



TC-ANet 1: A temporally joint model for post-processing ensemble weather forecasts for in-situ locations and multiple target variables

Peter Mlakar^{1,2} and Jana Faganeli Pucer²

¹Slovenian Environment Agency, Slovenia

²University of Ljubljana, Faculty of Computer and Information Science, Slovenia

Correspondence: Peter Mlakar (peter.mlakar@fri.uni-lj.si) and Jana Faganeli Pucer (jana.faganelipucer@fri.uni-lj.si)

Abstract. Ensemble forecast post-processing techniques are often applied to improve the reliability of the weather forecast. These methods issue a future predictive distribution which is better calibrated and as sharp as possible. While the field of forecast post-processing has been rapidly advancing, especially with the advent of machine learning, many challenges remain: post-processing models often assume the target distribution or require specialised distributions for each target variable, they require separate models for each lead time or location, and frequently forgo the explicit modelling of the temporal correlation structure in the target variable. In this work, we aim to address the outlined shortcomings by proposing a novel post-processing method which combines normalising spline flows and flow matching. This combination allows us to efficiently estimate marginal distributions and generate temporally correlated post-processed forecasts, useful for downstream applications. We evaluate our model on the openly available EUPPBench dataset, post-processing three target weather variables: temperature, precipitation, and wind gusts. Our method achieves competitive results for both marginal and joint distribution post-processing for all considered weather variables, rivalling current state-of-the-art techniques. Furthermore, we show that our joint lead-time post-processing approach extends its performance margin in a predictor-limited environment compared to per-lead-time methods.

1 Introduction

Numerical weather predictions (NWP) form the backbone of today's weather forecasting pipeline, impacting multiple aspects of our society, ranging from agriculture to transportation (Bauer et al., 2015). These forecasts are frequently issued as an ensemble, its plurality conveying an estimate of uncertainty. NWPs are increasingly reliable and accurate, yet they still exhibit biases and systematic errors (Vannitsem et al., 2021), which result from estimation errors in the initial conditions, sub-grid parameterisations, and resolution constraints. We can partially address these forecast shortcomings by statistically post-processing raw ensemble weather forecasts, such that they better represent measurements of relevant weather variables. The post-processed forecast is often explicitly or implicitly provided as a probability distribution, describing the uncertainty of the forecast at a specific time and location. The roster of post-processing methods is extensive, and we direct the reader to Vannitsem et al. (2021) for a general overview of the field.

Machine learning techniques have recently become more prevalent in the field of forecast post-processing (Rasp and Lerch,



2018; Schulz and Lerch, 2022; Mlakar et al., 2024; Chen et al., 2024; Feik et al., 2024; Bülte et al., 2025; Landry et al., 2026), producing state-of-the-art results and frequently outperforming other conventional approaches in terms of accuracy and reliability. Machine learning methods span multilayer perceptron (MLP) models (Rasp and Lerch, 2018), generative models (Chen et al., 2024), graph neural networks (Feik et al., 2024; Bülte et al., 2025), and transformer-based architectures (Landry et al., 2026). The ability to incorporate a wide range of predictors and generate spatially dependent forecasts is one of the features bolstering their growing popularity.

However, many challenges remain in this rapidly developing field. The majority of state-of-the-art deep learning post-processing techniques operate on a per-lead-time basis, requiring multiple models per lead time. Furthermore, if a model were to post-process multiple lead times concurrently, it would usually neglect the potential temporal correlation structure of the forecast, which could be exploited to improve its performance. This also implies that the estimation of temporally correlated post-processed forecasts using deep learning has, to the best of our knowledge, likewise not been explicitly addressed. Lastly, post-processing different target weather variables often requires tailored distribution models to account for their idiosyncrasies, slowing development. This is especially true for complex distributions such as the one describing the frequency of observed precipitation amounts, which is a part-discrete, part-continuous distribution. In this work, we propose the Temporally Correlated Atmosphere Network (TC-ANet), a novel post-processing method which addresses the above-mentioned shortcomings. Our main three contribution are:

- **Extracting cross-lead-time information** using a novel forecast encoder network and identifying contexts where such an approach yields the highest predictive gain.
- **Flexible censored marginal distribution modelling** via a modified normalising spline flow (Durkan et al., 2019) training procedure, enabling the model to correctly approximate zero-inflated point mass distributions such as that of precipitation.
- **Joint temporal forecast modelling** through a combination of continuous and discrete flow matching (Lipman et al., 2023), tailored for a robust estimate of both continuous and zero-inflated point mass joint distributions.

A detailed description of each contribution is provided in the following three subsections.

1.1 Combining lead time information

Recently, multiple post-processing techniques have been proposed which produce probabilistic forecasts by combining forecasts at different spatial locations. For example, Feik et al. (2024) apply a graph neural network (GNN) to encode relationships between spatial locations, with multiple message-passing layers exchanging information between connected nodes in the graph. This enables the method to leverage information from other stations when post-processing forecasts at a different station, improving its performance. While this method post-processes forecasts at all stations concurrently, it has to be applied separately for each lead time. Landry et al. (2026) investigate post-processing multiple lead times using deep learning, yet they do not explicitly consider inter-forecast dependencies. On the other hand, Wessel et al. (2024) include multiple lead times as covariates when post-processing temperature and wind speed forecasts using non-homogeneous Gaussian regression. They found



that adding lead times as covariates matches and even improves performance in data-scarce scenarios. Mlakar et al. (2024) also evaluate this aspect using a neural network post-processing technique. Crucially, both studies were conducted in a predictor-limited setting, using only the ensemble forecast of the target variable itself. Such approaches might not be scalable when multiple predictors become available, nor is it known how a lead-time dependent post-processing method would perform in such a context.

In this work, we propose a scalable neural network architecture which merges information from different lead times using a static attention mechanism, improving the predictive performance of the post-processing model. Additionally, we will demonstrate that the benefits of using multiple lead times as predictors scale inversely with the number of available predictor fields, making this method especially suited for scenarios with a limited number of predictors.

1.2 Flexible marginal distribution modelling for censored variables using normalizing flows

Success in distribution modelling is largely contingent on the correct choice of the target distribution. This is especially true for skewed or mixed distributions, the distribution of observed precipitation amounts being an example of the latter. There, a large portion of observations equate to zero, producing a point mass in the distribution and a continuous part in $\mathbb{R}^+$. Scheuerer and Hamill (2015) suggest that such distributions can be successfully modelled using a censored shifted gamma distribution (CSG) where the parameters of the distribution are estimated inside the framework of Ensemble Model Output Statistics (EMOS) (Gneiting et al., 2005), with an offset variable controlling the degree of censoring. Scheuerer et al. (2020) model the probabilities of distinct precipitation realisations as anomalies relative to the precipitation climatology using neural networks. This approach provides good results, yet is limited to a predetermined number of forecast bins, requiring interpolation to generate a continuous forecast. On the other hand, Berrocal et al. (2008) propose a two model approach for precipitation, consisting of a classifier and an intensity predictor. While performing well, the framework requires two separate models for approximating the censored distribution. Another approach might be to directly modify the raw forecast members (Van Schaeybroeck and Vannitsem, 2015; Bouallègue et al., 2024), yielding an effective and efficient method, albeit limited to the input forecast's ensemble size. Bremnes (2023) proposed one of the most flexible post-processing models, which estimates the quantile function of the precipitation distribution using censored Bernstein polynomials. This method enables flexible modelling of the distribution's quantile function, albeit with an image spanning a compact interval (which might be desirable in certain circumstances). Bülte et al. (2025), building on the graph neural network architecture of Feik et al. (2024), construct a three-part distribution for describing precipitation: a point mass at zero, a normal distribution for modelling intermediate, log-transformed precipitation levels, and a generalised Pareto Distribution for describing the distribution of log-transformed precipitation above a certain threshold. While achieving good overall results, this approach demonstrates the relative complexity of modelling highly skewed, part-discrete, part-continuous distributions. Another example of a skewed distribution, although to a lesser degree than precipitation, is the one describing wind gust measurements. Schulz and Lerch (2022) tested different distributional models and found that Bernstein polynomial quantile regression performs well in this setting.

As is evident, the choice of distributional model is important when designing a post-processing framework and depends heavily on the variable being modelled. A wrong choice can lead to suboptimal results and introduce additional biases. Furthermore, to



model complex distributions (precipitation), practitioners often tweak or design specific distribution models, which introduces another point of friction in the development process. This is especially true for precipitation post-processing, where multiple different approaches have been proposed, each one with its strengths and drawbacks. In this work, we propose a single, flexible normalising flow model which successfully approximates distributions, spanning from normal-like to highly skewed, even point-mass distributions, by altering the default training procedure through the inclusion of a censoring loss term. This removes the distribution design friction point, facilitating the development process and reducing complexity. Furthermore, univariate normalising spline flows yield an analytic description of the target distribution, allowing an explicit estimation of its quantiles and other properties of interest.

1.3 Joint temporal distribution modelling

Finally, we tackle the issue of producing temporally correlated probabilistic forecasts. Currently, the majority of neural-network-based approaches tackle the task of lead time forecasting in one of two ways: either by generating a separate model for each lead time (Rasp and Lerch, 2018; Bülte et al., 2025), or by providing a marginal forecast for each lead time at once using a single model (Mlakar et al., 2024; Landry et al., 2026). In these cases, the temporal correlation is not directly modelled.

Still, temporally correlated forecasts are of interest for downstream applications, such as estimating the minimal and maximal temperature or counting the number of exceedance events in a time interval. In the case of precipitation post-processing, this would facilitate the identification of severe rain events and provide additional information in the decision-making chain.

