\sin(2\pi v_i \mathbf{n}_j \cdot \mathbf{p}), \cos(2\pi v_i \mathbf{n}_j \cdot \mathbf{p}))$$

with frequency scales  $v_i = 2^i$ ,  $i \in \mathbb{N}$ , and axis-aligned unit directions  $\mathbf{n}_j$ ,  $j \in \{x, y, z\}$ . Note here, that similar embeddings with randomly chosen frequencies and orientations have been proposed by Tancik *et al.* [TSM\*20], but did not yield better results in our experiments. Additionally, we utilize an axis-aligned, regular grid of multi-dimensional feature vectors [CLI\*20]. The grid has a pre-set coarse spatial resolution (compared to the resolution of the original data grid) and captures non-local variability in the data. During inference, the features are interpolated trilinearly to match the input position. The network weights and the feature grid are optimized jointly during training. More elaborate multi-resolution feature grids have been proposed recently [MESK22], but were found to not improve the compression-accuracy trade-off of our architectures. For the auto-decoder network, we employ multi-layer perceptrons (MLPs) with  $l$  fully-connected layers with  $c$  hidden channels. Each layer performs an affine transformation with non-linear activation. Following Weiss *et al.* [WHW21], we use the *SnakeAlt* activation in all but the last layer, and discretize the network weights using half-precision floats and the latent grid using 8 bit per channel. Multi-parameter data is represented by augmenting the output dimension of the decoder models.

**Encoding of the ensemble dimension** To inform the V-SRN about which ensemble member to reproduce, we explore two different ensemble encoding strategies. First, a separate grid of feature vectors is allocated for each ensemble member and the auto-decoder network is shared between ensemble members. This is similar to how the time dimension is encoded in fV-SRN [WHW21], and replicates the approach of Park *et al.* [PFS\*19] in the limit of vanishing spatial resolution. We term this architecture the *multi-grid* configuration. Second, we consider SRNs with a single feature grid, which is shared among all ensemble members, and a separate auto-decoder per ensemble member. The intuition is, that the ensemble information is stored in the shared feature grid, and the separate decoders learn to extract member-specific features from the common grid, thus allowing for efficient reuse of model parameters. We term this variant the *multi-decoder* configuration. As a baseline comparison method, we consider training a separate V-SRN with a single decoder and a single feature grid for every member.



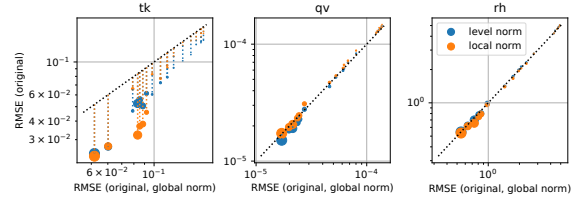
**Figure 3:** Reconstruction accuracy of multi-parameter models for parameters  $u$ ,  $v$  and  $w$ . Results are shown for single-parameter models (solid lines), two-parameter models ( $u$  and  $v$ , dashed lines), and three-parameter models ( $u$ ,  $v$  and  $w$ , dotted lines). Colors indicate different model configurations, chosen appropriately to have good reconstruction accuracy at each compression ratio.

**Training** At training time, we draw  $6 \times 10^6$  random positions uniformly distributed in  $[0, 1]^3$  and sample the original member volumes using trilinear interpolation. We choose a fixed number of samples per ensemble member proportional to the number of samples in the original data grid. The network predictions are matched against the ground truth using the  $L_1$  loss function and stochastic gradient descent. In every mini-batch, we balance the number of samples evenly between all ensemble members to ensure equally distributed gradient variances for all member models. We use the Adam optimizer with an initial learning rate of  $10^{-2}$  and learning rate decay of 0.2 after every 20 epochs. Training lasts for a total of 50 epochs with resampling of the training data applied after every 10 epochs. Loss-adaptive resampling strategies, as described by Weiss *et al.* [WHW21], were found to increase training stability for high-capacity models and slightly improve overall model accuracy. Switching to  $L_2$  loss or omitting the balanced sample distribution among ensemble members led to inferior results.

## 5. Single-member experiments

To assess the performance of V-SRNs, we first examine the reconstruction accuracy of models which are trained to represent parameter fields from single ensemble members, without accounting for the ensemble dimension. To guarantee proper gradient backpropagation, we fix the decoder architecture as a three-layer MLP and vary the number of channels per layer as well as the resolution and the number of channels in the feature grid. Models were trained separately for dataset parameters  $tk$ ,  $rh$ ,  $qv$ ,  $u$ ,  $v$  and  $w$ , and separately for multiple ensemble members. Exemplary compression-accuracy curves for parameters  $u$ ,  $v$  and  $w$  with level-wise min-max normalization are shown in Fig. 3 (solid lines).

**Impact of model parameterization** For all parameters, we find that the details of the decoder architecture have a minor effect on the reconstruction accuracy compared to the parameterization of the latent grid, which is consistent with prior work by Weiss *et al.* [WHW21]. We note that the compression and reconstruction performance of the models depends crucially on an appropriate choice of the grid resolution in horizontal and level direction.

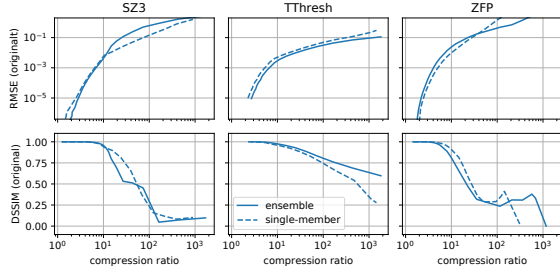


**Figure 4:** Impact of data normalization on reconstruction accuracy (RMSE) for models with varying complexity. Point size indicates model complexity (larger point  $\rightarrow$  bigger model). Global min-max normalization is considered as baseline. Points below the dashed diagonal line indicate an improvement.

Given a fixed grid resolution, we observe a sigmoid-shaped dependence of the reconstruction accuracy on the number of grid feature channels. This indicates that an increasing number of grid channels can partially compensate for a reduction in spatial grid resolution, but not indefinitely. This behavior is observed in qualitatively the same way for various decoder complexities.

**Impact of data normalization** To evaluate the impact of data normalization on the training outcomes, we retrain single-parameter models on target data to which we apply different normalization schemes. We consider global min-max normalization as a baseline and investigate the effect of applying level-wise or local min-max normalization instead. Specifically, we train model configurations with a three-layer MLP ( $c = 32$ ), and set the latent grid resolution to a fraction of  $1/2$  to  $1/8$  of the original data grid in all directions. We consider grid feature dimensions of 4 or 8. Fig. 4 illustrates the outcome of such experiments for three parameters with different statistical distributions in height (cf. Fig. 1). For  $qv$  and  $rh$ , only minimal improvements can be observed from global min-max normalization to level-wise min-max, independent of the model configuration. For  $tk$ , which exhibits a much stronger variation of value distribution with height (see Fig. 1, left), both local and level-wise min-max normalization help to reduce the reconstruction error. Local normalization performs better than level-wise min-max only for models with high parameter complexity. We attribute this to the fact that local normalization improves uniformity of the data, but potentially destroys spatial coherence patterns due to high-frequency components in the minimum- and maximum-value fields (see Fig. 2, middle column). Due to the preferable compression rate vs. accuracy trade-off, we use level-wise min-max normalization as a default for all further experiments. More generally, we conjecture that the importance of appropriate data normalization arises due to the inability of the  $L_1$  loss function to properly resolve multi-scale effects. For the field parameter  $tk$ , the pronounced field gradient in the vertical direction provides a strong learning signal, while the variability of the data within each level is weighted as relatively less important. Differences in variability, as seen in parameter  $qv$  (see Fig. 1, middle), appear less problematic.

**Multi-parameter models** To evaluate the ability of V-SRNs to fit multi-parameter data, we select a triplet of field parameters –  $u$ ,  $v$  and  $w$ , i.e. 3D wind components – for which strong inter-



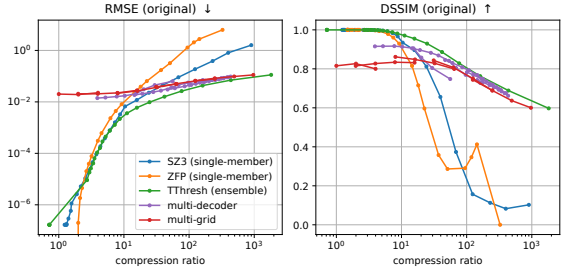
**Figure 5:** Reconstruction accuracy vs. compression rate for classical compression algorithms, applied to ensemble data separately for each 3D member volume (solid line) or to a 4D array of stacked member volumes (dashed line).

parameter correlations can be expected due to physical reasoning. We train model configurations of different complexity on predicting single parameters one at a time, all jointly, or only the horizontal winds. We use model configurations identical to those of the single-parameter experiments, except for adapting the final model layer to the number of required model outputs. Results of the trainings are shown in Fig. 3. The multi-parameter models show qualitatively the same accuracy-compression trade-off as the single parameter models. In particular, the grid parameterization is found to be more important than the decoder complexity. Models with wider decoder layers did not yield higher accuracy than shown in Fig. 3. At low compression ratios (blue curves),  $u$  and  $v$  are predicted best by the two-parameter model, which suggests that knowledge of both parameters supports accurate reconstruction. At the same time, the three-parameter configuration yields the largest reconstruction error, indicating that joint prediction of unsuitable pairs may hamper high reconstruction accuracy. The parameter is difficult to predict even by single-parameter models, as seen from the low DSSIM values, and thus disturbs the reconstruction of  $u$  and  $v$ . Only at very high compression ratios, the three-parameter model yields the highest reconstruction accuracy on all parameters.

## 6. Ensemble experiments

For all subsequent experiments, we use a subset of 64 members of the original ensemble, if not stated otherwise. Experiments are carried out using data for the parameter  $tk$ , subject to level-wise min-max normalization.

