GRAPH NEURAL NETWORKS FOR ENHANCING ENSEMBLE FORECASTS OF EXTREME RAINFALL

Christopher Bülte*, **Sohir Maskey, Philipp Scholl, Jonas von Berg**
Ludwig-Maximilians-Universität München
Munich Center for Machine Learning (MCML)
Munich, Germany
{buelte, maskey, scholl, berg}@math.lmu.de

Gitta Kutyniok
Ludwig-Maximilians-Universität München
University of Tromsø
DLR-German Aerospace Center
Munich Center for Machine Learning (MCML)
Munich, Germany
kutyniok@math.lmu.de

ABSTRACT

Climate change is increasing the occurrence of extreme precipitation events, threatening infrastructure, agriculture, and public safety. Ensemble prediction systems provide probabilistic forecasts but exhibit biases and difficulties in capturing extreme weather. While post-processing techniques aim to enhance forecast accuracy, they rarely focus on precipitation, which exhibits complex spatial dependencies and tail behavior. Our novel framework leverages graph neural networks to post-process ensemble forecasts, specifically modeling the extremes of the underlying distribution. This allows to capture spatial dependencies and improves forecast accuracy for extreme events, thus leading to more reliable forecasts and mitigating risks of extreme precipitation and flooding. [1]

1 INTRODUCTION

The increasing impacts of climate change have led to more frequent and severe extreme precipitation events, posing significant risks to infrastructure, agriculture, and public safety (Trenberth, 2011). Accurate and well-calibrated precipitation forecasts are critical for mitigating these risks, especially concerning important applications such as disaster management, urban planning, and water resource management (IPCC, 2021). Ensemble prediction systems (EPS) have become a cornerstone of modern weather forecasting, generating probabilistic predictions by running numerical weather predictions (NWPs) under varied initial conditions. Despite their widespread use, ensemble forecasts often exhibit biases, a lack of sharpness, and difficulty capturing the extreme tail behavior of precipitation distributions, which limits their utility for decision-making under the increased risks associated with climate change (Tabari, 2020; Trenberth, 2011).

Post-processing techniques have been developed to address the limitations of raw ensemble forecasts by refining them into more accurate probabilistic predictions. Statistical methods such as ensemble model output statistics (Gneiting et al., 2005), random forests (Muschinski et al., 2023), or nonparametric regression (Bremnes, 2019) have been widely used to improve the calibration and sharpness of ensemble forecasts. More recently, neural-network-based post-processing has shown promise by leveraging machine learning to learn high-dimensional relationships directly from data (Rasp & Lerch, 2018; Schulz & Lerch, 2022). These post-processing approaches have been extended to

*Corresponding author.
[1] Our code is available on GitHub.

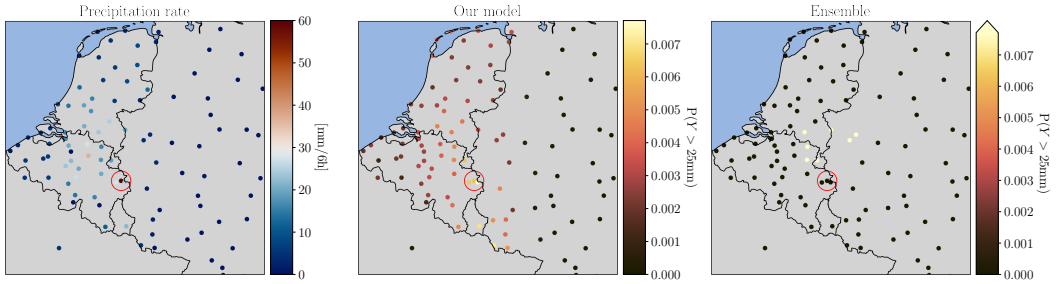

Figure 1: The left panel shows the precipitation rates on April 29, 2018 with a highlighted extreme precipitation occurrence of about 70mm over 6 hours. The two right plots show the threshold exceedance probability $\mathbb{P}[Y > 25\text{mm}]$ for our model and the ensemble prediction. In contrast to the ensemble, our model assigns a nonzero probability (0.63%) to the corresponding event.

convolutional (Horat & Lerch, 2024) and graph neural networks (Feik et al., 2024) and to post-processing of neural network prediction systems (Bülte et al., 2025). While these approaches are effective for general forecasting tasks, they often fail to capture the complex spatial dependencies and heavy-tailed characteristics of precipitation data, particularly during extreme weather events.

In this work, we introduce a novel framework for improving precipitation forecasts by post-processing ensemble predictions. Our method addresses key issues in forecasting extremes by explicitly accounting for the frequent occurrence of dry periods with a point mass and using a generalized Pareto distribution to capture the tail behavior associated with heavy rainfall. To enhance spatial accuracy, we employ graph neural networks (GNNs), which represent weather stations and ensemble forecasts with regards to their spatial dependence structure. This graph-based approach improves the model's ability to identify patterns and dependencies in extreme events that often span across regions (Feik et al., 2024). We validate our framework on a benchmark dataset for ensemble post-processing methods in medium-range weather forecasting.