While neural network-based temporally correlated post-processing has yet to be explored, recently proposed methods consider the spatial correlation structure of the forecast. These approaches, while not entirely analogous, offer insight into how one might address the issue at hand. Chen et al. (2024) used a generative post-processing model for two-meter temperature and ten-meter wind gust forecasts, producing spatially correlated results. The model optimises the energy score, which forces it to consider spatial dependencies, resulting in correlated post-processed forecasts. Similarly, Landry et al. (2026) propose a flow-matching-based generative method for computing spatially correlated forecasts of temperature and wind gusts using a transformer neural network. While similar to our approach, their method does not explicitly consider the temporal correlation structure of the target variable. Additionally, TC-ANet also handles the modelling of censored distributions while Landry et al. (2026) focus on temperature and wind gust post-processing. While generative models expanded the capabilities of modelling high-dimensional correlated variables (Ravuri et al., 2021), one can still recover the joint structure of the variable from the raw input weather forecast. This procedure, ensemble copula coupling (ECC) (Scheffzik et al., 2013), enables univariate post-processing models to issue correlated forecasts. However, using ECC limits the post-processed output to the number of ensemble members in the input forecast.

Therefore, to explicitly address the temporal correlation structure of the target variable, we propose a novel flow-matching-based method for providing joint temporal forecasts. We combine continuous and discrete flow matching to rigorously approximate zero-inflated distributions. The post-processed distribution can then be accessed by sampling the resulting generative model and applying the censoring mask where appropriate. However, when sampling from a multidimensional joint distribution, a substantial number of samples is usually required to properly represent the individual marginal distributions. This might



not be feasible in computationally restricted contexts. To overcome these hurdles, we combine a joint model with a separate, low-cost marginal distribution model. The latter provides efficient access to the marginal distribution at each lead time, while we use the former to extract the joint temporal dependence structure, which we apply, through ECC, to the marginal forecast. This results in a temporally correlated forecast with better calibrated marginals at the same ensemble size. This removes ECC's ensemble size constraint since we use the post-processed joint sample instead of the raw input weather forecast.

2 Notation

Variables	Description
B, F, D	Training batch size, number of output features of a neural network layer
D, d	The number of forecast fields D and its corresponding index $d \in [1, D]$.
S, s	The number of station S and its corresponding index $s \in [1, S]$.
R, r	The number of run times R and its corresponding index $r \in [1, R]$.
T, t	The number of lead times T in six hour steps and its corresponding index $t \in [1, T]$.
M, m	The number of ensemble members M and its corresponding index $m \in [1, M]$.
$\beta^{(S)}$	A 20-dimensional vector of per-station predictors, where $\beta^{(S)} \in \mathbb{R}^{20}$.
$\beta^{(T)}$	A 10-dimensional vector of per-lead-time predictors, where $\beta^{(T)} \in \mathbb{R}^{10}$.
$\beta^{(Y)}$	A 2-dimensional vector of per-day-of-year predictors, where $\beta^{(Y)} \in [-1, 1]^2$.
$\mathbf{X}^{(s,r)}$	A tensor of ensemble forecasts for station s and run time r containing T forecasts.
$\mathbf{o}^{(s,r)}$	A T -dimensional vector of observations, either temperature, precipitation, or wind gusts, for station s and run time r .

Table 1. Summary of notation used throughout manuscript.

Let us first introduce the notation that will underpin the remainder of this work. Throughout, we will consider forecasts and observations for S in-situ measurement locations (referred to simply as "stations"). Each forecast corresponds to a unique run time R and contains forecasts for T lead times. The tensor $\mathbf{X}^{(s,r)}$ corresponds to a specific forecast at station s and run time r , containing a prediction for each ensemble member m , each lead time t , and for each forecast field d . This tensor is of dimension $[M, T, D]$. Then, for all three variables of interest, two-meter temperature, precipitation, and wind gusts, let $\mathbf{o}^{(s,r)}$ denote the vectors of observations for location s and run time r . These vectors contain T measurements, each corresponding to the lead times covered by $\mathbf{X}^{(s,r)}$. Table 1 summarises this notation.

The remainder of this manuscript is structured as follows: in Section 3 we describe the proposed post-processing model. Then, in Section 4.1, we present the dataset used for model training and evaluation, and the metrics used in Section 4.2. We present and discuss the results of our evaluation in Section 5.

3 Neural network for marginal and joint lead time post-processing

Our goal is to model the random weather variable $\mathbf{o}^{(s,r)} \in \mathbb{R}^T$ which is distributed according to some unknown multivariate probability distribution $\mathbf{o}^{(s,r)} \sim \mathcal{O}^{(s,r)}$. To this end, we can leverage generative modelling techniques, which allow us to implicitly approximate the target distribution by producing samples from it. Furthermore, we can inform our generation process by conditioning it on the weather forecast $\mathbf{X}^{(s,r)}$. In cases where the joint distribution is not required, and one is only interested in the marginal, we circumvent the sampling procedure by also issuing a per-lead-time marginal distribution.

To achieve this, we devise a three-part model:

- An encoder network which transforms the input forecast into a high-dimensional embedding tensor, one embedding tensor for each lead time.
- A MLP decoder network which produces the marginal lead time distribution of the target weather variable.
- A convolutional decoder network which produces the joint lead time distribution of the target weather variable.

The network architecture is displayed in Figure 1 and the constituents of the individual modules in Figure 2.

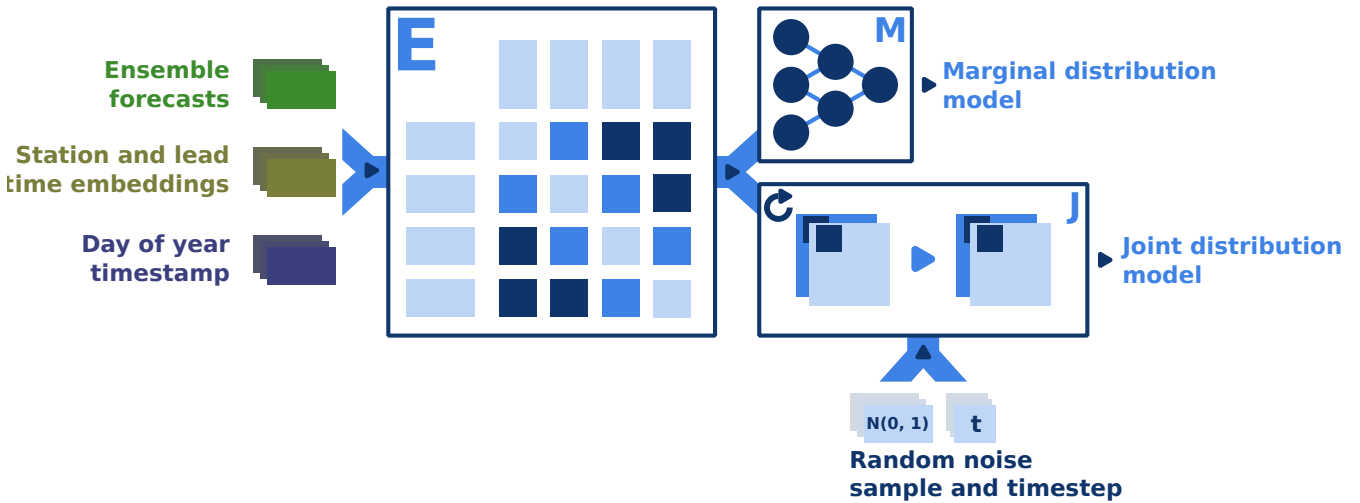


Figure 1. The proposed post-processing neural network architecture. The forecast encoder is denoted by the block labelled **E**, the marginal decoder by the block labelled **M**, and the joint decoder by the block labelled **J**. The joint decoder is invoked multiple times during integration to produce temporally correlated forecasts, denoted by the spinning symbol in the top left of the block. **E** consists of multiple layers of attention and residual layers and an initial linear layer which projects the input forecasts and predictors into embedding space, **M** is implemented as a shallow fully connected network, and **J** contains multiple sequential convolution layers followed by a fully connected layer producing the final output.

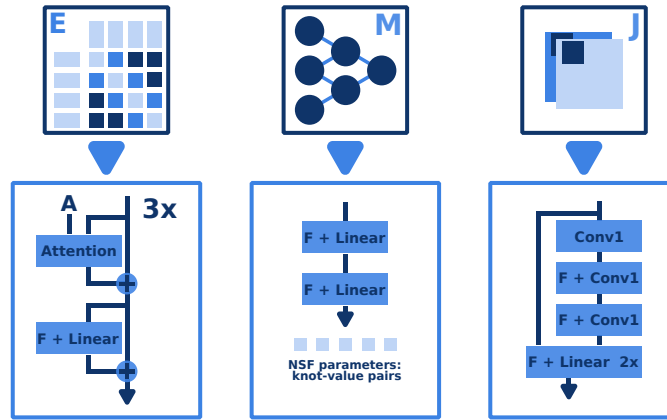


Figure 2. A more detailed presentation of the individual modules ("E" forecast encoder, "M" marginal decoder, "J" joint decoder) in terms of their more basic components. "Linear" denotes a fully connected layer, "Conv1" denotes a one-dimensional convolutional operation, and $\mathbf{A}$ the attention matrix computed by following Equation (A2). The symbol "F" denotes the SiLU activation function (Hendrycks and Gimpel, 2023). In total, the encoder uses three attention blocks, where all of them share the same attention matrix $\mathbf{A}$.

