Karlsruhe Institute of Technology

Parsimonious data-driven mid-latitude precipitation forecasting

Master's Thesis of

Amit Vincent Sextus

At the KIT Department of Physics
IMK-TRO – Institute of Meteorology and Climate Research Tropospheric Research

First examiner:	Prof. Dr. Peter Knippertz
Second examiner:	Prof. Dr. Andreas Fink
First advisor:	Dr. Dwaipayan Chatterjee

15. December 2024 – 16. September 2025

Karlsruher Institut für Technologie
Fakultät für Meteorologie und Klimaforschung
76131 Karlsruhe

Parsimonious data-driven mid-latitude precipitation forecasting (Master's Thesis)

I declare that I have developed and written the enclosed thesis completely by myself. I have not used any other than the aids that I have mentioned. I have marked all parts of the thesis that I have included from referenced literature, either in their original wording or paraphrasing their contents. I have followed the by-laws to implement scientific integrity at KIT.

Karlsruhe, 16. September 2025

. . . . . .

(Amit Vincent Sextus)

Abstract

Reliable, well-calibrated day-ahead precipitation forecasts are pivotal. While physically based numerical weather prediction (NWP) has gained roughly one extra day of useful medium-range skill per decade, and recent global AI surrogates (e.g., Pangu-Weather, GraphCast) have matched or exceeded leading deterministic NWP on many mid-tropospheric targets, 24 h precipitation remains difficult due to its intermittency, skewness, and multiscale controls. Building on a parsimonious two-stage paradigm developed for northern tropical Africa Walz et al. [30], this thesis adapts a moderate-sized U-Net that predicts daily rainfall fields from process-aware inputs and then calibrates the point forecasts into full predictive distributions using a special case of isotonic distributional regression(IDR) called Easy Uncertainty Quantification(EasyUQ). The system is trained on a Euro–Atlantic domain but optimized for Germany using a spatially tapered loss; verification uses cosine–latitude area-weighted proper scores against a monthly probabilistic climatology baseline (MPC). Using 2007–2019 for training/validation and 2020 for testing, the precipitation-only baseline with regional loss weighting (outside-weight w=0.90) improves Germany-mean CRPS by ∼1.16% versus no weighting and yields overall Continuous Ranked Probability Skill Score (CRPSS) ≈0.246. Adding ERA5 multi-level winds and specific humidity (300/500/700/850 hPa at 18 UTC, with three-day lags) further reduces Continuous Ranked Probability Score (CRPS) by ∼1.13% (CRPSS ≈0.254), with strongest relative gains in winter when synoptic-scale dynamics dominate. Single-level additions- mean sea-level pressure(MSLP), surface pressure(SP), 2 meter temperature (T2M) and total column water vapour (TCWV) provide no statistically discernible extra skill. A seasonal analysis highlights limited sensitivity in JJA (convective regime) and largest mean CRPS in SON (mixed regime with frequent AR influence and occasional tropical remnant interactions). To place these scores in the context of an operational NWP benchmark, we performed an illustrative out-of-sample comparison for JJA 2016. Over the EFAS Europe domain, ECMWF's dual-resolution ensemble exhibited a domain-mean day +1 CRPS of ≈ 1.3 mm that summer; when retraining our system only on 2007–2015 and verifying over Germany (MSWEP on a 1° grid), the day-ahead CRPS was 1.52 mm with CRPSS ≈ 0.29 relative to MPC. These results demonstrate that a compact, physics-aware, regionally-focused AI + calibration pipeline can compete with state-of-the-art NWP in delivering rainfall guidance for Germany at exceptionally low computational cost.

Contents

Abstract i

1. Introduction 1

2. Literature Review 5
2.1. Scientific Foundations of Weather Forecasting . . . 5
2.2. AI Transformation of Weather Forecasting: From NWP to Deep Learning . 7
2.3. Convolutional Neural Network with Isotonic Distributional Regression . . 12
2.3.1. Performance in Tropical Africa . . . 14
2.3.2. Limitations and Rationale for the Present Study . . . 15
2.4. Drivers of German Daily Precipitation . . . 16

3. Methodology 21
3.1. Data Preprocessing . . . 22
3.2. Input Preparation . . . 25
3.3. Neural Network Architecture Design . . . 30
3.3.1. U-Net Architecture Overview . . . 30
3.4. Training Framework and Optimization . . . 33
3.4.1. PyTorch Lightning and Loss Architecture . . . 33
3.4.2. Model interface with Training Framework . . . 35
3.5. Probabilistic Post-Processing and Evaluation . . . 36
3.5.1. Deterministic Performance Assessment . . . 36
3.5.2. Isotonic Distributional Regression Theory and Implementation . . 39
3.5.3. Probabilistic Evaluation Metrics . . . 40
3.5.4. Summary . . . 42

4. Results 43
4.1. Overall Metrics and Comparison with NWP . . . 43
4.2. Optimal Spatial Context . . . 46
4.2.1. The Search for Optimal Regional Weighting . . . 46
4.2.2. Seasonal Sensitivity to Spatial Context . . . 48
4.3. Enhancing Predictions with Atmospheric Information . . . 51
4.3.1. Physical Interpretation of Improvements . . . 51
4.3.2. Visual Analysis . . . 55
4.4. Hyperparameter Tuning . . . 61

5. Conclusion 65

Bibliography **67**

A. Appendix **71**
A.1. Detailed overview Encoder and Decoder 71
A.2. Network Components and Regularization 73
A.3. Spatial Resolution Handling 75
A.4. Data Augmentation and Invariance 76

List of Figures

2.1. Accuracy of forecasts has improved by roughly one day per decade, shown using 500 hPa geopotential height skill. Figure by Hannah Ritchie, *Our World in Data*[24], based on ECMWF data[8]. Licensed under CC BY 4.0. . 7
2.2. Figure 2a shows Z500 RMSE vs lead time (0–10 days), with GraphCast below ECMWF-HRES across nearly all ranges. Source: GraphCast[15] 8
2.3. Spatial structure of the CRPS skill score for probabilistic forecasts of precipitation accumulation with a) EPS, b) EPS+EMOS, c) HRES+EasyUQ, d) DIM-base, e) DIM-full, f) CNN+EasyUQ, and g) the Hybrid forecast, relative to MPC as baseline, for season JAS and combined evaluation folds from 2011 to 2019 Source: Walz et al. [30] 10
2.4. Full extended Euro-Atlantic domain with German subdomain of interest/evaluation highlighted in grey. 19

3.1. Full extended Euro-Atlantic domain with German subdomain and neighbor cells for a discretely tapered loss 32

4.1. NWP CRPS JJA 2016. Domain-mean 24-h precipitation CRPS of the ECMWF dual-resolution ensemble (EFAS Europe domain) for various lead times in JJA 2016 (Gascón et al. [10]). 44
4.2. JJA 2016 CRPSS. Spatial distribution of our model's CRPS skill score for day-ahead precipitation forecasts over Germany in JJA 2016. Skill is calculated against a monthly probabilistic climatology baseline (MPC) 45
4.3. Overall Mean CRPS vs outside weight 48
4.4. Seasonal CRPS as a function of outside weight, demonstrating stronger sensitivity in winter when synoptic-scale dynamics dominate. 49
4.5. Effect of training-set length (expanding window) on Germany-mean CRPS. Each point is the fold-mean CRPS when training on the years up to the abscissa value and validating on the immediate following year. The dashed line shows a fitted linear trend. 55
4.6. Visualisation of model forecast, baseline (w=0.9), DJF 56
4.7. Visualisation of model forecast, baseline (w=0.9), SON 57
4.8. Visualisation of model forecast, baseline (w=0.9), SON, full domain 58
4.9. Visualisation of model forecast, u,v,q, (w=0.9), DJF 59
4.10. Visualisation of model forecast, u,v,q, (w=0.9), SON 60
4.11. Visualisation of model forecast, u,v,q, (w=0.9), SON, full domain 60

List of Tables

4.1. Impact of outside weight on model performance. All experiments use precipitation-only input with lag features and IDR post-processing. Evaluation performed over Germany with cosine-latitude area weighting. 47
4.2. Performance comparison of ERA5 predictor configurations. All models use outside weight = 0.9. 51

1. Introduction

Over the past four decades, physically based numerical weather prediction (NWP) has advanced markedly, adding roughly a day of useful medium-range skill per decade through denser global observing systems, sophisticated data assimilation, finer grids, and improved physics [2]. In parallel, a new generation of global, data-driven "foundation models" (e.g., Pangu-Weather, GraphCast) has demonstrated striking skill and orders-of-magnitude faster inference by learning atmospheric evolution directly from reanalyses [5, 15]. Yet, despite these advances, precipitation remains one of the hardest variables to predict: it is intermittent, highly skewed, and controlled by a multiscale interplay between synoptic forcing, moisture transport, orography, and warm-season convection i.e. properties that stress both dynamical models and generic deep networks. Notably, even the pioneering global AI models initially sidestepped direct precipitation prediction in their headline results, underscoring the need for tailored approaches at this variable and time scale [5, 15].

The success of global AI surrogates, however, comes with trade-offs that matter for precipitation: large up-front training costs and energy use; reduced physical transparency; and (so far) a focus on mid-tropospheric, quasi-Gaussian targets rather than mixed discrete–continuous rainfall. This has motivated a complementary line of work that seeks *parsimonious* models which capture the essential predictors and scales with modest network size, clear process priors, and efficient training. In this "modest game," deep learning is used where it adds unique value, but is constrained by meteorological structure rather than by brute-force capacity. This perspective naturally points to regional, process-aware systems for precipitation that are inexpensive to train and inspect, and whose scope can be matched to the physics and users of interest.

A prominent example of this paradigm is the two-stage CNN+EasyUQ framework of Walz et al. [30] for northern tropical Africa: a moderate-sized U-Net produces deterministic daily rainfall, then non-parametric calibration via Easy Uncertainty Quantification (EasyUQ), a special case of isotonic distributional regression (IDR), yields statistically calibrated predictive distributions from single-valued outputs [30, 29, 11]. In a convection-dominated region where operational ensembles struggle, their framework delivered state-of-the-art probabilistic skill—substantially outperforming raw and post-processed NWP for both occurrence and amount, and achieving large positive CRPS skill relative to a climatological baseline. These results provide a compelling proof-of-concept that a compact, distribution-aware AI system can surpass state-of-the-art alternatives when the physics and scales are aligned with the model design. While this parsimonious AI approach excelled in the tropical regime, it is not obvious how well it will fare in a baroclinic extra-tropical setting.

The tropical and extra-tropical precipitation problems differ significantly. In the deep tropics, coherent, *linear* wave disturbances (e.g., African easterly waves, Kelvin waves) modulate the background in which convection (highly stochastic at grid-point scale) is triggered. Forecast quality hinges on connecting these slowly propagating, quasi-periodic modulations to sub-mesoscale convective responses; NWP systems often struggle to bridge that scale gap because the interaction is mediated by parameterized convection and microphysics. A data-driven model can, in principle, learn this cross-scale mapping from the recent spatio-temporal rainfall pattern and large-scale context, which explains why CNN+EasyUQ shines in the Walz setting.

In the extra-tropics, focusing on Germany as an example, dynamics during most of the year are governed by baroclinic waves and their *non linear* life-cycles: jet-streak forcing, frontal ascent, warm-conveyor belts, and frequent moisture-plume/atmospheric-river preconditioning. Orography (Alps, Black Forest, Mittelgebirge) further shapes totals and spatial patterns [20, 6, 16, 22]. These dynamics are *well captured* by NWP because they arise directly from resolved balance relations and baroclinic instability. Summer introduces some convective intermittency and mesoscale organization, creating a partial resemblance to the tropical regime but winter returns to largely stratiform, synoptic-scale control. This contrast motivates the central research question(RQ) that lies at the heart of the our thesis and some downstream research questions that further guide our path.

RQ1. **Can a parsimonious CNN+EasyUQ system, with a proven track-record over the tropics, deliver well-calibrated 24-hour precipitation forecasts over the mid-latitudes?**

In the tropical setting of Walz et al. [30], the input and verification domains largely overlapped and convection was strongly modulated by slowly propagating, quasi-linear waves i.e. conditions under which a compact, regional CNN plus non-parametric calibration excelled. But since the next-day precipitation over Germany is organized by baroclinic waves, fronts and warm-conveyor belts with upwind precursors that originate far west over the North Atlantic and evolve non-linearly. Two practical design issues therefore follow from RQ1: (i) we must train over a wider Euro–Atlantic domain to expose the model to those precursors while still prioritizing skill *in* Germany; and (ii) we must determine how much atmospheric context (beyond lagged rain) this parsimonious model can benefit from and what are the implications for forecast skill. These considerations lead directly to RQ2 and RQ3 below.

RQ2. **How should wide synoptic context be balanced with regional focus during training?**

RQ3. **How much incremental skill arises from adding atmospheric context beyond lagged rain?**

We adapt the two-stage CNN+EasyUQ framework of Walz et al. [30] to the mid-latitude problem as follows. A moderate U-Net ingests spatial tensors built from three-day lagged precipitation and a compact, process-informed subset of ERA5 fields (winds u, v and q at

300/500/700/850 hPa; additionally mean sea-level pressure, surface pressure, 2-m temperature, and total column water vapour) and outputs a deterministic day-ahead precipitation field. A cell-wise IDR/EasyUQ calibration then maps these single-valued forecasts to full predictive distributions, ensuring statistical calibration without parametric assumptions. To address RQ2, training is performed on an extended Euro–Atlantic domain with a spatially tapered loss that emphasizes Germany while retaining but down-weighting penalties elsewhere to maintain coherent synoptic learning. To address RQ3, we conduct ablations from a precipitation-only baseline to various configurations of added ERA5 predictor variables. Targets are gridded daily precipitation from MSWEP. Skill is verified against a monthly probabilistic climatology using proper scoring rules (CRPS/CRPSS), with cosine–latitude area weighting and seasonal stratifications.

With the research questions and their operationalization in place, Chapter 2 synthesizes the scientific background and related work that motvate or support RQ1–RQ3. Chapter 3 details design implications i.e. data, the U-Net and loss design, the IDR/EasyUQ calibration, and the verification protocol, explicitly mapping choices to each research question. Chapter 4 presents results organized by the questions—transferability and overall skill for RQ1, effects of regional weighting for RQ2, and value of ERA5 predictors for RQ3—together with seasonal diagnostics and case analyses. Chapter 5 synthesizes findings, discusses limitations, and outlines avenues for future work.

Additionally, the full code implementation of this work is publicly available on

https://github.com/amit-sextus/precip

2. Literature Review

2.1. Scientific Foundations of Weather Forecasting

Reliable weather forecasts with well-calibrated probabilities support many stakeholders: flood early-warning and river-basin managers (anticipating both stratiform multi-day events and short intense episodes), civil protection authorities (issuing graded alerts), and energy-system operators (hydropower scheduling, demand/supply balancing under weather uncertainty). Additionally, in these decision contexts, a probabilistic forecast conveys actionable risk information, for example the probability of exceeding an operationally relevant threshold.

This is a problem of predicting how the current atmospheric conditions will evolve. Unlike climate prediction (which deals with long-term averages and boundary conditions), weather forecasting is fundamentally an initial-value problem in fluid dynamics where the evolution of the atmosphere is highly sensitive to the initial conditions provided. One way to forecast the weather is to start with an accurate representation of the atmosphere's current state and then solve the governing physical equations forward in time. Both the former and latter half of that process is far more complex than it may seem at first glance.

In practice an enormous volume of observational data is collected from around the globe from surface weather stations, weather balloons, aircraft, ships, radars, and satellites. The quality-control and merging of these heterogeneous observations to produce a coherent picture of the atmosphere on a three-dimensional grid is termed 'data-assimilation'. This analysis serves as the initial condition for a numerical weather model. In the model integration phase, a computer model solves a system of mathematical equations (the primitive equations of atmospheric motion and thermodynamics) to simulate how the atmospheric state changes over time. This step involves time-stepping the equations forward, often on supercomputers, to produce a forecast for several hours or days into the future. Finally, a post-processing stage generates user-friendly forecast products and often applies statistical corrections or downscaling to improve the raw model output The end result is a set of forecast maps and variables (temperature, winds, precipitation, etc.) that meteorologists can analyze and disseminate to end-users.

It is worth noting that weather forecasting is conducted on multiple time scales and using different methods appropriate to those scales. Nowcasting refers to very short-range forecasts (on the order of 0–6 hours) and often relies on radar and satellite data extrapolation, assuming weather systems move and evolve in the near term similarly to current trends. Short- to medium-range forecasting (out to a week or two) is dominated by global and

regional NWP models as described above. For longer lead times (monthly or seasonal outlooks), the problem transitions towards climate prediction, which involves different techniques (often treating it as a boundary-value problem with ensemble climate models and statistical tools). It is in the short- and medium-range window that the distinction between weather and climate is clear: weather prediction hinges on getting today's state just right and watching it play out, whereas climate outlooks focus on average anomalies and boundary influences.

A fundamental challenge underlying all weather forecasting is the atmosphere's chaotic nature. The atmospheric system is famously sensitive to initial conditions; tiny uncertainties or errors in the starting state can amplify over time. As a result, there is an inherent limit to predictability: even with perfect models, forecast uncertainty grows with forecast lead time.

Modern forecasting systems explicitly tackle this issue by employing ensemble forecasting techniques. Instead of relying on a single deterministic forecast, ensemble prediction systems run the model many times with slight variations in initial conditions and model parameters. These multiple simulations produce a range of possible outcomes, reflecting the uncertainties in the initial state and model physics. For example, the European Centre for Medium-Range Weather Forecasts (ECMWF) operates a 51-member global ensemble where each member simulates the future weather under slightly perturbed conditions. The spread of solutions in an ensemble provides probabilistic information – forecasters can estimate the likelihood of certain events (e.g., a 70% chance of rain) and gauge confidence in a forecast based on how clustered or divergent the ensemble members are. Ensemble forecasting has thus become indispensable for quantifying forecast confidence and conveying risk to users, especially for critical applications like emergency management.

Over the decades, continuous improvements in all components of the forecasting system have led to steady gains in accuracy. Better observational coverage (e.g. advanced satellites monitoring every corner of the globe) and sophisticated data assimilation algorithms have reduced initial condition errors. Meanwhile, advancements in numerical models like finer grid resolutions, more advanced physical parametrizations of sub-grid processes (like cloud microphysics or turbulence), and coupling of atmosphere with ocean and land models have made simulations more realistic. Additionally, increased computational power has enabled national weather services and research centers to run these complex models at higher resolutions and more frequently. As a consequence, forecasts that once were skillful only a day or two out can now be trusted for five to seven days in many regions, and the skillful range continues to inch forward. For instance, what was a state-of-the-art 24-hour forecast in the 1980s is comparable in accuracy to a 3-day forecast by the early 2000s and to roughly a 5-day forecast by the 2020s(See Figure 2.1), highlighting the remarkable progress in NWP. This progress is illustrated by metrics such as the improving 500-hPa geopotential height forecast skill over time, often cited by operational centers to demonstrate how today's 5-day forecast is as good as the 2-day forecast of a few decades ago (a result of better models and data). Though these improvements are impressive, the predictability barrier associated

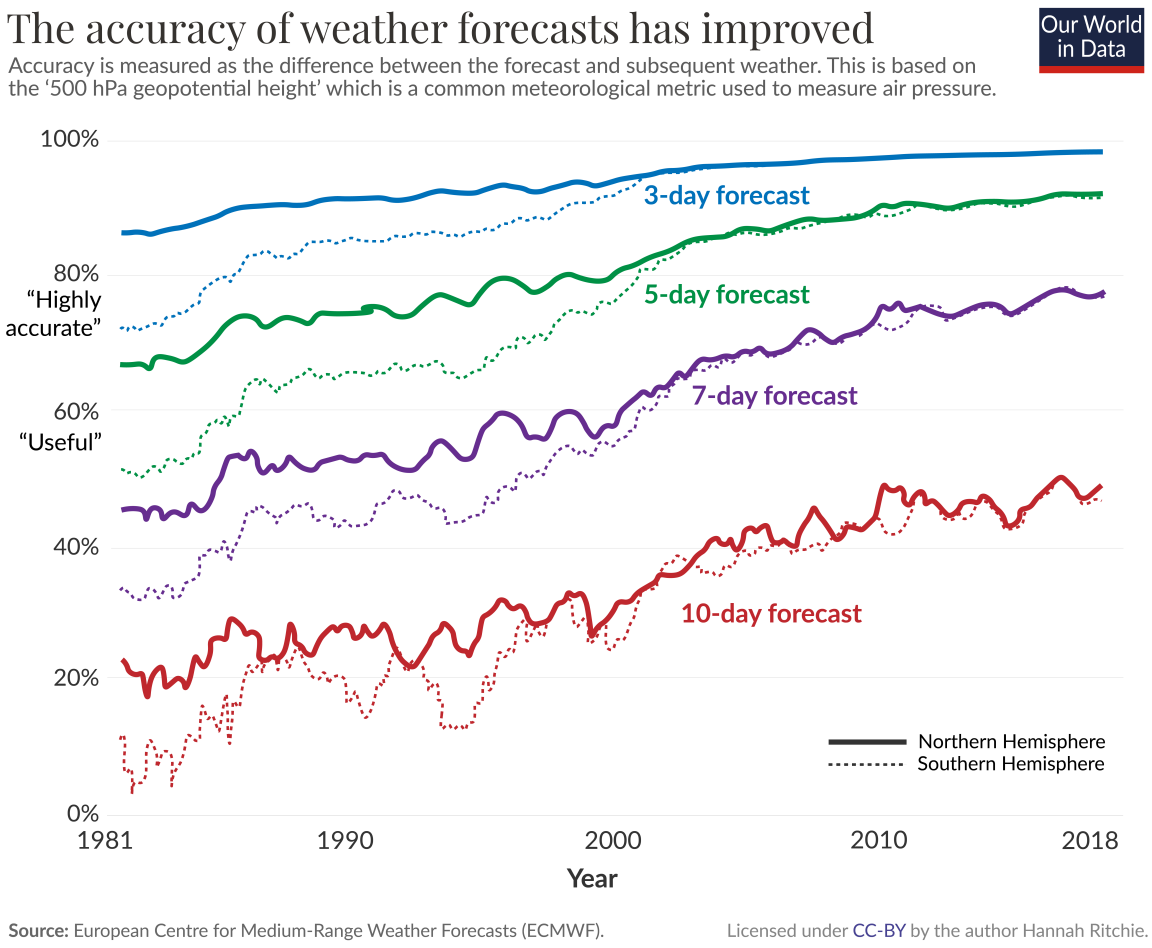

Figure 2.1.: Accuracy of forecasts has improved by roughly one day per decade, shown using 500 hPa geopotential height skill. Figure by Hannah Ritchie, *Our World in Data*[24], based on ECMWF data[8]. Licensed under CC BY 4.0.

with chaos means that entirely eliminating forecast error is impossible i.e there will always be a point at which atmospheric uncertainty dominates. This realization has encouraged researchers to explore new approaches, including advanced statistical methods and machine learning, to further extend predictive capabilities.

2.2. AI Transformation of Weather Forecasting: From NWP to Deep Learning

The past decade has seen an unprecedented transformation in weather forecasting as artificial intelligence (AI) and machine learning (ML) methods have upended the traditional NWP paradigm. For over half a century, NWP—solving physical equations on supercomputers has been the foundation of forecasting, with steady improvements (on the order of one additional day of useful skill per decade) driven by finer models and better data assimilation

[2]. However, this physically based approach is computationally intensive and struggles with processes like deep convection and precipitation. In recent years, a new class of data-driven *global* forecast models has emerged, training on decades of past weather data to learn the evolution of atmospheric patterns directly. This shift in paradigm was dramatically highlighted by the introduction of ultra-large AI forecast models, which have demonstrated that ML can rival, and in some aspects surpass, the accuracy of leading operational NWP systems. The landscape of weather prediction has thus rapidly evolved from one dominated by hand-crafted physical models to a hybrid scene in which brute-force deep learning plays an increasingly prominent role.