2 Data

To compare to existing methods, we utilize the EUPPBench dataset, a benchmark dataset for ensemble post-processing (Demaeyer et al., 2023). The dataset comprises 122 weather stations across Europe and includes medium-range ensemble forecasts from the European Centre for Medium-Range Weather Forecasts (ECMWF) and corresponding station observations, spanning from 1997 to 2018. The dataset includes both, typical forecasts, but also reforecasts, which are numerical weather prediction (NWP) models run for past dates. In total, EUPPBench includes 730 daily operational forecasts with 51 ensemble members and 4180 reforecasts with 11 ensemble members and a total of 31 variables each (compare Demaeyer et al., 2023). We follow the setup in Feik et al. (2024), where the model is trained on reforecast data from 1997-2013 and evaluated on reforecasts from 2014-2017, as well as on forecast data from 2017-2018. For modeling precipitation, we focus on predicting the *TP6* variable, the total precipitation in mm accumulated over six hours.

3 Methodology

We base our approach on a distributional regression network (DRN) (Rasp & Lerch, 2018), a benchmark for station-based post-processing, that has been successfully applied to various domains, such as wind gusts (Schulz & Lerch, 2022) or atmospheric rivers (Chapman et al., 2022). The central idea is to use a neural network that outputs the parameters of a specified predictive distribution. The model is then trained by minimizing the Continuous Ranked Probability Score (CRPS), defined as $\text{CRPS}(F, y) := \int_{-\infty}^{\infty} (F(x) - \mathbb{1}_{y \leq x})^2 \, dx$, where F denotes the cumulative distribution function and y denotes the realized outcome (Gneiting & Katzfuss, 2014). Similar to Feik et al. (2024), we employ a graph neural network to account for spatial dependencies and in addition choose a predictive distribution specifically designed to account for the characteristics of precipitation data.

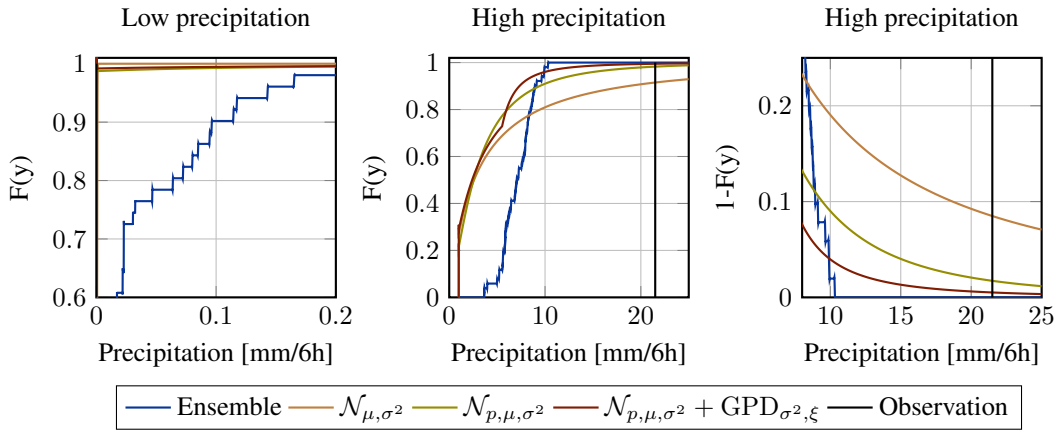

Figure 2: The figure compares a sample prediction of the different modeling methods. In our proposed approach, the low precipitation scenarios are modeled via a discrete point mass that accounts for the event of no precipitation, while the GPD models the tail behavior over a certain threshold.

3.1 Probabilistic Precipitation Modeling

While precipitation is one of the most critical meteorological variables, as its extremes have significant environmental and societal impacts, its highly variable nature and complex spatiotemporal characteristics make forecasting difficult. A dominant modeling approach in the literature is based on the mixed lognormal distribution (Kedem et al., 1990; Cho et al., 2004), which incorporates an additional point mass at zero with probability p, accounting for instances of no precipitation. A lognormal distribution describes a random variable Y such that its logarithm, $\log(Y)$, follows a normal distribution. While this has shown to perform well in practice (Kedem et al., 1990; Syed Jamaludin et al., 2011), the underlying data is heavily skewed, making neural network training difficult. Therefore, we propose to transform the underlying data space by applying a log-transform to the data, i.e., $h(y) = \log(y + \varepsilon)$, where we add $\varepsilon > 0$ for numerical stability (see Appendix B for more details). By definition, the transformed precipitation can now be modeled with a normal distribution with a discrete point mass at the left endpoint. As this distribution only accounts for the lower extreme of no precipitation, we further employ the Peaks-Over-Threshold approach to account for extreme high precipitation. By the Pickands-Balkema-De Haan theorem (Pickands, 1975), we know that under some regularity conditions, the excess function $\mathbb{P}(Y \leq x + u \mid Y > u)$ for a threshold u, converges to the generalized Pareto distribution (GPD). This allows to explicitly model the upper tails of the underlying distribution. To sum up, we model the (shifted and log-transformed) precipitation as