3.1 Ensemble forecast encoder

155 The encoder network's core is mainly composed of a sequence of attention layers Vaswani et al. (2017), which transfer information between different lead times. The outline of the network's architecture is presented in Figure 1. Initially, the input forecast $\mathbf{X}^{(s,r)}$ is augmented with three additional predictors: the per-station, per-lead time embeddings, and day of year predictors. The per-station embeddings enable the network to distinguish individual stations, resulting in a single predictive model for all. An individual per-station embedding is a vector $\beta^{(S)} \in \mathbb{R}^{20}$. We determined its size empirically through a grid search

160 over multiple values. Each per-station embedding is adapted during training, enabling the model to store information relevant to the post-processing at its corresponding location. We follow the same approach as Rasp and Lerch (2018) and sample the per-station embeddings from a normal distribution. The per-lead-time embeddings serve a similar purpose but enable the network to differentiate different lead times. The per-lead-time embedding is a vector $\beta^{(T)} \in \mathbb{R}^{10}$, unique for each of the T lead times. This vector is equivalent to the positional encoding used by transformers (Vaswani et al., 2017), often represented

165 by sinusoidal embeddings that become more dissimilar as the distance between positions increases. To retain the randomness aspect of Rasp and Lerch (2018) and the positional correlation of embeddings in Vaswani et al. (2017), we sample our random lead time embedding vectors recursively using a random walk. The recursive relation is defined in Equation (A1) in Appendix A. This procedure ensures that each $\beta^{(T)}$ follows a 10-dimensional normal distribution, and the similarity between embedding vectors i and j increases on average with $|i - j|$. Finally, we represent the per-day-of-year predictor as a vector of two elements

170 $\beta^{(Y)} \in \mathbb{R}^2$, where we define the two elements as $[\cos(2\pi \frac{y}{366}), \sin(2\pi \frac{y}{366})]^\top$, where y denotes the day of year $y \in [1, 366]$. The embedding and predictor vectors are concatenated with the ensemble average and standard deviation of each forecast field. Therefore, the resulting augmented forecast input tensor has a shape equalling $[B, T, 2D + 32]$. We denote this tensor as $\bar{\mathbf{X}}^{(s,r)}$. After augmentation, we pass $\bar{\mathbf{X}}^{(s,r)}$ through the encoder network. An initial linear layer transforms the tensor from input space



to embedding space of size F , resulting in a tensor of shape $[B, T, F]$. We apply a sequence of three residual attention trans-
 175 former blocks that transfer information between embedded forecasts at different lead times. For a detailed description of the
 transformer block implementation, we refer the reader to Appendix A. All attention blocks share the same attention matrix,
 which contains the influence a specific lead time has on the output of another. The attention matrix is defined by Equation (A2)
 in the Appendix. Examples of the attention matrices for random batches of forecasts are displayed in Figure 3.

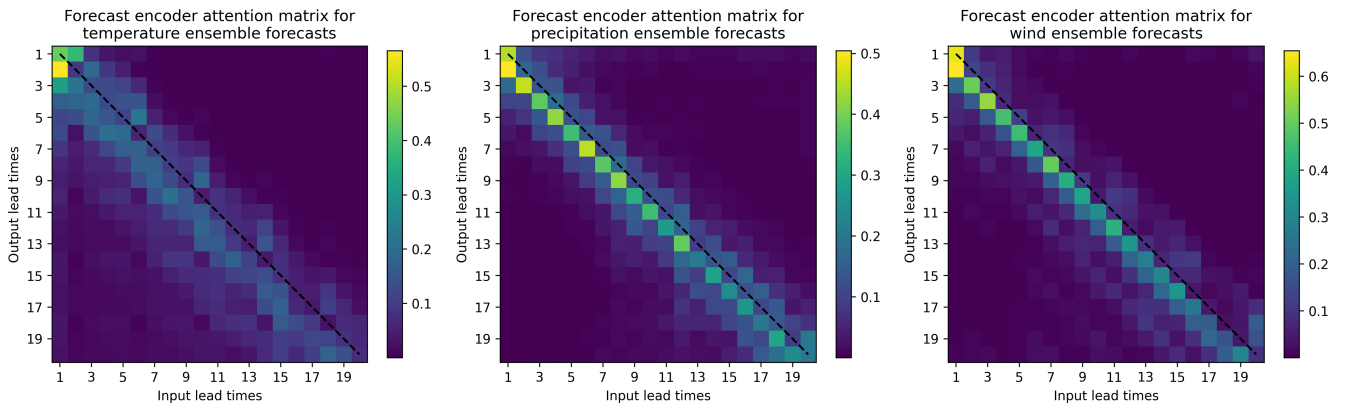


Figure 3. The forecast encoder’s attention matrix, averaged over a batch of ensemble forecasts. Each row of each matrix contains influence weights between pairs of lead times, a higher weight implying a stronger influence of the lead time in the column. It is immediately clear that frequently, for all three variables, the most impactful input lead time for a specific output lead time is the one directly preceding it. This is expected since attention is residual, the output of each block being added back to the input. Therefore, attention seeks relevant information from other lead times if available. Furthermore, the attention matrix for temperature is more spread out than precipitation and wind, which is partially explained by its higher degree of autocorrelation. Conversely, while wind gusts feature stronger autocorrelation compared to precipitation, the wind gust attention matrix concentrates its weight more, somewhat contradicting the above conjecture.

The output tensor of the final attention block represents the output of the forecast encoder. We denote this output tensor, of
 180 shape $[B, T, F]$, as $\mathbf{E}^{(s,r)}$.

3.2 Marginal and joint distribution modelling

After we construct the forecast embedding vectors $\mathbf{E}^{(s,r)}$, we pass those to each of the two decoder blocks: the marginal and the joint decoder (denoted by "M" and "J" respectively in Figure 1). The first provides an explicit description of the marginal distributions at each lead time, while the second models the target joint temporal distribution implicitly via sampling. The
 185 reason for our choice of a split model design now becomes clearer: the decoupling of the forecast encoder from the two decoders facilitates efficient sampling, as we encode forecasts only once, while executing the joint decoder multiple times during joint sampling. We present the individual decoders in more detail in the following sections, starting with the marginal model.



3.2.1 Marginal distribution modelling

190 Post-processed forecasts are commonly issued on a marginal, per-lead-time basis. Separate distribution models are estimated for each lead time, and the temporal correlation structure is not explicitly modelled. The parameters of the distribution, often chosen a priori by the practitioner (a normal distribution for temperature observations), are estimated by a regression model conditioned on a set of lead time predictors.

Our goal is to model the joint temporal distribution that encompasses the marginals, raising the question of whether we should
 195 model them separately. The marginal distribution could be recovered by generating multiple samples from the joint distribution and estimating the marginal quantiles from that sample. We found, however, that a large sample is usually required to obtain reliable estimates of the marginal distribution quantiles, especially the tails. Our encoder-decoder structure facilitates a simple solution to this problem: instead of generating many samples from the joint distribution, we add a shallow marginal decoder network that takes as input the forecast embeddings $\mathbf{E}^{(s,r)}$ and produces the per-lead-time parameters of a marginal
 200 distribution. This reduces the overall computational complexity of estimating the marginals while simultaneously increasing the quality of the estimate. We represent the marginal distribution as a normalising spline flow (NSF) (Durkan et al., 2019), based on the approach outlined in (Mlakar et al., 2024). NSF operate by transforming samples from a simple base distribution to those of a more complex target distribution. The transformation is parametrised by a diffeomorphic monotonic rational quadratic spline. A detailed description of the marginal model structure and loss function is available in Appendix B. For more
 205 details on normalising spline flows and their application to weather forecast post-processing, we refer the reader to Durkan et al. (2019); Mlakar et al. (2024).

Adapting NSF to censored distribution modelling is straightforward for univariate random variables. The only change required is a modification of the training procedure. Instead of minimising just the log-likelihood loss, we include an additional censoring loss for observations below a threshold (which is zero in our case), penalising the model for assigning probability to
 210 values above the threshold in those cases. The modified loss function is defined in Equation (B1) of Appendix B. This enables us to easily and economically model both censored and uncensored univariate random variables without the need to pick a variable-specific distribution or construct a separate classification model.

3.2.2 Joint distribution modelling

Our marginal distribution model does not account for the joint temporal distribution of weather variables. To this end, we add
 215 a decoder module to generate said distribution using flow matching (FM) (Lipman et al., 2023). Flow matching is a framework for the efficient estimation of Continuous Normalising Flows (CNF) (Grathwohl et al., 2018; Chen et al., 2019; Lipman et al., 2024). CNFs approximate the target distribution by transforming samples from a simple prior to samples from the distribution of interest. The transformation is not instantaneous (as with NSF) but iterative, performed over multiple autoregressive steps. At each step, the joint decoder module provides an estimate of the *vector field* which conducts this transformation, conditioned
 220 on the previous step's output and the forecast embedding vectors. A detailed description is available in the Appendix C. We refer the reader to Lipman et al. (2023, 2024) for an exhaustive treatise on Flow Matching.



Therefore, to produce the required vector field, our joint decoder module takes the forecast embeddings $E^{(s,r)}$ and concatenates them with $\mathbf{o}_l^{(s,r)}$ and l feature-wise, resulting in a tensor of shape $[B, T, F+2]$. Here, $\mathbf{o}_l^{(s,r)}$ denotes the target multivariate random variable at step l . We pass the augmented embeddings tensor through a sequence of three one-dimensional convolution
225 layers followed by a final, two-layer linear projection with a skip connection, producing the vector field v_l^θ . Using this framework, we can generate samples from the target distribution $O^{(s,r)}$ by picking multiple noisy initial samples $\mathbf{o}_0$ and, using an ODE solver, propagating them through the flow from step $l = 0$ to $l = 1$. We use the Euler solver in our implementation since higher-order solvers failed to improve results. The decoder module architecture is displayed in Figure 2.

The reason for designing the joint decoder as a convolutional neural network is entirely pragmatic. Highly optimised convolution
230 tion kernels do not require as much memory during inference as attention. The latter has a quadratic memory requirement w.r.t T , imposing stricter limitations on memory-constrained GPUs. Furthermore, the use of attention did not improve performance compared to convolution while requiring more computational resources. This was especially noticeable when sampling a large ensemble in parallel.