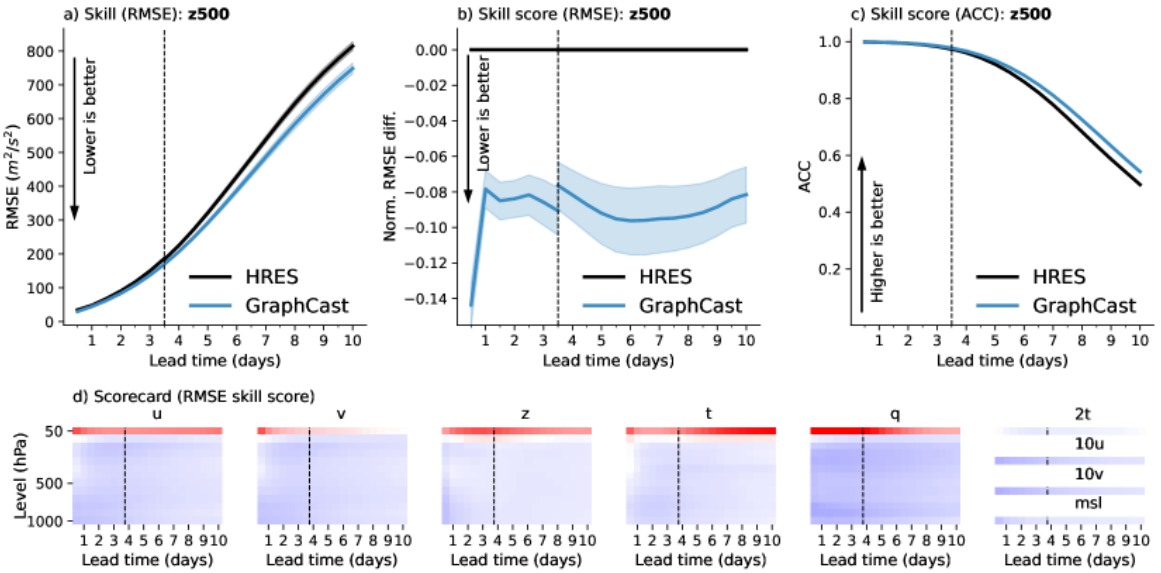

Figure 2.2.: Figure 2a shows Z500 RMSE vs lead time (0–10 days), with GraphCast below ECMWF-HRES across nearly all ranges. Source: GraphCast[15]

At the forefront of this shift are the so-called **global foundation models for weather**. Pangu-Weather [5] and GraphCast [15] are exemplary large-scale AI models that ingest historical reanalysis data and learn to predict the future state of the atmosphere with remarkable skill. These models leverage deep neural network architectures (3D Vision Transformers in Pangu, graph neural networks in GraphCast) to encode the complex spatial-temporal patterns of weather, and are trained on roughly 40 years of past global data to optimize multi-day forecasting accuracy. For instance, Pangu-Weather was trained on 39 years of ERA5 reanalysis and comprises a hierarchy of neural networks specialized by lead time (1 h to 24 h); when iterated, it produces a 7+ day forecast at 0.25° resolution. GraphCast similarly learned from 4 decades of data, but uses a multi-scale mesh and graph neural network approach to directly simulate 10-day global forecasts at 0.25° in one go. Despite differences in design, both models achieve comparable or better accuracy than state of the art NWP. For example, Pangu-Weather's 5-day forecast of 500 hPa geopotential height (a key mid-tropospheric variable) has about 10% lower error (RMSE ≈ 297 gpm) than the European Centre's high-resolution model, even with a single forecast member. GraphCast, likewise, was shown to *significantly outperform* the operational ECMWF deterministic forecast on

about 90% of standard verification targets (spanning multiple variables, levels, and lead times). These are remarkable results, effectively demonstrating AI's ability to capture large-scale atmospheric dynamics. Figure 2.2 illustrates this performance edge, showing that across nearly all forecast ranges up to 10 days the deep learning models maintain lower errors or higher skill scores than the NWP benchmark.

Not only are these AI models accurate, they are incredibly fast in production. Once trained, they generate forecasts in mere seconds on commodity hardware, whereas traditional NWP requires expensive supercomputers running for hours. GraphCast, for example, can produce a 10-day global forecast in under one minute on a single TPU chip, compared to the ~1 hour runtime on thousands of CPU cores needed by ECMWF's model. Pangu-Weather similarly achieves over four orders of magnitude speed-up in inference (1.4 s per forecast on a GPU) relative to the numerical model. This tremendous efficiency gain has major implications: forecasts can be updated more frequently and run at lower cost, potentially democratizing access to high-quality predictions. It is worth noting, however, that these benefits come only after an enormous upfront training effort. Training such large models is a resource-intensive endeavor, typically requiring weeks of computation on specialized hardware. The GraphCast team, for instance, utilized *32 TPU-v4 cores for roughly one month* to train the model (including fine-tuning). This implies a substantial energy and carbon footprint for model development. While the operational use of AI models is thus far more computationally efficient than NWP, their development hinges on big-data and big-compute whhich is a cost feasible for tech giants or major forecast centers, but a potential barrier for smaller institutions. In addition, these models remain largely black boxes, learning implicit correlations in data without providing the explicit physical insight that NWP models offer by construction. These limitations make us wonder how sustainable and generalizable the brute-force deep learning approach will be in the long run (e.g., for extreme events or climate non-stationarity).

Another important caveat is that current large AI weather models have focused on deterministic forecasts of continuous variables (e.g. winds, heights, pressure) and sidestepped some of meteorology's trickiest aspects. Notably, precipitation forecasting was deliberately *excluded* from both the Pangu-Weather and GraphCast experiments because of its intermittency and non-Gaussian distribution. Neither model directly predicts precipitation occurrence/amount from scratch in their seminal results, recognizing that special treatment would be needed for such highly skewed fields. This gap is significant, since precipitation is often the most societally important variable and one that traditional NWP struggles with, especially in the tropics. In response, more targeted AI approaches have arisen to handle probabilistic precipitation prediction. A prime example is the work of Walz et al. [30], who developed a specialized data-driven method for 24-hour rainfall forecasts over Africa. Walz et al. take a two-step approach: first training a moderate-sized convolutional neural network (U-Net) to forecast daily rainfall, then applying "EasyUQ" post-processing to generate full probabilistic prediction distributions. This method, termed CNN+EasyUQ, was tested on 1-day precipitation in northern tropical Africa—a region where operational NWP (e.g. the ECMWF ensemble) has notoriously low skill due to convective rainfall complexity. The results demonstrate that a focused AI model can indeed outperform the traditional methods in this scenario. Walz et al.'s CNN-based system achieved significantly

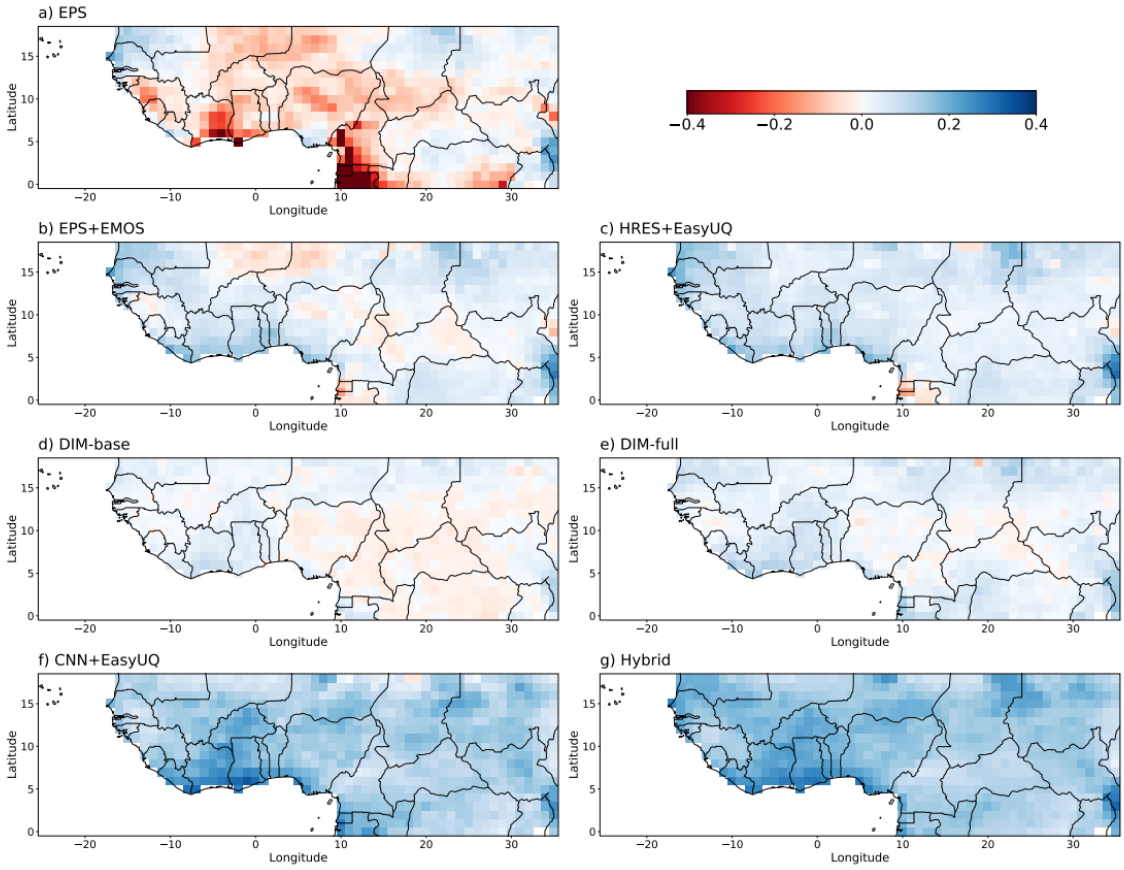

Figure 2.3.: Spatial structure of the CRPS skill score for probabilistic forecasts of precipitation accumulation with a) EPS, b) EPS+EMOS, c) HRES+EasyUQ, d) DIM-base, e) DIM-full, f) CNN+EasyUQ, and g) the Hybrid forecast, relative to MPC as baseline, for season JAS and combined evaluation folds from 2011 to 2019 Source: Walz et al. [30]

lower continuous ranked probability score (CRPS) than the ECMWF ensemble and various statistical baselines, indicating better overall probabilistic accuracy. In fact, their approach yielded positive CRPS skill scores up to 30% when measured against a climatology baseline, markedly surpassing both raw and post-processed NWP ensemble forecasts for both rainfall occurrence and amount. It is relevant to note that a 30% positive skill score against climatology is significantly less impressive in the mid-latitude context unlike the tropical context for meteorological reasons we will discuss later. Figure 2.3 exemplifies this improvement, showing the spatial distribution of CRPS skill where the CNN+EasyUQ method (and a hybrid that blends it with NWP) outshines all other competitors across the West African domain. Such findings are encouraging, as they illustrate how well designed ML models can address specific weaknesses of NWP (here, tropical convective rain) without the need for a globe-spanning, billion-parameter network. It also highlights a different philosophy from the all-encompassing GraphCast/Pangu: Walz et al. focus on a regional, high-impact variable with a relatively parsimonious model, rather than attempting a full Earth system simulation. A brief critique is that this study's scope is limited (one region, one-day lead),

so its approach might need adaptation for longer-range or other locations. Nonetheless, the success of CNN+EasyUQ in an area where NWP performance is poor underscores the value of targeted ML solutions alongside global models.

In line with approach taken by Walz et al., a number of researchers have argued for a more *parsimonious* approach to AI weather prediction i.e one that seeks a middle ground between the elegance of physical insight and the power of deep learning. Durran and colleagues[14], in particular, have championed what might be called a "modest game" in contrast to the brute-force game of ever-larger models. Instead of using hundreds of variables and enormous networks, their strategy is to design leaner neural models that capture the essential dynamical core of the atmosphere with far fewer degrees of freedom. For example, Weyn, Durran, and Caruana [31] showed that a simple convolutional neural network trained on global 500 hPa height, temperature, and wind fields could achieve useful 5-day forecast skill on a coarse (~200 km) grid. Although that early data-driven model was less accurate than operational NWP (and indeed than the later GraphCast), it proved that a reasonably small network can learn to propagate large-scale waves and even outperform persistence and climatology beyond a couple of days. Building on this, Durran's team recently introduced an updated parsimonious model on an icosahedral HEALPix grid that predicts a minimal set of seven atmospheric variables with 3-hour time resolution [14]. Remarkably, this compact model can be iterated forward *indefinitely* in time, essentially simulating the global atmosphere in a stable manner without explicit numerical physics. While its horizontal resolution (200 km) and variable set are far simpler than a full NWP, it nonetheless reproduces large-scale weather evolution and even shows skill in extended forecasts (the authors report being able to forecast seasonal anomalies such as El Niño events months in advance by coupling the system to a learned ocean model). The philosophical contrast to GraphCast is clear: rather than throwing maximal data and complexity at the problem, the "modest" approach tries to distill what is truly necessary for predictive skill. These models require dramatically less training data and computational expense (e.g., training can be done on a handful of GPUs rather than TPU pods) and offer greater interpretability. For instance, one can inspect how well the network learned known dynamical modes or "model physics". They also align more closely with the notion of a traditional forecast system that can run freely for climate-length simulations, something that current data-driven models struggle with beyond 10–15 days due to drift or lack of conservation. However, the parsimonious models so far do not reach the raw forecast accuracy of GraphCast or Pangu at medium-range lead times where it is essentially a trade-off between skill and simplicity. Karlbauer et al. [14] acknowledge that their 7-variable model cannot match ECMWF's precision, but they emphasize its strengths in efficiency and stability, and suggest that incorporating more physical knowledge or moderate increases in complexity could narrow the gap. More broadly, Karlbauer et al. [14] argue that purely data-driven forecasts should seek synergy with physical reasoning, rather than purely scaling up network size, to achieve robust performance in a resource-conscious way.

In summary, the advent of large AI models has undeniably revolutionized weather forecasting, yet it also raises new scientific and practical questions. Pangu-Weather and GraphCast have proven that deep learning can serve as a fast and accurate surrogate for numerical models in the medium range, but their advent comes at the cost of enormous training

requirements and a loss of the transparency inherent in physics-based modeling. On the other hand, targeted efforts like Walz's precipitation study and Durran's parsimonious frameworks highlight that there are alternate paths forward i.e ones focusing on specific phenomena or emphasizing model efficiency and interpretability. This thesis is motivated by the need to explore such paths of parsimony. Investigating simpler or hybrid modeling approaches is not just an academic exercise, but a scientifically and societally important endeavor. If we can achieve competitive forecasts with orders-of-magnitude fewer resources, it could democratize advanced forecasting capability and reduce the environmental footprint of prediction. Moreover, a deeper understanding of what minimal model complexity is sufficient for various aspects of weather prediction can yield insights into the atmosphere itself, potentially revealing which processes and inputs are truly predictive. Even if a pared-down model cannot yet top the leaderboard on global skill scores, it can provide valuable lessons in robustness, adaptability, and physical consistency.

2.3. Convolutional Neural Network with Isotonic Distributional Regression

Our study builds directly upon the pioneering work of Walz et al. [30], who addressed the challenge of day-ahead precipitation forecasting in the tropics. Numerical weather prediction (NWP) models have struggled to skillfully forecast convection-driven rainfall in regions like West Africa, often barely outperforming climatology. The West African monsoon features an exceptionally high degree of convective organization, with mesoscale convective systems and tropical waves that are difficult to capture with traditional NWP parameterizations. This context motivated Walz et al. to explore data-driven alternatives. Indeed, a simple logistic regression for 24-hour rain occurrence had already outscored NWP in that region, implying untapped predictive signal in patterns of recent rainfall[30]. Precipitation accumulation is also widely deemed "the most difficult weather variable to forecast" [7] due to its intermittency and non-Gaussian distribution. As discussed, even cutting-edge AI-based forecasts like Pangu-Weather and GraphCast deliberately excluded precipitation, citing its sparsity and complexity. In light of these challenges, Walz et al. [30] proposed a novel solution: a two-stage framework that leverages deep learning for pattern recognition and a statistical technique for uncertainty quantification. This framework which is referred to here as *CNN+EasyUQ* forms the foundation upon which we build our thesis.

Framework

In the CNN+EasyUQ approach, a convolutional neural network (CNN) with a U-Net architecture is first trained to produce a deterministic precipitation forecast, which is then post-processed into a full predictive distribution using the EasyUQ technique. The CNN component is a deep learning model adept at learning spatial patterns from gridded data. Walz et al. [30] employed a U-Net, a CNN architecture with symmetric downsampling

and upsampling paths joined by skip connections, which enables multi-scale feature learning. This choice allows the model to capture the spatial organization of rainfall systems: unlike pointwise statistical models, the CNN inherently learns the relationships between neighboring grid points. In practice, Walz et al. provided the CNN with inputs of recent rainfall maps. Specifically, three consecutive days of prior precipitation accumulation (lags of 1, 2, and 3 days) over the entire domain were supplied as input channels, replacing the hand-crafted lag predictors used in earlier logistic models. This way, the CNN could deduce the propagation of rain systems from one day to the next, essentially learning the same spatio-temporal coherence that logistic regression had to be told explicitly. The output of the CNN is a single-valued 24-hour rainfall prediction at each grid point (for the next day). While relatively standard in architecture and training (Walz et al. note that they stuck to conventional choices like a quadratic loss function and 3×3 conv. kernels), this CNN serves as a high-resolution nowcasting tool for daily rainfall.

The second stage, EasyUQ, is applied to convert the CNN's deterministic output into a probabilistic forecast. EasyUQ (*Easy Uncertainty Quantification*) is a recently developed calibration technique introduced by Walz et al. [29]. It can be seen as a special case of isotonic distributional regression (IDR) [11], a nonparametric approach that uses the pool-adjacent-violators algorithm to fit outcome distributions under a monotonicity constraint. In essence, EasyUQ takes the pairing of model outputs and observed outcomes in the training data and learns a stepwise CDF for predictions. The result is a discrete probabilistic forecast: for any new CNN prediction x, EasyUQ produces a distribution concentrated on observed rainfall values, with masses such that the CDF is nondecreasing with x. Importantly, this method is *distribution-free* and requires no explicit error model or knowledge of the CNN's inner workings where only past forecast–observation pairs are needed. As Walz et al. [29] emphasize, EasyUQ yields statistically calibrated distributions that are optimal in finite samples (under stochastic monotonicity) and it does so with a fully automated workflow (no hyperparameters to tune). In practice, Walz et al. applied EasyUQ separately at each grid point after the CNN was trained. Thus, for every grid cell, the deterministic CNN prediction was turned into a local probability distribution for 24-hour rain amount. This two-stage design cleanly separates the pattern prediction task from the uncertainty quantification task. The CNN (with a U-Net) excels at capturing where and how much rain is expected given recent conditions, while EasyUQ then ensures the forecast probabilities are empirically calibrated to the frequencies of outcomes seen in training. The CNN+EasyUQ framework is compelling because it tackles the non-Gaussian nature of precipitation in a pragmatic way: instead of trying to train a complex deep network to output a full distribution (which would require special loss functions or assumptions), one can train a simpler deterministic model and post-process it. This "plug-and-play" uncertainty quantification is true to its name as it is easy to implement and computationally lightweight. As Walz et al. put it, the approach provides "an elegant and computationally highly efficient" solution for handling precipitation's mixed distribution. Compared to running a large NWP ensemble (which demands significant supercomputing resources), calibrating a single CNN forecast with EasyUQ is extremely efficient. The entire CNN+EasyUQ pipeline can be trained on historical data and then used to generate probabilistic forecasts very quickly, making it attractive for operational use in data-sparse regions.

2.3.1. Performance in Tropical Africa

Applying the CNN+EasyUQ framework, Walz et al. demonstrated remarkable performance in one-day-ahead precipitation forecasting over northern tropical Africa. In their 2011–2019 hindcast experiment, CNN+EasyUQ consistently outperformed an array of state-of-the-art benchmarks. These benchmarks included a monthly probabilistic climatology, the European Centre's 51-member ensemble predictions (both raw and statistically post-processed), and traditional statistical models using up to 25 predictors (e.g., logistic and distributional regression models). The CNN+EasyUQ approach achieved the highest probabilistic skill for both rainfall occurrence and accumulation. For instance, in terms of Brier score for occurrence, the CNN+EasyUQ forecast was better than even a heavily tuned logistic regression using all available predictors, and it handily beat the calibrated ensemble forecast from ECMWF [30]. In terms of rainfall amount, measured by the continuous ranked probability score (CRPS), CNN+EasyUQ again delivered the lowest error among all methods. Notably, a hybrid forecast that simply averaged the CNN+EasyUQ distribution with the NWP (HRES) EasyUQ distribution showed only marginal improvement. This indicates that the machine-learning approach had already extracted most of the predictive signal; the physics-based model added little except in a few cases. Figure 7 of Walz et al. [30] vividly illustrates these results: the CNN+EasyUQ (and the Hybrid, nearly overlapping) yield the best scores in every season, well above climatology and appreciably better than the next-best methods (which were typically the statistical models). Over the peak monsoon months, CNN+EasyUQ achieved CRPS skill scores up to 40% higher than the climatological baseline (an "extended probabilistic climatology"). Even against the sophisticated 51-member ensemble, the gains were striking. In some areas, particularly the Guinea Coast where convective storms are most organized, the CNN-based approach dramatically outperformed the ensemble, which struggled there. This suggests that the CNN learned complex relationships in the data (e.g. coupling to synoptic waves or moisture surges) that the ensemble forecast, with its model errors, could not capture. Such results are unprecedented for data-driven daily rainfall forecasts; they underscore the potential of combining modern ML with robust calibration.

Another important advantage reported by Walz et al. is computational efficiency. Once trained, the CNN+EasyUQ model can generate a probabilistic forecast in a fraction of a second on standard hardware, whereas producing an ensemble forecast (let alone calibrating it) is orders of magnitude more expensive. Walz et al. explicitly note that calibrating a deterministic model via EasyUQ sidesteps the "large computational costs" of running full NWP ensembles.

In short, their framework provides superior probabilistic skill *and* speed. These features make CNN+EasyUQ an attractive blueprint for operational forecasting, especially in regions where computational resources or real-time data assimilation capabilities are limited. Indeed, Walz et al. conclude that CNN+EasyUQ could improve operational tropical rainfall forecasts and "potentially even beyond" the tropics. This suggestion that the methodology may generalize to other climates directly motivates our work. Before leaping to mid-latitudes, however, it is crucial to examine the assumptions and conditions of Walz et al.'s study, to understand how transfer might be achieved.

2.3.2. Limitations and Rationale for the Present Study

While Walz et al.'s CNN+EasyUQ results were groundbreaking, their study also has clear boundaries that inspire the questions and contributions of this thesis. First, the Walz et al. (2024b) experiment is region-specific, focused on convection-dominated tropical Africa. The precipitation regime in that region (driven by diurnal convection, African Easterly Waves, etc.) differs greatly from mid-latitude precipitation in Germany, which is more often associated with frontal systems, extratropical cyclones, and orographic influences. This raises a question of transferability: would the CNN+EasyUQ approach yield similar success in a mid-latitude setting, or are modifications needed to account for the different meteorological dynamics? Walz et al. themselves acknowledge that their method is proven in the tropics and only *hypothesized* to work "beyond". Our thesis directly tackles this question by attempting the framework in Germany's climate, necessitating careful consideration of the differences in precipitation characteristics.

A second important limitation is that Walz et al.'s primary goal was to demonstrate the power of the EasyUQ calibration, rather than to optimize the underlying CNN forecast. In their words, the "usage of EasyUQ in concert with the CNN is novel" and accordingly they "employ standard choices" for the CNN architecture and training. In practice, Walz et al. kept the network architecture and hyperparameters relatively generic (e.g., using a default U-Net configuration, a mean squared error loss, etc.), without exhaustive tuning or customization to maximize deterministic accuracy. This conservative setup was intentional, ensuring a clean demonstration that even a fairly vanilla CNN can be boosted to state-of-the-art performance via IDR calibration. However, it also implies that the CNN's raw forecast skill was likely not fully realized. This opens an opportunity: by investing effort into hyperparameter tuning and model improvements, one might obtain an even stronger deterministic baseline, which after calibration would translate to even better probabilistic forecasts. This might be a far more relevant exercise to conduct for the mid-latitude contexts where the NWP-forecasts do not struggle the same way that they do over the tropics. We explore this opportunity in our thesis.

In summary, two key opportunities emerge from the Walz et al. study:
(1) the untested generalization to a different climate regime, and
(2) the possibility of boosting forecast skill by improving the CNN component. These gaps directly inform the objectives of this thesis. We aim to translate the CNN+EasyUQ framework from tropical Africa to Germany. Achieving this requires bridging meteorological differences and implementing technical adaptations. However, it is quite important to note how much of an advantage we gain starting from the source code of Walz et al. Leaving the CNN architecture and statistical processing largely unchanged, we benefit from targeted adaptation in the training pipeline(for example, loss implementation) and choice of domain and inputs.

In conclusion, Walz et al.'s CNN+EasyUQ framework represents a state-of-the-art baseline for data-driven probabilistic precipitation forecasting. Its innovation lies in marrying a CNN's pattern recognition power with the rigorous uncertainty quantification of IDR, yielding forecasts that are both skillful and well-calibrated. This framework's success in

a convective tropical environment sets the stage for our study. However, the regional specificity and design choices of the original work left open questions that we have begun to address, namely, how to adapt and improve the approach for a different meteorological regime. The next section will delve into the meteorological foundations of precipitation over Germany and related literature, which is critical for informing the adaptations we implemented. Understanding the nature of mid-latitude precipitation (seasonality, driving processes, predictability) will further clarify why certain modifications to CNN+EasyUQ were necessary and how our approach is tailored to the German context. This background will solidify the rationale for our methodology in Chapter 3 and set the scene for evaluating its performance in later chapters.

2.4. Drivers of German Daily Precipitation

This section synthesizes the synoptic–mesoscale processes that control *daily* precipitation over Germany and distills them into concrete implications for our later design choices (predictor families, pressure levels, temporal lags, and training domain). We deliberately postpone some discussions and details until we can appropriately contextualize them within the framework of our methodology. Where relevant, we point forward to those sections to keep the literature review focused on physical understanding rather than methodology.

Guiding idea Mid-latitude precipitation at a daily lead is primarily organized by baroclinic dynamics (cyclones, fronts, and their jet-related forcing), modulated by moisture transport and orography; summertime convection superposes additional intermittency. These process controls map naturally to a compact set of large-scale predictors (winds, humidity, mass fields, and simple surface context) sampled with multi-day lags and over an extended Euro-Atlantic domain.