$$F_Y(y) := \begin{cases} 0, & y < c \\ \tilde{F}(y) := p + (1-p)\Phi_{\mu,\sigma^2}(y), & y \in [c, u] \\ \tilde{F}(u) + (1 - \tilde{F}(u))\text{GPD}_{u.\sigma_u^2,\xi}(y), & y > u, \end{cases} \tag{1}$$

where p denotes the discrete probability mass assigned to the left endpoint of the distribution $c := \log(\varepsilon)$. We then use the neural network to predict the parameters $\{p, \mu, \sigma^2, \sigma_u^2, \xi, u\}$ per individual station with various meteorological variables from the NWP ensemble as input. The optimal parameters are obtained by minimizing the CRPS(F_Y, y), for which a closed-form expression is given in Equation 11 in Appendix C.

3.2 Spatial Dependence Modeling with Graph Neural Networks

So far, the modeling has focused on station-specific features, with a single predictive distribution for each station. To apply graph-based learning methods, the data is transformed into a graph representation, where each station is treated as a node in the graph. Let N denote the total number of stations, and define the distance matrix $D \in \mathbb{R}^{N \times N}$ based on geodesic distances between stations. An edge is created between nodes i and j if $D_{ij} \leq d_{\text{max}}$, where d_{max} is a predefined distance threshold. To incorporate ensemble forecasts, every node is associated with a $n_{\text{ens}} \times F$-dimensional feature matrix, where n_{ens} is the number of ensemble members and F denotes the corresponding number of

Model	24h			72			120h		
Metric	CRPS	Brier	$QS_{0.99}$	CRPS	Brier	$QS_{0.99}$	CRPS	Brier	$QS_{0.99}$
ENS	0.662	0.180	0.108	0.699	0.180	0.106	0.797	0.200	0.117
$\mathcal{N}_{\mu,\sigma^2}$	0.515	0.316	0.299	0.640	0.384	0.381	0.782	0.337	0.558
$\mathcal{N}_{p,\mu,\sigma^2}$	**0.467**	**0.092**	**0.077**	**0.569**	**0.114**	**0.092**	0.682	0.139	0.117
$\mathcal{N}$-GPD$_{\sigma_u^2}$	0.470	0.093	0.084	0.577	0.116	0.096	**0.678**	0.138	**0.113**
$\mathcal{N}$-GPD$_{u,\sigma_u^2}$	**0.467**	**0.092**	0.082	0.597	0.119	0.099	**0.678**	**0.137**	**0.113**

Table 1: The table shows the evaluation metrics for the different models and some selected lead times on the forecasting task. The best model is highlighted in bold.

meteorological features. Edge weights $w_{i,j}$ between nodes i and j are defined based on normalized geodesic distances, capturing spatial relationships between stations.

Since the ensemble members are interchangeable, the ensemble dimension introduces permutation symmetry. To account for this, the input features are embedded using a DeepSet (Zaheer et al., 2017), ensuring permutation invariance. Specifically, we compute the initial node embedding as $h_v^{(0)} = \Psi\left(\sum_{n=1}^{n_{\text{ens}}} \rho\left(x_{v,n}\right)\right)$, where $x_{v,n} \in \mathbb{R}^F$ is the node feature vector for station v and ensemble member n. Here, ρ and Ψ are both two-layer multilayer perceptrons (MLPs). This setup allows the model to aggregate ensemble features while respecting their permutation-invariant nature. Unlike Feik et al. (2024), we perform this step before the GNN to reduce the dimensionality. The GNN processes the graph with input node features $h_v^{(0)}$ by iteratively aggregating features from neighboring nodes. In addition, residual connections are applied to stabilize learning: $h^{(t)} = h^{(t-1)} + \sigma\left(\text{GNN}\left(h^{(t-1)}\right)\right)$, where $h^{(t)}$ represents the hidden representation at layer t, and σ is the ReLU activation function. Furthermore, we employ a Graph Isomorphism Network (GINE) (Xu et al., 2019; Hu* et al., 2020), which incorporates both node features and edge weights to effectively model interactions between neighboring nodes. After aggregation, the resulting features predict the station-specific parameters $\{p, \mu, \sigma^2, \sigma_u^2, \xi, u\}$. To ensure parameter constraints, we use a softplus activation for σ^2, σ_u^2, sigmoid activation for p and ξ, and linear activation for μ, u.