There is a snag, however: we identified that the lead time marginal distributions recovered from the joint model lag behind
235 those provided by the marginal model at small ensemble sizes. Since we sample the marginal distribution at equidistantly spaced quantiles, those forecasts frequently better cover the tails of the distribution compared to sampling at random from the joint distribution. To improve the marginal quality of the joint forecast, we combine the two distribution modules of TC-ANet for joint predictions. First, we sample the per-lead time marginal distributions using the marginal decoder. Then, we sample the joint distribution, generating an equal number of samples. We combine the two by applying ECC (Scheffzik et al., 2013) to
240 the marginal forecast using the joint sample lead time dependence. ECC is a technique for restoring the dependence structure of an uncorrelated marginal forecast by reshuffling the sampled members according to the ranks of the raw input forecast ensemble. The downside of this approach is that it limits the size of the target post-processed ensemble by the size of the input forecast ensemble. Our proposed approach circumvents this issue as both distribution models enable sampling arbitrarily sized ensembles. This enables us to generate forecasts with well-dispersed marginals while also retaining the joint temporal
245 correlation structure. This is the default forecasting mode of TC-ANet used in the evaluation. However, depending on the use case, TC-ANet can provide either the raw joint distribution or the ECC corrected marginal. The post-processing pipeline is displayed in Figure 4.

Joint modelling of censored values

One final question we need to answer is how to model censored joint distributions. Again, special care is required in such
250 cases. Treating censored values as other, non-censored realisations would result in an improper distribution estimate around the censoring threshold, reducing the usability and reliability of the post-processed forecast when forecasting events at or around said threshold. Yet, it is difficult to construct an appropriate vector field which accounts for censoring, as it is not clear what the censored $\mathbf{o}^{(s,r)}$ should be. Consider lead times where no precipitation is recorded: those corresponding entries of $\mathbf{o}^{(s,r)}$ could practically take on any value below or equal to the censoring threshold $\{\mathbb{R}^- \cup 0\}$ and still result in a valid vector
255 field. Therefore, we do not have a single target to interpolate with, let alone the vector field itself (albeit the choice for the latter

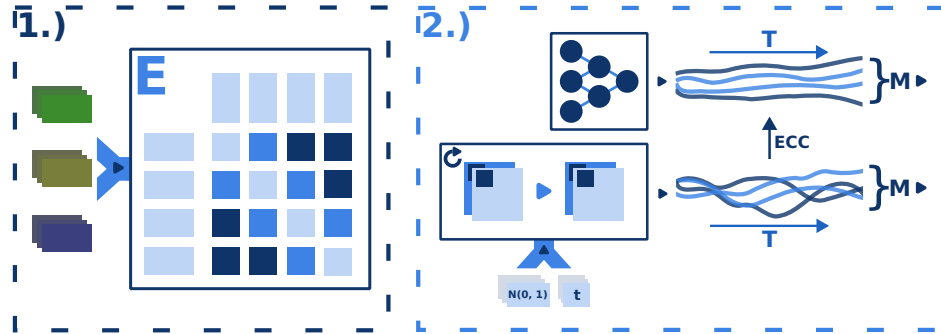


Figure 4. The TC-ANet post-processing pipeline. In 1.) the raw input forecast is encoded using the forecast encoder. The encoder is run once per forecast. In 2.), the forecast embeddings are passed to the marginal and the joint distribution decoder. Both produce an M member ensemble, the former described by equidistantly sampled quantiles and the latter by samples from the joint distribution. The marginal forecast is ECC corrected using the sampled joint distribution. The joint distribution and the ECC corrected marginal distribution can be issued as outputs depending on the downstream application.

is easier once the target is determined). To circumvent this conundrum, we expand our joint decoder into a two-part model when dealing with censored variables, by adding a "classifier" which determines, for a given sample, which lead times should be censored and which should not. However, we achieve this by not adding a classification model to the joint distribution decoder, but by modifying the flow itself to also produce a discrete flow (DFM) (Campbell et al., 2024; Lipman et al., 2024)

260 output, which complements its continuous part. This discrete flow estimates whether the t -th lead time in $\mathbf{o}^{(s,r)}$ is censored, and its output is computed concurrently with the continuous flow. A detailed description of DFT is provided by Appendix C. The described DFM approach requires little implementation overhead. Both $\mathbf{o}_l^{(s,r)}$ and $\mathbf{d}_l^{(s,r)}$, the latter being the DFM mask estimate at step l , are concatenated into a single tensor of shape $[B, T, F + 3]$ and passed to the joint decoder which now outputs two features instead of one: one regressing the FM vector field and the other regressing the DFM output. To produce

265 the joint censored forecast sample we multiply, element-wise, the discrete and continuous samples at their final integration steps. Both the discrete and continuous flow matching losses are optimised concurrently. Furthermore, inference for both parts is performed simultaneously, without the need for a separate classification model or an additional integration loop. For more details on DFM, we refer the reader to Campbell et al. (2024); Lipman et al. (2024).

3.3 Training procedure

270 The TC-ANet model is trained in two steps. First, the forecast encoder and the marginal decoder modules are trained concurrently. After this stage of training concludes, we freeze the forecast encoder's and marginal decoder's weights, and the joint decoder training stage begins using the forecast encoder's output embeddings. Each training stage ends after the validation loss fails to improve for a pre-specified number of epochs. Training both decoders simultaneously resulted in slightly inferior performance compared to the two-step approach. Surprisingly, a joint decoder trained in tandem with the forecast decoder

275 performs worse than the joint decoder provided by the two-step procedure. In the latter case, the model is pretrained to post-



process marginal distribution, possibly enabling the joint model to focus on the joint dependence structure, as the forecast embeddings already contain rich marginal information.

The same TC-ANet architecture is used for all three weather variables. Appendix E provides more details of the exact model implementation (number of parameters) and the choice of relevant hyperparameters, including those of BQN and MBM.

280 4 Evaluation

4.1 Dataset

We evaluate our model on the openly accessible EUPPBench (Demaeyer et al., 2023) dataset. The dataset consists of biweekly ECMWF reforecasts for a period of twenty years (from 1997 up to and including 2016) and daily forecasts over two years (2017 and 2018). The station data contain observations from Germany, France, the Netherlands, Belgium, and Austria, totalling
 285 $S = 122$ stations. All forecasts and reforecasts have a temporal extent of $T = 20$ lead times (excluding the analysis at time zero) in six-hour increments. The reforecasts contain a reduced ensemble count of $M = 11$ members while the forecasts contain $M = 51$ members. Both the reforecasts and forecasts contain the same $D = 29$ forecast fields. Following H hlelein et al. (2024), we divide the data into four separate datasets:

- **Training dataset:** The training dataset contains only reforecasts from 1997 up until, and including, 2009, resulting in a
 290 dataset with $R = 2612$. All considered models are trained using this dataset.
- **Validation dataset:** The validation dataset contains reforecasts from 2010 up until, and including, 2013, resulting in a dataset with $R = 836$. This dataset is used during training to terminate the process if the loss on this dataset fails to improve for a set number of epochs. Furthermore, we use the validation dataset to optimise the neural network hyperparameters and architecture.
- 295 – **Reforecast-based test dataset:** The test reforecast dataset contains reforecasts from 2014 up until, and including, 2017, resulting in a dataset with $R = 732$. We used this dataset to gauge the generalisation performance of our model.
- **Forecast-based test dataset:** The test forecast dataset contains forecasts from 2017 up until, and including, 2018, resulting in a dataset with $R = 730$. This dataset serves as the evaluation dataset in this manuscript.

4.1.1 Data preprocessing

300 The forecasts are all standardised using the training dataset sample mean and standard deviation. The same holds true for temperature observations, while precipitation and wind gusts are transformed by

$$\ln\left(\frac{o^{(s,r)}}{\hat{\sigma}_{o^{(s,r)}}} + 1\right),$$

$\hat{\sigma}_{o^{(s,r)}}$ being the observations training dataset sample variance. Every operation is performed element-wise on $o^{(s,r)}$.

Furthermore, when evaluating the post-processing performance for precipitation on the validation dataset, we found that the



305 methods underperformed at the tail of the distribution. Removing precipitation events with magnitudes over 40 mm from the training dataset only incidentally improved performance (in terms of the threshold weighted continuous ranked probability score) at the tails on the validation dataset. This cut-off threshold was determined empirically by evaluating multiple different values. This could be due to the difficulty of forecasting such events, which usually manifest as isolated, spiky occurrences, essentially introducing noise in the training procedure. Removing such events from this training setting (low resolution, per station post-processing) stabilised performance at the tail of the distribution, decreasing the amount of false positive forecasts. Furthermore, we found one station contained no precipitation observations for the entire training period and was therefore removed from the training and evaluation loop.

4.2 Metrics

To evaluate probabilistic forecasts Gneiting et al. (2007) suggest that one should consider the sharpness of the distribution subject to calibration. We describe the metrics used to evaluate different aspects of our distributional forecasts, both marginal and joint.

4.2.1 Univariate metrics

– Continuous ranked probability skill score (CRPSS)

320 The continuous ranked probability score (CRPS) (Matheson and Winkler, 1976) is a metric frequently used for evaluating the quality of probabilistic weather forecasts (Hersbach, 2000). It is a strictly proper scoring rule, meaning that, in expectation, it is minimised only when the forecast distribution equals the true observation-generating distribution. By considering the CRPS relative to a baseline, we compute the CRPSS, given in per cent improvements at lead time t . The CRPS is defined in Equations (D1),(D2) in Appendix D.

– Threshold-weighted Continuous Ranked Probability Skill Score (twCRPSS)

325 When dealing with heavy-tailed distributions, one might be interested in investigating the forecast quality on a specific range of outcomes. However, evaluating specific outcomes in isolation, even when using a proper scoring rule, can result in misleading conclusions (Lerch et al., 2017). To this end, weighted variants of proper scoring rules have been proposed (Gneiting and Ranjan, 2011). These can be used to evaluate the performance on specific outcomes by remaining proper. One such score is the threshold-weighted CRPS (twCRPS), which we use to evaluate the predictive performance of our models at different precipitation and wind gust levels. We consider the twCRPS skill score relative to a baseline, computed for different threshold values. The twCRPS is defined in Equation (D4) in Appendix D.