**Classical compression baseline** To set a baseline for achievable compression ratios from parameter sharing in the ensemble dimension, we select three commonly used compression algorithms from the literature and evaluate compression performance for ensemble member volumes compressed separately and jointly. We choose SZ3 [DC16] as an example of predictor-based compression algorithms, ZFP [Lin14] as an algorithm with block-wise transform coding, and TThresh [BRLP19], which is based on the Tucker decomposition of tensor data. To allow for a fair comparison between the algorithms, we apply all algorithms with a suitable set of thresholds on absolute error, record the achieved compression ratio and measure the resulting reconstruction accuracy in terms of root mean

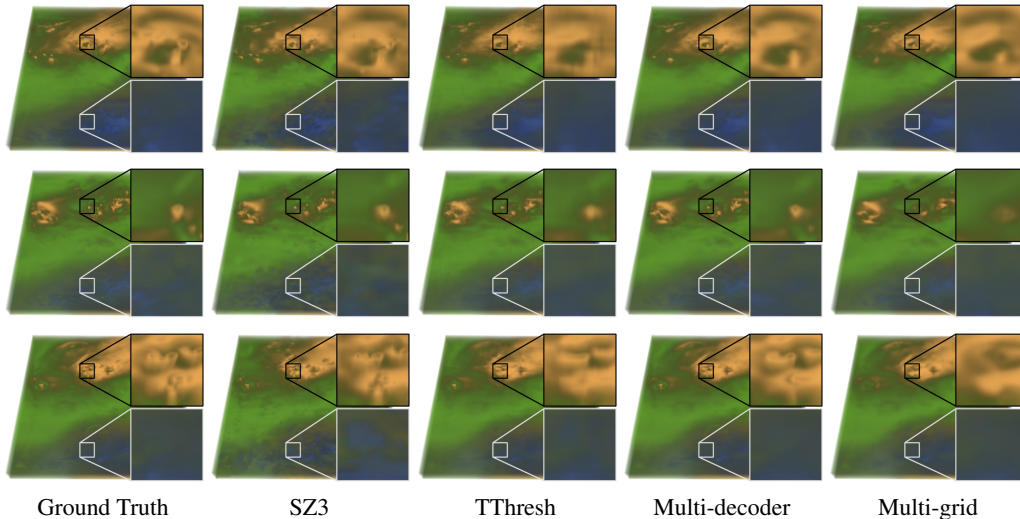


**Figure 6:** Comparison of compression algorithms, averaged over all ensemble members. For the baseline compression methods, the best configuration from Fig. 5 are selected. Every point represents a trained network or an invocation of a baseline compression method. Arrows indicate improving quality.

square error (RMSE) and DSSIM [BPH22]. To evaluate the ability of compression algorithms of exploiting inter-member similarities, we propose a test setting, where an ensemble of 3D volumetric scalar fields is first compressed in a member-by-member configuration (i.e. one 3D volume at a time) and subsequently with all members in common. The comparison of the required storage space per member allows to draw conclusions about whether similarities between ensemble members are exploited efficiently.

Fig. 5 depicts the trade-off between reconstruction accuracy vs. compression ratio found in this procedure. ZFP does not take advantage of between-member similarities. For most accuracy settings, a higher compression ratio is obtained when the 3D volumes are compressed separately. ZFP generally yields poor quality for compression ratios above 30x, but single-member compression is generally preferable. Ensemble compression is favorable with the SZ3 algorithm, in the case of low error thresholds and low-ratio compression. For intermediate and highly lossy compression single-member compression yields lower errors at a given compression ratio. We therefore select the single-member configuration as a baseline for comparison against V-SRN models. TThresh yields overall the best reconstruction accuracy, and is the only algorithm to take advantage from the ensemble dimension throughout the whole range of reconstruction accuracies. We therefore select the ensemble-wise compression for further comparisons.

**Ensemble V-SRNs** We apply both V-SRN configurations under the same conditions as the classical compressors. For both architectures, we train model variants with three- and four-layer MLP decoders and 32, 64 and 128 channels per layer, and find that models with four layers and 32 channels yield the best balance between reconstruction quality and compression rate. For multi-decoder models, higher decoder capacity is needed to achieve good accuracy, in comparison to V-SRNs trained on single member volumes. This can be seen as a consequence of sharing local feature vectors between multiple decoders. Due to the decoder being unique for every member, increases in decoder size limit the achievable compression ratio. For the multi-grid models, we note that four-layer MLPs with 32 channels yield similar reconstruction accuracy as three-layer architectures with 64 channels, at less than half the storage cost. Fur-



**Figure 7:** Qualitative comparison when compressing three member volumes of the  $tk$  parameter using SZ3, TThresh, and our two methods “multi-decoder” and “multi-grid”. The models were trained on 64 ensemble members, the first three are shown along the rows. Compression ratios of the methods are as follows: SZ3: 248.81x, TThresh: 253.25x, Multi-decoder: 251.88x, Multi-grid: 248.00x. For colorbar, see Fig. 2.

ther increase of decoder complexity led to only marginal accuracy improvements at significant additional storage cost. Given a fixed decoder configuration, feature grid resolution and channel number of the investigated architectures are determined empirically to optimize reconstruction accuracy.

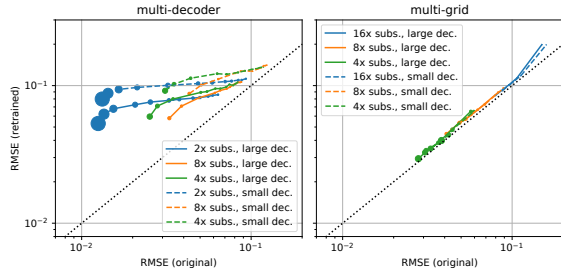
Fig. 6 illustrates the complexity-accuracy trade-off for different configurations of both architectures in comparison to the selected variants (single-member or ensemble-wise compression) of the classical compressors. Multi-grid models allow for higher compression ratios in our test because the majority of parameters is concentrated in the feature grid, which is stored in 8 bit format, thus requiring only half the memory space of an identical number of half-precision network parameters. The accuracy of multi-grid models reaches an optimum around compression ratios of 10x, and is limited by stochastic noise in the optimization of the grid parameters at lower compression ratios. For intermediate and high compression ratios, above 20x, both model variants outperform SZ3 and ZFP in reconstruction accuracy with respect to both RMSE and DSSIM. Given a fixed storage budget, the multi-decoder configuration achieves slightly better reconstruction accuracy. At compression ratios above 200x, both architectures come close to the accuracy of TThresh.

In Fig. 7, we compare the visual quality of renderings of three ensemble members, obtained from reconstructions with compression ratios around 250x. ZFP has been omitted from this comparison due to very low reconstruction quality. The multi-decoder model preserves visual high-frequency structures the best. Both TThresh and the multi-grid model show a tendency to smooth fine-scale details, with TThresh additionally introducing stripe-like artifacts. SZ3 is found to preserve high-frequency field structures in regions of high variability, but introduces fine-granular noise in regions, where the fields should be smooth. The V-SRN models, in contrast rather have a tendency to smooth out fine details, which can be seen

as another advantage, depending on the subsequent analysis task. A significant advantage of the V-SRN-based approaches lies in their decompression speed. To reconstruct the full-resolution voxel grid, the reference implementations of TThresh and SZ3 require 50 ms and 10 ms, respectively, on an Ubuntu 20.04 workstation with Intel Xeon W-2133 CPU (3.60GHz), 32 GB RAM, and Nvidia Titan RTX GPU. Our proposed multi-grid and multi-decoder models sample the full-resolution data in less than 2 ms, and allow for rendering and random data access directly out of the compressed data structure.

We note that the performance benefit of V-SRNs over classical compression algorithms in our application appears comparatively smaller at first sight than was reported in earlier works, such as [WHW21, LJLB21]. We attribute this to the properties of the data that we use for our experiments. The meteorological data differs from previously studied datasets with respect to data size and distribution of variability. In particular the low voxel number and high-frequency variability in the vertical direction prevent the grid-based V-SRNs from achieving higher reconstruction accuracy, because subsampling of the feature grid vertically impedes reconstruction accuracy. Additionally, many of the datasets in earlier studies possess areas of constant field values. Closest to our example is the Hurricane Isabel dataset [isa] as studied by Lu *et al.* [LJLB21], finding that V-SRNs perform similar to TThresh at compression rates around 500x. We expect larger storage savings for simulation data at sub-kilometer resolution, where the fields are determined by low-frequent variability, and for data with higher resolution in the vertical direction, which would simplify the exploitation of data coherence along a third spatial dimension.

**Generalization to new ensemble members** In Fig. 8, we investigate whether the shared representations of the proposed models encode information that is representative of the full ensemble. For this, we re-used the trained models from previous experi-



**Figure 8:** Reconstruction accuracy on unseen ensemble members after retraining of only the member-specific model parts for different model configurations. Marker size encodes model complexity, black dotted line indicates identity.

ments, fixed the parameterization of the shared model components, and retrained the member-specific components from scratch. Multi-decoder models with higher-capacity decoders (solid lines) achieve better reconstruction accuracy than models with simpler decoders (dashed lines). Nevertheless, all multi-decoder models fail to retain the same accuracy for new members. Models with higher complexity in the latent features perform comparatively worse in fitting unseen ensemble members. The pattern is apparent for model configurations that build on the highest-resolution grid in the test (Fig. 8, blue lines, 2x subsampling), and which exhibit the largest reconstruction error compared to the remaining configurations. We conjecture that a lack of complexity in the latent grid forces the models to learn more abstract and generalizable representations, thus providing better starting conditions for training on unseen members. For the multi-grid configuration, all models are able to achieve almost identical loss levels on unseen members as on the original member set, which confirms the intuition that member-specific information is stored in the feature grids.

**Impact of ensemble size** Experiments with different numbers of ensemble members were conducted for ensemble sizes between 2 and 128 members. The results indicate that multi-grid models are not affected significantly by changes in ensemble size, suggesting once more that member-specific information is stored in the feature grids. Multi-decoder models yield comparable accuracy for various ensemble sizes at equal compression rates, as well. However, for large ensembles, the evaluation of multi-decoder models is constrained by the memory capacity of the GPU, since the shared feature grid for the full ensemble must be held in device memory, or streamed from system memory or disk.

## 7. Conclusion

We have analyzed how volume scene representation networks (V-SRNs) can be used to transform a meteorological multi-parameter ensemble into compact neural data representations. We compared two model architectures, which exploit relationships between different field parameters and between ensemble members. Our findings suggest that V-SRNs, in particular in the multi-grid configuration (see Sec. 4), yield promising performance at high compression ratios, where they outperform the classical compressors

SZ3 or ZFP in reconstruction accuracy. We found that in meteorological applications the accuracy of V-SRNs may be affected by the choice of hyper parameters and peculiarities of the data distribution. We demonstrate that the latter can be counteracted with appropriate data normalization. However, the necessity of tuning grid resolution and feature channels currently remains a drawback of V-SRN-based data compression. Nevertheless, V-SRNs come with a significant advantage in reconstruction speed and flexibility on multi-parameter data, compared to classical floating-point compressors. This makes them appealing for visual analytics tasks, where an interactive exploration of large multi-parameter ensembles is paramount, using parallelizable statistical evaluations on the whole dataset.