Mid-latitudes vs. Tropics In the mid-latitudes, strong horizontal temperature gradients and vertical wind shear support baroclinic instability, jet streaks, and frontal circulations embedded in migratory extratropical cyclones. Resulting precipitation fields exhibit large spatial coherence ($O(10^2 - 10^3)$ km) and multi-hour to multi-day durations; orographic lifting along windward slopes (Alps, Black Forest, low mountain ranges) further modulates totals and spatial patterns [13, 26, 22]. In contrast, tropical rainfall is dominated by moist convection (isolated cells, squall lines, MCSs) with weaker baroclinicity, shorter lifetimes, and stronger diurnal modulation [17, 9, 19]. In this mid-latitude context, daily precipitation is orchestrated by synoptic-scale baroclinic dynamics. The jet stream, baroclinic waves, and their frontal circulations set the stage on which mesoscale processes play out. By contrast with tropical, trigger-sensitive convection, small perturbations embedded within a strong, sheared jet can—over 12–36h—amplify into a trough–frontal system and widespread rain, including episodes of rapid deepening ("explosive cyclogenesis"). A forecasting system that sees only yesterday's rainfall has little direct access to this stored dynamical potential energy: convective showers not pre-conditioned by the jet tend to decay, whereas jet-supported disturbances may grow rapidly. Consequently, a parsimonious yet physically informed

predictor set must expose the model to the evolving dynamics and moisture supply that precede precipitation, rather than asking it to infer them from rainfall alone. This motivates the compact, process-aware choices developed below (F1–F3) and the upstream training context adopted for Germany's storm-track setting.

Climatological structure Germany sits near the downstream end of the North Atlantic storm track. Mean annual precipitation displays a pronounced orographic imprint: higher totals along the windward Alpine foreland and Black Forest, moderate values across the central uplands, and drier leeward/NE lowlands. Winter (DJF) is dominated by stratiform frontal precipitation with broad spatial coverage; summer (JJA) totals are patchier due to convection, with local enhancements in orographically and boundary-layer-favored regions [22]. These patterns frame where large-scale predictors (moisture transport, pressure gradients) should be most informative and where convective intermittency may challenge daily skill.

Synoptic archetypes of heavy daily rain Four recurrent drivers produce a large share of heavy daily accumulations in Germany:

(a) *Atlantic cyclones and frontal zones* advected by the westerly jet, often with slow warm fronts and embedded rainbands in DJF/MAM [26].

(b) *Vb-type cyclones* (Genoa/Adriatic origin crossing the Alps toward Central Europe) that tap Mediterranean moisture and stall against orography; though relatively infrequent, they are disproportionately associated with Central European extremes [18].

(c) *Moisture plumes/atmospheric rivers* from the North Atlantic/subtropics that, upon landfall and interaction with quasi-stationary fronts and orography, yield widespread heavy rain, notably in SON [16].

(d) *Cut-off lows and quasi-stationary convective clusters* that sustain training convection and long-lived rainbands, particularly in late spring and summer.

Event attribution and compositing studies show that a large fraction of heavy daily precipitation in Central Europe is cyclone- or front-related; cold fronts are especially prominent for extremes [21, 6]. These findings motivate predictors that encode (i) synoptic structure and ascent, (ii) moisture supply and convergence, and (iii) orographic modulation.

Seasonal processes DJF is stratiform-dominated (fronts, large-scale ascent), JJA features a higher convective fraction with strong late-day modulation and mesoscale organization ahead of shallow fronts/troughs, and MAM/SON are transitional (with SON often featuring AR-assisted slow fronts) [26, 16]. For daily forecasting, this implies stronger incremental value from upper-level dynamical indicators in DJF and from low-level moisture transport and shear in JJA/SON.

Extratropical cyclones rarely follow a single textbook template. The canonical sequence (incipient baroclinic wave with a warm front ahead and cold front trailing, progressive frontal wrapping, and eventual occlusion) admits substantial variability in frontal geometry,

phase speed, interaction with orography, and embedded bands of ascent. An interesting question for a data-driven model is whether it can learn both the canonical evolution and these "wild" departures that strongly modulate next-day rainfall and hence would greatly affect forecast skill.

The geostrophic and thermal-wind framework motivates an emphasis on winds across the 300-850hPa layers and co-located moisture information. Under (quasi-)geostrophic balance in the mid-latitudes, the Coriolis force approximately balances the horizontal pressure-gradient force, so winds tend to follow isobars. The vertical shear of this geostrophic wind is, by the thermal-wind relation, proportional to horizontal temperature gradients; practically, stronger westerlies with height on the cold side of a frontal zone signal pronounced baroclinicity. Thus, the vertical structure of the wind field implicitly encodes aspects of the mass and thermal fields, while sharp shear zones identify regions where small perturbations can extract potential energy and grow. Providing the network with this dynamical scaffold (together with humidity that constrains available moisture) gives it the ingredients needed to represent frontal kinematics, cyclone propagation, and their precipitation footprints.

From Processes to Predictors The literature above supports three predictor families that together capture the governing controls at daily lead:

F1. **Moisture transport and convergence** via multi-level winds (u, v) and specific humidity q (emphasizing 700–850 hPa), complemented by total-column water vapour (TCWV) to represent the column reservoir. This stack lets the network infer IVT-like structures and low-level convergence without sacrificing vertical detail [16, 12].

F2. **Large-scale dynamical forcing** through jet-level and mid-tropospheric winds (300/500 hPa) and sea-level pressure (MSLP) gradients, which together proxy for upper-level divergence/PVA and near-surface geostrophic flow that organize frontal ascent [13, 20, 26].

F3. **Near-surface context** with surface pressure (SP) and 2-m temperature (T2M) to encode mass-field consistency, boundary-layer thermodynamics, and seasonal phase. These features help disambiguate similar synoptic patterns with different precipitation outcomes in DJF vs. JJA.

Pressure-Levels A four-level slice balances ascent signals aloft with moisture transport and convergence near the boundary layer:

- **300 hPa** (jet core): entrance/exit regions and ageostrophic circulations associated with upper-level divergence that pre-conditions frontal ascent—most valuable in DJF [20].
- **500 hPa**: troughs/ridges and mid-level humidity (dry intrusions vs. moist layers) that modulate stratiform vs. mixed-phase regimes.
- **700 hPa**: deformation and frontogenesis layers, pre-frontal ascent, and moist layers in stratiform events.

- **850 hPa**: low-level jets, warm advection, and orographic moisture flux convergence that anchor rainbands, especially in JJA/SON [6].

Seasonally, we expect upper-level winds (300/500 hPa) to contribute more to DJF skill, whereas low-level q and winds (700/850 hPa) and TCWV provide larger relative gains in JJA/SON when moisture transport governs [26, 16].

Temporal Lags and Training Domain Mid-latitude systems typically traverse 500–1000 km per day across the NE Atlantic into Central Europe; fronts require ~ 12–36 h from France and Benelux to Germany, and IVT anomalies often precede heavy daily rain by ~ 6–24 h [26, 16]. A lag design that samples atmospheric predictors on the preceding 2–3 days, with a fixed synoptic time that preserves causal ordering relative to the 00–00 UTC accumulation, therefore captures both preconditioning and approach of dynamically forced precipitation while avoiding same-day leakage.

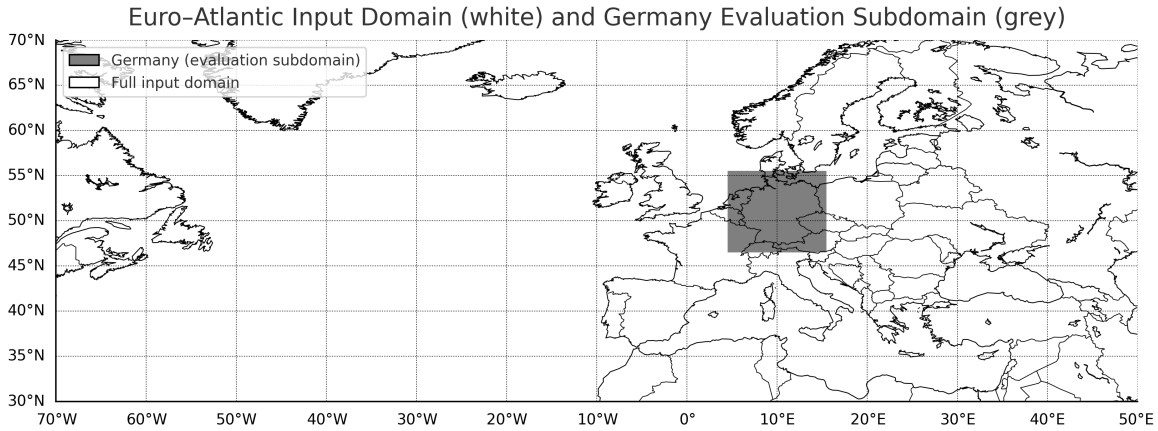

Figure 2.4.: Full extended Euro-Atlantic domain with German subdomain of interest/evaluation highlighted in grey.

In addition, training over an extended Euro-Atlantic domain (upstream of Germany) allows the model to "see" cyclone/IVT precursors before they affect Germany. This, however, poses a new question that Walz et al. did not have to answer; If the training domain is far larger than the region of interest, should we be emphasizing Germany in during training? Can this be achieved by reducing the impact of penalties over the region outside Germany such that we align learning with the target region without discarding upstream structure? Such questions will be better explored in conjunction with the methodological choices that enable our assessment of their validity in the next chapter.

In conclusion, sections 2.1–2.4 draw a coherent arc that provides a solid foundation on which to base our methodology:
(i) modern global AI systems (e.g., Pangu-Weather, GraphCast) prove that deep networks can emulate large-scale dynamics with extraordinary *inference* efficiency but at immense *training* cost and with notable gaps for precipitation;
(ii) the CNN+EasyUQ framework of Walz et al. [30] offers a compelling, *parsimonious* alternative for day-ahead rainfall that cleanly separates spatial pattern prediction (U-Net)

from distributional calibration (IDR/EasyUQ);
(iii) Germany's mid-latitude atmospheric conditions are governed by baroclinic dynamics, moisture transport, and orography, implying different sources of day-ahead predictability than the convection-dominated tropics.

3. Methodology

Overview This chapter details the methodology for developing a parsimonious data-driven daily precipitation forecasting system over Germany. In contrast to physics-based numerical weather prediction, our approach relies on extracting patterns from historical data. The quality and preparation of these data are therefore critical. Recent advances in deep learning have demonstrated that data-driven models can achieve competitive skill in weather prediction by learning from extensive reanalysis datasets [30] [4] [12]. Our work leverages multiple meteorological datasets and modern neural network architectures to forecast precipitation in a purely data-driven manner. We aim to integrate heterogeneous data sources (spanning gridded observations and reanalysis) into a unified modeling pipeline that learns the relationship between atmospheric conditions and subsequent rainfall.

We examine each aspect of our methodology in depth beginning with the raw data we use from various sources, how we apply preprocessing to obtain standardized gridded datasets, and how we use data processing and aggregation steps to convert these into training-ready tensors. Next, we examine the deep neural network (based on a U-Net architecture) and how it ingests these tensors and produces deterministic precipitation predictions along with its training and optimization framework. Finally, we explore the isotonic regression-based post-processing step (EasyUQ/IDR) that is applied to generate probabilistic forecasts, and the comprehensive evaluation framework that assesses the model performance. The full operational pipeline from raw data to performance assessment is shown below:

$$
\begin{aligned}
\text{Raw Data} &\xrightarrow{\text{Preprocessing}} \text{Standardized Tensors} \\
&\xrightarrow{\text{Input Preparation}} \text{Training Batches} \\
&\xrightarrow{\text{Neural Network}} \text{Deterministic Predictions} \\
&\xrightarrow{\text{Training Pipeline}} \text{Optimized Models} \\
&\xrightarrow{\text{IDR Post-Processing}} \text{Probabilistic Forecasts} \\
&\xrightarrow{\text{Evaluation Framework}} \text{Performance Assessment}
\end{aligned}
$$

3.1. Data Preprocessing

The first step of the methodology is to convert raw meteorological and precipitation datasets into a common format suitable for machine learning. This preprocessing stage resolves differences in spatial resolution, temporal frequency, and coordinate reference systems among the datasets, producing standardized data that maintain physical consistency. All datasets are reprojected and resampled to a unified grid and time frame, ensuring that subsequent model inputs are directly comparable across data sources.

Our forecasting framework draws on two datasets: a global atmospheric reanalysis and a complementary precipitation dataset. The reanalysis provides a physically consistent record of atmospheric variables, while the precipitation product serves as observations of the target variable (rainfall). In the following, we describe each dataset and its relevance to the study.

ERA5 Reanalysis Data ERA5 is the fifth-generation global reanalysis produced by the European Centre for Medium-Range Weather Forecasts (ECMWF) [12]. It provides hourly atmospheric fields at approximately 31 km horizontal resolution (0.25° grid). ERA5 is generated by assimilating a wide array of observations into a weather forecast model, yielding a globally complete and physically coherent dataset from 1979 to present. Importantly, precipitation in ERA5 is a model-derived field that is *not* directly constrained by observations during assimilation [12]. In other words, ERA5 precipitation is produced by short-term model forecasts within each assimilation cycle, rather than being adjusted to rain gauge or radar measurements. This approach can lead to systematic biases in ERA5 precipitation, especially for extreme events, since the model may not perfectly capture all local rainfall processes. Nevertheless, ERA5 provides a broad-scale representation of the atmospheric state (pressure, temperature, humidity, winds, etc.) that is critical for data-driven prediction of precipitation. We use ERA5 as the source of predictor variables, as described below.For this study, ERA5 data were retrieved via the Copernicus Climate Data Store API for the period 2007–2020. We focus on daily synoptic conditions at 18:00 UTC (6 hours prior to the target precipitation at 00:00 UTC), in line with previous data-driven setups that allow a lead time for atmospheric precursors [30].

Multi-Source Weighted-Ensemble Precipitation (MSWEP) MSWEP is a globally gridded precipitation dataset that merges rain gauge observations, satellite estimates, and reanalysis model outputs to provide high-quality precipitation estimates at every location. It is specifically designed for hydrological applications [4, 3]. MSWEP version 2 (used here) has a native resolution of 0.1° (11 km) and a 3-hourly temporal frequency, covering 1979 to present. By blending multiple sources, MSWEP capitalizes on the strengths of each: ground gauges for accuracy, satellite sensors for spatial coverage, and reanalysis for consistency in data-sparse regions. In practice, MSWEP adjusts for gauge reporting biases (undercatch, timing) and weights the different inputs based on gauge density and historical performance [4]. This process yields a precipitation product that generally outperforms single-source datasets in

both well-gauged and ungauged regions. Notably, the reanalysis component of MSWEP ensures that even areas with few gauges (such as oceanic or remote regions) receive physically plausible estimates, while gauge corrections anchor the values to observed totals.

For our purposes, MSWEP serves as the primary source of precipitation data for both training targets and lagged predictors. We obtained MSWEP data at its full 3-hourly, 0.1° resolution for the period 2007–2020. Due to the finer resolution relative to our analysis grid, two preprocessing steps were required: (1) temporal aggregation of 3-hourly accumulations into daily totals, and (2) spatial regridding from 0.1° to 1.0° resolution. The temporal aggregation was done by summing 3-hourly MSWEP values over each UTC day (aligning to 00:00–00:00 daily totals). The spatial downsampling was performed via conservative remapping to ensure that the total volumes of precipitation were preserved. Through these steps, the high-resolution MSWEP data were converted into daily precipitation fields on the common 1° grid. We emphasize that using a globally consistent product like MSWEP (as opposed to raw model precipitation) is crucial, because it grounds the training data in observed precipitation patterns. MSWEP's inclusion of ERA-Interim/ERA5 as one of its inputs means it inherits some large-scale structure from reanalysis, but its bias corrections and gauge adjustments make it a more reliable target for model learning [4].

Target Grid Specification A unified target grid was established to ensure spatial consistency across all datasets (see Figure 2.4). We defined a regular latitude-longitude grid covering the North Atlantic–European sector, chosen to include both the region of interest (Germany) and the upwind areas influencing it. The grid specifications are:

- **Projection and grid type**: Equidistant latitude-longitude (geographic coordinates).
- **Spatial extent**: 70°N to 30°N in latitude, and 70°W to 50°E in longitude. This encompasses the North Atlantic, Western and Central Europe, and the Mediterranean—capturing weather systems that travel into Central Europe.
- **Resolution**: 1.0° in both latitude and longitude.
- **Grid dimensions**: 41 latitudinal points × 121 longitudinal points (approximately 4,961 grid cells in total).

All data were regridded to this $1^\circ \times 1^\circ$ grid. While this resolution is coarser than the native data, it balances computational tractability with the need to resolve synoptic-scale features. Moreover, using a common grid avoids mismatches when combining variables and ensures that the neural network can learn coherent spatial relationships.

Temporal Framework The modeling period spans 14 years, from 2007 through 2020, covering a range of hydro-meteorological conditions. The data were split into training and testing periods in chronological order:

- **Training period**: 2007–2019 (13 years, totaling 4,748 daily samples).
- **Final Testing period**: 2020 (1 year, 366 daily samples, including Feb 29, 2020).

Please note that the section 3.2 also describes how yearly evaluation is built into the training window. No future data were used in training (strict chronological split), to mimic real forecasting conditions. All datasets were aligned to a once-daily frequency. Specifically, ERA5 predictor variables are taken at 18:00 UTC each day (as described at the start of section 3.2), and precipitation targets correspond to the 00:00–00:00 UTC accumulation centered at that date. This temporal alignment is maintained throughout preprocessing and ensures a 6-hour lead time between predictors and targets exactly consistent with the lead time used by Walz et al. [30].

Data Standardization Bringing the disparate datasets to the unified 1° grid required careful interpolation methods tailored to the nature of each variable. We distinguished between extensive quantities (like precipitation) that should conserve area-integrated values, and intensive quantities (like temperature or pressure) that vary continuously in space.

Precipitation is an extensive quantity (e.g., total rainfall over an area). When resampling precipitation data to a new grid, it is essential to conserve the volumetric total to preserve physical meaning (particularly for accumulated fields). We applied first-order conservative remapping to all precipitation fields. This method computes the overlapping area between source and target grid cells and distributes the source cell's total value proportionally to overlapping target cells.

Most meteorological variables from ERA5 (temperature, pressure, humidity, wind) are intensive quantities defined at points or as smooth fields. For these, a bilinear interpolation is appropriate, as it yields a weighted average of the four nearest source grid points to estimate the value at the target grid point.

Data Handling and Verification After regridding and alignment, we performed additional transformations and checks to ensure the datasets are not only consistent in format but also suitable in value ranges and completeness for machine learning.

Several quality control measures were applied:

- **Dimension consistency**: We programmatically verified that all variables (precipitation and ERA5 features) share the same spatial grid dimensions (latitude × longitude = 41×121) and the same temporal indices for training and testing periods. Mismatches in array shapes or time steps would raise exceptions in the preprocessing pipeline.
- **Range and physical sanity**: We computed summary statistics (min, max, mean) for each variable over the training period and compared them to known climatological bounds. For instance, T2M values in our region should roughly range from −30°C to +40°C; any value outside this (after unit conversion to Celsius) would indicate an error. Similarly, specific humidity at 850 hPa should not exceed a plausible mixing ratio (on the order of 20 g/kg). All variables fell in expected ranges, aside from some known biases (e.g., ERA5 has a slight wet bias in precipitation, which is accepted since it's not used as direct input here).

- **Missing data checks**: We checked for any missing values (NaNs) in the regridded data. ERA5 reanalysis is complete by design (no gaps), and MSWEP is also globally complete.

Any anomalies detected in this phase were addressed by revisiting the source data or interpolation method. For example, early tests showed extremely small negative precipitation values in a few MSWEP aggregated days (an artifact of how CDO handles summing of very small floating errors); we clipped those to zero.

Output Data Structure The final output of preprocessing consists of self-describing NetCDF files that are organized by data split and variable type. Specifically, we produced:

- A set of daily precipitation files for training (2007–2019) and testing (2020), each containing MSWEP precipitation on the 1° grid.
- A set of daily ERA5 files for training and testing, containing all chosen ERA5 variables on the 1° grid at the 18:00 UTC time stamp corresponding to each date.
- Each NetCDF file includes metadata from the original sources: units, variable long names (e.g., "2 metre temperature"), and attributes documenting the preprocessing steps (via the NetCDF history attribute).
- Coordinates are CF-compliant: we include latitude and longitude arrays for the grid, with appropriate attributes (e.g., standard-name, units in degrees) and the coordinate reference system if needed. Time is recorded in standard format (days since a reference date, with calendar).

By structuring the data in this way, we facilitate easy loading in Python (using xarray or PyTorch data loaders) and ensure reproducibility (any future researcher could inspect the NetCDF metadata to understand how the data were derived). The processed dataset is now "analysis-ready": all variables line up in space and time, have been checked for errors, and are in physically meaningful units (e.g., precipitation in mm/day, temperature in K, etc.) with proper scaling.

With the data preprocessing completed, we next describe how these standardized datasets are transformed into inputs for the neural network model.

3.2. Input Preparation

Building upon the standardized datasets from preprocessing, the data preparation serves as an interface between the meteorological data and the neural network. This process is responsible for loading the NetCDF files, constructing the spatio-temporal tensors needed for training, and applying any additional transformations (like normalization or lagged sampling).

Temporal Lag Structure Precipitation processes often exhibit memory on multiple time scales. For instance, soil moisture from rainfall in previous days can modulate subsequent convective activity, and slow-moving low-pressure systems can cause multi-day rain events. To capture such dependencies, we include a sliding window of previous days' precipitation as part of the input. Specifically, for each target day t, we take the precipitation from three preceding days $(t-3, t-2, t-1)$ as input features. This 3-day lag was chosen based on exploratory analysis and literature suggesting that most of the short-term autocorrelation in midlatitude daily rainfall is contained within the past 2–3 days [27]. By including three days, we capture immediate persistence (yesterday's rain) as well as the tail of any longer event or wet spell.

Mathematically, we denote $P(t)$ as the precipitation field on day t. The lagged precipitation input $\mathbf{X}_{\text{precip}}(t)$ is constructed as:

$$\mathbf{X}_{\text{precip}}(t) = \left[\, P(t-3),\ P(t-2),\ P(t-1) \,\right], \tag{3.1}$$

where the brackets indicate channel-wise stacking. Each $P(t-i)$ is a 41×121 grid, so $\mathbf{X}_{\text{precip}}(t)$ has shape $(3, 41, 121)$ before further concatenation with other variables. This formulation allows the model's first convolutional layer to automatically learn filters that operate across both space and the lag dimension (interpreting it similarly to how one might treat different input image channels). For example, the network could learn a 3D filter that detects patterns like "rainfall increasing day by day in a given area," which might precede a flood, or "rainfall moving eastward with time," which could indicate a propagating frontal system.

For the ERA5 variables, we analogously include 3 lags, but recall that ERA5 is already shifted 6 hours behind the precipitation day. Thus, to be consistent, the ERA5 lags for day t correspond to days $t-4, t-3, t-2$ at 18:00 UTC. In other words, if the precipitation lags cover the past 3 days, the ERA5 lags cover the past 3 days *plus one extra day behind* due to the offset. This yields the ERA5 input tensor:

$$\mathbf{X}_{\text{ERA5}}(t) = \left[\, E(t-4),\ E(t-3),\ E(t-2) \,\right], \tag{3.2}$$

where $E(\tau)$ is the collection of all chosen ERA5 variable fields at time τ. Each $E(\tau)$ is itself multi-channel (since it includes, say, all pressure-level variables at that time), but at this stage we keep the temporal dimension separate. The data module will later flatten the time dimension into the channel dimension after applying any needed transformations to each slice.

This temporal structure—precipitation at $t-3, t-2, t-1$ and atmosphere at $t-4, t-3, t-2$—establishes a cause-effect window where atmospheric conditions lead up to precipitation events. It prevents any information from time t or later from leaking into predictors for time t. The length of the window (3 days of lag) was a balance between capturing enough history and maintaining a reasonable number of input channels.

Seasonality Encoding To provide the model with awareness of the seasonal cycle (crucial since precipitation climatology varies strongly by season) we include two ancillary channels representing the day-of-year. We encode the seasonal cycle as two sinusoidal terms with an annual period:

$$s_{\sin}(t) = \sin\left(\frac{2\pi \cdot \mathrm{DOY}(t)}{365.25}\right), \tag{3.3}$$

$$s_{\cos}(t) = \cos\left(\frac{2\pi \cdot \mathrm{DOY}(t)}{365.25}\right), \tag{3.4}$$

where $\mathrm{DOY}(t)$ is the day-of-year of date t (with a fractional part if needed to account for leap years; using 365.25 in the denominator approximates the annual cycle including leap years). These two channels are simply 2D grids of the same shape as precipitation, but constant in space (each grid cell gets the same value for $s_{\sin}$ and $s_{\cos}$ on a given day). Essentially, they act as time-dependent bias fields that tell the network what time of year it is. By using sine and cosine, we ensure a smooth cyclical representation (Dec 31 and Jan 1 are near each other in this feature space). The inclusion of seasonality features helps the model differentiate, for example, a given atmospheric pattern in summer versus winter. Without this, the model might struggle with ambiguities (e.g., a particular pressure pattern might lead to rain in winter but not in summer due to different moisture availability). By explicitly providing the season, we reduce the burden on the model to infer it from other variables like temperature. This approach of encoding cyclic time features is common in time-series machine learning to avoid discontinuities at year boundaries.