– Quantile skill score (QSS)

335 We also investigate the relative improvement of the quantile loss (QL) (Koenker and Bassett, 1978) for 51 equidistantly spaced quantiles on the interval $(0,1)$. This provides insight into how well each post-processing algorithm improves forecasts at different quantile levels, identifying performance characteristics at different parts of the distribution. We



mainly investigate the QL skill score relative to a baseline over the different quantile levels. The QL is defined by Equation (D6) in Appendix D.

– *Composite sharpness* (SH50, SH95)

If a probabilistic forecast is calibrated, we would also prefer it to be as sharp as possible, concentrating the bulk of uncertainty in a tight interval. To this end, we measure the forecast sharpness by the length of its 50% and 95% coverage intervals. Furthermore, to not bias against forecasters capable of producing multimodal forecast distributions, we employ the composite interval approach proposed by Bremnes (2019).

– *Brier skill score* (BSS)

To evaluate the individual model's performance in predicting binary events (discriminating between rain and no-rain), we employ the Brier score (BS) (Brier, 1950), a proper scoring rule, which penalises the model based on the difference between the predicted probability of occurrence and the value of the indicator variable of the actual occurrence. We also consider the Brier skill score over different thresholds. The BS is defined in Equation (D7) in Appendix D.

– *Rank histogram and random rank histogram*

The rank histogram is a visual tool for inspecting the reliability of a distributional forecast (Hamill, 2001). In case where a post-processed forecast is provided in terms of equidistantly spaced quantiles, a uniform rank histogram implies that the forecast is reliable as an equal amount of observations was observed at each equally spaced quantile bin or rank. Deviations from uniformity can reveal forecast shortcomings, such as over- or underspreading. We provide the rank histogram for 51 equidistantly spaced quantiles.

When dealing with precipitation and wind gusts, we use the random rank histogram to assess the forecast's reliability. If the model forecasts with probability p_c that the target variable will be below or equal to the censoring threshold c , then a rank inside the domain covering the interval $[0, p_c]$ is chosen at random, and the observation is assigned to that rank. Each rank histogram is equipped with the average mean deviation from uniformity, defined by Equation (D9) in Appendix D.

– *Reliability diagram*

A similar tool to the rank histogram is the reliability diagram (Hamill, 1997). We utilise this tool when evaluating the classification reliability of the models to discriminate between precipitous and dry events. By considering all forecasts where a probability p of there being no rain is issued and their corresponding observations, we compute the observed fraction of no-precipitation events p_o . For each p , one can plot its corresponding p_o . Looking at all $p_i \in [0, 1]$ and their corresponding $p_{o,i}$, a perfectly reliable classifier implies $p_i = p_{o,i}, \forall p_i$, while deviations indicate under- or overconfidence at specific probabilities.

4.2.2 Multivariate metrics

– *Energy skill score* (ESS)

The energy score (ES) is a multivariate variant of the CRPS (Gneiting and Raftery, 2007) often used for evaluating the



joint dependence quality of forecasts (Landry et al., 2026; Chen et al., 2024). The ES is defined in Equation (D9) in Appendix D.

4.2.3 Statistical significance

To determine the statistical significance of each method's forecasts, we employ the Diebold-Mariano (DM) test (Diebold and Mariano, 1995). The normalised average of the test score differentials follows a standard normal distribution under the null hypothesis of equal predictive performance. Large deviations from zero imply that one method outperformed the other, depending on the sign of the difference. We report the results as the percentage of stations where each method outperforms its counterparts. The DM is defined in Equation (D10), Appendix D.

4.3 Baselines

Two baseline methods are included in our comparison: the Bernstein Quantile Network (BQN) (Bremnes, 2020) and the Member-by-Member method (MBM) (Van Schaeybroeck and Vannitsem, 2015). Both methods represent popular post-processing approaches and have been widely applied in different post-processing settings (Scheffzik, 2017; Bremnes, 2020; Schulz and Lerch, 2022; Lerch et al., 2024; Bremnes et al., 2024). We also contextualise the performance of TC-ANet relative to other recently proposed state-of-the-art deep learning methods, focusing on publications that utilise the EUPPBench dataset.

4.3.1 Bernstein Quantile Network with Ensemble Copula Coupling (BQN)

The BQN method proposed by Bremnes (2020) is a neural network-based post-processing approach, where the network parametrises a quantile function describing the distribution of the target weather variable. By representing the quantile function in terms of Bernstein polynomials, BQNs offer a flexible model for approximating many different distributions without the need for selecting a specially tailored distribution model a priori. For precipitation Bremnes (2023) suggest a modification to the BQN training procedure, enabling the model to adapt to censored distributions. We employ this approach as a counterpart to our own censored NSF.

We implement a three-layer fully connected neural network for parametrising the quantile function. The network operates by post-processing each lead time and station separately; however, similar to Landry et al. (2026), we use a single model for all lead times and stations. To achieve this, we augment and transform the input forecast in the same manner as with TC-ANet, which resulted in the augmented forecast tensor $\tilde{\mathbf{X}}^{(s,r)}$, enabling the model to differentiate between different stations, lead times, and seasons. For more details on the augmentation procedure, we refer the reader to Subsection 3.1. The only difference is that the per-lead-time embeddings are not temporally correlated but just sampled from the standard normal distribution. Furthermore, we increased the number of parameters of BQN's parameter network to approximately match that of TC-ANet.



Each of the three layers is a mapping of type

$$\begin{aligned} l_1 : \mathbb{R}^{2D+32} &\mapsto \mathbb{R}^{128}, \\ l_2 : \mathbb{R}^{128} &\mapsto \mathbb{R}^{128}, \\ 400 \quad l_3 : \mathbb{R}^{128} &\mapsto \mathbb{R}^9, \end{aligned}$$

where the dimensionality of the image of mapping l_3 represents the number of parameters required by the quantile function, which is a Bernstein polynomial of degree eight. The enlarged network improved BQN's performance and, at the same time, somewhat equalised the playing field between the baseline and TC-ANet, minimising model size as a potential explanation for observed performance discrepancies. The parameters of BQN are tuned by minimising the QL as described in Equation (D6).

405 We construct an ensemble of 10 BQN networks to overcome neural network initialisation variability. This procedure presents little additional overhead due to the small size of the individual BQN models while significantly improving the performance compared to using a single model only. The individual BQN quantile forecasts are combined into a single post-processed forecast via quantile averaging. BQN, on its own, does not provide a joint temporal forecast. To restore the temporal dependence of the post-processed marginal ensemble, we use ECC, reshuffling the marginal forecast according to the ranks of the raw

410 ECMWF forecast.

4.3.2 Member-by-member post-processing (MBM)

A method specifically tailored for retaining the inter-ensemble correlation structure in the post-processed forecast is the Member-by-member post-processing approach (MBM) (Van Schaeybroeck and Vannitsem, 2015). By using the same correction coefficients for each ensemble member at lead time t

$$415 \quad P_{m,t}^{(s,r,d)} = p_{1,t}^{(s,r)} + p_{2,t}^{(s,r)} \langle X_t^{(s,r,d)} \rangle + \mathbb{S}(p_{3,t}^{(s,r)})(X_{m,t}^{(s,r,d)} - \langle X_t^{(s,r,d)} \rangle), \quad \mathbb{S}(x) = \ln(1 + e^x),$$

the model corrects for biases with three forecast dependent coefficients $p_{1,t}^{(s,r)}, p_{2,t}^{(s,r)}, p_{3,t}^{(s,r)}$ while preserving the inter-ensemble dependence structure in the post-processed ensemble at lead time t , whose members are denoted by $P_{m,t}^{(s,r,d)}$. The variable $X_t^{(s,r,s)}$ denotes the d th forecast field tensor at lead time t , while angle brackets represent averaging over the ensemble. The symbol $\mathbb{S}$ denotes the softplus function.

420 We use BQN's neural network backbone to regress MBM's coefficients. We refer the reader to Section 4.3.1 for more details regarding its implementation. Our decision to combine neural nets with MBM is based on work by Lerch et al. (2024) who evaluated a neural network framework in conjunction with member-by-member post-processing and concluded that such a combination offers good results. The parameters of the network are tuned by minimising the marginal CRPS, defined in Equation (D2). An ensemble of ten models is trained to curb neural network initialisation variability.

425 MBM has been applied to the post-processing of all three variables. And while our implementation performed well in the former two cases, the model exhibited issues in terms of reliability in the case of precipitation post-processing. We therefore decided not to include MBM's precipitation results in this study.



4.3.3 State-of-the-art deep learning post-processing methods

We provide additional context as to how well TC-ANet compares to recently proposed state-of-the-art neural network post-processing methods. Specifically, we focus on results reported by Höhle et al. (2024); Feik et al. (2024); Bülte et al. (2025); Landry et al. (2026), all of them using the EUPPBench dataset. While all four works use varying metrics to gauge their respective method's performance, one metric is included in all works, that being the CRPS, computed on a per-lead time basis or, in the case of Landry et al. (2026), aggregated over all lead times. While the CRPS is reported as a summary statistic, it, being a proper scoring rule, still offers a compressed measure of the quality of a probabilistic forecast. Therefore, we examine the reported CRPS from each work to contextualise TC-ANet's performance:

- Höhle et al. (2024); Feik et al. (2024) provide temperature CRPS scores at three lead times, averaged over all stations and run times: 24, 72, and 120. In the case of Höhle et al. (2024), we chose the scores reported for the ST-BQN method as it achieved the smallest CRPS at most lead times.
- Landry et al. (2026) provide the mean CRPS for temperature and wind gust forecasts aggregated over all stations, run times, and lead times.
- Bülte et al. (2025) report mean precipitation CRPS for lead times at 24, 72, and 120 hours.