4 RESULTS

We evaluate our approach on the EUPPBench dataset, comparing it against three baselines: ensemble prediction (ENS), a normal distribution ($\mathcal{N}_{\mu,\sigma^2}$), and a normal distribution with a point mass ($\mathcal{N}_{p,\mu,\sigma^2}$). Implemented in PyTorch with early stopping (compare Appendix D), our method uses two threshold selection strategies: (1) a global threshold, u, set as the 90th percentile of the training data ($\mathcal{N}$-GPD$_{\sigma_u^2}$); (2) and learned station-specific thresholds, u_i ($\mathcal{N}$-GPD$_{u,\sigma_u^2}$). Due to numerical instability when optimizing the CRPS with respect to the GPD shape parameter, ξ, we fixed ξ at 0.5 after a small hyperparameter search. Performance is evaluated using the CRPS, the Brier score for the binary event of no rain (= precipitation less than $0.01mm/6h$), and the quantile score (QS) at $\alpha = 0.99$ for extreme precipitation. These metrics collectively evaluate the entire predictive distribution, including both the lower and upper tails.

Table 1 presents the results for the forecast task, with additional reforecast results available in the supplementary materials. The $\mathcal{N}_{p,\mu,\sigma^2}$ model performs well for most lead times, whereas the additional GPD modeling achieves similar performance and tends to outperform other methods at the 120h lead time. As the zero probability p accounts for much of the data, we cannot expect the metrics to be too different. Figure 2 visualizes the different probabilistic predictions for a high precipitation and zero precipitation event at a selected station, demonstrating a good fit between the predictive distribution and the realized event. In addition, Figure 1 shows a spatial visualization of a selected extreme precipitation event. In contrast to the ensemble prediction, our approach leads to a higher probability of the extreme event occurring. In addition, it correctly assigns low probability to sites that are spatially separated and where no precipitation occurred. Although more prominent for the lower tail, the above results suggest that our proposed approach can model the full range of precipitation, by explicitly considering the extremes on both ends of the support of the distribution.

5 CONCLUSION

We propose a predictive modeling framework for precipitation post-processing that focuses directly on the underlying extremes. By combining the modeling framework with a powerful graph neural network architecture, we can provide improvements in predictions regarding different baselines and with a focus on the extremes of the precipitation, allowing for more thorough prediction of extreme precipitation, mitigating risks of climate-change related events such as floods. Possible future research might revolve around combining our modeling approach with an end-to-end neural network model, such as Graphcast (Lam et al., 2023) or RainNet (Ayzel et al., 2020), to work with direct forecasting tasks. In addition, more detailed investigation of the extreme modeling with the GPD approach is required, especially with regards to the choice of the threshold u and the shape parameter ξ. Further improving the tail modeling provides an interesting direction of research, regarding analysis of precipitation extremes.

ACKNOWLEDGEMENTS

C. Bülte and G. Kutyniok acknowledge support by the DAAD programme Konrad Zuse Schools of Excellence in Artificial Intelligence, sponsored by the Federal Ministry of Education and Research.

S. Maskey acknowledges support by the NSF-Simons Research Collaboration on the Mathematical and Scientific Foundations of Deep Learning (MoDL) (NSF DMS 2031985).

P. Scholl and G. Kutyniok acknowledge support by the project "Genius Robot" (01IS24083), funded by the Federal Ministry of Education and Research (BMBF), as well as the ONE Munich Strategy Forum (LMU Munich, TU Munich, and the Bavarian Ministery for Science and Art).

J. Berg and G. Kutyniok acknowledge support by the gAIn project, which is funded by the Bavarian Ministry of Science and the Arts (StMWK Bayern) and the Saxon Ministry for Science, Culture and Tourism (SMWK Sachsen).

G. Kutyniok acknowledges partial support by the Munich Center for Machine Learning (BMBF), as well as the German Research Foundation under Grants DFG-SPP-2298, KU 1446/31-1 and KU 1446/32-1. Furthermore, G. Kutyniok is supported by LMUexcellent, funded by the Federal Ministry of Education and Research (BMBF) and the Free State of Bavaria under the Excellence Strategy of the Federal Government and the Länder as well as by the Hightech Agenda Bavaria.

REFERENCES

G. Ayzel, T. Scheffer, and M. Heistermann. Rainnet v1.0: a convolutional neural network for radar-based precipitation nowcasting. *Geoscientific Model Development*, 13(6):2631–2644, 2020. doi: 10.5194/gmd-13-2631-2020. URL https://gmd.copernicus.org/articles/13/2631/2020/.

John Bjørnar Bremnes. Ensemble postprocessing using quantile function regression based on neural networks and bernstein polynomials. *Monthly Weather Review*, 148, 10 2019. doi: 10.1175/MWR-D-19-0227.1.

Christopher Bülte, Nina Horat, Julian Quinting, and Sebastian Lerch. Uncertainty quantification for data-driven weather models. *Artificial Intelligence for the Earth Systems*, 2025. doi: 10.1175/AIES-D-24-0049.1. URL https://journals.ametsoc.org/view/journals/aies/aop/AIES-D-24-0049.1/AIES-D-24-0049.1.xml.