Logarithmic Transformation for Precipitation Precipitation amounts have a highly skewed distribution with a long tail (many days of little or no rain, and a few days of very heavy rain). Training a neural network on raw precipitation values (in mm) can be problematic: large values dominate the mean squared error, and the network might learn less from the numerous small rainy days. To alleviate this, we apply an optional logarithmic transformation to the precipitation data (both the target and the lagged inputs). Specifically, we define:

$$\tilde{P} = \log(P + \epsilon)\,, \tag{3.5}$$

where P is the precipitation in mm/day and ϵ is a small constant (we use $\epsilon = 0.1$ mm) to handle $P = 0$ cases. This transformation compresses the range: for example, $P = 0$ becomes $\log(0.1) \approx -2.30$, a moderate negative number, and $P = 50$ becomes $\log(50.1) \approx 3.91$. Heavy rainfall events (e.g., 100 mm) that would strongly influence a linear scale are brought to a more benign range ($\log(100.1) \approx 4.61$). The effect is that the difference between 0 and 10 mm is amplified relative to the difference between 50 and 60 mm, which can help the model pay attention to lighter precipitation as well.

We implemented this via a custom transformer class (‘TargetLogScaler‘). During training, all target precipitation values are logged before computing loss, and the model’s outputs are interpreted in log-space. For inference or evaluation, we exponentiate the predictions and subtract ϵ to get back to mm. A few safeguards are built in:
- After inverse transforming (exp), any negative values are clamped to 0, ensuring physical non-negativity.

- If the model predicts extremely large log-values (corresponding to >100 mm in linear scale), we issue a warning (such extremes were not present in training data, so they likely indicate an extrapolation).
- The value of $\epsilon = 0.1$ was chosen as a small fraction of a millimeter to be negligible for moderate/heavy rain, but large enough to avoid $\log(0)$ issues.

In summary, the log transform makes the distribution closer to Gaussian and the learning problem more balanced. It is a common technique in precipitation modeling (e.g., [28] used a similar approach for post-processing ensemble rainfall forecasts). All results reported later are converted back to original units (mm) for interpretation.

ERA5 Variable Standardization Each ERA5 variable has its own units and scale (for instance, T2M in Kelvin 290, TCWV in kg/m^2 20, winds in m/s 10, etc.). To prevent any single variable from unduly dominating the gradient updates due to scale, we standardize each variable to zero-mean and unit-variance based on the training data distribution:

$$\tilde{E}_{ij} = \frac{E_{ij} - \mu_E}{\sigma_E}, \tag{3.6}$$

where E_{ij} is a particular ERA5 field (for example, 500 hPa humidity at a certain pixel i, j on a certain day, though in practice we compute μ_E over all i, j, t in training for that field), and μ_E, σ_E are the mean and standard deviation of that field over the entire training dataset. This is done separately for each variable type. For multi-level variables, we could standardize each level separately, but we chose to standardize across all levels together for simplicity (since values at different pressure levels can differ—e.g., humidity at 850 hPa is generally higher than at 300 hPa in absolute terms). In practice, because we feed each level as a separate channel, treating them together or separately doesn't change the per-channel normalization result, as we still compute a mean and std over all those values.

After standardization, most ERA5 inputs lie in a range roughly $[-3, +3]$ (since atmospheric variables roughly follow Gaussian-like distributions after removing seasonal cycles). This puts them on a comparable footing with the log-precipitation inputs, which also end up roughly within a few standard deviations (e.g., log precipitation might roughly range -2 to $+4$ for 0 to 50 mm). The seasonal sine/cosine features are already in $[-1, 1]$ by construction, which is fine. We did not standardize those as they are already zero-mean over a year.

We compute these normalization statistics (μ_E, σ_E for each variable) from the training set at initialization. We then apply the normalization to training data on the fly, and store the values so that the exact same normalization can be applied to validation and test data. This ensures no information from validation/test leaks into the normalization (which could happen if, say, we mistakenly normalized using the global mean including test).

Data splitting Given the relatively limited number of years available (14 years) and the possibility of year-to-year climate variability, we adapt a similar temporal cross-validation strategy as Walz et al. [30] with the a few minor changes:

We implemented a year-based expanding-window, which is a form of time series cross-validation that respects chronological order by splitting the training data into overlapping

chunks called **folds**. The idea is to simulate a forecasting scenario for multiple consecutive validation years, each time training on all data prior to that year. Formally, if we denote by Y_k the year corresponding to the k-th fold's validation set, then:

$$\text{Training set}_k = \{2007, 2008, \ldots, Y_k - 1\}, \tag{3.7}$$

$$\text{Validation set}_k = \{Y_k\}, \tag{3.8}$$

with $Y_0 = 2010$, $Y_1 = 2011$, ..., up to $Y_9 = 2019$ (assuming we hold out one year at a time from 2010 onward as validation). The final year, 2020, is reserved as an out-of-sample test set for the end. This results in 10 folds (if we start validation at 2010) or more folds if we start earlier. In Walz et al. [30]'s setup, the initial model is trained on roughly a decade of data (2001 up to November 2010), and then they incrementally extend the training period to include that year, retrain, and forecast the following year, and so on. The philosophy is the same as ours: use an expanding window to utilize all past data and respect causal ordering, thereby emulating how a real-world forecasting system would be updated over time. Walz et al. [30] explicitly note that this annually growing window reflects an operational setting where one continually retrains as new data become available. Despite the conceptual similarity, there are a few notable differences in implementation in our approach:

1. Initial training span: We began with only 3 years of training data (2007–2009) for the first fold, whereas Walz et al. started with approximately 10 years (2001–2010) before their first evaluation. The larger initial window in their case reflects the greater data availability in their study and ensures a well-trained model from the outset. In our case, data constraints led to a smaller initial training period, which then grew fold by fold. This difference might influence early-fold performance (see 4.5) but by later folds the gap narrows as our training set catches up in size. Indeed, both approaches ultimately leverage more than a decade of data by the final evaluation.

2. Model re-initialization vs. warm-start: In our fold progression, we "carry over" the model state from one fold to the next, effectively fine-tuning an existing network with new data. However, in Walz et al. [30]'s design, at each fold, the model is retrained from scratch on the expanded dataset. Retraining afresh for each window ensures that the model fully re-optimizes to the data available in that fold, without any potential bias from earlier training cycles. Our warm-start approach, on the other hand, is a pragmatic choice to expedite training convergence, especially important given the computational cost of repeatedly training deep networks for each fold. Warm-starting assumes that the optimal weights for the $(k + 1)$-th training period are close to those from the k-th period, which is reasonable in a slowly expanding dataset scenario. The risk is that if the initial fold's model converged to a suboptimal solution, continuing from those weights might carry that history forward; however, we mitigated this risk by explicitly comparing both approaches and finding no statistical differences in final performance metrics. Our code allows us to switch freely between the two approaches.

This expanding approach has several benefits:
- Temporal integrity: Each validation year comes chronologically after all training years,

mimicking a true forecasting scenario. We never train on future data relative to the validation target.
- Increasing training size: As k increases, the training period grows (from 3 years up to 12 years in our case), allowing us to evaluate how more data improves performance and whether the model suffers from any overfitting or underfitting that changes with sample size.
- Climate non-stationarity assessment: By comparing performance across validation years, we can see if the model does consistently well or if certain periods (e.g., an extremely dry year or an extremely wet year) pose challenges, which might indicate climate regime differences.

From a meteorological perspective, this strategy captures the seasonal cycle every year in validation (so metrics aren't biased by evaluating on only part of a year) and ensures that patterns like the North Atlantic Oscillation phases or El Niño years that occur in validation were not seen in training (if they happened after). It thereby tests generalization to unseen climatic conditions to some extent

Finally, with all the lag structures built and standardization and normalization applied, we prepare a tensor of shape $(C, 41, 121)$ for each sample, where $C = 5 + 3 \times N_{\text{ERA5}}$ (with $N_{\text{ERA5}} = 0$ in the case of precipitation-only). The code adaptively handles all the channel concatenation logic so that the rest of the pipeline can remain agnostic to how many features are used which makes it easy to experiment with adding or removing predictors without altering the core model code.

3.3. Neural Network Architecture Design

With the input tensors prepared, we design a neural network architecture to map these multi-channel spatio-temporal inputs to an output precipitation field. Our architecture is based on the U-Net convolutional neural network, chosen for its proven ability to handle image-like data and capture both local and global features [25]. We adapt the vanilla U-Net to our meteorological forecasting context by incorporating domain-specific modifications, such as handling a non-square grid, focusing the loss on a sub-region, and providing for a variable number of input channels.

3.3.1. U-Net Architecture Overview

Encoder-Decoder Framework The U-Net is an encoder–decoder CNN originally developed for biomedical image segmentation [25]. It has since been widely applied to other image-to-image tasks, including precipitation nowcasting [1]. Its key characteristic is the symmetric "U" shape: an encoder (downsampling path) that progressively reduces spatial resolution to capture context, and a decoder (upsampling path) that reconstructs the output at full

resolution, with skip connections from encoder layers to corresponding decoder layers to preserve fine details.

For precipitation forecasting, the encoder allows the model to recognize large-scale weather patterns (e.g., a broad low-pressure area or a moisture plume) by compressing the image, while the decoder ensures that the output is delivered on the original spatial grid (41×121) to match observational data. The skip connections help retain information about small-scale features (like localized convective regions) that might otherwise get lost in the compression. This exploits the particular weakness of NWP identified in chapter 2 where they struggle to capture multi-scale connections (for example mesoscale convection and planetary waves).

Considering the scope of a Master Thesis, we avoided exploring various different neural network architectures and directly adapted the U-Net with four levels of down/upsampling from the experiment design of Walz et al. [30] with only minor changes to the input channel configuration and tensor shape adaptation to account for change in domain size. Specifically:
- Input layer: C input channels, image size 41×121. - Level 1 (encoder): Two 3×3 convolutions (with padding to preserve dimensions), each followed by InstanceNorm and LeakyReLU activation, then a 2×2 max pooling, reducing spatial size to $\sim 20 \times 60$.
- Level 2 (encoder): Two convolutions (with filters doubled), then pool to 10×30.
- Level 3 (encoder): Two convolutions (filters doubled again), then pool to 5×15.
- Level 4 (encoder): Two convolutions (filters doubled again), then pool to 3×8 (after padding, since 5×15 pooling yields 2.5×7.5, we treat it as 3×8).
- Bottleneck: Two convolutions at the smallest scale (filters doubled again).
- Decoder: At each level, an up-convolution (transpose conv) doubles the spatial dimensions, then concatenation with the corresponding encoder feature map (skip connection), then two convolutions (with filters halved relative to the previous decoder layer).
- Final output: A 1×1 convolution to reduce to a single output channel (the predicted precipitation). No activation is applied in the final layer because we work in (transformed) regression space; the network can output any real number, which corresponds to log-precipitation if we use the log transform, or mm if not. We later convert log predictions to mm.

The skip connections are crucial: they inject high-resolution features from the encoder into the decoder. For instance, if during encoding the model detects a small-scale feature (like a sharp gradient indicating a front or coastline) in Level 1, that information can be passed directly to the decoder's Level 1 (last upsampling) to help reconstruct the precipitation pattern sharply at those boundaries. Without skip connections, decoders sometimes produce overly smooth outputs because they rely purely on the coarse feature maps.

Another advantage of U-Net is parameter efficiency. By reusing the encoder features via skip connections, the network can achieve good performance with fewer parameters than an equivalent plain encoder-decoder. This was appealing given our channel count variability and moderate dataset size.

For a more detailed look at the Encoder-Decoder logic, please see section A.1

Spatial Loss Weighting Mask (discrete but tapered) During training, we want the model to focus learning on Germany, but not to the extent of totally neglecting surrounding areas (because those areas' weather may influence Germany's weather, and because the CNN will propagate information across the domain). To achieve this, we designed a spatial weighting mask for the loss function, $\mathbf{W}_{\text{spatial}}(i, j)$, that assigns different weights to different regions. For example:

$$\mathbf{W}_{\text{spatial}}(i,j) = \begin{cases} 1.0 & \text{if } (i,j) \text{ is inside Germany (target region),} \\ \frac{1+w}{2} & \text{if } (i,j) \text{ is within 3 grid cells of Germany,} \\ \text{w} & \text{if } (i,j) \text{ is more distant (chosen value for outside weight).} \end{cases}$$

The idea is to impose a soft focus: errors over Germany count the most (full weight 1.0), errors in the immediate vicinity of Germany (neighboring grid cells up to 3° away, which roughly covers countries just around Germany) count somewhat (arithmetic mean of full weight and outside weight w), and errors far away (like Spain or mid-Atlantic) count only a fraction (w) as much where the chosen value of w should like between 0 and 1. This encourages the model to still learn general weather patterns (since ignoring the outside completely could harm its ability to, say, simulate a cyclone approaching Germany), but it will preferentially adjust to fix errors in Germany.

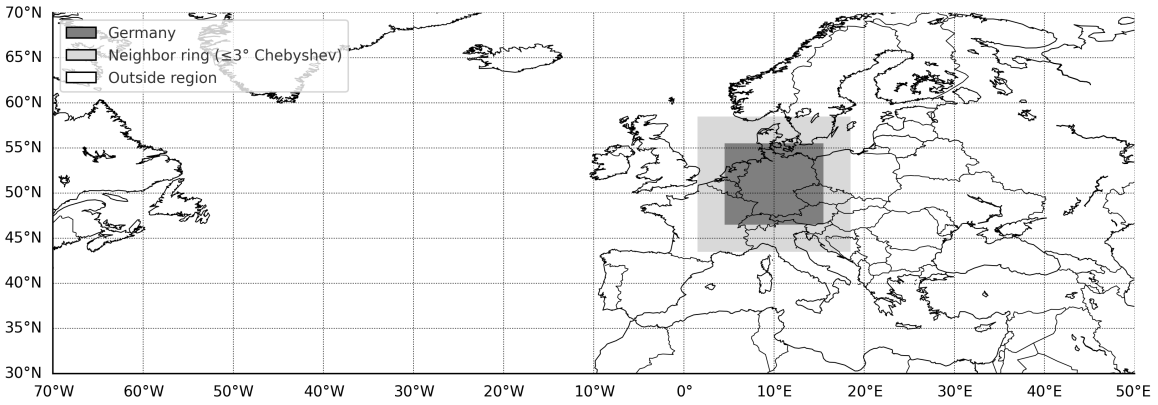

Figure 3.1.: Full extended Euro-Atlantic domain with German subdomain and neighbor cells for a discretely tapered loss

Suppose we chose an outside weight $w = 0.2$ leading to a neighbour weight of 0.6. 0.2 is small but not zero, so the model still sees a penalty for large errors far away. 0.6 gives intermediate importance to near-Germany; one could argue for a gradual decay rather than a step, but for simplicity we did a two-tier outside region. The distance of 3 grid cells (3°) was chosen because weather systems typically have influence radius of a few hundred kilometers; anything beyond 300 km we considered "far" in this context. We used Chebyshev distance (max norm) for convenience on the grid (so a square of side 2*3+1 around Germany is "neighbor"). In practice, we calculate the correct weight for the neighbor cells by keeping the mathematical relation between the outside weight and neighbor weight consistent for any chosen value of outside weight (between 0 and 1). Hence outside weight is the only parameter out of these three that one needs to choose at the start of the run.

During training, when computing the loss over the whole 41×121 grid, we multiply the squared error (or other loss per grid cell) by $\mathbf{W}_{\text{spatial}}(i, j)$ before averaging. Effectively, the loss becomes:

$$\mathcal{L}_{\text{weighted}} = \frac{\sum_{i,j} \mathbf{W}_{\text{spatial}}(i, j)\, \ell(\hat{y}_{i,j}, y_{i,j})}{\sum_{i,j} \mathbf{W}_{\text{spatial}}(i, j)},$$

where ℓ is the element-wise loss (e.g., squared error) and the denominator is just a normalizing constant to keep the magnitude of loss roughly consistent (we normalize by sum of weights so that the learning rate etc. need not change).

This weighting was implemented efficiently by precomputing the weight mask and doing an element-wise multiplication with the loss matrix. In PyTorch, we took advantage of broadcasting to apply it to the whole batch.

The spatial weighting is only used for training loss. When validating or testing, we compute metrics either over Germany (with evaluation mask) or unweighted over whatever region of interest, but not using these fractional weights. The rationale is that we want the model to learn in a guided way, but at evaluation we can examine how it did specifically in Germany which is our main goal.

For a detailed look at regularization, spatial resolution handling and data augmentation see section A.2, section A.3 and section A.4.

3.4. Training Framework and Optimization

The training framework connects the neural network architecture to an efficient optimization and monitoring pipeline. Built on **PyTorch Lightning**, it encapsulates the model, loss, and optimizer in a modular class for cleaner training logic. This Lightning module wraps the U-Net model ($M_{\text{U-Net}}$), the composite loss function (L_{loss}), the chosen optimizer (O_{opt}), and learning-rate scheduler (S_{sched}) into a single trainable unit:

3.4.1. PyTorch Lightning and Loss Architecture

Modular training architecture. The `UNetLightningModule` encapsulates the U-Net model, objective, and optimization:

$$\mathcal{F}_{\text{Lightning}} = \{\mathcal{M}_{\text{U-Net}},\ \mathcal{L}_{\text{loss}},\ \mathcal{O}_{\text{optimizer}},\ \mathcal{S}_{\text{scheduler}}\}, \tag{3.9}$$

where each component exposes a clean interface while remaining replaceable (e.g., switching from Adam to AdamW).

Forward Pass Implementation The framework handles channel dimension management so that the output maintains spatial consistency with the input grid while producing single-channel precipitation predictions.

$$\hat{\mathbf{Y}} = \mathcal{M}_{\text{U-Net}}(\mathbf{X}) \tag{3.10}$$

$$\text{where } \mathbf{X} \in \mathbb{R}^{B\times C\times 41\times 121} \tag{3.11}$$

$$\hat{\mathbf{Y}} \in \mathbb{R}^{B\times 41\times 121} \tag{3.12}$$

Multi-Objective Loss Design The training framework implements various loss functions that are appropriate for precipitation forecasting allowing us to test various configurations and compare them directly in SECTION???? to determine the best choice(s) for our particular case. Each loss function addresses different aspects of the precipitation prediction problem:

Mean Squared Error (MSE) Loss The standard MSE loss serves as the baseline optimization objective, operating in log-transformed space when target scaling is enabled:

$$\mathcal{L}_{\text{MSE}}(\hat{\mathbf{Y}}, \mathbf{Y}_{\log}) = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_{\log,i})^2 \tag{3.13}$$

where $\mathbf{Y}_{\log} = \log(\mathbf{Y}_{\text{orig}} + \epsilon)$ represents the log-transformed target precipitation and N denotes the number of valid (non-NaN) elements.

Weighted MSE Loss To address the heavy-tailed distribution of precipitation, the weighted MSE loss applies intensity-dependent penalties:

$$\mathcal{L}_{\text{Weighted}}(\hat{\mathbf{Y}}, \mathbf{Y}) = \frac{1}{N}\sum_{i=1}^{N} w(y_i)\cdot(\hat{y}_i - y_i)^2 \tag{3.14}$$

where the weight function follows:

$$w(y) = \begin{cases} 1.0 & \text{if } y = 0 \text{ (no precipitation)} \\ 2.0 & \text{if } 0 < y \leq 0.1 \text{ (light precipitation)} \\ 5.0 & \text{if } 0.1 < y \leq 1.0 \text{ (moderate precipitation)} \\ 10.0 & \text{if } 1.0 < y \leq 5.0 \text{ (heavy precipitation)} \\ 20.0 & \text{if } y > 5.0 \text{ (extreme precipitation)} \end{cases} \tag{3.15}$$

Asymmetric MSE Loss Recognizing that underestimation of precipitation events is typically more problematic than overestimation from a meteorological perspective, the asymmetric loss penalizes these errors differently:

$$\mathcal{L}_{\text{Asymmetric}}(\hat{\mathbf{Y}}, \mathbf{Y}) = \frac{1}{N} \sum_{i=1}^{N} \begin{cases} \beta \cdot (\hat{y}_i - y_i)^2 & \text{if } \hat{y}_i < y_i \text{ (underestimation)} \\ (\hat{y}_i - y_i)^2 & \text{if } \hat{y}_i \geq y_i \text{ (overestimation)} \end{cases} \tag{3.16}$$

where $\beta = 2.0$ provides increased penalty for missed precipitation events.

Focal MSE Loss The focal loss variant emphasizes difficult examples by weighting the loss according to the prediction error magnitude:

$$\mathcal{L}_{\text{Focal}}(\hat{\mathbf{Y}}, \mathbf{Y}) = \frac{1}{N} \sum_{i=1}^{N} ((\hat{y}_i - y_i)^2 + \epsilon)^{\gamma/2} \cdot (\hat{y}_i - y_i)^2 \tag{3.17}$$

where $\gamma = 2.0$ controls the focusing strength and $\epsilon = 10^{-8}$ ensures numerical stability.

Huber Loss For robust optimization in the presence of outliers, the Huber loss combines MSE and MAE characteristics:

$$\mathcal{L}_{\text{Huber}}(\hat{\mathbf{Y}}, \mathbf{Y}) = \frac{1}{N} \sum_{i=1}^{N} \begin{cases} \frac{1}{2}(\hat{y}_i - y_i)^2 & \text{if } |\hat{y}_i - y_i| \leq \delta \\ \delta(|\hat{y}_i - y_i| - \frac{1}{2}\delta) & \text{otherwise} \end{cases} \tag{3.18}$$

with $\delta = 1.0$ defining the threshold between quadratic and linear regimes.

3.4.2. Model interface with Training Framework

Atmospheric Variable Selection Strategy The framework enables systematic evaluation of different atmospheric predictor combinations:

Minimal Configuration ($C = 5$): Precipitation-only baseline **Standard Configuration** ($C = 13$): Wind only (biggest lever)

- Horizontal winds: u, v at 300, 500, 700, 850 hPa

Comprehensive Configuration ($C = 53$): Full atmospheric state

- Multi-level winds: u, v at 300, 500, 700, 850 hPa
- Multi-level moisture: q at 300, 500, 700, 850 hPa
- Surface variables: MSLP, T2M, TCWV, SP

Meteorological Variable Justification The atmospheric variable selection reflects established understanding of precipitation processes:

Moisture Transport: Specific humidity q at multiple levels captures the vertical moisture profile essential for precipitation formation, while total column water vapor (TCWV) provides integrated moisture availability.

Dynamic Forcing: Horizontal winds u, v at multiple pressure levels represent atmospheric dynamics that drive moisture transport and convergence patterns leading to precipitation.

Synoptic Environment: Mean sea level pressure (MSLP) and surface pressure characterize large-scale pressure patterns and their evolution, which control precipitation-generating weather systems.

3.5. Probabilistic Post-Processing and Evaluation

The evaluation and post-processing framework receives the deterministic models from the training pipeline and transforms their outputs into calibrated probabilistic forecasts and also provides performance assessment across multiple scales and metrics. This component integrates not only deterministic evaluation metrics (SEEPS) but also isotonic distributional regression (IDR) for uncertainty quantification [11, 29], verifies them using proper scores (CRPS, BS) [30] and provides diagnostic tools for model validation and improvement. This section formalizes theory and documents the *operational* evaluation protocol we implemented, such as, cell-wise IDR fitting, daily-first CRPS with *cos(lat)* area-weighting, a monthly probabilistic climatology (MPC) baseline and skill scores (CRPSS)

3.5.1. Deterministic Performance Assessment

Standard Accuracy Metrics The evaluation framework computes comprehensive deterministic performance metrics using the inverse-transformed predictions in physical units (mm/day):

Root Mean Square Error (RMSE):

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2} \tag{3.19}$$

Mean Absolute Error (MAE):

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i| \tag{3.20}$$

Bias:

$$\text{Bias} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i) \tag{3.21}$$

All metrics are computed both for the ful domain as well as the German subdomain to ensure regional relevance.

Intensity-Stratified Evaluation Recognition that precipitation prediction challenges vary significantly across intensity ranges motivates intensity-stratified metric computation:

$$\text{Metric}_{\text{intensity}}(I) = \text{Metric}(\{(\hat{y}_i, y_i) : y_i \in I\}) \tag{3.22}$$

where intensity categories follow meteorological conventions:

- **Light precipitation**: $0.1 \leq y < 1.0$ mm/day
- **Moderate precipitation**: $1.0 \leq y < 5.0$ mm/day
- **Heavy precipitation**: $5.0 \leq y < 20.0$ mm/day
- **Extreme precipitation**: $y \geq 20.0$ mm/day

This stratification reveals differential model performance across the precipitation intensity spectrum, enabling targeted model improvements.

Conditional Performance Analysis The framework implements conditional performance analysis that examines forecast skill as a function of:

Seasonal Variations:

$$\text{Metric}_{\text{seasonal}}(s) = \text{Metric}(\{(\hat{y}_i, y_i) : t_i \in \text{Season } s\}) \tag{3.23}$$

where $s \in \{\text{DJF}, \text{MAM}, \text{JJA}, \text{SON}\}$ represent meteorological seasons.

Forecast Lead Time Dependencies: Though the current implementation focuses on 24-hour forecasts, the framework supports lead-time dependent evaluation for extended forecast horizons.

Stable Equitable Error in Probability Space (SEEPS) The SEEPS evaluation framework provides categorical assessment specifically designed for precipitation forecasting applications. Unlike traditional contingency table approaches, SEEPS accounts for the climatological difficulty of precipitation prediction at each location.