5 Results

The aggregate scores for the *forecast-based test dataset* for all three target variables and compared methods are reported in Table 2. By averaging the metrics over all stations, run and lead times, we present an overall measure of performance on this dataset. All methods improve upon the raw ECMWF forecast, with TC-ANet leading in the majority of metrics. The ECMWF forecast is, on average, the sharpest; however, at the expense of calibration, as is evident by its relatively high CRPS values. The margins between the post-processing methods are slim, especially those of TC-ANet and BQN. We supplement this analysis by including the DM predictive performance analysis in Table 3. There, we consider the percentage of stations at which one method produced significantly better CRPS results than another. TC-ANet more frequently produces lower CRPS forecasts at a statistically significant level compared to the other two approaches, where the null hypothesis is only rarely rejected in the other direction. Precipitation forecasts have the highest ratio of stations in favour of TC-ANet, with wind gust showing the least improvement in this regard. Still, a more in-depth analysis is required to elucidate other differences between the two post-processing approaches.

Temperature [K]	CRPS ↓	ES ↓	SH50 ↓	SH95 ↓	
ECMWF	1.189 ($\sigma = 1.227$)	6.421 ($\sigma = 3.579$)	0.875 ($\sigma = 0.543$)	2.981 ($\sigma = 1.816$)	
BQN	0.768 ($\sigma = 0.689$)	4.462 ($\sigma = 1.562$)	1.804 ($\sigma = 0.698$)	4.500 ($\sigma = 1.616$)	
MBM	0.773 ($\sigma = 0.697$)	4.467 ($\sigma = 1.563$)	1.264 ($\sigma = 0.515$)	4.290 ($\sigma = 1.646$)	
TC-ANet	0.762 ($\sigma = 0.669$)	4.447 ($\sigma = 1.512$)	1.838 ($\sigma = 0.745$)	4.432 ($\sigma = 1.710$)	
Precipitation [mm]	CRPS ↓	ES ↓	BS($\mathcal{T} = 0$) ↓	SH50 ↓	SH95 ↓
ECMWF	0.407 ($\sigma = 1.398$)	3.807 ($\sigma = 4.799$)	0.297 ($\sigma = 0.357$)	0.256 ($\sigma = 0.548$)	1.543 ($\sigma = 2.394$)
BQN	0.375 ($\sigma = 1.456$)	3.709 ($\sigma = 4.999$)	0.119 ($\sigma = 0.221$)	0.189 ($\sigma = 0.697$)	1.812 ($\sigma = 2.898$)
TC-ANet	0.369 ($\sigma = 1.422$)	3.658 ($\sigma = 4.877$)	0.114 ($\sigma = 0.202$)	0.206 ($\sigma = 0.745$)	1.843 ($\sigma = 2.945$)
Wind gusts [$\frac{m}{s}$]	CRPS ↓	ES ↓	SH50 ↓	SH95 ↓	
ECMWF	1.470 ($\sigma = 1.444$)	7.996 ($\sigma = 3.787$)	1.266 ($\sigma = 0.793$)	4.255 ($\sigma = 2.601$)	
BQN	1.016 ($\sigma = 0.944$)	5.955 ($\sigma = 2.148$)	2.427 ($\sigma = 0.948$)	6.177 ($\sigma = 2.303$)	
MBM	1.021 ($\sigma = 0.964$)	5.951 ($\sigma = 2.148$)	1.666 ($\sigma = 0.663$)	5.855 ($\sigma = 2.297$)	
TC-ANet	1.012 ($\sigma = 0.945$)	5.948 ($\sigma = 2.143$)	2.403 ($\sigma = 0.987$)	5.921 ($\sigma = 2.274$)	

Table 2. Aggregate performance scores over the forecast-based test dataset for all three target variables. Lower values represent better results for the CRPS, ES, BS, SH50, and SH95. Highlighted in bold is the best-performing model for each score.



Temperature	BQN	MBM	TC-ANet	Precipitation	BQN	MBM	TC-ANet	Wind gusts	BQN	MBM	TC-ANet
BQN ↑	×	42.6	3.3	BQN ↑	×	×	3.3	BQN ↑	×	59.8	0.0
MBM ↑	0.8	×	2.5	MBM ↑	×	×	×	MBM ↑	5.7	×	4.1
TC-ANet ↑	25.4	58.2	×	TC-ANet ↑	73.8	×	×	TC-ANet ↑	10.7	62.3	×

Table 3. The Diebold-Mariano (Diebold and Mariano, 1995) test of predictive performance in terms of the **CRPS**. Each entry represents the percentage of stations for which the method in the row outperforms its counterpart in the column in a statistically significant manner. The percentage $100 - \mathcal{M}_{i,j} - \mathcal{M}_{j,i}$ provides the percentage of equal performance cases.

Temperature	CRPS @ 24h [K] ↓	CRPS @ 72h [K] ↓	CRPS @ 120h [K] ↓
Höhlein et al. (2024)	0.62	0.80	1.13
Feik et al. (2024)	0.60	0.78	1.09
TC-ANet	0.62	0.77	1.09
Temperature and wind gusts	CRPS [K] ↓	CRPS [$\frac{m}{s}$] ↓	
Landry et al. (2026)	0.77	1.03	
TC-ANet	0.76	1.01	
Precipitation	CRPS @ 24h [mm] ↓	CRPS @ 72h [mm] ↓	CRPS @ 120h [mm] ↓
Bülte et al. (2025)	0.467	0.597	0.678
TC-ANet	0.468	0.570	0.676

Table 4. Aggregate performance scores over the forecast-based test dataset of TC-ANet relative to other recently proposed state-of-the-art approaches. Highlighted in bold are the best-performing models for each score.

Table 4 contains the comparison of TC-ANet to other, recently proposed state-of-the-art deep learning methods in terms of the CRPS. We can see that TC-ANet competes favourably with other approaches, exhibiting similar or better performance. A more notable exception is the method by Feik et al. (2024) with a lower CRPS at the lead time of 24 hours. This difference could be due to Feik et al. (2024) combining information from multiple stations, something that TC-ANet does not consider. Furthermore, TC-ANet is While this comparison is limited as we consider a single metric in a set of limited scenarios, we believe that it contextualises TC-ANet as a competent method in today’s post-processing environment.

Continuing our baseline comparison, Figure 5 provides further insight into the performance of individual methods. It contains the CRPSS and QSS, the former stratified over lead times and the latter over quantiles. Plots in the first two rows display the two metrics relative to the raw ECMWF forecast. To better highlight the differences between the three post-processing baseline approaches, rows three and four in Figure 5 display the same scores, but relative to BQN. Contemplating rows one and two first, we can observe a periodic pattern in the CRPSS for all three variables with an overall decreasing trend in improvement as lead times increase. Due to the under-dispersed nature of the ECMWF forecast (as is evident from the rank histograms in



Figure 6), the QSS is most strongly improved for the tail quantiles. Rows three and four in the same figure better highlight specific differences between the individual post-processing models. For temperature and wind gusts, TC-ANet improves over BQN more strongly in early lead times, where the improvement slowly tapers off, and at the end, slightly dips below BQN. The exception to this is TC-ANet's performance at the first lead time, which is smaller than the improvement at adjacent lead
470 times. This might be due to a lack of exogenous information from other lead times, as shown in Figure 3, leading to a non-homogeneous post-processing scenario, forcing the network to adapt its behaviour in this specific case. The QSS indicates that TC-ANet better estimates the right tails of the temperature and precipitation distribution. In the latter case, this might be due to BQN producing a bound quantile function, essentially assigning zero probability to events below and above some threshold. This is further supported by the spike in the precipitation rank histogram for BQN in Figure 6. Wind gust shows a trend
475 reversal, with TC-ANet slightly underperforming BQN at the extreme quantiles while still improving performance at the lower and higher quantiles. When considering aggregate scores, one should keep in mind that for precipitation, the forecast-based dataset consists of 75% no-precipitation events, so the majority improvement in both CRPSS and QSS is heavily influenced by those events. This holds true for wind gusts as well, although to a lesser extent.