In the future, we intend to shed light on the embedding of network-based compression of multi-parameter ensembles into visual data analysis workflows. For large ensembles comprising billions of data points with many parameters per point, visual analysis techniques like parallel coordinates plots or scatter plot matrices cannot be realized on the GPU due to memory limitations. The fast random access capabilities of V-SRNs allow to overcome these limitations, while at the same time preserving the spatial structure of the data, so that linked 3D spatial data views can be integrated. To improve usability, we will analyze how to design generalizing V-SRNs to limit retraining for new datasets. For this, we consider V-SRNs as a mapping from a latent space representation to an ensemble, and explore speeding up training through direct prediction of the feature representation for new ensembles. Another promising approach could be the combination of V-SRN decoders with generative network architectures, such as variational auto-encoders or generative adversarial networks, which could help to circumvent storage of member-specific feature grids by generating the required features efficiently on demand. Furthermore, we plan on exploring improved methods for hyper-parameter selection, which will enable a higher level of automation and adaptivity, and will improve accessibility of V-SRNs for practical compression applications.

## 8. Acknowledgments

This study was conducted within the subproject B5 of the Transregional Collaborative Research Center SFB/TRR 165 Waves to Weather funded by the German Research Foundation (DFG). The authors acknowledge crucial contributions by Juan Ruiz and the RIKEN Data Assimilation Research Team for conducting the 1000-member ensemble simulation.

## References

- [BHM\*16] BAKER A. H., HAMMERLING D. M., MICKELSON S. A., XU H., STOLPE M. B., NAVEAU P., SANDERSON B., EBERT-UPHOFF I., SAMARASINGHE S., DE SIMONE F., ET AL.: Evaluating lossy data compression on climate simulation data within a large ensemble. *Geoscientific Model Development* 9, 12 (2016), 4381–4403. 1, 2
- [BPH22] BAKER A. H., PINARD A., HAMMERLING D. M.: Dssim: a structural similarity index for floating-point data. *arXiv preprint arXiv:2202.02616* (2022). 2, 5
- [BRLP19] BALLESTER-RIPOLL R., LINDSTROM P., PAJAROLA R.: Tlthresh: Tensor compression for multidimensional visual data. *IEEE transactions on visualization and computer graphics* 26, 9 (2019), 2891–2903. 1, 2, 5

- [CDL\*19] CAPPELLO F., DI S., LI S., LIANG X., GOK A. M., TAO D., YOON C. H., WU X.-C., ALEXEEV Y., CHONG F. T.: Use cases of lossy compression for floating-point data in scientific data sets. *The International Journal of High Performance Computing Applications* 33, 6 (2019), 1201–1220. 1
- [CLF\*20] CHABRA R., LENSSEN J. E., ILG E., SCHMIDT T., STRAUB J., LOVEGROVE S., NEWCOMBE R.: Deep local shapes: Learning local sdf priors for detailed 3d reconstruction. In *European Conference on Computer Vision* (2020), Springer, pp. 608–625. 1, 2, 3
- [CZ19] CHEN Z., ZHANG H.: Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), pp. 5939–5948. 1, 2, 3
- [DC16] DI S., CAPPELLO F.: Fast error-bounded lossy hpc data compression with sz. In *2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS)* (2016), IEEE, pp. 730–739. 1, 2, 5
- [DCG19] DELAUNAY X., COURTOIS A., GOULLON F.: Evaluation of lossless and lossy algorithms for the compression of scientific datasets in netcdf-4 or hdf5 files. *Geoscientific Model Development* 12, 9 (2019), 4099–4113. 2
- [DLB19] DÜBEN P. D., LEUTBECHER M., BAUER P.: New methods for data storage of model output from ensemble simulations. *Monthly Weather Review* 147, 2 (2019). 2, 3
- [DMG20] DÍAZ J., MARTON F., GOBBETTI E.: Interactive spatiotemporal exploration of massive time-varying rectilinear scalar volumes based on a variable bit-rate sparse representation over learned dictionaries. *Computers & Graphics* 88 (2020), 45–56. 2
- [FM07] FOUT N., MA K.-L.: Transform coding for hardware-accelerated volume rendering. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (2007), 1600–1607. 2
- [GG16] GUTHE S., GOESELE M.: Variable length coding for gpu-based direct volume rendering. In *Proceedings of the Conference on Vision, Modeling and Visualization* (2016), pp. 77–84. 2
- [GIGM12] GOBBETTI E., IGLESIAS GUITIÁN J. A., MARTON F.: COVRA: A compression-domain output-sensitive volume rendering architecture based on a sparse representation of voxel blocks. In *Computer Graphics Forum* (2012), vol. 31, Wiley Online Library, pp. 1315–1324. 2
- [HSB\*20] HOANG D., SUMMA B., BHATIA H., LINDSTROM P., KLACANSKY P., USHER W., BREMER P.-T., PASCUCCI V.: Efficient and flexible hierarchical data layouts for a unified encoding of scalar field precision and resolution. *IEEE Transactions on Visualization and Computer Graphics* 27, 2 (2020), 603–613. 2
- [HW22] HÖHLEIN K., WEISS S.: Evaluation of Volume Representation Networks for Meteorological Ensemble Compression: Code for Experiments, Sept. 2022. doi:10.5281/zenodo.7054427. 2
- [HWK\*13] HÜBBE N., WEGENER A., KUNKEL J. M., LING Y., LUDWIG T.: Evaluating lossy compression on climate data. In *International Supercomputing Conference* (2013), Springer, pp. 343–356. 2
- [isa] Hurricane Isabel dataset, IEEE Visualization challenge 2004. <http://sciviscontest-staging.ieeevis.org/2004/data.html>. Accessed: 2022-07-05. 6
- [KRD\*21] KLÖWER M., RAZINGER M., DOMINGUEZ J. J., DÜBEN P. D., PALMER T. N.: Compressing atmospheric data into its real information content. *Nature Computational Science* 1, 11 (2021), 713–724. 2
- [LCA08] LEE M.-C., CHAN R. K., ADJEROH D. A.: Fast three-dimensional discrete cosine transform. *SIAM Journal on Scientific Computing* 30, 6 (2008), 3087–3107. 2
- [Lin14] LINDSTROM P.: Fixed-rate compressed floating-point arrays. *IEEE transactions on visualization and computer graphics* 20, 12 (2014), 2674–2683. 1, 5
- [LJLB21] LU Y., JIANG K., LEVINE J. A., BERGER M.: Compressive neural representations of volumetric scalar fields. *Computer Graphics Forum* 40, 3 (2021), 135–146. 1, 2, 6
- [MAG19] MARTON F., AGUS M., GOBBETTI E.: A framework for gpu-accelerated exploration of massive time-varying rectilinear scalar volumes. In *Computer Graphics Forum* (2019), vol. 38, Wiley Online Library, pp. 53–66. 2
- [MESK22] MÜLLER T., EVANS A., SCHIED C., KELLER A.: Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.* 41, 4 (July 2022), 102:1–102:15. 2, 3
- [MHBB22] MISHRA A., HAZARIKA S., BISWAS A., BRYAN C.: Filling the void: Deep learning-based reconstruction of sampled spatiotemporal scientific simulation data. doi:10.31219/osf.io/aw7rf. 2
- [MLL\*21] MARTEL J. N. P., LINDELL D. B., LIN C. Z., CHAN E. R., MONTEIRO M., WETZSTEIN G.: Acorn: adaptive coordinate networks for neural scene representation. *ACM Transactions on Graphics (TOG)* 40, 4 (Aug 2021), 1–13. 2
- [MON\*19] MESCHEDER L., OECHSLE M., NIEMEYER M., NOWOZIN S., GEIGER A.: Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), pp. 4460–4470. 1, 2, 3
- [MRNK21] MÜLLER T., ROUSSELLE F., NOVÁK J., KELLER A.: Real-time neural radiance caching for path tracing. *ACM Trans. Graph.* 40, 4 (Aug. 2021), 36:1–36:16. 2
- [MST\*20] MILDENHALL B., SRINIVASAN P. P., TANCİK M., BARRON J. T., RAMAMOORTHY R., NG R.: NeRF: Representing scenes as neural radiance fields for view synthesis. In *Computer Vision – ECCV 2020* (2020), pp. 405–421. 3
- [NGW\*20] NECKER T., GEISS S., WEISSMANN M., RUIZ J., MIYOSHI T., LIEN G.-Y.: A convective-scale 1,000-member ensemble simulation and potential applications. *Quarterly Journal of the Royal Meteorological Society* 146, 728 (2020), 1423–1442. 1, 2
- [PFS\*19] PARK J. J., FLORENCE P., STRAUB J., NEWCOMBE R., LOVEGROVE S.: DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), pp. 165–174. 1, 2, 3
- [SCH\*14] SON S. W., CHEN Z., HENDRIX W., AGRAWAL A., LIAO W.-K., CHOUDHARY A.: Data compression for the exascale computing era. *Supercomputing frontiers and innovations* 1, 2 (2014), 76–88. 1
- [SW03] SCHNEIDER J., WESTERMANN R.: Compression domain volume rendering. In *IEEE Visualization, 2003. VIS 2003.* (2003), pp. 293–300. 2
- [TFT\*20] TEWARI A., FRIED O., THIES J., SITZMANN V., LOMBARDI S., SUNKAVALLI K., MARTIN-BRUALLA R., SIMON T., SARAGIH J., NIESSNER M., ET AL.: State of the art on neural rendering. In *Computer Graphics Forum* (2020), vol. 39, Wiley Online Library, pp. 701–727. 2
- [TLY\*21] TAKIKAWA T., LITALIEN J., YIN K., KREIS K., LOOP C., NOWROUZEZHAI D., JACOBSON A., MCGUIRE M., FIDLER S.: Neural geometric level of detail: Real-time rendering with implicit 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), pp. 11358–11367. 2
- [TSM\*20] TANCİK M., SRINIVASAN P., MILDENHALL B., FRIDOVICH-KEIL S., RAGHAVAN N., SINGHAL U., RAMAMOORTHY R., BARRON J., NG R.: Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems* 33 (2020), 7537–7547. 3
- [WHW21] WEISS S., HERMÜLLER P., WESTERMANN R.: Fast neural representations for direct volume rendering. *arXiv preprint* (2021). doi:10.48550/arXiv.2112.01579. 1, 2, 3, 4, 6
- [YL95] YEO B.-L., LIU B.: Volume rendering of dct-based compressed 3d scalar data. *IEEE Transactions on Visualization and Computer Graphics* 1, 1 (1995), 29–43. 2

[ZDL\*20] ZHAO K., DI S., LIANG X., LI S., TAO D., CHEN Z., CAPPELLO F.: Significantly improving lossy compression for hpc datasets with second-order prediction and parameter optimization. In *Proceedings of the 29th International Symposium on High-Performance Parallel and Distributed Computing* (New York, NY, USA, 2020), HPDC '20, Association for Computing Machinery, pp. 89–100. [2](#)





# Attribution 4.0 International Creative Commons

## Deed – reformatted for display in this thesis

### You are free to:

1. **Share** — copy and redistribute the material in any medium or format for any purpose, even commercially.
2. **Adapt** — remix, transform, and build upon the material for any purpose, even commercially.
3. The licensor cannot revoke these freedoms as long as you follow the license terms.