Local Climatology Construction: For each grid cell, local precipitation climatology is established using training data:

$$p_1 = P(\text{Precipitation} = 0) \tag{3.24}$$
$$p_2 = P(\text{Light Precipitation}) \tag{3.25}$$
$$p_3 = P(\text{Heavy Precipitation}) \tag{3.26}$$

where category boundaries are determined by local percentiles:

$$\text{Light-Heavy Threshold} = 90\text{th percentile of non-zero precipitation} \tag{3.27}$$

SEEPS Error Matrix: The SEEPS score employs a 3×3 error matrix S where element S_{ij} represents the penalty for forecasting category j when category i is observed:

$$\mathbf{S} = \begin{pmatrix} 0 & \frac{p_1}{p_1+p_2} & 1 \\ \frac{p_2}{p_1+p_2} & 0 & \frac{p_2}{p_2+p_3} \\ 1 & \frac{p_3}{p_2+p_3} & 0 \end{pmatrix} \tag{3.28}$$

This matrix design ensures:

- Perfect forecasts receive zero penalty
- Symmetric penalties for adjacent categories
- Maximum penalty (1.0) for the most severe errors (dry/heavy mismatches)
- Climatologically adjusted penalties reflecting local precipitation characteristics

SEEPS Implementation and Aggregation The SEEPS computation follows a systematic protocol:

Algorithm 1: SEEPS Score Calculation

for each grid cell $(i, j) \in \Omega_{\text{Germany}}$ **do**
  Compute local climatology from training data
  Construct cell-specific error matrix $\mathbf{S}_{i,j}$
  for each validation sample k **do**
    Categorize observation: $o_k \in \{1, 2, 3\}$
    Categorize forecast: $f_k \in \{1, 2, 3\}$
    Compute SEEPS score: $\text{SEEPS}_k = \mathbf{S}_{i,j}[o_k, f_k]$
  end for
end for
Aggregate: $\text{SEEPS}_{\text{mean}} = \frac{1}{N} \sum_{k=1}^{N} \text{SEEPS}_k$

3.5.2. Isotonic Distributional Regression Theory and Implementation

Mathematical Foundation Isotonic Distributional Regression (IDR) provides a non-parametric approach to uncertainty quantification that transforms deterministic point forecasts into calibrated probability distributions. The method addresses the fundamental limitation that neural networks, when trained with standard loss functions, produce uncalibrated uncertainty estimates that may not reflect true forecast reliability.

The IDR framework solves the optimization problem:

$$\hat{F}_{Y|X}(y|x) = \arg\min_{F} \mathbb{E}[\text{CRPS}(F, Y) + \lambda \cdot \text{Isotonic Penalty}(F)] \tag{3.29}$$

where F represents the predictive cumulative distribution function, Y denotes observed precipitation, and the isotonic penalty enforces monotonicity constraints that preserve the ordering relationships inherent in deterministic predictions.

Isotonic Constraint Formulation For a set of deterministic predictions $\{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n\}$ with corresponding training observations $\{y_1, y_2, \ldots, y_n\}$, the isotonic constraint ensures:

$$\text{If } \hat{y}_i \leq \hat{y}_j, \text{ then } \hat{F}(t|\hat{y}_i) \geq \hat{F}(t|\hat{y}_j) \text{ for all } t \tag{3.30}$$

This constraint reflects the meteorological principle that higher precipitation predictions should generally correspond to higher probabilities of exceeding any given threshold.

Optimization Algorithm The IDR optimization employs a combination of the Pool-Adjacent-Violators Algorithm (PAVA) and quadratic programming to solve the constrained optimization problem efficiently:

Step 1 - Empirical CDF Computation:

$$\hat{F}_{\text{empirical}}(t|x_i) = \frac{1}{n_i} \sum_{j:x_j = x_i} \mathbf{1}_{y_j \leq t} \tag{3.31}$$

Step 2 - Isotonic Regression:

$$\hat{F}_{\text{isotonic}}(t|\cdot) = \text{PAVA}(\hat{F}_{\text{empirical}}(t|\cdot), \text{order constraints}) \tag{3.32}$$

Step 3 - Continuous CDF Construction:

$$\hat{F}(t|x) = \text{Interpolation}(\hat{F}_{\text{isotonic}}(t|\cdot), x) \tag{3.33}$$

Spatial Heterogeneity Modeling and Meteorological justification Rather than applying a single global calibration, the methodology performs independent IDR fitting for each grid cell:

$$\hat{F}_{i,j}(t) = \text{IDR}(\{\hat{y}_{i,j}^{(k)}\}_{k=1}^{N_{\text{train}}}, \{y_{i,j}^{(k)}\}_{k=1}^{N_{\text{train}}}) \tag{3.34}$$

where (i, j) denotes the spatial grid indices and k indexes the temporal training samples.

This cell-wise approach reflects several key meteorological considerations:

Local Climate Regimes: Different regions within the domain exhibit distinct precipitation climatologies, with coastal areas experiencing different seasonal patterns than inland regions, and topographic effects creating localized precipitation enhancement or suppression.

Model Bias Heterogeneity: Neural network predictions may exhibit spatially varying biases due to differential representation of local-scale processes, land-surface interactions, or orographic effects that require location-specific calibration.

Forecast Skill Variations: The relationship between deterministic predictions and observed precipitation may vary spatially depending on local meteorological complexity, data density, or the effectiveness of the model's learned representations.

Implementation Framework The cell-wise calibration follows a systematic protocol:

Algorithm 2: Cell-Wise IDR Calibration

1: **for** each grid cell $(i, j) \in \Omega_{\text{domain}}$ **do**
2: Extract training data: $\{\hat{y}_{i,j}^{(k)}, y_{i,j}^{(k)}\}_{k=1}^{N_{\text{train}}}$
3: Fit IDR model: $\mathcal{M}_{i,j} = \text{IDR}(\text{training data})$
4: Extract validation predictions: $\{\hat{y}_{i,j}^{(k)}\}_{k=1}^{N_{\text{val}}}$
5: Generate probabilistic forecasts: $\{\hat{F}_{i,j}^{(k)}\}_{k=1}^{N_{\text{val}}} = \mathcal{M}_{i,j}(\text{validation predictions})$
6: Compute evaluation metrics for cell (i, j)
7: **end for**

3.5.3. Probabilistic Evaluation Metrics

Daily-first CRPS computation and area weighting We aggregate first *daily*, second *spatially* and lastly *temporally* which is fully aligned with [30]. A key difference however is that unlike over the tropics, our mid-latitude domain requires us to account for area-weighting due to different latitudes covering unequal areas[12].

Step 1: per-cell, per-day CRPS from IDR For each valid day t and cell $(i, j) \in \Omega_{\mathrm{DE}}$,

$$\mathrm{CRPS}_{t,i,j} = \int_{-\infty}^{\infty} \left(F_{t,i,j}(z) - \mathbf{1}\{y_{t,i,j} \le z\}\right)^2 \mathrm{d}z,$$

computed exactly from the IDR CDF $F_{t,i,j}$ and the realized $y_{t,i,j}$.

Step 2: $\cos(\varphi)$ area-weighted Germany mean (per day) Let φ_i be the latitude at row i. Define weights

$$w_{i,j} = \frac{\cos\varphi_i \, \mathbf{1}\{(i,j) \in \Omega_{\mathrm{DE}}\}}{\sum_{(m,n)\in\Omega_{\mathrm{DE}}} \cos\varphi_m}, \tag{3.35}$$

and the daily Germany-mean CRPS

$$\overline{\mathrm{CRPS}}_t^{\mathrm{DE}} = \sum_{(i,j)\in\Omega_{\mathrm{DE}}} w_{i,j}\,\mathrm{CRPS}_{t,i,j}\,. \tag{3.36}$$

The same weighting is used for BS, MAE, RMSE, etc. Area-weighting is best practice for lon–lat grids [12].

Step 3: temporal averaging (overall and seasonal)

$$\mathrm{CRPS}^{\mathrm{overall}} = \frac{1}{N}\sum_{t=1}^{N} \overline{\mathrm{CRPS}}_t^{\mathrm{DE}}, \tag{3.37}$$

$$\mathrm{CRPS}^{(s)} = \frac{1}{N_s}\sum_{t\in s} \overline{\mathrm{CRPS}}_t^{\mathrm{DE}}, \quad s \in \{\mathrm{DJF}, \mathrm{MAM}, \mathrm{JJA}, \mathrm{SON}\}\,. \tag{3.38}$$

This ensures that the overall mean equals the *time-weighted* seasonal means (verified numerically with tight tolerance).

Note A prior implementation pooled cell–day pairs without area weighting, leading to inconsistencies and *artificially low* CRPS. The updated protocol above resolves these issues and is the basis for all results reported in Chapters 4.

Monthly probabilistic climatology (MPC) and CRPSS For each cell (i, j) and calendar month m, the MPC is the empirical distribution of training-year observations in month m; MPC PoP is $1 - F^{\mathrm{MPC}}_{m,i,j}(0.2)$. This mirrors Walz et al. [30], adapted to our splits (2007–2019 train/val; 2020 test).

Skill score For any proper score $S \in \{\mathrm{CRPS}, \mathrm{BS}\}$,

$$\mathrm{CRPSS} = 1 - \frac{S_{\mathrm{model}}}{S_{\mathrm{MPC}}}, \tag{3.39}$$

computed with the same spatial weights and temporal aggregation as above. Positive values indicate improvement over climatology. We report CRPSS overall and by season.

3.5.4. Summary

The latter half of our methodology (post-processing and evaluation) converts deterministic CNN outputs into calibrated probabilistic forecasts using IDR, and rigorously evaluates them with proper scoring rules. By introducing daily-aggregated, area-weighted scoring and using a climatological baseline, we ensure that the reported skill metrics are physically meaningful and aligned with standard verification practice. All these steps are implemented in a reproducible workflow, lending credibility to the results presented in the next chapter and allowing future researchers to build on or audit our approach. The careful design of experiments and verification in this chapter lays the groundwork for interpreting the model performance in the Results and Discussion that follow.

4. Results

This chapter reports the empirical findings of our study in order to answer the research questions RQ1, RQ2 and RQ3. We begin with the overall day-ahead probabilistic score and skill over Germany contextualized by a comparison to a numerical-weather-prediction (NWP) benchmark RQ1. We then quantify how the training loss balanced wide synoptic context against a regional evaluation focus RQ2, and finally measure the value of adding atmospheric predictors beyond lagged rain RQ3. A final section details the optimal loss function and hyperparameters for our precipitation-only baseline which served as the foundation for all subsequent inquiry.

Note: All values of CRPS are in units of mm while CRPSS is unitless by design. Both CRPS and CRPSS are averaged across all evaluation years from 2010 to 2020 unless stated otherwise.

4.1. Overall Metrics and Comparison with NWP

We begin by summarizing the overall performance of the best model and then contextualize these results with a comparison to an NWP benchmark (addressing RQ1). We find the optimal configuration to include not only lagged daily precipitation but also ERA5 multi-level winds and specific humidity at 300, 500, 700, and 850 hPa. The model is trained on the period 2007-2019, with regional weighted loss tuned to emphasize Germany via an outside-weight of $w = 0.9$ (see 4.1). In this overall sample, the mean CRPS of the best model is about 1.26 mm, and the Continuous Ranked Probability Skill Score reaches CRPSS ≈ 0.254 (i.e. about 25.4% lower CRPS than the baseline climatology).

To put these results into context, we compare against the published ECMWF ensemble precipitation forecast skill in summer 2016. Gascón et al. [10] report that over the EFAS Europe domain, the domain-mean CRPS for 24-h precipitation at day+1 lead time in JJA 2016 was approximately 1.3 mm. For direct comparison with ECMWF, our best model configuration ($w = 0.9$ with multi-level winds and specific humidity) was trained only on 2007-2015 data (to keep 2016 independent) and achieved a CRPS of 1.52 mm (averaged over German sub-domain) in the JJA season of 2016, corresponding to 29% skill relative to the monthly probabilistic climatology baseline (see 4.2. Superficially, this places our system's performance between the ECMWF ensemble's day+3 and day+5 CRPS values for JJA 2016 (see 4.1). However, it must be emphasized that this is not an apples-to-apples comparison. There are several configuration differences and caveats that complicate any direct equivalence between our data-driven model and the NWP benchmark. Figure 4.1

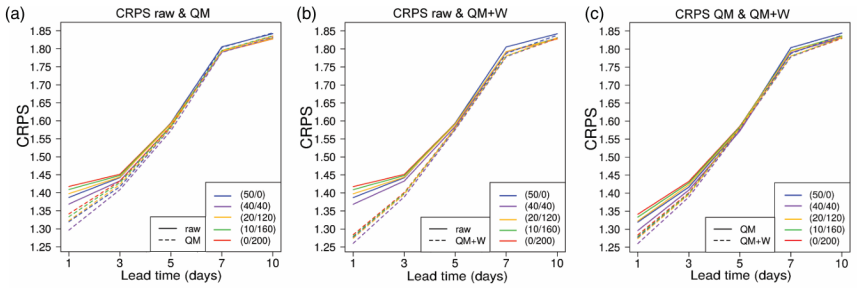

Figure 4.1.: NWP CRPS JJA 2016. Domain-mean 24-h precipitation CRPS of the ECMWF dual-resolution ensemble (EFAS Europe domain) for various lead times in JJA 2016 (Gascón et al. [10]).

illustrates the reference NWP scores: it shows the CRPS of ECMWF's dual-resolution ensemble forecasts in JJA 2016 as a function of lead time, with CRPS at 1.3 mm at day+1 and slowly degrading by day+10. Our model's CRPS roughly matches that of a 4-day lead forecast in that particular comparison.

Several factors need to be considered when assessing why our model's CRPS can appear relatively comparable to NWP benchmark, aside from genuine forecast skill. First, domain differences play a role: the ECMWF results are evaluated over the expansive EFAS domain (pan-Europe with dense station coverage), whereas our verification is restricted to the Germany domain. Germany in summer is, on average, somewhat drier than southern Europe and mountainous regions included in EFAS, which affects absolute error magnitudes. Second, we verify forecasts at a coarser spatial scale i.e. our model outputs are on a 1° grid, and we verify against the gridded MSWEP analysis. In contrast, Gascón et al. [10] verify against high-resolution rain-gauge-based analyses. Verifying at a coarser grid averaging tends to reduce the absolute CRPS (since spatial averaging smooths extreme localized errors) relative to station-level verification.

On the other hand, we specifically chose 2016 for evaluation to match the period used by Gascón et al. [10]. This meant our model for this test did not benefit from the full 2007-2019 training span, using only 9 years of training (2007-2015) which is suboptimal for performance, and likely a reason our CRPS is higher than it would be with a larger training set. Nonetheless, we deemed this restriction necessary for a fair comparison, since using a different evaluation year (especially if notably wetter or drier than 2016) would itself alter the CRPS.

Taking all these caveats together, the broad message positively answers the research question that lies at the heart of our Thesis (RQ1) i.e. **our parsimonious data-driven day-ahead forecast achieves comparable CRPS values to a 3–5 day operational NWP forecast in this mid-latitude summer scenario.** But this comparison comes with important qualifiers and is not a strict head-to-head comparison of superiority or deficiency.

We now turn to the spatial distribution of skill in our model's summer forecasts. Figure 4.2 shows a map of the CRPS skill score (CRPSS) for our day-1 precipitation forecasts over Germany in JJA 2016, evaluated against the monthly climatology baseline. Skill is clearly heterogeneous across the country. Notably, our model performs best in southern Germany,

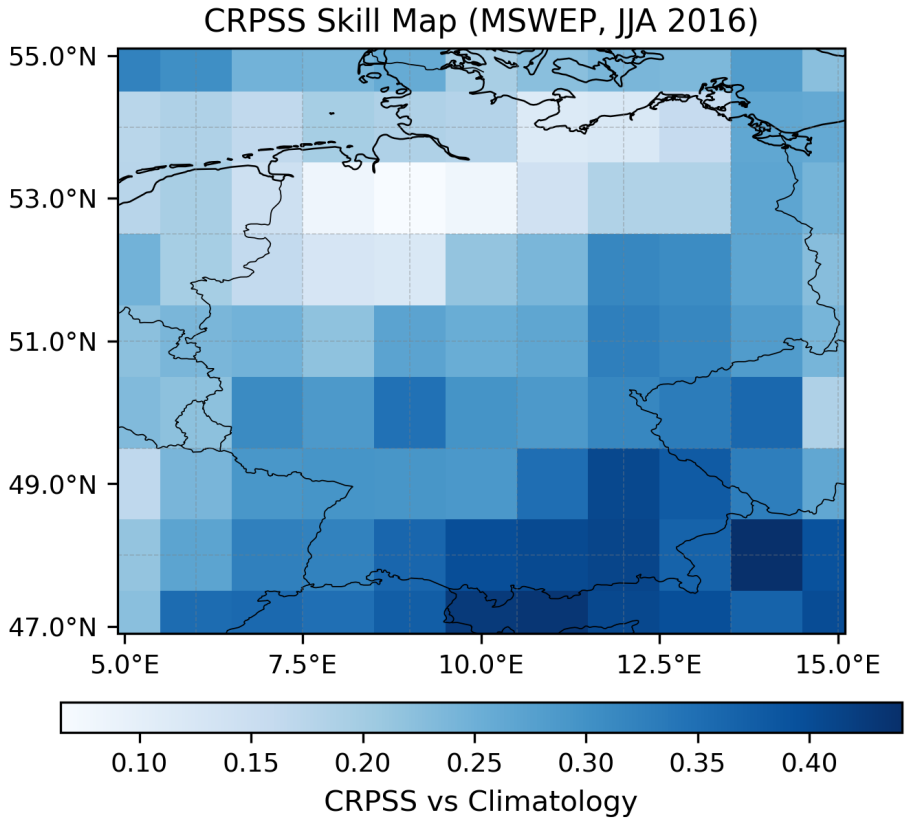

Figure 4.2.: JJA 2016 CRPSS. Spatial distribution of our model's CRPS skill score for day-ahead precipitation forecasts over Germany in JJA 2016. Skill is calculated against a monthly probabilistic climatology baseline (MPC)

with local CRPSS values reaching upwards of 0.3–0.4 in parts of Bavaria and southwest Germany. By contrast, skill drops off in the north, and in portions of northeastern Germany the CRPSS is near zero (indicating the model did not outperform climatology there).

The high skill in southern Germany can be attributed to that region's convective and orographic rainfall dynamics. In summer, precipitation in the south (e.g. along the Alpine foothills and uplands of Bavaria and Baden-Württemberg) often arises from mesoscale convective events – afternoon thunderstorms and heavy downpours triggered by orography or boundary-layer convergence. These events are intermittent and localized. Our model, however, can learn associations between large-scale conditions (moisture flux, instability proxies, etc.) and the likelihood of convective storms, allowing it to outperform the baseline in these scenarios. By contrast, the low or negative skill in the northeast likely reflects a more stable summer rainfall regime. Northeastern Germany (e.g. parts of Brandenburg and Mecklenburg) experiences relatively drier and less variable summer weather, often dominated by persistent high-pressure systems with only occasional rainfall. In such areas, the monthly climatology baseline (which might, for example, simply predict a high probability of zero rain on any given summer day) is already quite hard to beat. The model does not significantly improve upon that already-good baseline forecast. It is important to

clarify that a low CRPSS in those regions does not necessarily mean the model's forecasts are terrible in absolute terms; rather, it means that climatology could be a strong forecast there(as we have seen for the tropical convective regime in Walz et al. [30]).

It is also insightful to consider these results in light of mid-latitude summer meteorology and our model's inherent limitations. Summer convective precipitation in Germany is notoriously challenging to predict: it is patchy and often governed by small-scale processes (e.g. boundary-layer heating, local convergence, orography) that are only weakly constrained by the large-scale atmospheric state. Our model's predictors – even with the inclusion of upper-level winds and humidity – primarily represent the synoptic-scale conditions. As a result, they carry less predictive power for convective thunderstorms, which helps explain why our JJA skill improvements are modest.

In summary, our best data-driven model delivers a CRPSS on the order of 25–30% against a monthly climatology baseline in summer, and its day-ahead precipitation forecasts for Germany attain a CRPS comparable to medium-range NWP forecasts (albeit under non-identical conditions but useful for a rough order-of-magnitude comparison). These results highlight the strength of the approach, providing tangible skill in a mid-latitude convective season at extremely low computational cost. The next sections will explore the results from the downstream research questions RQ2 and RQ3 that arise when considering design choices to answer RQ1, namely, the spatial training focus and the inclusion of additional predictors respectively.

4.2. Optimal Spatial Context

Our first set of experiments on the road to answering RQ1 addressed a critical design choice: How should wide synoptic context be balanced with regional focus during training? (RQ2). As described in chapter 3, we introduced an "outside weight" parameter to control the relative importance of grid cells within Germany versus the surrounding region. This investigation was motivated by the hypothesis that mid-latitude precipitation over Germany cannot be understood in isolation from the broader atmospheric circulation. However, our objective was producing skilled precipitation forecasts over Germany, not the whole domain.

4.2.1. The Search for Optimal Regional Weighting

We trained the U-Net on the Euro–Atlantic domain using a piecewise spatial loss mask that assigns weight 1.0 to Germany, a three-cell neighbor ring at the midpoint between 1.0 and the outside weight, and a tunable outside weight everywhere else (see 3.1. This design preserves upstream synoptic context while emphasizing errors over Germany during optimization.

We systematically evaluated outside weight values across a broad range to understand how spatial context influences forecast quality. The experiments tested weights of between 0.2, and 1.0

This range between 0.2 and 0.7 performed slightly poorly compared to not having any regional focus at all. This could suggest that the model's ability to learn synoptic scale features is noticeably affected in this range. However, the range 0.75 to 0.9 performed better than having no regional focus i.e setting the outside weight to 1 thereby **empirically answering RQ2** with a possibility of further refinement by varying outside weight with various ERA5 configurations or by increasing total training data as well.

Table 4.1.: Impact of outside weight on model performance. All experiments use precipitation-only input with lag features and IDR post-processing. Evaluation performed over Germany with cosine-latitude area weighting.

Weight	Mean CRPS	CRPSS	Seasonal CRPS			
			DJF	MAM	JJA	SON
0.20	1.284	0.239	1.046	1.214	1.300	1.573
0.50	1.285	0.239	1.022	1.224	1.317	1.574
0.70	1.294	0.233	1.042	1.243	1.313	1.577
0.80	1.279	0.243	1.036	1.222	1.296	1.559
0.85	1.280	0.242	1.021	1.245	1.294	1.556
0.90	**1.273**	**0.246**	**1.024**	**1.201**	**1.292**	**1.572**
1.00	1.288	0.237	1.036	1.219	1.315	1.578

[—] indicates data not available in current evaluation set

Figure 4.3 summarizes the Germany-mean CRPS as a function of the outside weight w. The curve is not a smooth U-shape with a single global minimum but shows multiple local extrema across $w \in [0, 1]$. A physically consistent interpretation is that the *composite* mean CRPS aggregates four season-specific responses to w that differ in sign and amplitude; their superposition yields the observed undulations rather than a clean quadratic minimum. In winter and spring, modest down-weighting outside Germany (here, $w \approx 0.85$–0.90) preserves crucial upstream storm-track information while focusing gradients over the target; in summer the response is weak; in autumn the regime mixture broadens the error distribution and flattens the sensitivity.

A high outside weight maintains learning of upstream storm-track structure and moisture pathways while modestly prioritizing the verification subdomain; this could be interpreted as being consistent with mid-latitude dynamics (advecting fronts/WCBs and orographic inflow) and with our inductive-bias argument for pairing wide training context with regional emphasis. However, one could also argue that the minor improvements in performance are not meteorological in nature but instead solely a statistical artifact.

To breakdown and explore these results further, we assess the seasonal variation in CRPS caused by the choice of outside weight.

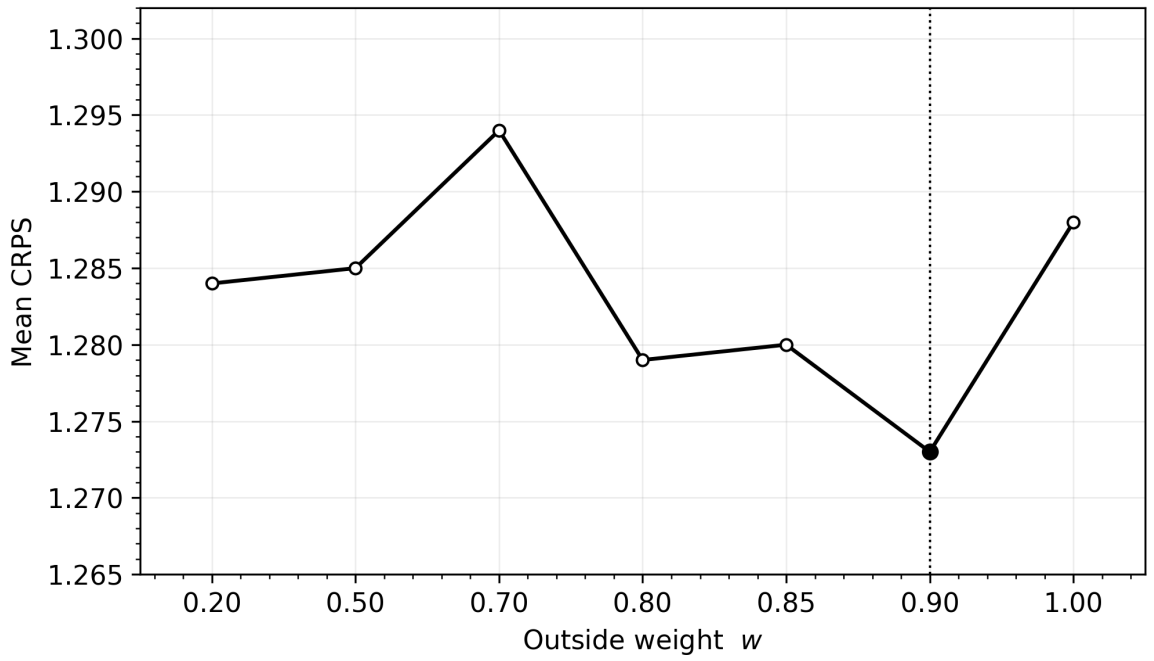

Figure 4.3.: Overall Mean CRPS vs outside weight

4.2.2. Seasonal Sensitivity to Spatial Context

The importance of regional context varies by season, reflecting different dominant mechanisms:

The outside-weight parameter, w, controls how strongly the loss "listens" to the circulation around Germany while the model learns to predict rain over Germany. In mid-latitudes, the pieces that assemble a precipitation event (moisture, ascent, and duration) are organized by flow features that often extend far beyond national borders—fronts and warm-conveyor belts within baroclinic cyclones, jet-streak divergence aloft, and, at times, atmospheric rivers (ARs) that deliver moisture from the Atlantic. The synoptic imprint of these features is seasonally distinct, so the amount of upstream context the model should emphasize to learn them optimally is also seasonal:

Winter (DJF) precipitation over Central Europe is dominated by synoptic-scale baroclinic systems with clear precursors in the North Atlantic jet and upper-level troughs. A moderate down-weighting outside Germany near 0.85 appears to be ideal: it keeps most of the surrounding flow information—crucial to represent quasi-geostrophic forcing for ascent from vorticity and temperature advection—without letting distant, ultimately irrelevant storms dominate the optimization.