William E. Chapman, Luca Delle Monache, Stefano Alessandrini, Aneesh C. Subramanian, F. Martin Ralph, Shang-Ping Xie, Sebastian Lerch, and Negin Hayatbini. Probabilistic predictions from deterministic atmospheric river forecasts with deep learning. *Monthly Weather Review*, 150(1):215 – 234, 2022. doi: 10.1175/MWR-D-21-0106.1. URL https://journals.ametsoc.org/view/journals/mwre/150/1/MWR-D-21-0106.1.xml.

Hye-Kyung Cho, Kenneth P. Bowman, and Gerald R. North. A comparison of gamma and lognormal distributions for characterizing satellite rain rates from the tropical rainfall measuring mission. *Journal of Applied Meteorology*, 43(11):1586 – 1597, 2004. doi: 10.1175/

JAM2165.1. URL https://journals.ametsoc.org/view/journals/apme/43/11/jam2165.1.xml.

J. Demaeyer, J. Bhend, S. Lerch, C. Primo, B. Van Schaeybroeck, A. Atencia, Z. Ben Bouallègue, J. Chen, M. Dabernig, G. Evans, J. Faganeli Pucer, B. Hooper, N. Horat, D. Jobst, J. Merše, P. Mlakar, A. Möller, O. Mestre, M. Taillardat, and S. Vannitsem. The euppbench postprocessing benchmark dataset v1.0. *Earth System Science Data*, 15(6):2635–2653, 2023. doi: 10.5194/essd-15-2635-2023. URL https://essd.copernicus.org/articles/15/2635/2023/.

Moritz Feik, Sebastian Lerch, and Jan Stühmer. Graph neural networks and spatial information learning for post-processing ensemble weather forecasts, 2024. URL https://arxiv.org/abs/2407.11050.

Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

Tilmann Gneiting and Matthias Katzfuss. Probabilistic forecasting. *Annual Review of Statistics and Its Application*, 1(Volume 1, 2014):125–151, 2014. ISSN 2326-831X. doi: https://doi.org/10.1146/annurev-statistics-062713-085831. URL https://www.annualreviews.org/content/journals/10.1146/annurev-statistics-062713-085831.

Tilmann Gneiting, Adrian E. Raftery, Anton H. Westveld, and Tom Goldman. Calibrated probabilistic forecasting using ensemble model output statistics and minimum crps estimation. *Monthly Weather Review*, 133(5):1098 – 1118, 2005. doi: 10.1175/MWR2904.1. URL https://journals.ametsoc.org/view/journals/mwre/133/5/mwr2904.1.xml.

Nina Horat and Sebastian Lerch. Deep learning for postprocessing global probabilistic forecasts on subseasonal time scales. *Monthly Weather Review*, 152(3):667 – 687, 2024. doi: 10.1175/MWR-D-23-0150.1. URL https://journals.ametsoc.org/view/journals/mwre/152/3/MWR-D-23-0150.1.xml.

Weihua Hu*, Bowen Liu*, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HJlWWJSFDH.

IPCC. Summary for policymakers. *Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*, pp. 3–32, 2021. doi: 10.1017/9781009157896.001.

Alexander Jordan. *Facets of forecast evaluation*. PhD thesis, Karlsruher Institut für Technologie (KIT), 2016.

Alexander Jordan, Fabian Krüger, and Sebastian Lerch. Evaluating probabilistic forecasts with scoringrules, 2018. URL https://arxiv.org/abs/1709.04743.

Benjamin Kedem, Long Chiu, and Gerald North. Estimation of mean rain rate - application to satellite observations. *J. Geophys. Res*, 95, 03 1990. doi: 10.1029/JD095iD02p01965.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, Alexander Merose, Stephan Hoyer, George Holland, Oriol Vinyals, Jacklynn Stott, Alexander Pritzel, Shakir Mohamed, and Peter Battaglia. Learning skillful medium-range global weather forecasting. *Science*, 382(6677):1416–1421, 2023. doi: 10.1126/science.adi2336. URL https://www.science.org/doi/abs/10.1126/science.adi2336.

T. Muschinski, G. J. Mayr, A. Zeileis, and T. Simon. Robust weather-adaptive post-processing using model output statistics random forests. *Nonlinear Processes in Geophysics*, 30(4):503–514, 2023. doi: 10.5194/npg-30-503-2023.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

James Pickands. Statistical Inference Using Extreme Order Statistics. *The Annals of Statistics*, 3(1): 119 – 131, 1975. doi: 10.1214/aos/1176343003. URL https://doi.org/10.1214/aos/1176343003.

Stephan Rasp and Sebastian Lerch. Neural networks for postprocessing ensemble weather forecasts. *Monthly Weather Review*, 146(11):3885 – 3900, 2018. doi: 10.1175/MWR-D-18-0187.1. URL https://journals.ametsoc.org/view/journals/mwre/146/11/mwr-d-18-0187.1.xml.

Benedikt Schulz and Sebastian Lerch. Machine learning methods for postprocessing ensemble forecasts of wind gusts: A systematic comparison. *Monthly Weather Review*, 150(1):235 – 257, 2022. doi: 10.1175/MWR-D-21-0150.1. URL https://journals.ametsoc.org/view/journals/mwre/150/1/MWR-D-21-0150.1.xml.