### Under the following terms:

1. **Attribution** — You must give appropriate credit, provide a link to the license, and indicate if changes were made . You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
2. **No additional restrictions** — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation.

No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material.

### Notice

This deed highlights only some of the key features and terms of the actual license. It is not a license and has no legal value. You should carefully review all of the terms and conditions of the actual license before using the licensed material.

Creative Commons is not a law firm and does not provide legal services. Distributing, displaying, or linking to this deed or the license that it summarizes does not create a lawyer-client or any other relationship.

### Deed Source / Canonical URL

<https://creativecommons.org/licenses/by/4.0/>



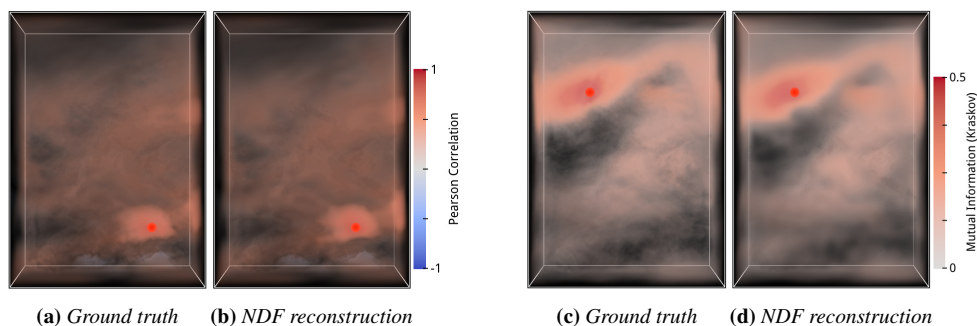
# Neural Fields for Interactive Visualization of Statistical Dependencies in 3D Simulation Ensembles

F. Farokhmanesh<sup>1</sup>, K. Höhle<sup>1</sup>, C. Neuhauser<sup>1</sup>, T. Necker<sup>2</sup>, M. Weissmann<sup>2</sup>, T. Miyoshi<sup>3</sup> and R. Westermann<sup>1</sup>

<sup>1</sup>Technical University of Munich, Department of Computer Science, School of Computation, Information and Technology, Germany

<sup>2</sup>University of Vienna, Department of Meteorology and Geophysics, Austria

<sup>3</sup>RIKEN Center for Computational Science, Kobe, Japan



**Figure 1:** Neural dependence fields have learned to infer 1500 billion point-to-point Pearson correlation (left) or mutual information estimates (right) in a 1000-member simulation ensemble. Inference of the dependencies between data values at an arbitrary grid vertex (red dot) to all other vertices in a  $250 \times 352 \times 20$  grid takes 9 ms on a high-end GPU. Ground truth volume renderings and network results, respectively, are shown in Figures 1a, 1c and 1b, 1d. The network requires only 1 GB at runtime.

## Abstract

We present neural dependence fields (NDFs) – the first neural network that learns to compactly represent and efficiently reconstruct the statistical dependencies between the values of physical variables at different spatial locations in large 3D simulation ensembles. Going beyond linear dependencies, we consider mutual information as an exemplary measure of non-linear dependence. We demonstrate learning and reconstruction with a large weather forecast ensemble comprising 1000 members, each storing multiple physical variables at a  $250 \times 352 \times 20$  simulation grid. By circumventing compute-intensive statistical estimators at runtime, we demonstrate significantly reduced memory and computation requirements for reconstructing the major dependence structures. This enables embedding the estimator into a GPU-accelerated direct volume renderer and interactively visualizing all mutual dependencies for a selected domain point.

## CCS Concepts

• Computing methodologies → Neural networks; Computer graphics; • Applied computing → Earth and atmospheric sciences;

## 1. Introduction

Estimating statistical dependencies between physical variables at different spatial locations is crucial for understanding physical systems in various scientific and engineering fields. An important application lies in meteorology, where accurate weather forecasting relies on extensive numerical simulations. Weather forecasts need to account for randomness, and ensembles of simulations with varying initial conditions and model specifications are used to

quantify uncertainty. It is essential to analyze statistical relations, such as spatio-temporal auto-correlations within forecast fields or correlations between different forecast variables, to translate volumetric ensemble fields into reliable forecasts. Studying statistical dependence between random variables is well-researched, and measures exist for assessing linear and non-linear relationships. However, determining relations in 3D ensemble fields presents challenges. Computing dependencies on the fly may be compu-

tationally costly, and storing all point-to-point correlations leads to an explosion in required memory. For example, in a simulation ensemble with 1000 members on a  $250 \times 352 \times 20$  grid, storing all correlations would need over 3 terabytes of memory, making it infeasible. Additionally, computing correlations between arbitrary point pairs on the fly requires the entire ensemble to fit into the working memory, and more complex measures like mutual information (MI) take roughly 16 minutes on a recent multi-core CPU, hindering interactive analysis of correlation structures.

In this work, we address these challenges by introducing neural dependence fields (NDFs), a novel compact representation of the major correlation structures in large multi-variable ensembles. For this, we propose to interpret fields of two-point correlation measures in 3D ensembles as scalar fields  $R$  over the domain of position pairs in 3D space, i.e.,  $R : \Omega \times \Omega \rightarrow \mathbb{R}$ , for  $\Omega \subset \mathbb{R}^3$ . Taking inspiration from recent progress in neural scene representations and multi-dimensional tensor decomposition, we design a neural network architecture that exploits self-similarity in the correlation fields to learn a compact representation thereof. At the same time, the network enables fast sampling out of the neural representation. Thus, we can avoid holding the ensemble in memory and are able to speed up the computation of correlation estimates significantly, especially for complex non-linear dependence measures. For the ensemble considered in this work, it takes roughly 9 ms to reconstruct dependencies between an arbitrary reference point and all other points in the domain. This allows embedding the network into an interactive volume rendering pipeline, which enables instant visualization and comparison of single-variable auto-correlation and inter-variable correlation fields. In summary, our contributions are:

- A compact neural network architecture to learn statistical point-to-point correlations in large ensemble fields.
- The embedding of neural network-based correlation reconstruction into direct volume rendering to enable interactive visual exploration of the dependencies in the 3D domain.
- A demonstration of interactive correlation analysis for a large meteorological 3D ensemble field.

The proposed method is agnostic towards the choice of correlation measure, such that both linear (e.g., Pearson correlation) and compute-intensive non-linear measures (e.g., MI) are supported. The network manages to reconstruct the major correlation structures faithfully, despite showing a tendency to smooth out fine details (cf. Fig. 1). In view of the complexity of the information to be learned, our results demonstrate the potential of network-based correlation learning and open the door for future research in this field, e.g., by looking into more powerful architectures or specialized loss functions. The code for the project is publicly available at [FH23, NS23].

## 2. Related work

**Scene representation networks and neural fields** Scene representation networks (SRNs) are neural networks trained to derive compact representations of 3D models and scenes. Originally, they were proposed for 2D or 3D position coordinate mapping. Early examples include encoding surface models as implicit functions or occupation maps using fully-connected neural networks [MON\*19, CZ19, PFS\*19]. Later, they evolved to encode

diverse volumetric scenery information, such as neural radiance fields [TSM\*20, MST\*21] and were named neural fields [XTS\*22]. A neural field is a neural network that learns a parametrization of spatio-temporal multi-dimensional physical fields over spatial coordinates. In inference, coordinates are transformed into latent-space representations and then decoded to obtain the physical quantity.

Recent work on neural fields has shown that domain-oriented input feature encodings can significantly boost reconstruction quality. Chabra et al. [CLI\*20] proposed laying out trainable parameters in a grid of latent features to learn spatial variations more directly. Refinements include adaptive data structures [MLL\*21], fixed multi-resolution grids [TLY\*21], and multi-resolution spatial hashing [MESK22, MRNK21a], enabling multi-scale learning of spatial feature maps. Comprehensive reviews of SRN-related literature focusing on neural scene representations are available [HSB\*20, TFT\*20].

In scientific data visualization, Lu et al. [LJLB21] introduced SRNs for volumetric data compression, which was sped up and refined by Weiss et al. [WHW22] through the use of trainable feature representations in combination with an efficient GPU implementation. Höhle et al. [HWW22] employ neural fields for compressing ensemble data by sharing model parameters between different ensemble members. Both works demonstrate the combination of volume rendering and network inference as used in this study.

**Correlation visualization** Volume rendering was chosen to demonstrate network-based reconstruction for correlation visualization in interactive workflows. However, alternative techniques for correlation visualization have been proposed, including clustering [PW12, LWS18, EHL21], correlation matrices [CWMW11, EHL21], diagram views [STS06, BDSW13, ZMZM14, LS16], and specific feature-based approaches like analyzing dependencies in flow fields with particle trajectories [BMLC19]. Correlation subsampling [GW10, CWMW11] identifies prominent features in correlation fields for analysis. These approaches complement our contribution, as they address significant structures in correlation fields or develop effective visual encodings. Our approach seamlessly integrates with these techniques, reducing memory and computation requirements for accessing correlations in large 3D fields.