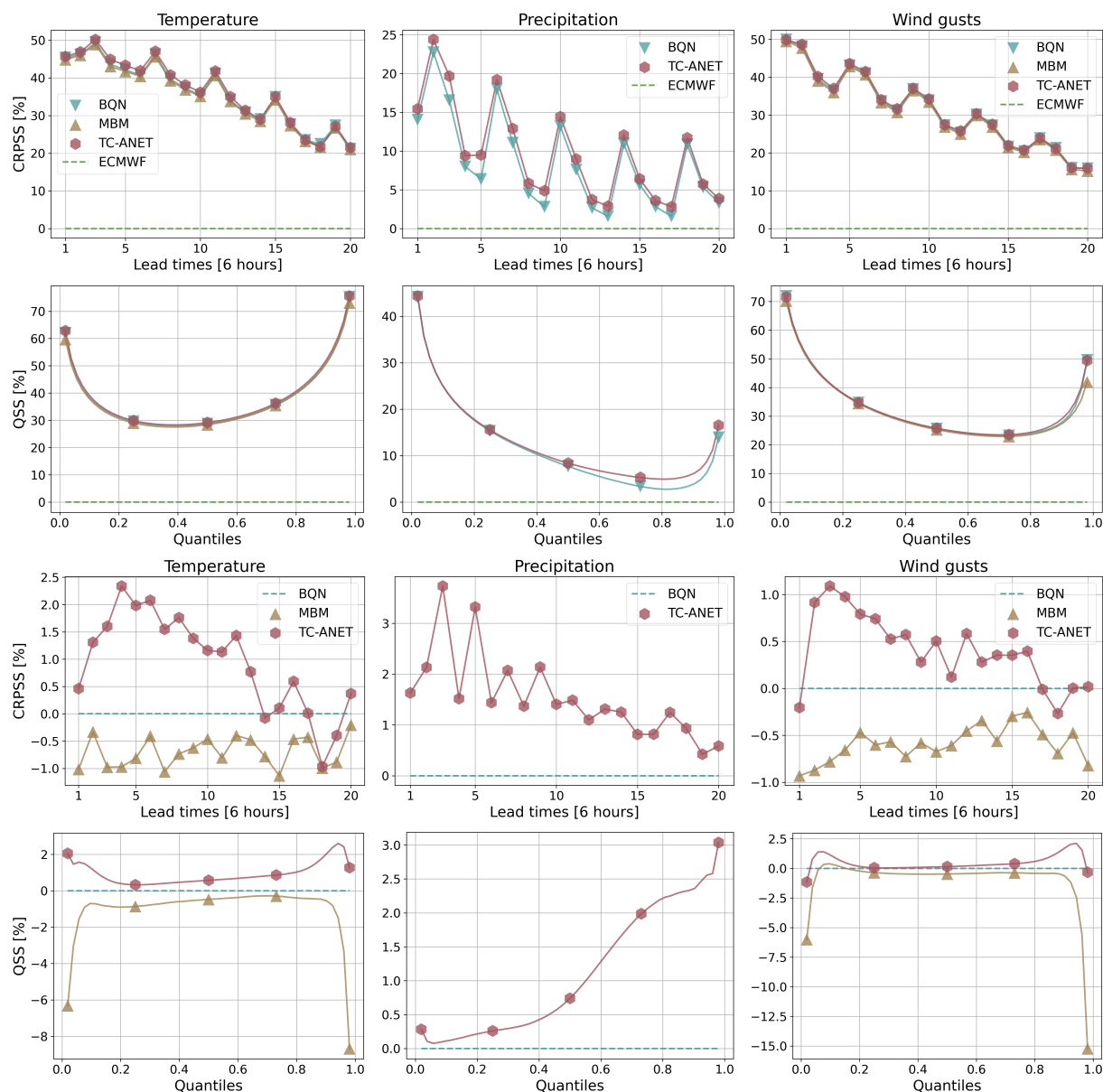


Figure 5. (First row and third): Per-lead time CRPSS for the forecast-based test dataset. (Second row and fourth): Per-quantile QSS for the forecast-based test dataset. All skill scores are computed relative to ECMWF in the first and third rows, and relative to BQN in the second and fourth rows.

To assess the reliability of individual methods, Figure 6 displays the rank histograms for each method and target variable.
 480 The ECMWF ensemble forecast is largely under-dispersed for all three variables. This is evident from the overpopulation in its extreme ranks. All methods improve on the base forecast (observe the difference in scale on the y axis between ECMWF and the post-processing methods), with MBM providing the most uniform rank histogram on average. However, it's overly sharp



forecast results in a high proportion of outliers not covered by the ensemble envelope, as is evident by the spikes in the extreme ranks. This is also true for BQN, where the limited support of the distribution somewhat contributes to this behaviour, albeit to a lesser extent. On the other hand, TC-ANet's distributions seem to be overdispersed as can be surmised from the inverted U shape of its rank histograms. We did not observe this behaviour on the reforecast-based dataset, which might suggest that the source of this discrepancy lies in the different ensemble counts of the two datasets, influencing how the post-processing model handles forecast uncertainty.

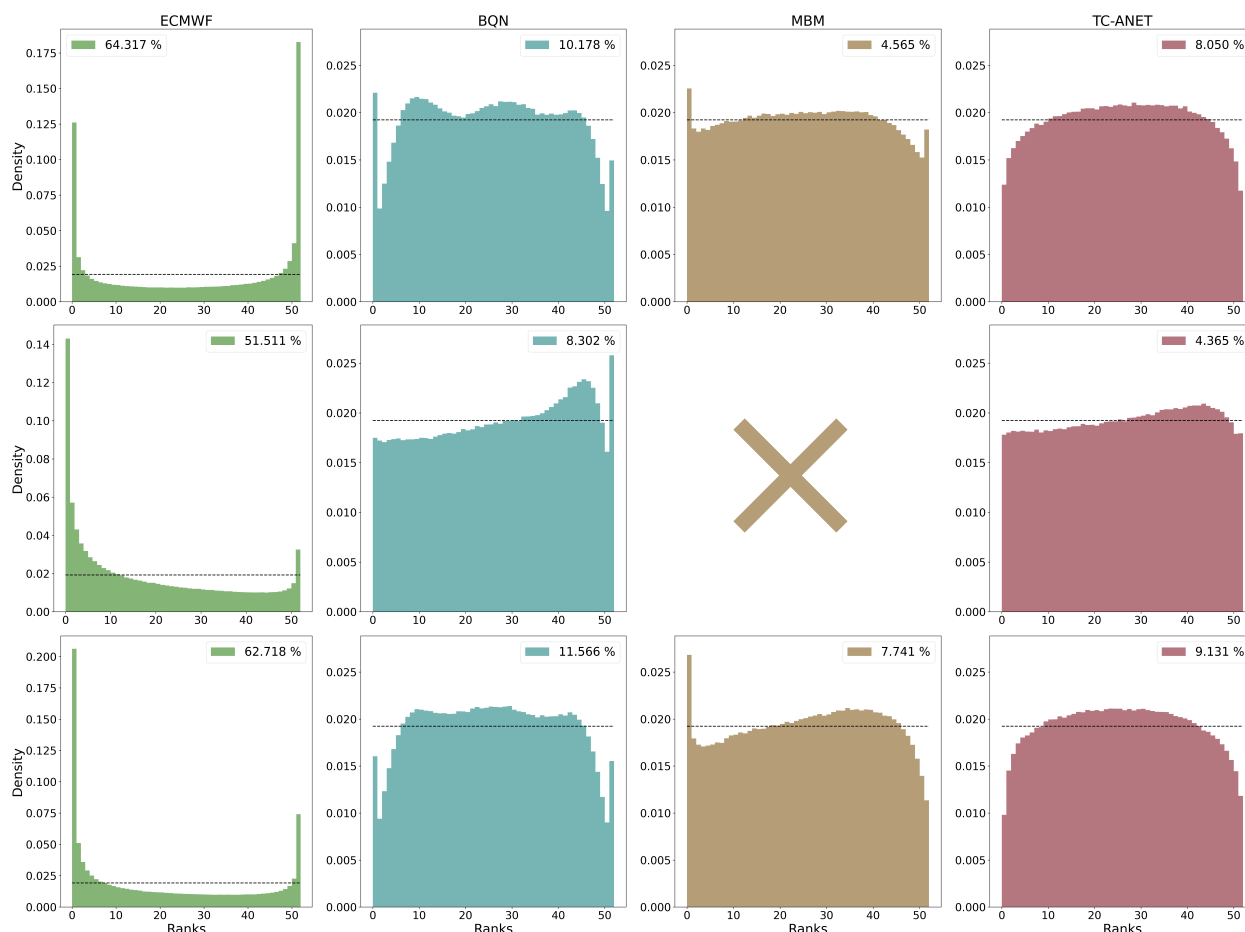


Figure 6. (Upper row): Temperature rank histograms. (Middle row): Precipitation random rank histograms. (Bottom row): Wind gust random rank histograms. A uniform rank is preferable and is denoted by the black dashed line. The average discrepancy between expected and observed rank histogram heights relative to expected height is provided by the number in the label for each model.

Continuing, let's consider Figures 7 and 8. These contain the twCRPSS and BSS for precipitation and wind gusts, respectively. Additionally, Figure 7 also contains the reliability diagram for precipitation at the zero millimetre threshold. The twCRPSS in the first row of Figure 7 reveals that both methods improve upon the ECMWF precipitation forecast, with TC-ANet



exhibiting greater improvement over all thresholds compared to BQN. The precipitation BSS plot in the same row displays a similar trend for all thresholds. One can observe a large spike in BS improvement compared to the ECMWF forecasts for small thresholds, with the reliability diagram elucidating the underlying cause: the ECMWF forecast frequently over-predicts the amount of precipitation. For example, in cases where the ECMWF forecast assigned a probability of 0.25 of there being no precipitation, the actual observed fraction of no-precipitation events equalled close to 0.78, meaning that it rained much less often than forecast. Both BQN and TC-ANet correct for this bias, introducing a bias of their own by being marginally dry. A twCRPSS and BSS comparison relative to BQN is presented in the second row of Figure 7 for the purpose of legibility.