Shariffah Suhaila Syed Jamaludin, Kong Ching-Yee, Fadhilah Yusof, and Foo Hui-Mean. Introducing the mixed distribution in fitting rainfall data. *Open Journal of Modern Hydrology*, 01, 01 2011. doi: 10.4236/ojmh.2011.12002.

Hossein Tabari. Climate change impact on flood and extreme precipitation increases with water availability. *Scientific Reports*, 10:13768, 08 2020. doi: 10.1038/s41598-020-70816-2.

Kevin Trenberth. Changes in precipitation with climate change. climate change research. *Climate Research*, 47:123–138, 03 2011. doi: 10.3354/cr00953.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=ryGs6iA5Km.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.

A ADDITIONAL RESULTS

Model	24h			72			120h		
Metric	CRPS	Brier	$QS_{0.99}$	CRPS	Brier	$QS_{0.99}$	CRPS	Brier	$QS_{0.99}$
ENS	0.757	0.196	0.208	0.854	0.216	0.221	0.965	0.233	0.270
$\mathcal{N}_{\mu,\sigma^2}$	0.585	0.319	0.376	0.735	0.385	0.507	0.915	0.341	0.804
$\mathcal{N}_{p,\mu,\sigma^2}$ mixed	**0.523**	0.103	0.089	**0.652**	**0.131**	0.105	**0.782**	0.158	0.120
$\mathcal{N}$-GPD$_{\sigma_u^2}$	0.530	0.104	**0.095**	0.656	0.132	**0.102**	0.786	0.159	0.133
$\mathcal{N}$-GPD$_{u,\sigma_u^2}$	0.524	**0.102**	0.096	0.676	0.136	0.117	**0.782**	**0.157**	**0.116**

Table 2: The table shows the evaluation metrics for the different models and some selected lead times on the reforecasting task. The best model is highlighted in bold.

B PRECIPITATION MODELING

Let Y denote the precipitation variable in the unit $[mm/6h]$. To simplify the training procedure, we apply two successive transformations to the data. First, we shift our data by a small constant value $\varepsilon > 0$[2]. We model this shifted variable with a mixture distribution of the following form:

$$F_Y(y) \coloneqq p + (1-p)\text{LN}_{\mu,\sigma^2}(y), \quad \varepsilon \leq y. \tag{2}$$

Here, $p \in [0,1]$ is the probability of the point mass at ε and LN denotes the log-normal distribution, defined as

$$\text{LN}_{\mu,\sigma^2}(x) \coloneqq \mathcal{N}_{\mu,\sigma^2}(\log(x)) = \Phi\left(\frac{\log(x)-\mu}{\sigma}\right). \tag{3}$$

with Φ the cumulative distribution function (CDF) of a centered normal distribution. The combined distribution in Equation 2 is also known as the mixed lognormal distribution (Kedem et al., 1990) and has been commonly applied to precipitation modeling (Kedem et al., 1990; Cho et al., 2004).

Log-transform To remove skewness from the data, we perform a second transformation $\tilde{Y} = g(Y) \coloneqq \log(Y)$, $Y \sim F_Y$. Note, for numerical stability, it is crucial that we have introduced the threshold $\varepsilon > 0$, thereby avoiding the singularity at $\log(0)$. We now want to analyze the resulting distribution $F_{\tilde{Y}}(y)$. Using the cumulative distribution function (CDF), we obtain:

$$F_{\tilde{Y}}(y) = P\{\tilde{Y} \leq y\} = P\{g(Y) \leq y\} = P\{X \leq g^{-1}(y)\} = F_Y(g^{-1}(y)).$$

Since $g^{-1}(y) = e^y$, we obtain for our transformed distribution:

$$F_{\tilde{Y}}(y) \coloneqq p + (1-p)\mathcal{N}_{\mu,\sigma^2}(y) \quad \log(\varepsilon) \leq y. \tag{4}$$

Modeling Pareto in transformed space To model the upper tails of the distribution, we utilize the Pickands-Balkema-de Haan theorem (Pickands, 1975), which states that, under some regularity assumptions, the excess distribution of a random variable $\mathbb{P}(\tilde{Y} \leq x + u \mid \tilde{Y} > u)$ for a certain threshold u can be approximated by a generalized Pareto distribution (GPD$_{u,\sigma_u^2,\xi}$), defined as

$$\text{GPD}_{u,\sigma_u,\xi}(x) \coloneqq \begin{cases} 1 - \left(1 + \frac{\xi(x-\mu)}{\sigma}\right)^{-1/\xi}, & \xi \neq 0, \\ 1 - \exp\left(-\frac{x-\mu}{\sigma}\right), & \xi = 0. \end{cases} \tag{5}$$

For our modeling, this leads to the final definition of the CDF as

$$F_{\tilde{Y}}(y) \coloneqq \begin{cases} \tilde{F}(y) \coloneqq p + (1-p)\mathcal{N}_{\mu,\sigma^2}(y), & \log(\varepsilon) \leq y \leq u, \\ \tilde{F}(u) + (1-\tilde{F}(u))\text{GPD}_{u,\sigma_u,\xi}(y), & y > u. \end{cases} \tag{6}$$

[2]For our experiments we chose $\varepsilon = 0.01$.