## 3. Statistical dependence in ensemble fields

We quantify statistical dependencies in ensemble datasets through bivariate correlation measures  $\rho : \mathbb{R}^N \times \mathbb{R}^N \rightarrow \mathbb{R}$  between vectors of paired random samples. For this, let  $\Omega \subset \mathbb{R}^3$  be a simulation domain in 3D space, and let  $\mathcal{E} = \{E_i : 0 \leq i < N\}$  be an ensemble of  $N$  multi-variable fields  $E_i : \Omega \rightarrow \mathbb{R}^d$ . The index  $i$  suggests a fixed but arbitrary enumeration of the members. For all  $i$  and  $0 \leq v < d$ , let  $E_i^v : \Omega \rightarrow \mathbb{R}$  denote the scalar field associated with variable  $v$  in member  $E_i$ . For a given position  $\mathbf{p} \in \Omega$ , we refer to the local sample of variable values as  $\mathbf{e}^v(\mathbf{p}) := (E_i^v(\mathbf{p}) : 0 \leq i < N) \in \mathbb{R}^N$ . Then, for variables  $\mu$  and  $v$ , we consider the field of  $\mu$ - $v$ -correlations,  $R_{\mu v} : \Omega \times \Omega \rightarrow \mathbb{R}$ , where for all pairs of positions  $(\mathbf{p}_\mu, \mathbf{p}_v) \in \Omega^2$  the field value is defined as  $R_{\mu v}(\mathbf{p}_\mu, \mathbf{p}_v) := \rho(\mathbf{e}^\mu(\mathbf{p}_\mu), \mathbf{e}^v(\mathbf{p}_v))$ . Note that the indexing of the position variables is used to imply that position  $\mathbf{p}_\mu$  (position  $\mathbf{p}_v$ ) alters the value of  $R_{\mu v}(\cdot, \cdot)$  by changing the reference

position in field  $\mu$  (field  $\nu$ ), respectively. Special attention is paid to the case  $\mu = \nu$ , which we refer to as the  $\mu$ -self-correlation field  $S_\mu := R_{\mu\mu}$ . Notably,  $S_\mu$  is symmetric under exchange of position coordinates, i.e.,  $S_\mu(\mathbf{p}_1, \mathbf{p}_2) = S_\mu(\mathbf{p}_2, \mathbf{p}_1)$  for all  $\mathbf{p}_1, \mathbf{p}_2 \in \Omega$ .

This work uses Pearson correlation and MI from the wide range of possible dependence measures. Both represent opposite sides of the spectrum of computational cost and indicate different kinds of dependence [BMLC19].

### 3.1. Pearson product-moment correlation coefficient

The Pearson correlation coefficient, or Pearson’s  $r$ , measures the linear correlation between random variable pairs. It is commonly used in data visualization to explore relationships between variables. Given a set of paired random samples,  $\mathbf{e}_1, \mathbf{e}_2 \in \mathbb{R}^N$ , the Pearson correlation coefficient is defined as

$$r = \frac{\text{cov}(\mathbf{e}_1, \mathbf{e}_2)}{\sqrt{\text{var}(\mathbf{e}_1) \text{var}(\mathbf{e}_2)}}, \quad (1)$$

wherein  $\text{var}(\cdot)$  and  $\text{cov}(\cdot)$  denote sample variance and covariance of the respective random samples. With a range of -1 to 1, Pearson correlation indicates correlation (+1), anti-correlation (-1), or the absence of correlation (0). It is easy to interpret, quantifying the strength and direction of the relationship between variables. However, caution is needed when the relationship is nonlinear or when outliers are present, as they can significantly affect the correlation coefficient.

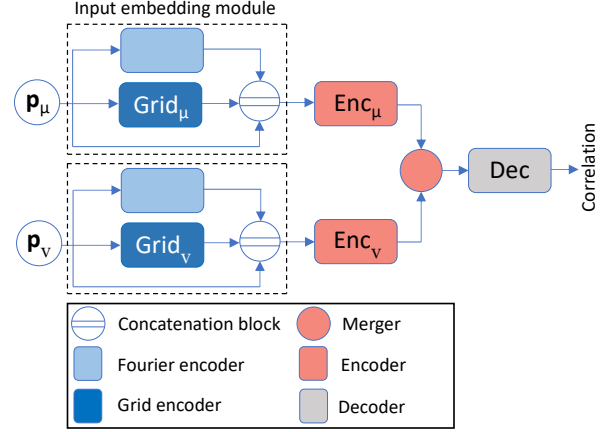
### 3.2. Mutual information

MI is widely used in machine learning, statistics, and information theory [CT\*91] to measure similarity or correlation between random variable pairs. Unlike linear correlation, MI can detect non-linear and non-monotonic dependencies that are not evident in covariance. Mathematically, this is expressed as

$$I(X_1; X_2) = H(X_1) - H(X_1|X_2) = H(X_2) - H(X_2|X_1), \quad (2)$$

wherein  $X_1$  and  $X_2$  are random variables,  $I(X_1; X_2)$  is the MI of  $X_1$  and  $X_2$ ,  $H(X_1)$  is the entropy of  $X_1$ , and  $H(X_1|X_2)$  is the conditional entropy of  $X_1$  given  $X_2$ . Note that MI is symmetric under the exchange of  $X_1$  and  $X_2$ .

Estimating MI from finite samples  $\mathbf{e}_1$  and  $\mathbf{e}_2$  of random variables  $X_1$  and  $X_2$ , i.e. computing  $I(\mathbf{e}_1, \mathbf{e}_2)$ , is computationally expensive. Existing algorithms struggle to scale with large sample sizes [KSG04, MRL95]. More recent copula-based and neural-network-based variational MI estimators offer better performance in high-dimensional data spaces but still pose computational challenges [ZD11, BBR\*18]. In our study, we compute ground truth MI fields using the nearest-neighbor-based estimator of Kraskov et al. [KSG04], implemented in parallel. However, detailed performance analysis in section 6 shows that estimating MI fields for visualization exceeds the time constraints of interactive data analysis. Mutual information neural estimation (MINE) is a recent method that uses neural networks to estimate MI [BBR\*18]. It trains a neural network to learn a lower bound on MI, which provides an estimate. MINE has shown better performance than traditional MI estimation methods on benchmark datasets. Nevertheless, in our sce-



**Figure 2: NDF architecture.**  $\mathbf{p}_\mu$  and  $\mathbf{p}_\nu$  are respectively the reference and query positions.  $\text{Grid}_\mu$  and  $\text{Grid}_\nu$ , respectively, are the hash grids. Variable-specific encoders  $\text{Enc}_\mu$  and  $\text{Enc}_\nu$  are MLPs. Since the feature grids involve trainable parameters, each encoder is equipped with a separate grid. Merger indicates the multiplication of the encoder outputs.

nario, the inefficiency arises because MINE operates on a member-wise basis and requires all members to be present in memory.

## 4. Neural dependence fields

To enable the use of large sets of point-to-point dependence measures, we perform the computationally expensive calculations of these measures in a preprocess and encode the two-point  $\mu$ - $\nu$ -correlation fields  $R_{\mu\nu}$  (as defined in section 3) with memory- and compute-efficient neural scene representations,  $\Phi_{\mu\nu} : \Omega \times \Omega \rightarrow \mathbb{R}$ . The network is obtained by solving the optimization problem

$$\Phi_{\mu\nu} := \arg \min_{\Phi} \mathbb{E}_{(\mathbf{p}_1, \mathbf{p}_2) \sim \mathcal{U}(\Omega^2)} [d(\Phi(\mathbf{p}_1, \mathbf{p}_2), R_{\mu\nu}(\mathbf{p}_1, \mathbf{p}_2))], \quad (3)$$

wherein  $\Phi(\cdot, \cdot)$  is the neural network,  $d(\cdot, \cdot)$  is a similarity metric, such as  $L_1$  or  $L_2$  loss, and the expectation  $\mathbb{E}[\cdot]$  is taken over samples of position pairs from a uniform distribution,  $\mathcal{U}(\Omega^2)$ , with support  $\Omega^2$ . The optimization is carried out iteratively using stochastic gradient descent. Using tractably sized batches of position pairs simultaneously avoids storing excessive amounts of correlation samples. After training,  $\Phi_{\mu\nu}$  is a compact correlation field encoding, enabling rapid sample reconstruction. Classical compression methods like TThresh and SZ cannot achieve similar compaction due to fixed discretization and computational infeasibility. Additionally, once the correlation network is trained, there is no need to keep the entire ensemble dataset in memory, allowing the approach to scale efficiently to large ensemble sizes.

### 4.1. Network architecture

NDFs differ from classical neural scene representations as they operate on a bi-spatial domain with six dimensions (6D). Due to the curse of dimensionality, training efficiency is lowered by the additional dimensions since covering the domain adequately with samples becomes exponentially more complex. To overcome this, we

construct NDFs to utilize sparse sampling information efficiently, which improves memory efficiency and avoids the computation of excessive amounts of correlation samples.

As shown in Figure 2, we propose a bipartite network architecture, which consists of two variable-specific encoder networks,  $\text{Enc}_\mu$ , and  $\text{Enc}_\nu$ , along with a shared decoder network. All networks are implemented as multi-layer perceptrons (MLPs) with  $l$  fully-connected layers and  $c$  hidden channels using *SnakeAlt* activation [WHW22]. Each encoder model receives information about one of the two positions between which the correlation should be reconstructed. Positions are translated into a latent feature vector, which is merged via element-wise multiplication and forwarded to the decoder for the final prediction. This architecture allows each encoder to be trained on only the marginal space  $\Omega \subset \mathbb{R}^3$ , which improves the training efficiency by increasing the effective amount of correlation samples per volume. In combination with the spatial coherence of the ensemble fields, this enables the model to infer correlations even for point pairs that were not seen during training, thus saving computation time and memory requirements.

The decomposition is similar in spirit to the approach used in TensorRF [CXG\*22] or K-planes [FKMW\*23], where 3D feature tensors are decomposed into linear combinations of tensor products between lower-dimensional feature vectors and matrices for higher parameter efficiency. In the proposed NDF, features over a 6D domain are decomposed into an outer product of fields over a 3D domain. The accuracy of predictions relies heavily on using multiplication for feature merging. Alternative combination methods, such as concatenation, addition or absolute difference, result in a substantial drop in prediction fidelity, which is in line with findings in [FKMW\*23]. Deviating from TensorRF and K-planes, we found applying MLPs before and after feature merging beneficial, which we validate in more detail in section 6.