Spring (MAM) is a transition season with frequent frontal waves and variable tracks; a slightly higher optimal near 0.9 is consistent with the need to retain even more of the basin-scale context to learn incipient cyclogenesis and jet-streak forcing that are still strongly dynamical but less repeatable than in deep winter.

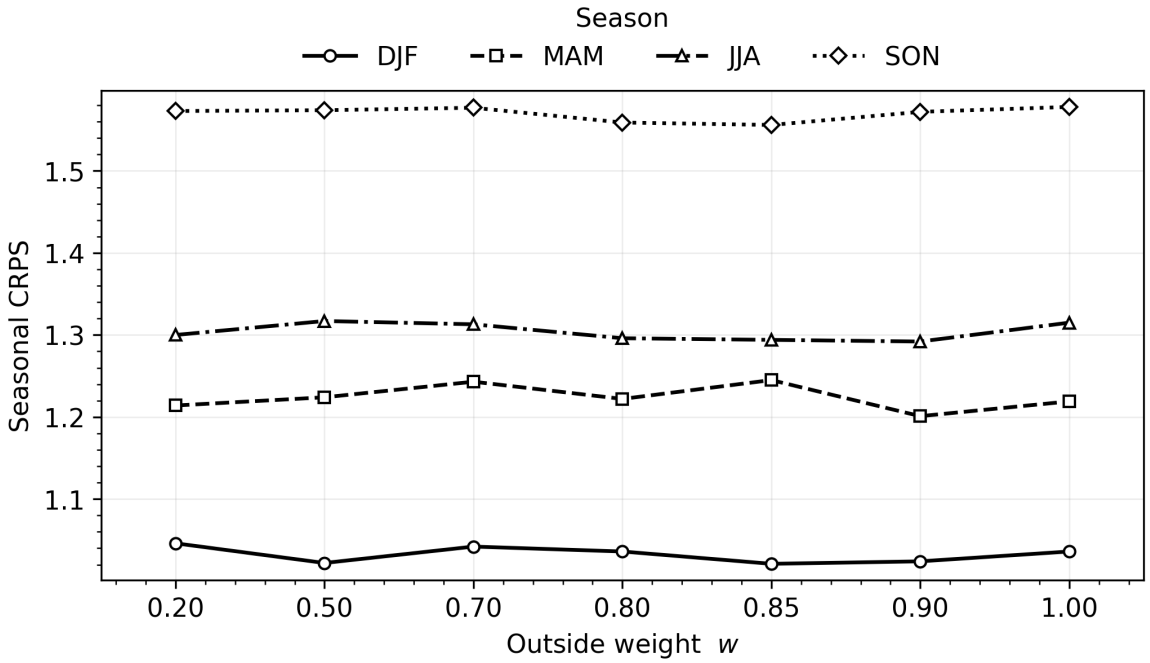

Figure 4.4.: Seasonal CRPS as a function of outside weight, demonstrating stronger sensitivity in winter when synoptic-scale dynamics dominate.

Summer (JJA) rain increasingly hinges on mesoscale convection and boundary-layer processes; large-scale precursors are weaker and more ambiguous. The curve's broad, shallow minimum (little sensitivity to w fits a regime where remote context helps only marginally because many events are triggered locally and stochastically.

Autumn (SON) returns to stronger baroclinicity and frequent AR influence on Europe's west flank; your shallow minimum around 0.85 is again consistent with needing substantial—but not equal—weight outside the target domain to encode moisture-flux corridors and frontogenetical zones before they arrive.

Across outside–weight choices, autumn exhibits the largest mean CRPS in our evaluation (Figure 4.4), and this seasonal minimum has a clear physical basis. In early–mid autumn (especially September but even sometimes October) North Atlantic and adjacent shelf-sea SSTs remain near their annual maximum, so upward latent and sensible heat fluxes are at their strongest. The column therefore carries abundant moist static energy and can support both convective outbreaks and vigorous frontal precipitation. At the same time, the wintertime jet and storm track are not yet fully established in latitude or intensity; baroclinicity and jet meanders fluctuate from episode to episode. The result is a mixed-regime season in which Germany alternates between summer-like, anticyclonic dry spells, transient thunderstorm situations, and early winter-like frontal systems. Such rapid regime switching broadens the distribution of next-day precipitation outcomes and weakens the persistence and synoptic precursors our model (and many post-processing schemes) typically exploit at a daily lead, inflating CRPS relative to winter and, to a lesser extent, summer.

A further autumn-specific source of forecast difficulty is the frequent recurvature and extratropical transition of Atlantic tropical cyclones. These systems inject air with anomalously high equivalent potential temperature and strong PV (potential vorticity) perturbations into the mid-latitude flow and can phase nonlinearly with baroclinic waves, producing rapid deepening, AR(atmospheric river)-assisted moisture plumes, and large downstream spread in track, timing, and rainfall amplitude. Small phase or displacement errors translate into large probability penalties at daily accumulation, again elevating CRPS in SON. Consistently, European "shoulder-season" verification often shows reduced NWP skill for precipitation-related metrics, so the SON minimum is meteorologically plausible and aligned with expectations.

The larger undulations in DJF and MAM indicate that forecast quality is sensitive to how much nonlocal information is emphasized during learning—exactly what one expects when precipitation depends on the correct representation of Rossby-wave packets, jet streaks, and fronts, whose signals are embedded in the multi-level wind field and evolve on continental scales. Small changes in w subtly alter how much the network prioritizes those upwind precursors and thus can produce noticeable CRPS differences.

By contrast, the near-flat JJA curve says: most of what limits daily precipitation skill in summer is not recoverable by reweighting nonlocal context. Warm-season rainfall over land is dominated by organized but locally triggered convection and mesoscale convective systems whose timing and placement are less predictable from synoptic fields alone; even when episodes propagate, their initiation and intensity are set by boundary-layer thermodynamics and mesoscale convergence that daily, large-scale predictors only weakly constrain. This intrinsic property of the convective regime is why JJA variations are modest and relatively insensitive to w.

If these minima were merely statistical artifacts of sample size or optimization noise, we would expect similar sensitivity across seasons (or random, unstructured fluctuations). Instead, the pattern aligns with well-documented differences in European precipitation physics:

DJF/MAM: Extreme and non-extreme precipitation alike are tightly linked to fronts, warm conveyor belts, and upper-level disturbances; many studies show the bulk of heavy precipitation events in Europe are associated with fronts and/or strong moisture transport, and often with Rossby-wave breaking upstream.

JJA: Warm-season rainfall predictability is lower; episodes are governed by mesoscale convective organization and boundary-layer processes, with weaker ties to synoptic patterns.

That said, we should be transparent: the amplitudes of the CRPS differences are modest, and a portion may reflect sampling variability. We consider further investigations and possibilities for improvement in chapter 5.

4.3. Enhancing Predictions with Atmospheric Information

Having established that an outside weight of 0.9 provides optimal performance, our quest for improving model forecast skill naturally leads to RQ3 i.e. How much incremental skill arises from adding atmospheric context beyond lagged rain?

We evaluated three configurations, each building upon the optimal regional weighting:

1. Baseline: Precipitation only (with w=0.9)
2. Wind Configuration: Adding u and v wind components at 300, 500, 700, and 850 hPa
3. Wind and Humidity: Wind components plus specific humidity (q) at the same levels
3. Comprehensive Configuration: Wind components plus specific humidity (q) at the same levels and all single level variables MSLP, TCWV, t2m, SP

Table 4.2.: Performance comparison of ERA5 predictor configurations. All models use outside weight = 0.9.

Configuration	Mean CRPS	CRPSS	Improvement	DJF CRPSS	JJA CRPSS
No ERA5 (w=0.9)	1.2731	0.2461	baseline	0.2893	0.2259
Wind Only	1.2687	0.2484	0.35%	0.3034	0.2313
Wind and Humidity	1.2587	0.2544	1.1%	0.3051	0.2352
All Variables	**1.2589**	**0.2543**	**1.1%**	**0.3051**	**0.2350**

The results demonstrate clear, statistically significant improvements with the addition of the first two chunks of atmospheric information. The wind-only configuration reduces CRPS by 0.35%, while the addition of humidity achieves a 1.1% reduction compared to the precipitation-only baseline. Thus, **we empirically answer RQ3** with an important caveat: The addition of all 4 single-level variables producing statistically indistinguishable differences is rather surprising.

4.3.1. Physical Interpretation of Improvements

Impact of Wind Predictors (u/v) Including wind fields from ERA5 at multiple pressure levels (300, 500, 700, 850 hPa) in the input feature set led to a measurable improvement in forecast accuracy. Quantitatively, adding these wind predictors yielded about a 0.35% reduction in CRPS (and a corresponding uptick in CRPSS) compared to the baseline model without ERA5 features. From a meteorological standpoint, this result is quite sensible. Wind fields are the carriers of the atmospheric state. They encapsulate the motion of weather systems, advection of air masses, and areas of convergence or divergence that can drive vertical motions. By providing the CNN with dynamic information about the atmosphere, we make it easier for the model to learn cause-and-effect relationships that lead to precipitation. For example, the 300 hPa wind (near the jet stream level) is indicative of jet streaks and upper-level troughs. A strong jet stream overhead often implies upper-level divergence downstream of the jet streak exit region, which encourages rising motion beneath it. Such

rising motion is a key ingredient for precipitation development (as lifting cools the air and condenses moisture). At the mid-levels (500 hPa), wind patterns show the positioning of troughs and ridges; a deep 500 hPa trough to the west of Germany, for instance, signals a likely large-scale ascent over Germany ahead of the trough, often corresponding to widespread precipitation. Lower-level winds, like at 700 hPa and especially 850 hPa, are even more directly tied to precipitation processes: they can delineate low-level jets and moisture transport pathways. An 850 hPa southerly flow from the Mediterranean or Atlantic can flood Europe with warm, moist air, setting the stage for heavy rainfall given a trigger for ascent. Moreover, low-level wind convergence (converging winds at 850 hPa) typically signifies air piling up and being forced upward which is a classic mechanism for generating preciipitation (such as along cold fronts or in upslope flow against mountains).

In essence, adding winds helped the CNN answer the "where" and "how" of precipitation: Where are the air masses coming from, and how are they converging or diverging? This dynamic context is critical. For example, a purely precipitation-based model might struggle to predict a heavy rain event if the rain hasn't started yet (no strong signal in yesterday's precipitation), but the wind fields could reveal a developing cyclonic circulation and moisture feed that will likely produce rain tomorrow.

Impact of Adding Humidity Predictors(q) When specific humidity (q) at the same pressure levels was incorporated alongside the winds, the forecast skill improved further. We observed roughly a 1.1% CRPSS gain over the baseline, effectively about twice the improvement seen with winds alone. This indicates that humidity was a particularly valuable addition. Meteorologically, this result is highly intuitive: moisture availability is a fundamental limiting factor for precipitation. No matter how strong the upward motion or how favorable the dynamics, without sufficient water vapor, you cannot get substantial rainfall. In fact, given a certain amount of lift in the atmosphere, the intensity of the resulting precipitation is largely determined by how much moisture is present to condense out. The inclusion of humidity variables gave the CNN direct insight into the atmospheric moisture content at various layers, essentially answering the question "Is there fuel for rain in this air parcel?" Rather than infer moisture indirectly (say, from temperature or other proxies), the model now knows the actual specific humidity, which sharpens its ability to predict rain.

Consider a scenario: a strong frontal system is approaching, and the wind fields might suggest uplift. However, if the air ahead of the front is dry, rainfall will be limited (perhaps just clouds). Conversely, a marginal dynamical situation can still produce heavy showers if the air is exceptionally moist (think of tropical moisture surges). By adding humidity data, the CNN can distinguish between these scenarios. The result is a further 0.75% improvement beyond winds which reflects the model's enhanced skill in gauging where and how much precipitation will fall, not just that some precipitation will occur. This aligns well with the conceptual model of precipitation forecasting: rain = moisture × lift. We gave the model both components explicitly (winds for lift, q for moisture), and it responded with better forecasts.

Lack of Further Improvement from Surface Variables (MSLP, TCWV, SP, T2M) After incorporating winds and humidity, we also experimented with adding various single-level or surface-based variables from ERA5: mean sea-level pressure (MSLP), 2-meter temperature (T2M), surface pressure (SP), and total column water vapor (TCWV, which is essentially the vertically integrated moisture). Perhaps surprisingly, the inclusion of these additional predictors did not yield any noticeable improvement in forecast skill; the CRPSS remained statistically unchanged with their addition. This outcome can be understood by considering the information content and redundancy of these variables relative to what the model already had. By the time we had winds and multi-level humidity in the mix, the CNN was already directly apprised of the essential dynamics and thermodynamics of the atmosphere.

Take MSLP as an example. Mean sea-level pressure is a diagnostic field that reflects the integrated effect of mass distribution in the atmospheric column. Large-scale storm systems (cyclones and anticyclones) show up clearly in the MSLP pattern. However, the model was already ingesting wind fields at 850 hPa, 700 hPa, etc., which themselves arise from pressure gradients (by geostrophic and ageostrophic wind relationships). If a deep low exists, the 850 hPa winds will be strong and convergent; if a strong high exists, the winds will tell that story as well. In other words, MSLP doesn't provide much new information if upper- and lower-level winds are known. The CNN with multi-level winds, humidity and seasonality(always present) likely already captured the presence of highs, lows, and frontal zones; adding MSLP could be another view of the same as it didn't change the overall picture the model had learned. Similarly, surface pressure (SP) is nearly redundant with MSLP (just altitude-adjusted), so it's no surprise that adding it did nothing unique.

Now consider 2-meter temperature (T2M). Near-surface temperature can influence precipitation in a few ways: it determines precipitation type (rain vs snow), and when coupled with moisture it can indicate low-level instability or boundary-layer humidity (via dew point depression). However, in a daily aggregated precipitation forecast over an entire country, fine distinctions of rain vs snow or minor stability tweaks may not play a dominant role in the score, especially since we did not target separate metrics for snow or extreme convective storms specifically. Also, the model already had some measure of seasonality(sin and cos of the day) in the baseline. It likely roughly knew if it was summer or winter, and hence a first-order approximation of temperature. Moreover, low-level temperature fields often vary on small spatial scales (especially in summer with diurnal heating), and if not paired with equally high-resolution data on, say, terrain or land use, they might just introduce noise. The CNN might have simply learned to ignore T2M as it wasn't strongly correlated with next-day precipitation occurrence beyond what other factors provided. In short, T2M did not provide a strong direct physical handle on precipitation amount for our purposes. The primary drivers (moisture and lift) were already accounted for, and the presence of T2M didn't alter those.

Total column water vapor (TCWV) is an interesting case: one might expect it to be very useful, since it's essentially a measure of how much moisture is in the entire column and higher TCWV often means greater rainfall potential. However, our model already had

specific humidity at four pressure levels. Those should allow it to infer the column moisture to a large extent (e.g. high humidity at 850 and 700 hPa is going to imply high TCWV, even if we didn't explicitly provide TCWV). In fact, TCWV is mostly a vertical integral of the humidity profile; adding it is somewhat redundant if the profile itself is known. There could be situations where TCWV might add information (say, if moisture mostly lies near the surface, and the chosen levels don't capture it perfectly), but apparently in our dataset those nuances were either rare or the model could interpolate them. It appears that once winds and q were in place, TCWV did not further constrain the problem in a way that the model could capitalize on.

From a machine learning perspective, these results illustrate the idea of diminishing returns and feature redundancy. After a certain point, adding more features that are correlated with existing ones yields little to no gain in predictive power. The model naturally focuses on the most informative features and can ignore those that don't help. In our case, winds and mid-level humidity were clearly informative (hence the gains). The surface variables were either derivable from those (MSLP, TCWV) or only weakly related to precipitation (T2M, given other inputs). The lack of CRPSS change tells us the CNN either learned nothing new from them or possibly couldn't learn effectively due to sample size. It's worth noting that our training dataset, while sizeable for a thesis project, might not be enormous by deep learning standards at least in relation to ten-fold increase in total input channels when all ERA5 predictors are included (compared to baseline). With such data volumes, a model will focus on the strongest signals. Subtler predictors might require more data to tease out their utility. There is a known phenomenon in statistics and machine learning regarding high-dimensional data: unless the dataset is sufficiently large, adding features can lead to overfitting or simply not improve generalization which is sometimes dubbed the "curse of dimensionality". In our case, adding these extra ERA5 fields increased the input dimensionality, but without a proportionate increase in training examples, the model did not find reliable new patterns associated with them. It's plausible that with an order of magnitude more data, one might start to see benefits from variables like T2M or MSLP. But within our scope, the dominant drivers were already captured, and the single-level extras were essentially superfluous. This outcome highlights that a parsimonious set of well-chosen predictors (here, rainfall history/seasonality plus winds and humidity aloft) can achieve nearly the maximum performance within the bounds of your data volume and hardware.

ERA5 augmentation and the role of sample size We also assessed sample-size sensitivity when using our expanding-window design (train up to year N, validate on N+1). Winds (u, v) and humidity (q) provide compact dynamical and moisture context (baroclinic forcing and IVT preconditioning) that the network cannot infer from lagged rain alone, hence the observed ~1.13% CRPS reduction. At the same time, our experiment with an expanding training window reveal a clear downward trend of mean CRPS as more years are included (Fig. 4.5): early folds trained on 3–4 yr exhibit CRPS $\gtrsim 1.4$, whereas later folds (~11–13 yr) settle near $\lesssim 1.23$ (linear trend shown). Part of this improvement can arise from our time-ordered validation (each fold verifies the next year), but the magnitude and persistence

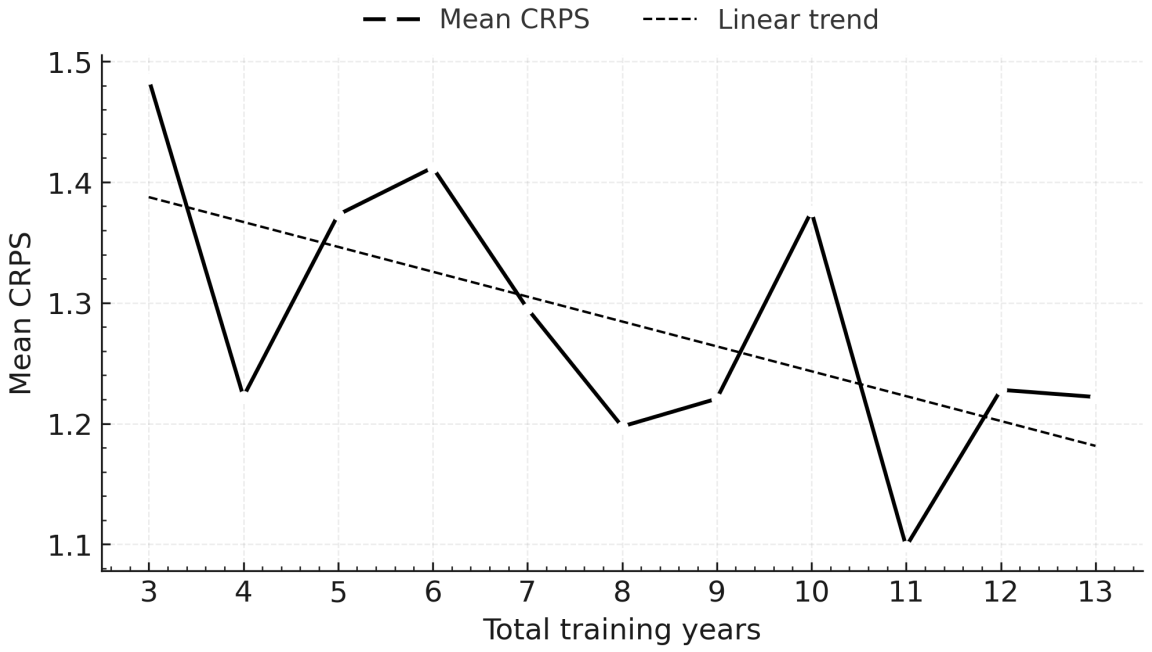

Figure 4.5.: Effect of training-set length (expanding window) on Germany-mean CRPS. Each point is the fold-mean CRPS when training on the years up to the abscissa value and validating on the immediate following year. The dashed line shows a fitted linear trend.

of the decline are also consistent with a *sample-complexity* effect: once ERA5 channels are introduced, the dimensionality of the predictor space increases greatly, and more years could be needed to reliably estimate the mapping. This could help explain why single-level fields (MSLP, TCWV, T2M, SP) did not add skill in our 14-yr setup; with a longer archive, their incremental value may emerge.

4.3.2. Visual Analysis

We present six deterministic plots that share a common layout and training setup. All six figures use the same regional loss weighting, with an outside weight of $w = 0.9$ Each figure is a 2×3 panel: the top row shows lagged MSWEP precipitation at $t-3$, $t-2$, $t-1$; the bottom row shows the Target $t = 0$, the model's Prediction, and a Difference panel visualising Prediction − Target to highlight pockets of over-/under-prediction. It is also important to note that the deterministic metrics highlighted in the figures like mean precipitation and bias are computed over the entire domain, not only over Germany.

We deliberately select two seasons that bracket model performance:

Winter (DJF), which has the lowest mean CRPS (i.e., highest skill) in our evaluation, and Autumn (SON), which has the highest mean CRPS (lowest skill). For comparability, each season is shown for a fixed verifying day used across configurations: 25.12.2020 for DJF and 22.10.2020 for SON. In winter we display the Focused Germany view; in autumn we

provide both the Focused Germany view and a Full-Domain Euro–Atlantic view to inspect synoptic-scale organisation. The six figures cover baseline i.e only precipitation in the first three figures vs ERA5-augmented configurations in the latter three figures where "ERA5-augmented" denotes adding u, v winds and specific humidity q at 300/500/700/850 hPa

Our target area is Germany and all quantitative verification (CRPS/CRPSS) is computed over the German. The full-domain panels are provided purely for qualitative synoptic context. Errors outside Germany or especially near the outer edges of the training domain are not diagnostic of German-domain skill. By construction the model has less upstream context beyond the domain boundary.

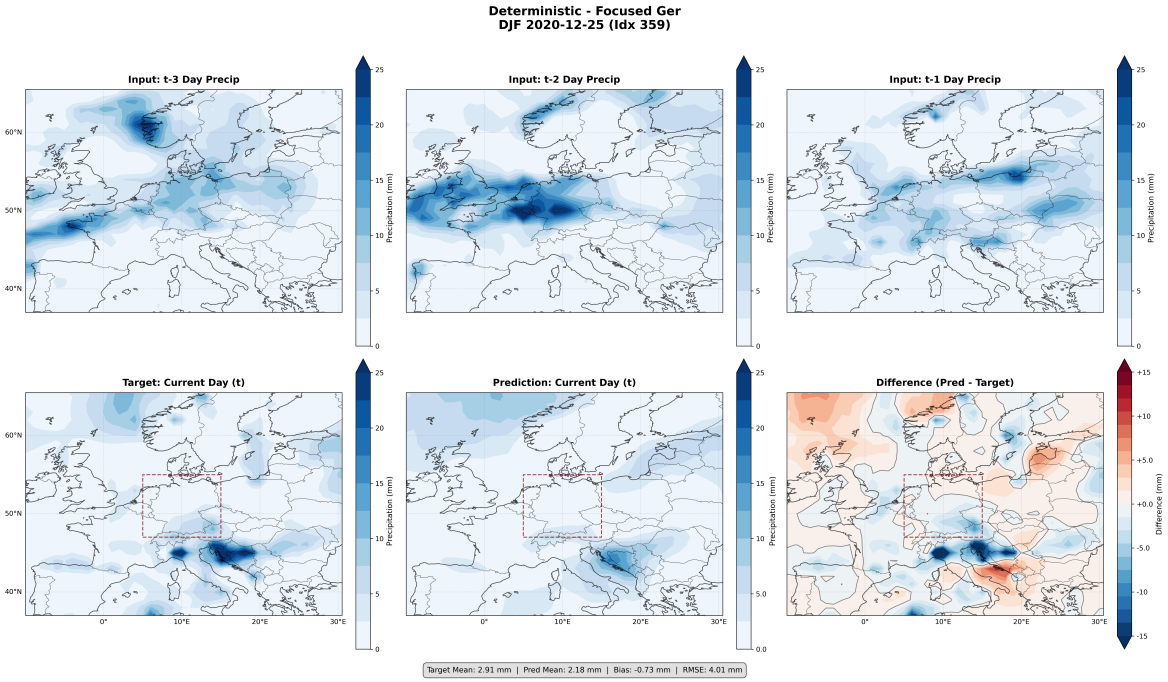

Figure 4.6.: Visualisation of model forecast, baseline (w=0.9), DJF

On 25.12.2020 (DJF) (Figure 4.6) the three lagged MSWEP inputs depict a west–east oriented precipitation band entering the domain from the eastern North Sea and Benelux and extending across western into central Germany. This feature progresses eastward and weakens, consistent with a decaying frontal rainband translated by the westerlies. The verifying MSWEP target retains this organization: a continuous rainband from the Alps into Slovenia and Croatia, indicating dissipation over Germany by the target day. The baseline forecast (precipitation lags + seasonality; no ERA5 predictors) reproduces the location and extent of the German rainband credibly. Over Germany proper, residual errors are small and spatially structured rather than systematic: a modest underprediction confined to the southwest and slight overprediction in parts of the northeast. In magnitude, most German-domain differences are on the order of only a few millimetres in the Pred–Target panel, with no large, domain-wide bias evident.