C CRPS OF THE MIXTURE DISTRIBUTION

As a loss function, we want to utilize the continuous ranked probability score (CRPS), which is a proper scoring rule and, therefore, can be used to measure the distance between a predictive distribution and a single data point Gneiting & Katzfuss (2014). It is defined as

$$\text{CRPS}(F, y) \coloneqq \int_{-\infty}^{\infty} (F(x) - \mathbb{1}_{y \leq x})^2 \, dx = \int_{-\infty}^{y} F^2(x) dx + \int_{y}^{\infty} (1 - F(x))^2 dx. \tag{7}$$

for a given CDF F and observation y. To compute it efficiently during training, it is important to compute the closed-form CRPS for the predictive distribution, shown in Equation 6.

For the ease of computation, we rewrite $F_{\hat{Y}}(y)$ as

$$F_{\hat{y}}(y) = \begin{cases} F_1(y), & y \leq u \\ F_2(y), & y > u, \end{cases} \tag{8}$$

where we assume that $y \geq c \coloneqq \log(\epsilon)$. Furthermore, we denote

$$F_1(y) = \begin{cases} 0, & y < c \\ p + (1-p)\Phi_{\mu,\sigma^2}(y), & y \in [c, u) \\ 1, & y \geq u, \end{cases} \tag{9}$$

where $\Phi_{\mu,\sigma^2}(y)$ denotes the CDF of the normal distribution with mean μ and standard deviation σ, and

$$F_2(y) = \begin{cases} 0, & y < u \\ F_1(u) + (1 - F_1(u))\text{GPD}_{u,\sigma_u,\xi}(y), & y \geq u. \end{cases} \tag{10}$$

Then, following Jordan (2016) we can decompose the CRPS into the following form:

$$\text{CRPS}(F_{\hat{y}}, y) = \begin{cases} \text{CRPS}(F_1, y) + \text{CRPS}(F_2, u), & y < u \\ \text{CRPS}(F_1, u) + \text{CRPS}(F_2, y), & y \geq u. \end{cases} \tag{11}$$

Equation 11 gives the loss of the predicted CDF proposed in this paper: a mixture of a truncated normal distribution with point mass and GPD for extreme values. The different parts of it are computed in Equation 12, 13, 14, and 17.

CRPS for F_2 Jordan et al. (2018) provide a closed-form expression for the Generalized Pareto distribution with point mass, which is given as

$$\text{CRPS}(F_2, y) = \sigma_u \left(\frac{y-u}{\sigma_u} - \frac{2(1-M)}{1-\xi} \left(1 - \left(1 - F_\xi \left(\frac{y-u}{\sigma_u} \right) \right)^{1-\xi} \right) + \frac{(1-M)^2}{2-\xi} \right), \tag{12}$$

where in our case $M = F_1(u)$. In addition, we obtain

$$\text{CRPS}(F_2, u) = \sigma_u \frac{(1-M)^2}{2-\xi}. \tag{13}$$

Note that the CRPS for the GPD distribution is only defined for $\xi < 1$.

CRPS for F_1 We use the following representation of the CRPS by Jordan (2016), which we first derive for the case of a standard normal distribution. Denote $F^u_{c,\mu,\sigma^2} \coloneqq F_1$, where μ and σ are the mean of the normal distribution. Then we have

$$\begin{aligned} \text{CRPS}(F^u_{c,0,1}, y) = \; & y(2F^u_{c,0,1}(y) - 1) - cP_c^2 + uP_u^2 + 2(1-p)G(c)P_c + 2(1-p)G(u)P_u \\ & - 2 \begin{cases} (1-p)G(c) - cP_c, & y < c, \\ (1-p)G(y), & c \leq y < u, \\ (1-p)G(u) + uP_u, & y \geq u, \end{cases} \\ & + 2(1-p)^2 \int_c^u G(x) f(x) \, dx, \end{aligned} \tag{14}$$

where $f = \varphi$, $P_x = F_1(x) - F_1(x^-)$, and $G(x) = \int_{-\infty}^{x} t\varphi(t)\,dt$ with φ denoting the density of a standard normal distribution. First, note that $G(x) = -\varphi(x)$. Then, we can consider each term individually:

$$\begin{aligned} P_c &= p + (1-p)\Phi(c) \\ P_u &= 1 - (p + (1-p)\Phi(u)) = (1-p)(1-\Phi(u)) \\ G(c) &= -\varphi(c) \\ G(u) &= -\varphi(u) \\ G(y) &= -\varphi(y) \end{aligned}$$