For self-correlation fields  $S_\mu$  (as defined in section 3) and the corresponding NDFs  $\Phi_{\mu\mu}$ , we further constrain the architecture to use identical encoders for both positions, i.e.,  $\text{Enc}_\mu = \text{Enc}_\nu$  (and  $\text{Grid}_\mu = \text{Grid}_\nu$ , see below for details). This helps to keep the models small and ensures symmetry of the learned fields under exchange of the query positions on an architectural level, i.e.,  $\Phi_{\mu\mu}(\mathbf{p}_1, \mathbf{p}_2) = \Phi_{\mu\mu}(\mathbf{p}_2, \mathbf{p}_1)$  is fulfilled trivially by design and does not need to be learned in expensive training iterations. To improve the capability of the encoders to learn high-frequency patterns as well as spatially distributed and multi-scale features, we employ Fourier features [MST\*21, TSM\*20] as well as multi-resolution hash-grids [MESK22] on the position coordinates, which are concatenated to the raw positions before being processed by the encoders. The input embedding modules are marked with dashed rectangles in Figure 2.

#### 4.2. Input embedding

For a given vector of input coordinates,  $\mathbf{p} = (p_x, p_y, p_z) \in \mathbb{R}^3$ , Fourier features increase the spread between spatially close positions by embedding the position information into a higher-dimensional space using the fixed feature mapping

$$\mathbf{f}_{ij} = (\sin(\omega_i \mathbf{n}_j \cdot \mathbf{p}), \cos(\omega_i \mathbf{n}_j \cdot \mathbf{p})), \quad (4)$$

wherein  $\omega_i = 2^i \pi$  for  $0 \leq i < L \in \mathbb{N}$ , and  $\mathbf{n}_j \in \mathbb{R}^3$  are the axis-aligned unit vectors for  $j \in \{x, y, z\}$ . With Fourier features, the model can better resolve high-frequency patterns while not affecting the number of trainable parameters (and thus memory consumption) due to the fixed functional form of the mapping. In our implementation, we empirically determined  $L = 12$  as a good choice for the number of Fourier frequencies.

Multi-resolution hash grids use hash tables filled with trainable feature vectors to populate the domain at various resolutions [MESK22]. By hashing 3D grid indices, vectors are assigned to regular grid positions at multiple scales, enabling retrieval of feature vectors at arbitrary positions through tri-linear interpolation. This allows the creation of virtual feature grids at any resolution with a fixed memory budget.

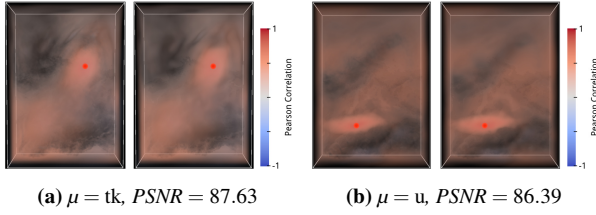
The feature vectors for different resolution levels are concatenated and trained jointly with subsequent model parts using stochastic gradient descent. Hash collisions equilibrate during training due to the pseudo-random hash mapping and multiple resolutions, ensuring adaptive and local feature capacity distribution. The critical parameter for the expressiveness of the hash grid is the hash table size,  $2^T$  for  $T \in \mathbb{N}$ , which also determines memory complexity. Other hyper-parameters include the dimension of the feature vectors per resolution level, the number of resolution levels, and the grid resolution on each level. We use 6 resolution levels with virtual grids of size  $16^3$  on the coarsest level, doubling with each finer level. These parameters ensure that the feature granularity matches the spatial resolution of the original dataset, avoiding higher memory consumption with finer levels while maintaining or improving the reconstruction accuracy.

#### 4.3. Training

During training, we use a rectangular simulation domain  $\Omega$  with data samples on a regular grid, rescaled to fill the symmetric unit cube, i.e.,  $\Omega = [-1, 1]^3$ . We generate  $10^6$  pairs of uniformly distributed random positions  $(\mathbf{p}_\mu, \mathbf{p}_\nu)$  in  $[-1, 1]^3$ , retrieve samples  $\mathbf{e}^\mu(\mathbf{p}_\mu)$  and  $\mathbf{e}^\nu(\mathbf{p}_\nu)$  using trilinear interpolation in the original ensemble dataset, and compute correlations  $R_{\mu\nu}(\mathbf{p}_\mu, \mathbf{p}_\nu)$ . The models are trained to optimize the  $L_1$  loss as a similarity measure, with  $L_2$  loss yielding similar results in our experiments. We use the Adam optimizer [KB14] with an initial learning rate of  $3 \times 10^{-4}$  and 1000 samples per batch. An adaptive learning rate scheduler reduces the learning rate by a factor of 0.1 after 5 passes without improvement in reconstruction accuracy. After every epoch, the training samples are renewed. The total training duration is 200 epochs.

#### 5. Correlation visualization

Once trained, the network can estimate dependencies for any position pair. Multiple queries can be batched efficiently for parallel processing on a GPU using the tiny-cuda-nn framework [Mü21], which features a fully-fused MLP implementation [MRNK21b] with fast 16-bit inference using tensor cores on NVIDIA GPUs and provides functionality for the multi-resolution hashed feature grids [MESK22]. The custom activation functions *Snake* and *SnakeAlt* [WHW22] were added to a fork of the library to support



**Figure 3:** Point-to-point Pearson self-correlations  $S_\mu$  for different variables  $\mu$  (temperature  $tk$  and longitudinal component  $u$  of wind speed) and reference positions  $\mathbf{p}_{ref}$ . Figures show ground truth (left) and NDF reconstruction (right). The reference is shown in red.

the model architecture, as simpler activation functions like ReLU led to inferior reconstruction accuracy.

Network training is performed in PyTorch using the Python bindings of the library. To enable interactive visualizations, the binary weights of the MLP encoder and decoder can then be loaded by a tiny-cuda-nn module into whatever GPU correlation visualization is used. In our primary use case, we access the network from a GPU-based volume renderer implemented in Vulkan [The23]. The renderer is tied to a graphical interface, in which the user is able to select reference points  $\mathbf{p}_{ref} \in [-1, 1]^3$ , for which correlation samples are reconstructed and displayed as a density field. Specifically, we consider the case of displaying volumetric correlation fields,  $\phi: [-1, 1]^3 \rightarrow \mathbb{R}$ , where  $\phi(\mathbf{p}) := \Phi_{\mu\nu}(\mathbf{p}, \mathbf{p}_{ref})$  or  $\Phi_{\mu\nu}(\mathbf{p}_{ref}, \mathbf{p})$ .

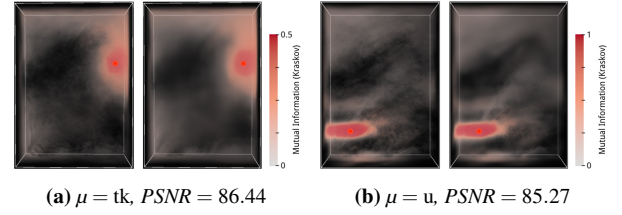
For visualization, the correlation fields are sampled on a grid with resolution  $X \times Y \times Z$ . A CUDA input buffer is prepared for the reference point  $\mathbf{p}_{ref}$  and passed to the tiny-cuda-nn encoder module to get the encoded reference vector. The grid is divided into  $\lceil (X \times Y \times Z)/M \rceil$  query batches of size  $M$ . Each batch is encoded, and the reference and query features are multiplied before being passed to the tiny-cuda-nn decoder module to obtain correlation estimates in an output buffer shared between Vulkan and CUDA, preventing race conditions with shared semaphores. Finally, the shared output buffer is copied to a 3D Vulkan image for visualization in the volume renderer.

## 6. Performance and Quality Analysis

Here, we showcase the NDF model for interactive visual analysis of spatial statistical dependencies in a large weather forecast ensemble. We examine the network’s reconstruction speed and memory requirements and compare the results to ground truth dependence fields obtained using Pearson correlation coefficients and MI on GPU and CPU.

### 6.1. Dataset

We validate our approach using a convective-scale multi-variable ensemble dataset (CSEns) by Necker et al. [NGW\*20]. It consists of 1000 numerical simulations of a 3D atmospheric dynamics model over a rectangular region in central Europe, with a grid size of  $250 \times 352$  nodes and 20 discrete height levels. The simulations span six hours, and we select the last time step with the most interesting features for visualization [HWW22]. The dataset



**Figure 4:** Point-to-point MI self-correlations  $S_\mu$  for different variables  $\mu$  (temperature  $tk$  and longitudinal component  $u$  of wind speed) and reference positions  $\mathbf{p}_{ref}$ . Figures show ground truth (left) and NDF reconstruction (right). The reference is shown in red.

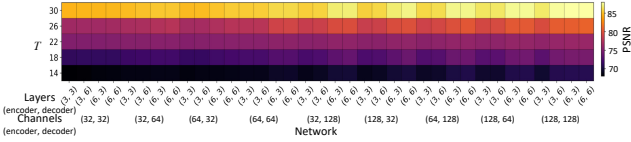
includes 3D data for nine meteorological variables. For validation, we compute ground truth correlation fields for temperature ( $tk$ ) and longitudinal wind ( $u$ ) using all 1000 ensemble members. The generalization of our approach to multiple time steps and inter-temporal dependencies will be explored in future work.

### 6.2. NDF model performance

To shed light on the performance of the NDFs, we conducted a comparative analysis of the runtime of NDFs against reference implementations of Pearson correlation and MI. The implementation of the MI estimator follows Kraskov et al. [KSG04]. All performance measurements are based on the CSEns dataset and were performed on an NVIDIA RTX 3090 GPU and a 6-core Intel Xeon W-2235 CPU, respectively. Notably, a parallel MI estimator is only available on the CPU, whereas we restricted ourselves to a GPU implementation of NDF-based reconstruction. Table 1 shows that our model is about 26x faster than the GPU implementation of the Pearson correlation coefficient. Compared to the CPU implementation of the MI estimator, the factor is  $114,106\times$ . Once the NDF model is trained, it takes 9 ms to reconstruct the dependencies between the data values at a selected grid point and all other grid points. The network model requires roughly 1 GB of memory while keeping the entire dataset in memory for one variable would amount to 7 GB. Training of the NDF takes approximately one hour on the aforementioned machine. Even though it is difficult to estimate the performance of a GPU implementation of the MI estimator, it can be assumed that even an optimized GPU-accelerated estimator will be significantly slower than the NDF model. The MI estimator must construct search structures for nearest-neighbour queries for all requested samples, a much more elaborate process than passing data through fully-fused MLP kernels.

**Table 1:** Performance comparison. For a selected grid point, the dependence measures are computed for all other grid points in a  $250 \times 352 \times 20$  simulation grid using a single variable. A GPU implementation of the MI estimator is not available.