South of the Alps, however, the difference panel highlights a separate precipitation maximum over northern Italy and the Alpine foreland that is markedly underpredicted. This feature is consistent with a Mediterranean disturbance and orographic precipitation on the Alpine flank, which the precipitation only model struggles to anticipate when it is not strongly foreshadowed in the preceding three days' rainfall over Germany. In other words, the German rainband, advecting and decaying within the westerly flows well captured by persistence-plus-translation learned from the lags, whereas a dynamically forced Mediterranean event, less coupled to prior German rainfall, is missed in amplitude. The rainband over the Croatian coast is correctly identified by the model but under-predicted in intensity.

Finally, patches of over/underprediction near the outer edges of the focused map fall largely outside the verification region and are not central to model assessment here. They are plausibly influenced by reduced upstream context beyond the training domain and do not contradict the main conclusion for this case: within Germany, the baseline forecast places and sizes the Christmas-day rainband well, with only small signed residuals, while substantially underestimating the contemporaneous Mediterranean precipitation to the south (Figure 4.2).

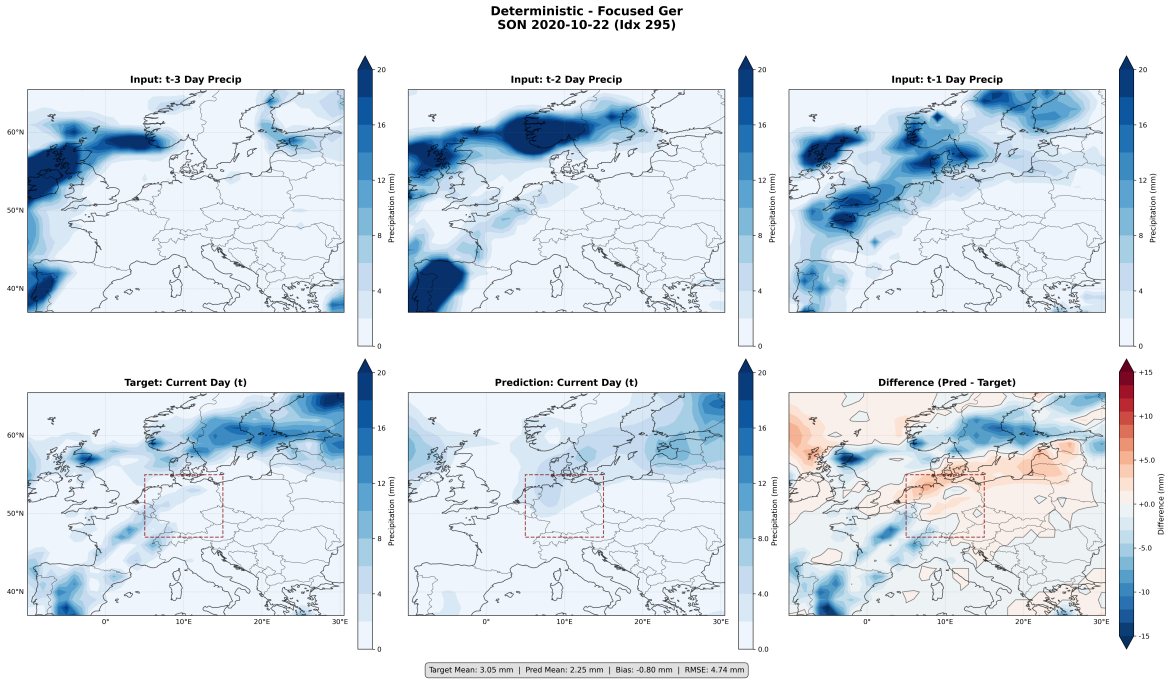

Figure 4.7.: Visualisation of model forecast, baseline (w=0.9), SON

Figure 4.7 shows a classic baroclinic rainband evolving across northwest Europe in the three lagged inputs. A southwest–northeast oriented precipitation corridor advances from the eastern North Atlantic across the British Isles into the North Sea and southern Scandinavia. Germany lies mostly on the drier, equatorward flank(with the exception of some northern states) of this corridor in the lags. On the verifying day, the MSWEP target retains this structure with the main precipitation maximum poleward of Germany (Denmark–southern

Scandinavia), and only light amounts within the German borders. The baseline forecast reproduces the synoptic-scale alignment over-predicts heavily in the northern half of Germany, leaking light to moderate over-prediction into Poland and and Lithuania. The PredTarget panel confirms a swath of over-prediction over southern Scandanavia as well as compensating under-prediction over the Baltic sea. Over northern Germany the over-prediction is rather large but structured, consistent with an amplitude error rather than a displacement of the frontal rainband.

This higher amplitude error in autumn both in under- and over-predictions compared to the modest (within a few millimeters) error we see in Figure 4.6 is further visual explanation as to why we observe a significantly worse autumn(SON) CRPS compared to winter(DJF).

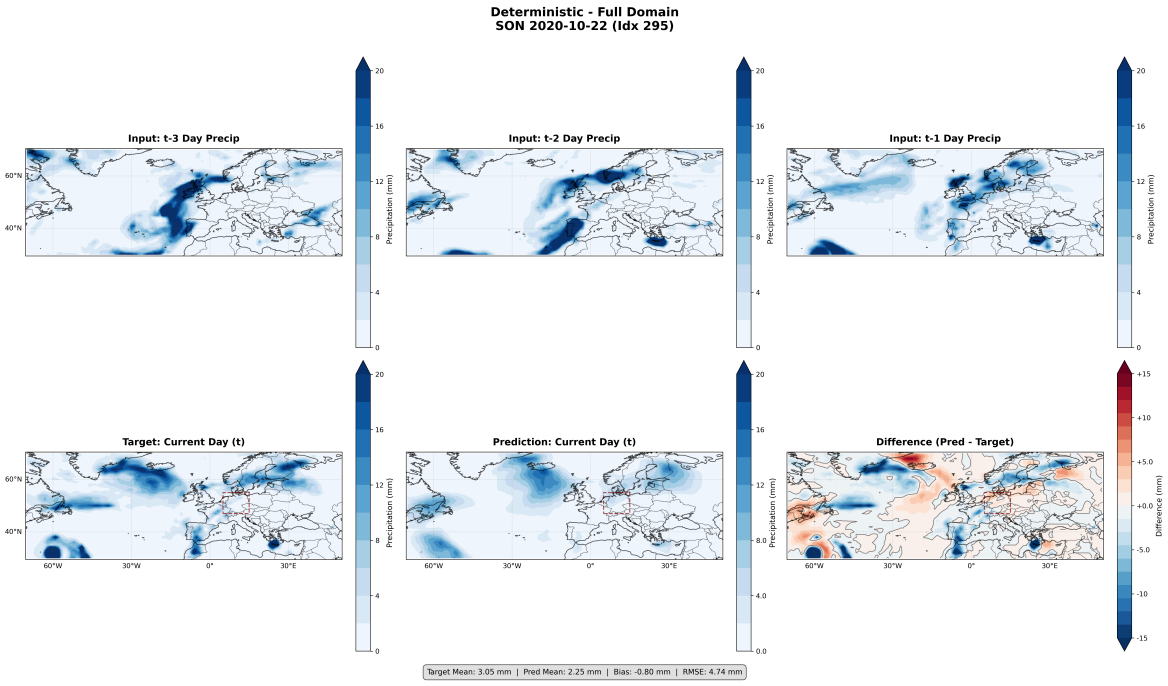

Figure 4.8.: Visualisation of model forecast, baseline (w=0.9), SON, full domain

Zooming out from the previous figure and assessing the full domain of the autumn forecast in Figure 4.8 we observe displacement errors at the far east of our domain, namely Newfoundland. We also observe consistent under-prediction and amplitude errors over the North Atlantic, Scandanavia and Eastern Russia despite relatively small displacement errors in these regions. This is consistent with our hypotheses that the outer edges of the full domain are bound to have big errors due to low upstream spatial context while an overall under-prediction (negative bias) is consistent with our expectation from ML-based deterministic forecasts.

A well-known pitfall in precipitation forecasts is the *double-penalty* problem: a small displacement error of a rainband or convective cell is punished twice by pointwise scores—once as a miss at the observed location and again as a false alarm at the forecast location. As a result, sharper, more realistic high-resolution forecasts can paradoxically score worse than smoother, low-resolution blurred fields.

In parallel, data-driven precipitation models often exhibit *underprediction* of intensities, especially for medium-to-heavy rain. Several mechanisms contribute. First, precipitation is intermittent and highly skewed (mixture of zeros and a heavy-tailed continuous component), so minimizing pixelwise losses (e.g., MSE/MAE) favors regression-to-the-mean and spatial smoothing, which damps peaks. This "blurring" of high-frequency, high-intensity structures is widely reported in nowcasting and related tasks and has motivated loss designs and generative approaches that better preserve extremes [23, 32]. Second, class imbalance (many no/light-rain pixels vs. few heavy-rain pixels) biases standard objectives toward getting the ubiquitous light rain correct at the expense of rare heavy rates; balanced/weighted losses or threshold-aware objectives reduce this tendency and improve high-intensity skill. Third, when forecasts are evaluated pointwise, the double-penalty itself encourages conservative amplitudes: sharper features risk both miss and false-alarm penalties under small displacements, so a learned compromise is a smoother, lower-amplitude field. Together, these effects explain the common underprediction signature in ML-based forecasts.

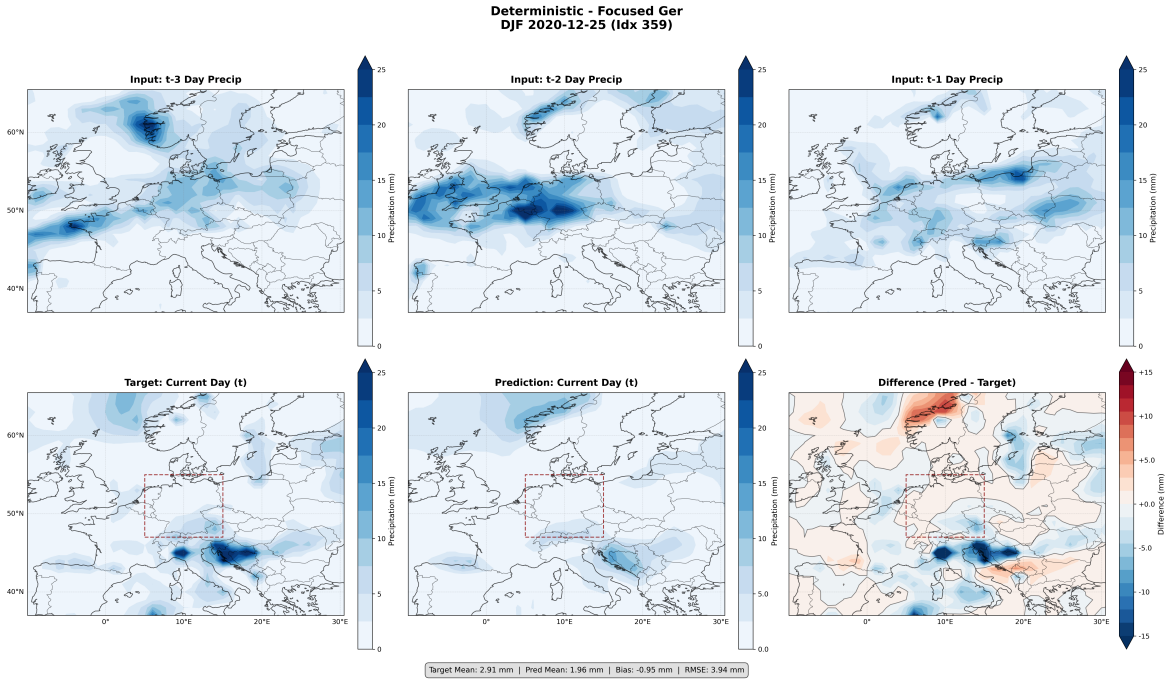

Figure 4.9.: Visualisation of model forecast, u,v,q, (w=0.9), DJF

Now, including the most impactful ERA5 predictors, namely u,v and q at 4 pressure levels, Figure 4.9 when directly compared to Figure 4.6 reveals similar displacement errors but even lower in amplitude, specifically over Germany. This is expected as we are well aware of the 1.1% lower CRPS over Germany, irrelevant of any variations in error we observe outside Germany. Notably, the over all deterministic bias is worse than in Figure 4.6.

Finally, Figure 4.10 and Figure 4.11 showcase once again the poorest forecast skill of autumn(SON). Large rainbands extending from Spain to Benelux are almost entirely missed by the model. Heavy under-predictions in low pressure systems outside our domain of interest and uniform under-prediction (however, lower in amplitude than the over-prediction we

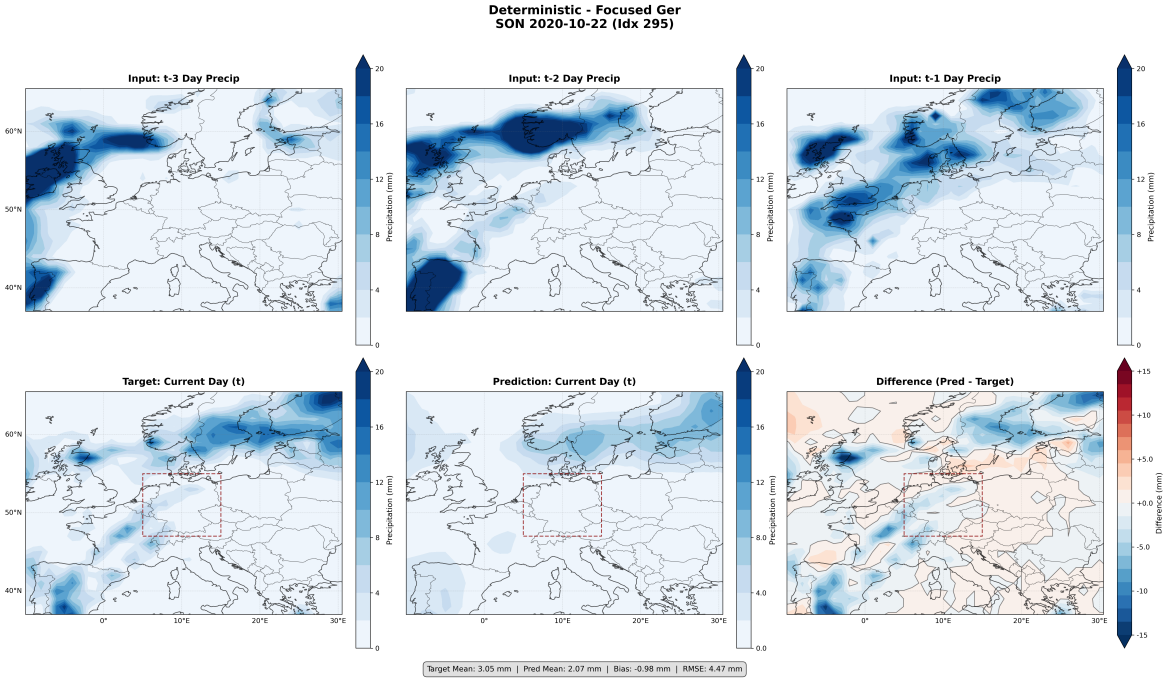

Figure 4.10.: Visualisation of model forecast, u,v,q, (w=0.9), SON

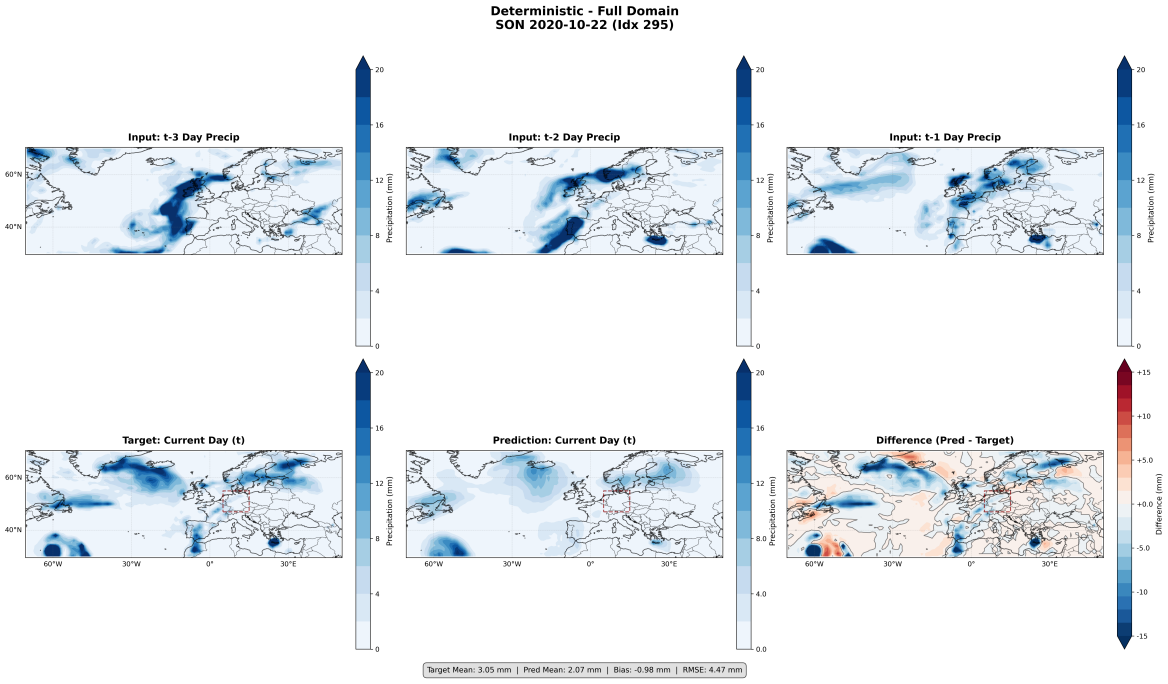

Figure 4.11.: Visualisation of model forecast, u,v,q, (w=0.9), SON, full domain

saw in the baseline model Figure 4.7) emphasize the seasonal variations in improvements of forecast skill with ERA5 augmentation just as we saw for the weighting scheme of the loss function. The contrast between the best CRPS (DJF) and worst CRPS(SON) is highly

evidently correlated with the qualitative and quantitative differences we observe in these determinisitc forecast plots.

4.4. Hyperparameter Tuning

In this section we outline the results of the extensive hyperparameter search. We thus showcase which configuration of hyperparmeters produced the best determinisitc forecasts and served as the baseline precipitation-only model for us to test regional weighting schemes and gains from adding ERA5 variables.

Metric Because probabilistic post–processing via isotonic distributional regression (IDR) and subsequent CRPS evaluation is computationally expensive for each candidate, hyperparameter selection is performed using a deterministic verification metric on $\mathcal{D}_{\text{val}}$. We use the Stable Equitable Error in Probability Space (SEEPS) score, denoted $S_{\text{SEEPS}}(\hat{y}, y)$, in its standard three–category (dry/light/heavy) formulation; lower values indicate better performance (see chapter 3 for implementation details).

Automated Search Execution The hyperparameter search framework implements a grid search with built-in result tracking and analysis:

Algorithm 3: Hyperparameter Search Protocol

foreach $\mathbf{c} \in C$ **do**
    Initialize run directory and logging;
    for $k \leftarrow 0$ **to** $K - 1$ **do**
        Configure data module with fold k;
        Initialize model with configuration $\mathbf{c}$;
        Train model with early stopping;
        Save best model and metrics;
    end
    Aggregate cross-validation metrics;
    Record configuration performance;
end
Rank configurations by validation performance;
return best configuration $\mathbf{c}^*$;

Search space and protocol Following the search logic outlined above, we executed a comprehensive cartesian sweep over the principal hyperparameters:

- **Loss function** $\mathcal{L}$: several pointwise regression losses (including MSE) formulated in log–space.
- **Optimizer** O: Adam and AdamW.

- **Learning–rate scheduler** $\mathcal{S}$: cosine annealing, cosine annealing with warm restarts, and other standard monotone schedules.
- **Learning rate** $\eta_0 \in \{10^{-2}, 10^{-3}, 10^{-3}, 10^{-5}\}$; note that 10^{-2} matches the setting used by Walz *et al.*.
- **Weight decay** $\lambda_{\text{wd}} \in \{10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}, 10^{-8}\}$; note that 10^{-8} matches the setting used by Walz *et al.*

Findings Across the grid, we observed the following, quantified via S_{SEEPS} on $\mathcal{D}_{\text{val}}$:

1. **Learning rate (dominant factor)** The initial learning rate η_0 was the *single most influential* hyperparameter. Suboptimal η_0 values degraded deterministic skill by $O(10\%)$. Concretely, using the Walz *et al.* hyperparameter package (including their η_0) produced $S_{\text{SEEPS}} = 0.66$ on our validation set, whereas our tuned configuration ($\eta_0 = 10^{-3}$ achieved $S_{\text{SEEPS}} = 0.58$, a $\approx 12.1\%$ relative reduction (lower is better).
2. **Loss function** Changing the pointwise loss had a negligible effect: pairwise differences in S_{SEEPS} were $\leq 0.1\%$ (relative), i.e., practically indistinguishable for our application. For methodological continuity with Walz *et al.*, we therefore adopt *MSE in log–space* for all subsequent experiments.
3. **Optimizer** Adam and AdamW produced indistinguishable validation performance. We *choose AdamW* to retain decoupled weight decay and consistency with Walz *et al.*
4. **Learning–rate scheduler.** Cosine annealing with and without warm restarts, yielded no material difference in S_{SEEPS}. We *use cosine annealing without restarts* for simplicity and reproducibility.
5. **Weight decay** Within the tested range $[10^{-8}, 10^{-3}]$, we found a shallow but consistent improvement when λ_{wd} is on the order of 10^{-5}, amounting to $\sim 2\%$ relative reduction in S_{SEEPS} compared with $\lambda_{\text{wd}} = 10^{-8}$ (the Walz *et al.* setting).

Summary Our tuned CNN achieved roughly a 12% reduction in error (increase in skill) over a configuration emulating Walz et al.'s "standard" settings. This is a non-trivial gain, but does not directly prove that Walz et al.'s choices leave headroom that can be exploited. The concrete example is the learning rate for training the CNN. Walz et al. used a relatively aggressive learning rate of 0.01 in their CNN optimization (as seen in their open-source code and confirmed in personal communication). In our experiments, such a high learning rate led to noisy underfitting and poorer forecasts on our system; a more moderate learning rate of 0.001 proved optimal for forecast skill. This difference may stem from various differences in the pipeline and hardware factors. For example, Walz's team might have used different batch normalization or mixed-precision training that tolerated 0.01, whereas our training pipeline on modern GPU hardware favored a smaller step size. It underlines how seemingly minor implementation details can affect model training, and it reinforces the value of re-assessing hyperparameters when transferring the model to a new context. Overall, recognizing the possibility that the original CNN was not fully optimized provided

a clear rationale for us to invest in model tuning as part of adapting CNN+EasyUQ to a mid-latitude application.

5. Conclusion

This thesis adapted the *CNN+EasyUQ* paradigm—deterministic U-Net forecasts calibrated to full predictive distributions via IDR/(discrete)EasyUQ—to a mid-latitude, baroclinic setting over Germany, training on a Euro–Atlantic domain and verifying with cosine-latitude area-weighted proper scores against a monthly probabilistic climatology (MPC) baseline. The resulting system yields robust positive skill year-round, with overall CRPSS values around 24%–25% (best with winds and humidity), thus establishing a parsimonious, physically consistent baseline for day-ahead probabilistic precipitation over Germany. Using the common JJA-2016 benchmark, ECMWF's dual-resolution ensemble reported day+1 CRPS 1.3 mm over EFAS-Europe; our model (trained only through 2015 to keep 2016 independent) yielded CRPS = 1.52 mm over Germany in JJA 2016—roughly comparable to an ECMWF lead of 3–5 days in that season, though not an apples-to-apples comparison due to domain, verification and resolution differences. Spatially, JJA skill concentrates in the Alpine/Black Forest south (CRPSS 0.3–0.4) and is lowest in the drier northeast.

What moved the needle Two design choices were central to our thesis. First, *regional loss weighting* during training—keeping Germany at unit weight while down-weighting the rest of the Euro–Atlantic domain—sharpened German-domain skill without discarding upstream synoptic context. In the precipitation-only configuration, the best outside weight ($w = 0.90$) reduced the Germany-mean CRPS from 1.288 (no weighting, $w = 1$) to 1.273, an **improvement of 1.16%**, with CRPSS increasing from 0.237 to 0.246 Second, adding ERA5 winds (u, v) and humidity (q) at 300/500/700/850 hPa provided additional, physically interpretable gains: relative to the precipitation-only baseline ($w = 0.90$), winds+ q lowered CRPS from 1.2731 to 1.2587 (**1.13%** reduction; CRPSS $0.2461 \rightarrow 0.2544$), while winds alone yielded a smaller 0.35% improvement.