The only remaining unknown term is the last integral, which can be expressed as

$$\int_c^u G(x)f(x)\,dx = -\int_c^u \varphi^2\,dx = -\frac{1}{2\sqrt{\pi}}\left(\Phi(\sqrt{2}u) - \Phi(\sqrt{2}c)\right) \tag{15}$$

Following Jordan et al. (2018), we know that for a location-scale transformation, we have

$$\text{CRPS}(F_1, y) = \sigma\text{CRPS}\left(F^{(u-\mu)/\sigma}_{(c-\mu)/\sigma,0,1}, \frac{y-\mu}{\sigma}\right) \tag{16}$$

Lastly, we can compute

$$\begin{aligned} \text{CRPS}(F^u_{c,0,1}, u) = u - cP_c^2 + uP_u^2 + 2(1-p)G(c)P_c + 2(1-p)G(u)P_u \\ - 2((1-p)G(u) + uP_u) - (1-p)^2\frac{1}{\sqrt{\pi}}\left(\Phi(\sqrt{2}u) - \Phi(\sqrt{2}c)\right) \end{aligned} \tag{17}$$

D Experimental Details

In this section, we describe the experimental setup in detail, covering the software libraries used, the graph construction process, the permutation-invariant ensemble embedding, and the specifics of our Graph Neural Network (GNN) architecture, including the Graph Isomorphism Network with Edge features (GINE). All experiments were implemented in PyTorch (Paszke et al., 2019) and PyTorch Geometric (PyG) (Fey & Lenssen, 2019). The code for our method will be made public upon acceptance.

Graph Construction.

Meteorological stations are modeled as nodes in a graph. For N stations, we first compute the geodesic distance matrix $D \in \mathbb{R}^{N\times N}$, where each element $D_{u,v}$ represents the geodesic distance between station u and station v. An edge is created between nodes u and v if

$$D_{u,v} \leq d_{\max},$$

with $d_{\max} = 300$ km in our experiments. Each node is assigned a feature matrix of dimensions $n_{\text{ens}} \times F$, where n_{ens} is the number of ensemble members and F denotes the number of meteorological features.

Edge weights $w_{u,v}$ are computed based on the inverse of the geodesic distance between the corresponding stations. Specifically, given the locations l_u and l_v for stations u and v, respectively, the weight is defined as:

$$w_{u,v} = \frac{1}{d(l_u, l_v)},$$

where $d(l_u, l_v)$ is the normalized geodesic distance between the two locations. After computing these weights, we normalize them and set the self-connection weight $w_{u,u} = 1$ for all nodes u.

We summarize the graph as $G = (V, E, X, D, W)$, where V is the set of nodes, E is the set of edges, X represents the node feature matrices, D is the distance matrix, and W is the weight matrix.

PERMUTATION-INVARIANT ENSEMBLE EMBEDDING

To address the permutation symmetry in ensemble forecasts, where the order of ensemble members is irrelevant, we incorporate a DeepSet architecture (Zaheer et al., 2017) into the preprocessing stage. The initial node embedding for station v is computed as:

$$h_v^{(0)} = \Psi\left(\sum_{n=1}^{n_{\text{ens}}} \rho(x_{v,n})\right),$$

where $x_{v,n} \in \mathbb{R}^F$ is the feature vector corresponding to ensemble member n at node v. The functions ρ and Ψ are implemented as two-layer multilayer perceptrons (MLPs), ensuring that the embedding remains invariant to the permutation of ensemble members.

GRAPH NEURAL NETWORK ARCHITECTURE

The core of our model is a Graph Neural Network with residual connection that processes the constructed graph to capture spatial dependencies. Specifically, we use a Graph Isomorphism Network with Edge features (GINE) (Xu et al., 2019; Hu* et al., 2020) to integrate both node and edge information.

For each node v at layer t, the GINE update the node feature via:

$$h_v^{(t)} = h_v^{(t-1)} + \text{MLP}^{(t)}\left((1+\epsilon^{(t)}) \cdot h_v^{(t-1)} + \sum_{u \in \mathcal{N}(v)} \text{ReLU}\Big(h_u^{(t-1)} + w_{u,v}\Big)\right),$$

where $\mathcal{N}(v) := \{u \in V \mid (v,u) \in E\}$.

OUTPUT LAYER AND PARAMETER PREDICTION

The final node representations are used to predict station-specific parameters $\{p, \mu, \sigma^2, \sigma_u^2, \xi, u\}$. To enforce appropriate parameter constraints, we apply:

- a *softplus* activation for σ^2 and σ_u^2 and
- a *sigmoid* activation for p and ξ and
- a *linear* activation for μ and u.

OPTIMIZATION AND TRAINING

Our model is trained using the Adam optimizer (Kingma & Ba, 2017) with a learning rate set to 0.0001. Training and evaluation are performed on NVIDIA Tesla RTX3090 GPUs with 24GB RAM. Each model is trained for 25 epochs, and we report the test CRPS at the epoch that achieves the lowest validation CRPS.