	NDF (ours)	Pearson	MI [KSG04]
CPU	–	4772 ms	1026957 ms
GPU	9 ms	234 ms	–



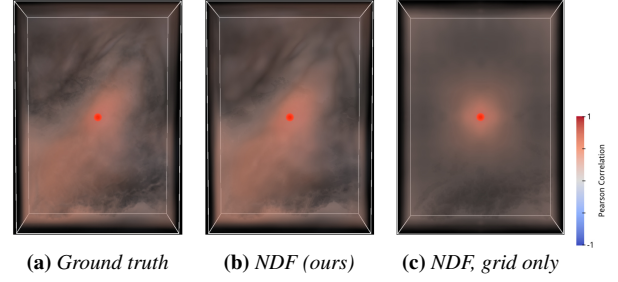
**Figure 5:** Impact of hash table size and MLP hyperparameters on NDFs’ reconstruction quality. The horizontal axis displays various encoder and decoder configurations, including the number of layers and hidden channels. The vertical axis shows the log-2 hash table size  $T$  from section 4.1. The plot considers variable temperature ( $tk$ ) and Pearson correlation, with similar behavior observed for other variables and similarity metrics.

### 6.3. NDF model accuracy

Figures 1, 3, and 4 visually compare the network’s ability to reconstruct major dependence structures in the variable fields. While fine details may not be equally well reproduced due to the limited number of correlation samples during model training, the proposed bipartite NDF architecture uses sample information efficiently by increasing the effective sampling density for the encoder parts of the model (see section 4.1). Furthermore, experiments with increased sample counts did not significantly improve reconstruction, indicating that sampling density is not the limiting factor for reconstruction accuracy. A trade-off between model capacity (memory size) and reconstruction quality is observed, suggesting that the quality is primarily bounded by the model’s ability to store training information rather than sample availability. The figures displayed dependence fields for selected reference points, representing 3D slices in a much larger 6D correlation space where the model was trained. Storing all possible two-point correlations in 32-bit floating point format would require 6 TB of memory space for the CSEns grid of size  $250 \times 352 \times 20$ . The resulting network size of 1 GB corresponds to an effective compression factor of over 6,000 $\times$ , showing promising results.

Figure 5 shows the trade-off between model size and reconstruction quality. NDFs are trained with various settings for the hash table size  $2^T$  and different complexities of encoder and decoder MLPs, using Pearson self-correlation fields of variable  $tk$ . PSNR values are computed on a set of  $10^6$  position pairs, sampled uniformly from the grid domain. Models are trained for a shorter duration of 50 epochs for efficiency. The figure indicates a minor advantage for models with more complex encoder and decoder MLPs. However, the most significant factor affecting achievable PSNR is the hash table size, which also strongly affects model memory consumption. Increasing the table size leads to higher PSNR values. A doubling of the table size  $2^T$  roughly doubles the model memory requirements, while the volume of the MLP parameterization is limited to only a few kB, making it negligible. In our implementation, we set the maximum table size to less than  $2^{32}$ . For further experiments, we choose  $T = 30$  with 6-layer encoders and decoders, each having 128 channels per layer.

To validate our design choices against TensorRF [CXG\*22] and K-planes [FKMW\*23], we compare the reconstruction quality of the proposed architecture against a TensorRF-like model, where the MLP in the encoder part is omitted, predicting based solely on the



**Figure 6:** Accuracy comparison between NDFs with MLP encoder (complete) and pure grid model (without encoder) using Pearson self-correlation fields for variable temperature ( $tk$ ).

outputs of the input embedding. Figure 6 displays reconstructed Pearson correlation fields for both approaches, highlighting the significant added value of using the MLP before feature merging.

### 6.4. Additional experiments

The following are quantitative results using NDFs for reconstructing Pearson correlation coefficients and MI values. We begin with experiments on variable self-correlation fields  $S_\mu$  and their corresponding networks  $\Phi_{\mu\mu}$ .

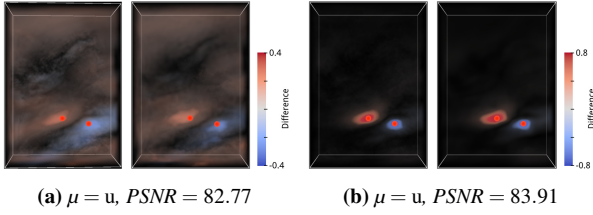
**Single-point experiment** Firstly, the user selects a point in the 3D domain, and the dependence field  $\phi(\mathbf{r})$  is instantly displayed via volume rendering (see supplementary video). Volumetric visualizations, including reconstruction and rendering, can be generated below 10 ms. Moving the reference point interactively highlights different regions with high correlation. This allows the user to identify areas of high internal correlation or observe how correlations change with distance from the reference. The transfer function can be adjusted interactively for better visibility of specific features during the analysis.

**Multi-point experiment** Secondly, the user selects two points in the domain, and the difference between the reconstructed dependence fields for each point is visualized (Figure 7a for Pearson correlation and Figure 7b for MI). The images show the correlation decay around the points and reveal additional structures in other areas within the fields. This allows users to efficiently analyze the differences in the dependence structures related to different points in the 3D domain.

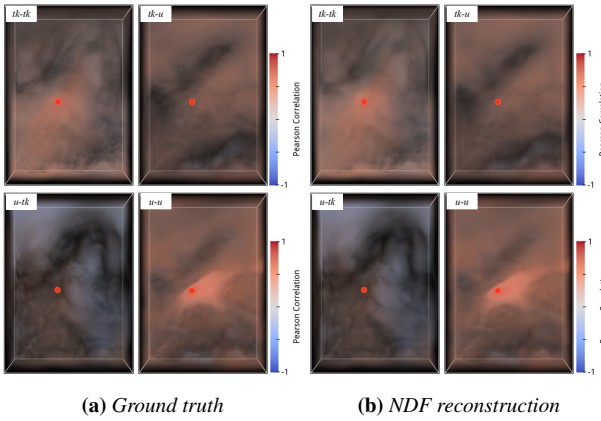
**Multi-variable experiment** Our last experiment demonstrates the use of NDFs for analyzing spatial dependencies between different variables. For a set of  $d \in \mathbb{N}$  variable fields, i.e.,  $V = \{v_1, v_2, \dots, v_d\}$ , this requires training of  $d(d+1)/2$  NDFs  $\Phi_{v_i v_j}$ , for each of the combinations  $v_i, v_j \in V$  with  $1 \leq i < j \leq d$ . For a pair of two different physical variables, such as  $v_1 = tk$  and  $v_2 = u$ , this amounts to training three NDFs,  $\Phi_{tk,tk}$ ,  $\Phi_{u,u}$  and  $\Phi_{tk,u}$ , which emulate the corresponding correlation fields. Note that no separate model is required for  $\Phi_{u,tk}$  if the underlying correlation measure  $\rho$  (see section 3) is symmetric under exchange of arguments, since due to the symmetric architecture of NDFs  $\Phi_{u,tk}(\mathbf{p}_1, \mathbf{p}_2) = \Phi_{tk,u}(\mathbf{p}_2, \mathbf{p}_1)$  for all  $\mathbf{p}_1, \mathbf{p}_2 \in \Omega$ .

Elaborating on the example of  $tk$  and  $u$ , we propose a matrix-like





**Figure 7:** Difference field visualization for multiple points in longitudinal component  $u$  of wind speed. a) Pearson correlation field for two different points, ground truth (left), model reconstruction (right). b) MI for two different points, ground truth (left), and model reconstruction (right).



**Figure 8:** Visualizing dependencies between variables temperature ( $tk$ ) and longitudinal component  $u$  of wind speed using Pearson correlation coefficients. a) Ground truth, b) NDF model reconstruction. Fields show correlations between variables  $tk$  or  $u$  at selected points to the same or different variable at other points.

arrangement of linked volumetric correlation visualizations in the spirit of standard correlation matrix visualizations, i.e., correlation volume matrices. For this, a single reference point is selected, and correlation fields with respect to this point are rendered for all NDF configurations, i.e., all combinations of variables. Figure 8 shows an example of this.

NDFs enable easier visualization with multiple variables as they reduce computation time and memory usage compared to on-the-fly computations using raw data. The standard method would require all variables’ data in GPU memory, and even high-end GPUs with 24 GB of memory can only load two variable fields simultaneously in practice. NDFs of 1 GB each allow networks for up to four variables to fit into the same memory. Exploring the reuse of variable-specific encoder grids in different networks could prevent the memory requirements of NDFs from growing quadratically with the number of variables.

## 7. Conclusion and Future Work

We have introduced and evaluated neural dependence fields (NDFs), a novel approach for encoding and visualizing statistical

dependencies in large 3D ensemble fields. NDFs infer spatial dependencies within single variables and in pairs of different variables. They offer compact representations of linear and non-linear dependence patterns in large ensembles, facilitating the rapid reconstruction of correlation samples from the compact representation. We demonstrated interactive visual analysis of 3D dependence structures through GPU-accelerated direct volume rendering.

Our evaluations show that NDFs faithfully encode and reconstruct the prominent dependence structures in 3D fields while smoothing out some details due to limited network capacities. In the future, we aim to enhance these capacities by adding network stages in the encoder and decoder, exploring alternative architectures like diffusion networks, and trying different loss functions for preserving fine details better (e.g., gradient regularization [LJLB21]). Additionally, we plan to extend NDFs to infer temporal dependence structures in time-varying ensemble fields by decomposing the data into independent fields with spatial variation. This extension would enable new application scenarios in scientific workflows, such as ensemble sensitivity analysis [KRRW19].

## Acknowledgments

The research leading to these results has been done within the sub-project “A7” of the Transregional Collaborative Research Center SFB / TRR 165 “Waves to Weather” ([www.wavestoweather.de](http://www.wavestoweather.de)) funded by the German Research Foundation (DFG). The authors acknowledge crucial contributions by Juan Ruiz and the RIKEN Data Assimilation Research Team for conducting the 1000-member ensemble simulation.