Given that $w = 0.90$ vs. $w = 1.00$ already delivers a ∼**1.16%** CRPS reduction—comparable in magnitude to the ∼**1.13%** gain from adding (u, v, q)—mask design is a meaningful lever for further skill, especially because it is model-agnostic and cheap to test.

The limited scope of our Thesis restricted the exploration of many interesting paths that have the potential to further the forecast skill of parsimonious data-driven models over the mid-latitudes:

1. **Explore neural network architecture** Various neural network architectures that leverage attention and multi-scale connections could prove more optimal over the mid-latitudes

2. **Hyperparameter search** We note that the systematic hyperparameter sweep was executed on the precipitation-only pipeline using deterministic targets (SEEPS) and without IDR, due to training-time constraints; optimal settings for ERA5-augmented and probabilistic training may therefore differ, reinforcing the case for a dedicated search in the full probabilistic setup for more optimal performance.

3. **Training archive** Given the consistent lowering of CRPS with increase in training years (Figure 4.5) it naturally follows that when not subject to hardware limitations, the model can greatly benefit from a bigger training archive, possibly with a variety of data-sources.

4. **Re-shape the spatial taper:** (i) set Germany > 1 to increase the contrast; (ii) include the neighbor ring in the unit-weight core and apply a steeper radial decay beyond; (iii) use anisotropic tapers aligned with mean storm-track vectors (west–southwest inflow sectors).

5. **Condition weight on season or regime:** define w (or the entire mask) as a function of DJF/MAM/JJA/SON or synoptic regime (e.g., AR/front composites), reflecting the demonstrably different seasonal sensitivities.

Final Outook In a mid-latitude, baroclinic regime, modest gains accrue from (i) regional weighting during training (here, ~1.16% CRPS improvement at $w = 0.90$ vs. no weighting) and (ii) a physically motivated ERA5 stack (u, v, q; ~1.13%). These levers—together with longer training archives and hyperparameter optimization targeted at the probabilistic pipeline—point to clear, meteorologically grounded pathways for further skill.

Bibliography

[1] Georgy Ayzel et al. “RainNet: A Deep Convolutional Neural Network for Radar-Based Precipitation Nowcasting”. In: *Environmental Modelling and Software* (2020). DOI: 10.1016/j.envsoft.2020.104815. URL: https://doi.org/10.1016/j.envsoft.2020.104815.

[2] Peter Bauer, Alan Thorpe, and Gilbert Brunet. “The quiet revolution of numerical weather prediction”. In: *Nature* 525 (2015), pp. 47–55. DOI: 10.1038/nature14956.

[3] Hylke E. Beck et al. “MSWEP V2 global 3-hourly 0.1° precipitation: methodology and quantitative assessment”. In: *Bulletin of the American Meteorological Society* 100.3 (2019), pp. 473–500. DOI: 10.1175/BAMS-D-17-0138.1. URL: https://doi.org/10.1175/BAMS-D-17-0138.1.

[4] Hylke E. Beck et al. “MSWEP: 3-hourly 0.25° global gridded precipitation (1979–2015) by merging gauge, satellite, and reanalysis data”. In: *Hydrology and Earth System Sciences* 21 (2017), pp. 589–615. DOI: 10.5194/hess-21-589-2017. URL: https://doi.org/10.5194/hess-21-589-2017.

[5] Kaifeng Bi et al. “Accurate medium-range global weather forecasting with 3D neural networks”. In: *Nature* 619.7970 (2023), pp. 533–538. DOI: 10.1038/s41586-023-06185-3.

[6] T. J. Catto and S. Pfahl. “The Importance of Fronts for Extreme Precipitation in Europe”. In: *Geophysical Research Letters* (2013). DOI: 10.1002/grl.50852. URL: https://doi.org/10.1002/grl.50852.

[7] Imme Ebert-Uphoff and Kyle Hilburn. “The outlook for AI weather prediction”. In: *Nature* (2023). DOI: 10.1038/d41586-023-02084-9. URL: https://doi.org/10.1038/d41586-023-02084-9.

[8] ECMWF. *Quality of our forecasts*. https://www.ecmwf.int/en/forecasts/quality-our-forecasts. Public content CC BY 4.0 unless otherwise stated. 2024.

[9] Andreas H. Fink, Susanne Pohle, and co-editors. *Meteorology of Tropical West Africa: The Forecasters’ Handbook*. Springer, 2017. DOI: 10.1007/978-3-319-47340-2. URL: https://doi.org/10.1007/978-3-319-47340-2.

[10] Estíbaliz Gascón et al. “Statistical postprocessing of dual-resolution ensemble precipitation forecasts across Europe”. In: *Quarterly Journal of the Royal Meteorological Society* 145.724 (2019), pp. 3218–3235. DOI: 10.1002/qj.3615.

[11] Alexander Henzi, Johanna Ziegel, and Tilmann Gneiting. “Isotonic Distributional Regression”. In: *Journal of the Royal Statistical Society: Series B* (2021). DOI: 10.1111/rssb.12464. URL: https://doi.org/10.1111/rssb.12464.

[12] Hans Hersbach et al. “The ERA5 Global Reanalysis”. In: *Quarterly Journal of the Royal Meteorological Society* 146.730 (2020), pp. 1999–2049. DOI: 10.1002/qj.3803. URL: https://doi.org/10.1002/qj.3803.

[13] James R. Holton and Gregory J. Hakim. *An Introduction to Dynamic Meteorology*. 5th ed. Academic Press, 2013. DOI: 10.1016/C2010-0-64866-3.

[14] Matthias Karlbauer et al. “Advancing Parsimonious Deep Learning Weather Prediction Using the HEALPix Mesh”. In: *Journal of Advances in Modeling Earth Systems* (2024). DOI: 10.1029/2023MS004021. URL: https://doi.org/10.1029/2023MS004021.

[15] Remi Lam et al. “Learning skillful medium-range global weather forecasting”. In: *Science* 382.6677 (2023), pp. 1416–1421. DOI: 10.1126/science.adi2336.

[16] David A. Lavers and Gabriele Villarini. “The Nexus between Atmospheric Rivers and Extreme Precipitation across Europe”. In: *Geophysical Research Letters* (2013). DOI: 10.1002/grl.50636. URL: https://doi.org/10.1002/grl.50636.

[17] Paul Markowski and Yvette Richardson. *Mesoscale Meteorology in Midlatitudes*. Wiley-Blackwell, 2010. DOI: 10.1002/9780470682104.

[18] Monika Messmer, Juan J. Gomez-Navarro, and Christoph C. Raible. “Climatology of Vb Cyclones, Their Tracks and Precipitation in Central Europe from a Multidecadal Perspective”. In: *Quarterly Journal of the Royal Meteorological Society* 141.689 (2015), pp. 1136–1147. DOI: 10.1002/qj.2417. URL: https://doi.org/10.1002/qj.2417.

[19] Stephen W. Nesbitt and Edward J. Zipser. “The Diurnal Cycle of Rainfall and Convective Intensity According to Three Years of TRMM Measurements”. In: *Journal of Climate* (2006). DOI: 10.1175/JCLI3923.1. URL: https://doi.org/10.1175/JCLI3923.1.

[20] Stefan Pfahl and Michael Sprenger. “On the Relationship between Extratropical Cyclones and Strong Jet-Related Forcing for Ascent”. In: *Geophysical Research Letters* (2016). DOI: 10.1002/2016GL068018. URL: https://doi.org/10.1002/2016GL068018.

[21] Stephan Pfahl. “Characterising the Relationship Between Weather Extremes in Europe and Synoptic Circulation Features”. In: *Natural Hazards and Earth System Sciences* 14 (2014), pp. 1461–1475. DOI: 10.5194/nhess-14-1461-2014. URL: https://nhess.copernicus.org/articles/14/1461/2014/.

[22] Monika Rauthe et al. “A Central European precipitation climatology – Part I: generation and validation of a high-resolution gridded daily data set (HYRAS)”. In: *Meteorologische Zeitschrift* 22.3 (2013), pp. 235–256. DOI: 10.1127/0941-2948/2013/0436. URL: https://doi.org/10.1127/0941-2948/2013/0436.

[23] Suman Ravuri et al. “Skilful precipitation nowcasting using deep generative models of radar”. In: *Nature* 597 (2021), pp. 672–677. DOI: 10.1038/s41586-021-03854-z. URL: https://doi.org/10.1038/s41586-021-03854-z.

[24] Hannah Ritchie. “Weather forecasts have become much more accurate; we now need to make them available to everyone”. In: *Our World in Data* (2024). URL: https://ourworldindata.org/weather-forecasts.

[25] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Proceedings of MICCAI*. 2015. DOI: 10.1007/978-3-319-24574-4_28. URL: https://doi.org/10.1007/978-3-319-24574-4_28.

[26] Ian Simmonds and Alexandre D. Michelsen. "Extratropical Cyclone Dynamics, Climatology and Impacts: A Review". In: *Tellus A* (2022). DOI: 10.1080/16000870.2022.2024321. URL: https://doi.org/10.1080/16000870.2022.2024321.

[27] Uwe Ulbrich, Gregor C. Leckebusch, and Joaquim G. Pinto. "Extra-tropical cyclones in the present and future climate: a review". In: *Theoretical and Applied Climatology* 96.1–2 (2009), pp. 117–131. DOI: 10.1007/s00704-008-0083-8. URL: https://link.springer.com/article/10.1007/s00704-008-0083-8.

[28] Peter Vogel et al. "Skill of Global Raw and Postprocessed Ensemble Predictions of Rainfall in the Tropics". In: *Weather and Forecasting* (2020). DOI: 10.1175/WAF-D-20-0082.1. URL: https://doi.org/10.1175/WAF-D-20-0082.1.

[29] Eva-Maria Walz et al. "Easy Uncertainty Quantification: Generating Predictive Distributions from Single-Valued Model Output". In: *SIAM Review* (2024). DOI: 10.1137/22M1510272. URL: https://doi.org/10.1137/22M1510272.

[30] Eva-Maria Walz et al. "Physics-Based vs Data-Driven 24-Hour Probabilistic Forecasts of Precipitation for Northern Tropical Africa". In: *Monthly Weather Review* (2024). DOI: 10.1175/MWR-D-24-0005.1. URL: https://doi.org/10.1175/MWR-D-24-0005.1.

[31] Jonathan Weyn, Dale Durran, and Rich Caruana. "Improving Data-Driven Global Weather Prediction Using Deep Convolutional Neural Networks on a Cubed Sphere". In: *Journal of Advances in Modeling Earth Systems* (2020). DOI: 10.1029/2020MS002109. URL: https://doi.org/10.1029/2020MS002109.

[32] Yuchen Zhang et al. "Skilful nowcasting of extreme precipitation with NowcastNet". In: *Nature* 619 (2023), pp. 526–532. DOI: 10.1038/s41586-023-06184-4. URL: https://doi.org/10.1038/s41586-023-06184-4.

A. Appendix

A.1. Detailed overview Encoder and Decoder

Contracting Path (Encoder) We tailored the encoder to our data shape. A challenge is that our grid is significantly wider (121 longitudes) than it is tall (41 latitudes). If we did symmetric pooling in both directions, after 4 levels the dimension 41 becomes about 3, and 121 becomes about 7. A non-square aspect ratio remains. We handled this by using a slightly asymmetric stride in the first pooling to adjust for the aspect ratio. Specifically, in the first downsampling, we used a pooling kernel of 2×3 (lat × lon) to reduce 41→20 and 121→40 (approximately). This is an uncommon trick, but it helped make the feature maps more balanced in size across levels. The subsequent poolings were 2×2 as usual. In the convolution layers themselves, we kept kernels 3×3 but padded appropriately at edges.

At each level, the number of feature filters grows:

- Level 1: C input channels are transformed to 64 feature maps after the first convolution block. (If $C < 64$, this is an expansion; if $C > 64$, the network will actually reduce dimensionality in first layer.)
- Level 2: 128 feature maps.
- Level 3: 256 feature maps.
- Level 4: 512 feature maps.
- Bottleneck: 512 feature maps (two convs here as well).

These choices were guided by standard U-Net configurations and some experimentation. We found 64 as a base width was sufficient; increasing to 128 didn't significantly improve skill but did slow training. Using fewer (32 base) hurt performance, likely because the model couldn't capture all complexity of the data (especially when many channels were used).

Each convolution in the encoder is followed by:
- Instance normalization (IN) instead of batch normalization. We chose IN because precipitation patterns can vary strongly in mean and variance across spatial locations (mountains vs plains) and across time; IN normalizes each sample and each channel independently of others, which tends to preserve spatial structures better for segmentation tasks [25]. In our case, it also means the model's convolution filters don't have to account for varying overall intensity in different samples—IN removes that, focusing on the pattern shape. We use IN with learnable affine parameters (so the network can rescale/shift back if needed).
- LeakyReLU activation (with a small negative slope $\alpha = 0.01$). We opted for LeakyReLU

over standard ReLU to avoid "dead" neurons that could occur given the sparse nature of precipitation data (ReLU could zero out whole feature maps if they get negative inputs often). The leaky slope is small enough not to affect positive values but prevents zero gradients for negative inputs.
- Dropout (spatial dropout) with probability $p = 0.2$ is applied after each convolution in the encoder (except maybe the first layer or so for stability). Spatial dropout, also known as feature map dropout, drops entire 2D feature maps at once (as opposed to independent pixels), which is more appropriate for convolutional features to avoid introducing unrealistic noise patterns. This regularizes the network and forces redundancy in feature encoding (so that not one single feature map is solely responsible for an important detail).

The max pooling layers do not have learnable parameters; they simply downsample by taking the max in non-overlapping regions. We ensured to handle boundaries by pooling with the appropriate kernel size (for the non-square initial pooling, we used padding or a slightly different window at the edge to cover the 41x121 properly).

Expanding Path (Decoder) The decoder mirrors the encoder. For each level from 4 down to 1, we perform:
- Upsampling of the feature map (using a 2×2 transposed convolution, a.k.a. deconvolution, with stride 2) to double the spatial dimensions. This learns how to interpolate the coarse features into finer resolution. We set the number of filters in each up-conv to half of the number of filters in the encoder at that level (e.g., going into level 3 decoder, we use 256 filters).
- Concatenation with the corresponding encoder feature map (skip connection). We actually experimented with a weighted skip connection where the encoder features are scaled by a learnable factor α before concatenation, to let the model decide how much encoder detail to use. In practice, we found α tended to remain near 1 (meaning it used the encoder features fully), so one can assume standard concatenation. We kept this mechanism in code for flexibility, but it did not qualitatively change results.
- Two 3×3 convolutions (with IN + LeakyReLU + Dropout as in encoder). These gradually refine the upsampled image using both the coarse context and the fine details from the encoder.

By the end of the decoder, at level 1, we have 64 feature maps of size 41×121. We then apply the final 1×1 convolution to reduce 64 to 1 feature map, which is the predicted value at each grid cell.

One challenge in our decoder was the non-integer upscale for some dimensions due to the initial asymmetric pooling. For example, after level 4, we had a 3×8 bottleneck. Upsampling that with 2×2 stride gives 6×16; concatenating with level 3's encoder feature (which was 5×15) requires alignment. We addressed this by cropping or padding the decoder features to match the encoder size. This is a common step in U-Net implementations because some dimensions may not divide evenly by 2 multiple times, the encoder feature may be one pixel larger than the upsampled decoder feature. We simply cropped the encoder feature

(or padded the decoder feature) by at most 1 pixel to ensure they align. This had negligible effect on results but is important to implement to avoid size mismatch.

A.2. Network Components and Regularization

We have already touched on some of the network components (InstanceNorm, Dropout, LeakyReLU) in context, but here we summarize the key choices and why they were made, in terms of regularization and stability.

Instance Normalization (IN) We opted for instance normalization layers after each convolution instead of the more commonly used batch normalization (BN). BN normalizes across the batch dimension and spatial dims, which is very effective in vision tasks with large and varied datasets. However, in our case, each sample is a full meteorological field with its own distribution, and batch sizes were relatively small (due to memory). BN could cause issues when batch size is small (the estimated mean/var might be noisy). More importantly, BN would mix information across different samples (days), which might not be desirable if different days have systematically different distributions (for example, summer vs winter).

Instance normalization normalizes each sample (each feature map) independently:

$$\mathrm{IN}(x_c) = \frac{x_c - \mu_c}{\sigma_c},$$

where μ_c, σ_c are the mean and std of feature map c for the current sample (averaged over spatial positions). This is similar to the concept of "per-image normalization" in style transfer, which IN was originally used for. In segmentation and generation tasks, IN helps remove instance-specific contrast while retaining structure. In our use, IN effectively says: "for each channel of features, remove the spatial mean and normalize variance for each sample." It can be interpreted as making the conv layers focus on spatial patterns rather than absolute magnitudes, since magnitude can be reintroduced by the affine parameters after normalization.

Meteorologically, using IN means the model might ignore absolute baseline differences between samples (like one day has a domain-wide wet bias because it's a generally rainy day) and focus on spatial anomalies (like where within the domain it's raining relative to the average of that day). We found that this improved training stability and convergence. A potential downside is losing global context (e.g., distinguishing a day where whole domain is wet vs whole domain is dry). However, the network can learn to encode that through multiple feature maps or through the affine parameters of IN. Also, the seasonal features and pressure patterns provide some global context.

We included learnable scale and bias in IN (as is standard), so the model can undo normalization if it finds it necessary for any channel.

Dropout Regularization Overfitting is a concern given the moderate dataset size (4.7k training days) and model capacity (7 million parameters). We used dropout to mitigate this. Specifically, we used spatial dropout (implemented as ‘nn.Dropout2d‘ in PyTorch) with $p = 0.2$ in most conv layers. Spatial dropout zeros out entire feature maps in a layer's output with probability 0.2. This means in each forward pass, 20% of the feature channels are removed (set to zero) uniformly at random. This forces the network to not rely solely on any one feature map; others must carry redundant information.

We applied dropout after the activation (and normalization) in each conv block, which is a typical placement. We did not apply dropout in the final output layer.

We experimented with $p = 0.3$ and $p = 0.1$; 0.3 gave slightly higher training error and slightly lower validation error (sign of stronger regularization), but 0.2 was a good middle ground. At 0.1, regularization was weaker and we saw a bit more overfit in later epochs.

Another form of regularization present inherently is the data augmentation (discussed later) and weight decay in optimizer, but dropout was the primary method to prevent the network from simply memorizing patterns (which it can't fully anyway due to the chaotic nature of weather, but could still over-adjust to specific years).

Activation Functions We chose LeakyReLU with $\alpha = 0.01$ as the activation after each convolution. ReLU (with $\alpha = 0$ for negative side) is very common, but we had a concern: precipitation data has many zero regions (especially on some days, e.g., dry days) which can lead to many zeros in feature maps, and subsequent layers might get zero input a lot. ReLU could exacerbate dying neurons. LeakyReLU allows a small gradient when the input is negative (0.01 times the input). This helps keep neurons "alive" even if initial weights produce negative outputs.

The choice $\alpha = 0.01$ is conventional. We didn't tune it heavily; something like 0.1 would let negative information through more strongly, which we didn't want, and 0.001 would be almost like ReLU.

In practice, using LeakyReLU gave slightly better validation results than plain ReLU in our tests. It's also computationally negligible overhead.

We considered other activations like ELU or GELU, but LeakyReLU was simpler and effective. Also, since we normalize inputs via IN, the distribution in each channel might be centered around 0, making symmetric activations like tanh less appropriate (also they saturate, which we avoid).

All hidden layers use LeakyReLU. The final output layer uses no activation because for regression we typically leave it linear (the loss can handle it). If we were predicting in original mm space and wanted to ensure nonnegativity, we could put a ReLU at the end. But since we predict in log space (if using log), that isn't needed. If we weren't using log transform, it might make sense to have a ReLU at the end to enforce nonnegative rain—however, we avoided that because it complicates training (ReLU at end could zero out gradient if prediction overshoots negative, plus we had other means to ensure physical output in post-processing).

A.3. Spatial Resolution Handling

Non-Square Grid Adaptation As noted, our input grid is 41×121, which is a 1:2.95 aspect ratio. CNNs themselves are agnostic to aspect ratio, but the pooling operations can lead to uneven downsampling if not considered. We addressed this partly by the initial pooling adjustment (using a 2×3 pool to reduce more in lon direction early). Another consideration is convolution stride. We generally used stride 1 in conv layers to preserve spatial dimensions within each level. Only pooling (or the one asymmetric stride at start) reduces dims. We also used appropriate padding to maintain output shape = input shape for convs.

Another measure: to avoid artifacts from wrapping around longitudinally (since our domain is not global, but it is quite wide), we did not treat the data as cyclic in longitude. We pad with zeros or replicate border for convolution at edges. Thus, at 70°W and 50°E edges, the conv sees a padded area outside domain. This might cause a slight decrease in accuracy near the very edges (since the conv can't get context beyond boundary), but those areas (far west or far east of domain) are anyway not our focus (and are down-weighted in loss if far outside Germany). We accept that. An alternative could have been to wrap around (like a periodic boundary at edges), but that would be physically incorrect (50°E does not wrap to 70°W in reality, as that would connect unrelated regions across the Pacific given our domain cut).

We ensured that after each pooling, the feature map sizes are: - After level1 pool: (41/2=20.5 $\rightarrow$ 21, 121/340.3 $\rightarrow$ 40 or 41). We actually got 21×41 if we used ceil on both. But we opted to go to 20×40 or 21×41? There was a small decision. We ended up with 21×41 after first pool (because we used floor for lat (20) and floor for lon (40) plus maybe one padded row/col to make it even). - After level2: about 10×20. - After level3: 5×10. - After level4: 3×5 (with some padding leading to 3×8 we had earlier). - Bottleneck 3×8.

We verified that the upsampling reversed these to get back to 41×121. The cropping we did in skip connections was minimal (like removing an extra row/col if present after upsampling).

The unusual aspect ratio did not pose major issues beyond that. The model can still use wide conv filters to cover longitudinal spread.

Output Resolution Recovery The final output of the U-Net has shape 41×121, exactly matching the input grid, due to the symmetric architecture. We used a transposed convolution (also called convolutional upsampling) for upsampling steps. One nuance with transposed conv is that it can sometimes introduce a checkerboard artifact if not carefully set. We addressed that by using kernel size divisible by stride and by using LeakyReLU after it which tends to smooth it out. Also, some frameworks prefer bilinear upsampling followed by conv to avoid that; we found transposed conv fine.

The predicted field is thus aligned grid cell to grid cell with the target precipitation. Because we train on log or scaled precipitation, the raw network output is $\hat{y}_{\log}(i, j)$ (if log used). We later exponentiate and subtract epsilon to compare with original $y_{\text{orig}}(i, j)$.

We highlight that the network does not explicitly enforce any constraints like nonnegativity or conservation. It purely learns to approximate the mapping from inputs to outputs by minimizing error. This means it could, in theory, output negative values in linear space or create artifacts. In practice, with log scaling and enough training data, it learns not to predict implausible negatives. On a few occasions we saw slight negative predictions in very low-rain situations (like -0.1 mm), which we clamp to 0. Also, the network might not inherently conserve water (if there was such a concept) across domain, but that's fine for a prediction task.

Maintaining the full resolution is crucial for our probabilistic post-processing step (IDR), which operates gridpoint-wise on residuals. If we had downscaled output (like predicting on a coarser grid), we'd lose spatial detail needed for evaluation. The U-Net structure ensures we output on the original resolution with the benefit of multi-scale learned features.

A.4. Data Augmentation and Invariance

Gaussian Blur Augmentation To make the model more robust to small spatial shifts or noise and to encourage it not to overly rely on exact pixel-level alignments, we introduced a simple data augmentation: random Gaussian blur on the input fields. During training, with 50% probability for each sample, we applied a Gaussian filter to all input maps (precip lags and ERA5) with a randomly chosen standard deviation σ between 0.1 and 2.0 degrees (since our grid spacing is 1°, $\sigma = 2$ means quite a heavy blur mixing across a few grid cells). This blur is applied channel-wise (the same blur to all channels to not disrupt their relationships). The effect is that occasionally the model sees a slightly smoothed version of the data.

This augmentation can improve the model's ability to handle slightly misaligned patterns. For example, if a rain band in ERA5 predictor is off by one grid cell relative to where it causes rain in MSWEP, a model without augmentation might get confused unless it sees many examples. Blur augmentation effectively says "don't be too sensitive to one-grid-cell differences; learn the broader pattern." We found that models with blur augmentation were a bit more stable and had slightly lower validation error, especially when verifying precipitation structures (they didn't predict spotty "grid-point" rain as much, they predicted smoother fields more consistent with observations).

We avoided other augmentations that are common in vision (like flips or rotations), because flipping a meteorological field left-right (east-west) or up-down is not physically meaningful (would correspond to mirroring the map, which changes geography). Rotation by 180° would scramble the map relative to true lat-lon. We did not want to violate physical coordinate meaning. Perhaps shifting could be considered (cyclic shift in lon?), but because our domain isn't the whole globe, that wouldn't hold. So blur was a safe augmentation that respects that patterns could be slightly more diffuse or shifted.

The augmentation is only applied during training batches, not during validation/test. We implement it by convolving the input tensor with a Gaussian kernel (we used a small 5×5

or 7×7 kernel approximating the desired sigma, normalized to sum to 1). The cost overhead was negligible relative to model forward pass.