## References

- [BBR\*18] BELGHAZI M. I., BARATIN A., RAJESHWAR S., OZAI S., BENGIO Y., COURVILLE A., HJELM D.: Mutual information neural estimation. In *Proceedings of the 35th International Conference on Machine Learning* (10–15 Jul 2018), Dy J., Krause A., (Eds.), vol. 80 of *Proceedings of Machine Learning Research*, PMLR, pp. 531–540. 3
- [BDSW13] BISWAS A., DUTTA S., SHEN H.-W., WOODRING J.: An information-aware framework for exploring multivariate data sets. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (2013), 2683–2692. doi:10.1109/TVCG.2013.133. 2
- [BMLC19] BERENJKOUB M., MONICO R. O., LARAMEE R. S., CHEN G.: Visual analysis of spatio-temporal relations of pairwise attributes in unsteady flow. *IEEE Transactions on Visualization and Computer Graphics* 25, 1 (2019), 1246–1256. doi:10.1109/TVCG.2018.2864817. 2, 3
- [CLI\*20] CHABRA R., LENSSEN J. E., ILG E., SCHMIDT T., STRAUB J., LOVEGROVE S., NEWCOMBE R.: Deep local shapes: Learning local sdf priors for detailed 3d reconstruction. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX 16* (2020), Springer, pp. 608–625. 2
- [CT\*91] COVER T. M., THOMAS J. A., ET AL.: Entropy, relative entropy and mutual information. *Elements of information theory* 2, 1 (1991), 12–13. 3
- [CWMW11] CHEN C.-K., WANG C., MA K.-L., WITTENBERG A. T.: Static correlation visualization for large time-varying volume data. In *2011 IEEE Pacific Visualization Symposium* (2011), pp. 27–34. doi:10.1109/PACIFICVIS.2011.5742369. 2
- [CXG\*22] CHEN A., XU Z., GEIGER A., YU J., SU H.: Tensorf: Tensorial radiance fields. In *Computer Vision–ECCV 2022: 17th European Conference, 2022, Proceedings, Part XXXII* (2022), Springer, pp. 333–350. 4, 6

- [CZ19] CHEN Z., ZHANG H.: Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 5939–5948. 2
- [EHL21] EVERS M., HUESMANN K., LINSEN L.: Uncertainty-aware Visualization of Regional Time Series Correlation in Spatio-temporal Ensembles. *Computer Graphics Forum* (2021). doi:10.1111/cgf.14326. 2
- [FH23] FAROKHMANESH F., HÖHLEIN K.: Neural Fields for Interactive Visualization of Statistical Dependencies in 3D Simulation Ensembles: Code for Experiments, July 2023. doi:10.5281/zenodo.8186686. 2
- [FKMW\*23] FRIDOVICH-KEIL S., MEANTI G., WARBURG F., RECHT B., KANAZAWA A.: K-planes: Explicit radiance fields in space, time, and appearance. *arXiv preprint arXiv:2301.10241* (2023). 4, 6
- [GW10] GU Y., WANG C.: A study of hierarchical correlation clustering for scientific volume data. In *Advances in Visual Computing: 6th International Symposium, ISVC 2010, 1, 2010, Proceedings, Part III 6* (2010), Springer, pp. 437–446. 2
- [HSB\*20] HOANG D., SUMMA B., BHATIA H., LINDSTROM P., KLACANSKY P., USHER W., BREMER P.-T., PASCUCCI V.: Efficient and flexible hierarchical data layouts for a unified encoding of scalar field precision and resolution. *IEEE Transactions on Visualization and Computer Graphics* 27, 2 (2020), 603–613. 2
- [HWW22] HÖHLEIN K., WEISS S., WESTERMANN R.: Evaluation of volume representation networks for meteorological ensemble compression. 2, 5
- [KB14] KINGMA D. P., BA J.: Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014). 4
- [KRRW19] KUMPF A., RAUTENHAUS M., RIEMER M., WESTERMANN R.: Visual analysis of the temporal evolution of ensemble forecast sensitivities. *IEEE Transactions on Visualization and Computer Graphics* 25, 1 (2019). doi:10.1109/TVCG.2018.2864901. 7
- [KSG04] KRASKOV A., STÖGBAUER H., GRASSBERGER P.: Estimating mutual information. *Phys. Rev. E* 69 (Jun 2004), 066138. doi:10.1103/PhysRevE.69.066138. 3, 5
- [LJLB21] LU Y., JIANG K., LEVINE J. A., BERGER M.: Compressive neural representations of volumetric scalar fields. In *Computer Graphics Forum* (2021), vol. 40, Wiley Online Library, pp. 135–146. 2, 7
- [LS16] LIU X., SHEN H.-W.: Association analysis for visual exploration of multivariate scientific data sets. *IEEE Transactions on Visualization and Computer Graphics* 22, 1 (2016), 955–964. doi:10.1109/TVCG.2015.2467431. 2
- [LWS18] LIEBMAN T., WEBER G. H., SCHEUERMANN G.: Hierarchical correlation clustering in multiple 2d scalar fields. *Computer Graphics Forum* 37, 3 (2018), 1–12. doi:https://doi.org/10.1111/cgf.13396. 2
- [MESK22] MÜLLER T., EVANS A., SCHIED C., KELLER A.: Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.* 41, 4 (jul 2022). doi:10.1145/3528223.3530127. 2, 4
- [MLL\*21] MARTEL J. N., LINDELL D. B., LIN C. Z., CHAN E. R., MONTEIRO M., WETZSTEIN G.: Acorn: Adaptive coordinate networks for neural scene representation. *arXiv preprint arXiv:2105.02788* (2021). 2
- [MON\*19] MESCHEDER L., OECHSLE M., NIEMEYER M., NOWOZIN S., GEIGER A.: Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2019), pp. 4460–4470. 2
- [MRL95] MOON Y.-I., RAJAGOPALAN B., LALL U.: Estimation of mutual information using kernel density estimators. *Physical Review E* 52, 3 (1995), 2318. 3
- [MRNK21a] MÜLLER T., ROUSSELLE F., NOVÁK J., KELLER A.: Real-time neural radiance caching for path tracing. *arXiv preprint arXiv:2106.12372* (2021). 2
- [MRNK21b] MÜLLER T., ROUSSELLE F., NOVÁK J., KELLER A.: Real-time neural radiance caching for path tracing. *ACM Trans. Graph.* 40, 4 (jul 2021). doi:10.1145/3450626.3459812. 4
- [MST\*21] MILDENHALL B., SRINIVASAN P. P., TANCIK M., BARRON J. T., RAMAMOORTHY R., NG R.: Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM* 65, 1 (2021), 99–106. 2, 4
- [Mü21] MÜLLER T.: tiny-cuda-nn, 4 2021. URL: <https://github.com/NVlabs/tiny-cuda-nn>. 4
- [NGW\*20] NECKER T., GEISS S., WEISSMANN M., RUIZ J., MIYOSHI T., LIEN G.-Y.: A convective-scale 1,000-member ensemble simulation and potential applications. *Quarterly Journal of the Royal Meteorological Society* 146, 728 (2020), 1423–1442. doi:https://doi.org/10.1002/qj.3744. 5
- [NS23] NEUHAUSER C., STUMPFEGGER J.: chrismile/Corrender: A correlation field renderer using the Vulkan graphics API, v2023-07-29, July 2023. doi:10.5281/zenodo.8195623. 2
- [PFS\*19] PARK J. J., FLORENCE P., STRAUB J., NEWCOMBE R., LOVEGROVE S.: DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2019), pp. 165–174. 2
- [PW12] PFAFFELMOSER T., WESTERMANN R.: Visualization of global correlation structures in uncertain 2d scalar fields. In *Computer Graphics Forum* (2012), vol. 31, Wiley Online Library, pp. 1025–1034. 2
- [STS06] SAUBER N., THEISEL H., SEIDEL H.-P.: Multifield-graphs: An approach to visualizing correlations in multifield scalar data. *IEEE Transactions on Visualization and Computer Graphics* 12, 5 (2006), 917–924. doi:10.1109/TVCG.2006.165. 2
- [TFT\*20] TEWARI A., FRIED O., THIES J., SITZMANN V., LOMBARDI S., SUNKAVALLI K., MARTIN-BRUALLA R., SIMON T., SARAGIH J., NIESSNER M., ET AL.: State of the art on neural rendering. In *Computer Graphics Forum* (2020), vol. 39, Wiley Online Library, pp. 701–727. 2
- [The23] THE KHROSOS VULKAN WORKING GROUP: Vulkan 1.3.245 - A Specification. <https://registry.khronos.org/vulkan/specs/1.3-extensions/html/vkspec.html>, 2023. Accessed: 2023-03-30. 5
- [TLY\*21] TAKIKAWA T., LITALIEN J., YIN K., KREIS K., LOOP C., NOWROUZEZHAI D., JACOBSON A., MCGUIRE M., FIDLER S.: Neural geometric level of detail: Real-time rendering with implicit 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 11358–11367. 2
- [TSM\*20] TANCIK M., SRINIVASAN P., MILDENHALL B., FRIDOVICH-KEIL S., RAGHAVAN N., SINGHAL U., RAMAMOORTHY R., BARRON J., NG R.: Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems* 33 (2020), 7537–7547. 2, 4
- [WHW22] WEISS S., HERMÜLLER P., WESTERMANN R.: Fast neural representations for direct volume rendering. In *Computer Graphics Forum* (2022), vol. 41, Wiley Online Library, pp. 196–211. 2, 4
- [XTS\*22] XIE Y., TAKIKAWA T., SAITO S., LITANY O., YAN S., KHAN N., TOMBARI F., TOMPKIN J., SITZMANN V., SRIDHAR S.: Neural fields in visual computing and beyond. In *Computer Graphics Forum* (2022), vol. 41, Wiley Online Library, pp. 641–676. 2
- [ZD11] ZENG X., DURRANI T.: Estimation of mutual information using copula density function. *Electronics letters* 47, 8 (2011), 493–494. 3
- [ZMZM14] ZHANG Z., MCDONNELL K. T., ZADOK E., MUELLER K.: Visual correlation analysis of numerical and categorical data on the correlation map. *IEEE transactions on visualization and computer graphics* 21, 2 (2014), 289–303. 2

# Attribution 4.0 International Creative Commons

## Deed – reformatted for display in this thesis

### You are free to:

1. **Share** — copy and redistribute the material in any medium or format for any purpose, even commercially.
2. **Adapt** — remix, transform, and build upon the material for any purpose, even commercially.
3. The licensor cannot revoke these freedoms as long as you follow the license terms.

### Under the following terms:

1. **Attribution** — You must give appropriate credit, provide a link to the license, and indicate if changes were made . You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
2. **No additional restrictions** — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation.

No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material.

### Notice

This deed highlights only some of the key features and terms of the actual license. It is not a license and has no legal value. You should carefully review all of the terms and conditions of the actual license before using the licensed material.

Creative Commons is not a law firm and does not provide legal services. Distributing, displaying, or linking to this deed or the license that it summarizes does not create a lawyer-client or any other relationship.

### Deed Source / Canonical URL

<https://creativecommons.org/licenses/by/4.0/>