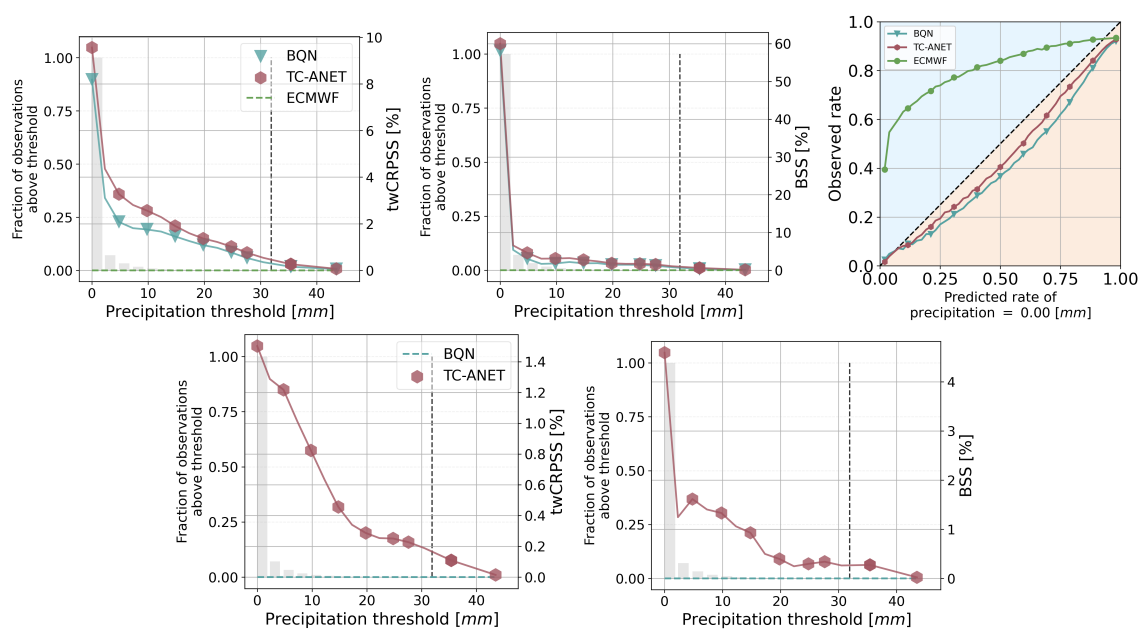


Figure 7. The twCRPSS (first row, left plot; second row, left plot), BSS (first row, middle plot; second row, right plot), and reliability diagram (first row; right plot) for precipitation post-processing on the forecast-based test dataset. (twCRPSS, BSS): The grey columns in twCRPSS and BSS represent the fraction of observations with values equal to or greater than that of the threshold. The black vertical dashed line represents the 99.9% quantile of positive observation values. The first row contains scores relative to ECMWF, the second relative to BQN. (Reliability diagram): The black, diagonal dashed line represents a perfectly reliable forecaster. Falling below the diagonal implies an under-prediction of precipitation occurrence (denoted by the light red colour of the half plane) while falling above the diagonal signifies precipitation over-prediction (denoted by the light blue colour of the half plane).

A similar visualisation is displayed in Figure 8 for wind gust post-processing. TC-ANet outperforms the other two baseline methods at nearly all thresholds in terms of twCRPS, only falling behind BQN at the very extreme of the wind gust distribution. For this specific case, the observed wind speed falls outside of TC-ANet's predicted ensemble envelope, under-predicting the realised wind speed. Evaluation on the validation dataset found this issue to be partially systematic, requiring further research. TC-ANet achieves the best BS across all thresholds, with the margin extending as thresholds increase. The exception is the



threshold at zero wind speed, where it falls slightly behind. This is due to TC-ANet predicting a zero speed wind gust event in
 505 0.02% cases where those events did not materialise.

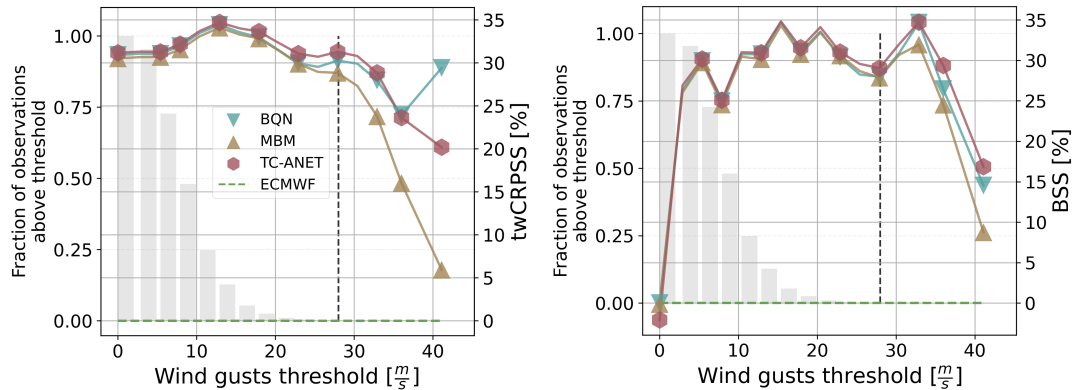


Figure 8. The twCRPS (left plot) and BSS (right plot) for wind gust post-processing over the forecast-based test dataset. The grey columns represent the fraction of observations with values equal to or greater than the threshold. The black vertical dashed line represents the 99.9% quantile of positive observation values across all stations, run times, and lead times.

Figure 9 displays examples of all three forecast types TC-ANet can provide: the marginal forecast, the FM temporally correlated forecast, and the combination of both, the ECC corrected marginal forecast. This example demonstrates the difference between the joint forecast and ECC corrected marginal forecast at small ensemble sizes. The median estimate, denoted by the dashed line, is more variable due to the small sample size. The ECC corrected model’s analytic estimate of the median (and
 510 in fact of every quantile) provides better scores than the joint model for the same ensemble size. This is further corroborated by Table 5, where we report the CRPS score for the joint forecast for multiple different values of M . As M increases, the CRPS decreases due to a better estimate of the expectation in Equation (D2) and a better coverage of the spread. Yet even at the largest ensemble size $M = 301$, the method does not reach the same CRPS as the marginal model (except precipitation). Therefore, if one is interested in a particular marginal distribution, it is prudent to query the marginal model. Still, generat-
 515 ing a $M = 301$ -member ensemble for the forecast dataset took approximately ten minutes on our system, largely due to the decoupled encoder-decoder structure and the small size of the TC-ANet model (even when in an ensemble of multiple models). Table 6 holds the percentage of stations at which a specific model produced a more significant ES result compared to other models. The marginals between the models are smaller compared to the results reported in Table 6, however, TC-ANet still outperforms its counterparts at more stations, where again the null hypothesis is often left rejected in favour of the other
 520 method. TC-ANet also benefits from being able to generate ensemble sizes of arbitrary size while still retaining competitive joint modelling performance.

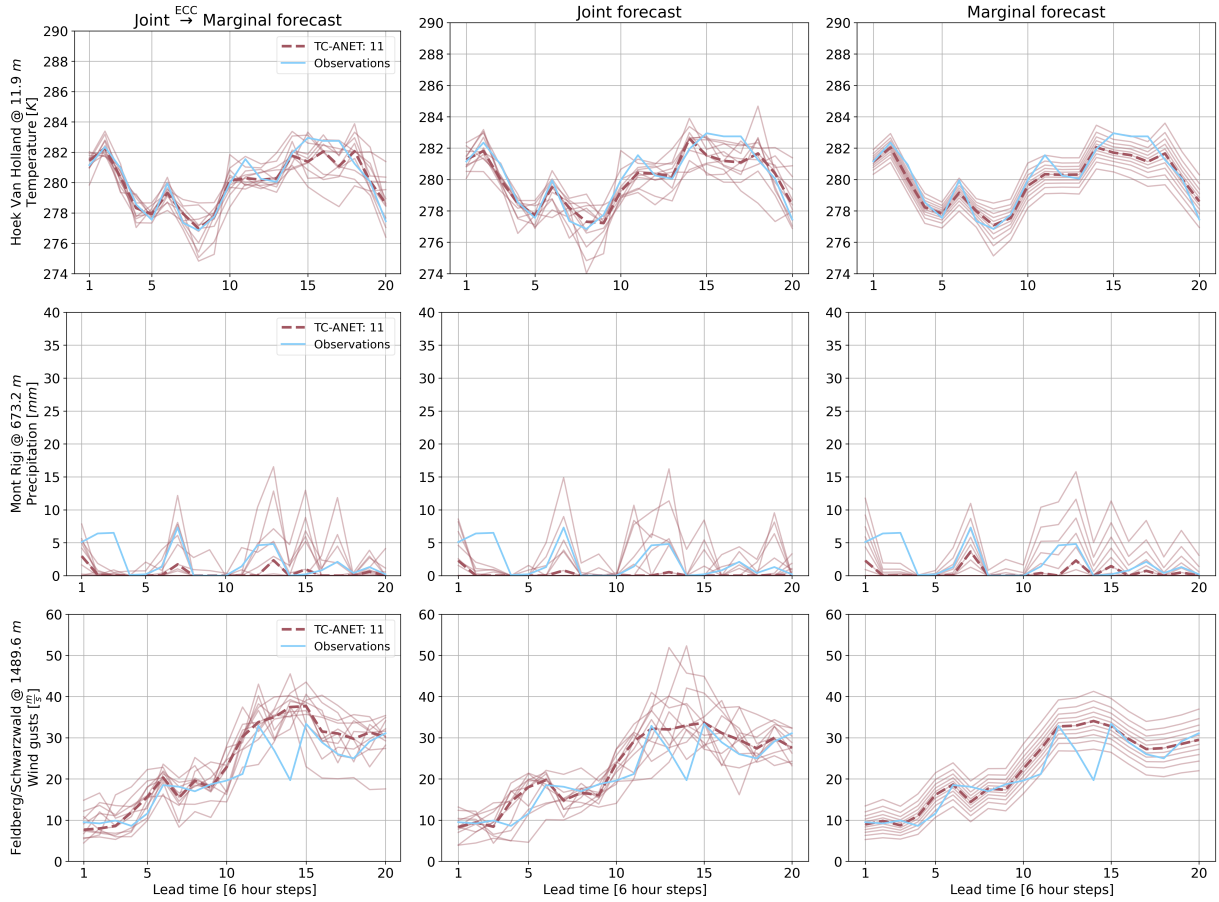


Figure 9. Examples of all three forecast types provided by TC-ANet. (Right plot) The marginal model forecast of M equidistantly sampled quantiles. Temporal correlation is not explicitly modelled. (Middle plot) The joint model forecast where M members are sampled via FM and censored according to the DFM output. (Left plot) The ECC corrected marginal forecast based on the joint forecast’s temporal dependence.

	CRPS @ $M = 51 \downarrow$	CRPS @ $M = 101 \downarrow$	CRPS @ $M = 201 \downarrow$	CRPS @ $M = 301 \downarrow$
Temperature[K]	0.779	0.772	0.768	0.767
Precipitation[mm]	0.373	0.370	0.368	0.367
Wind gust[$\frac{m}{s}$]	1.035	1.024	1.019	1.018

Table 5. The CRPS of TC-ANet’s joint model forecast as a function of M on the forecast test dataset. Lower values represent preferred scores.

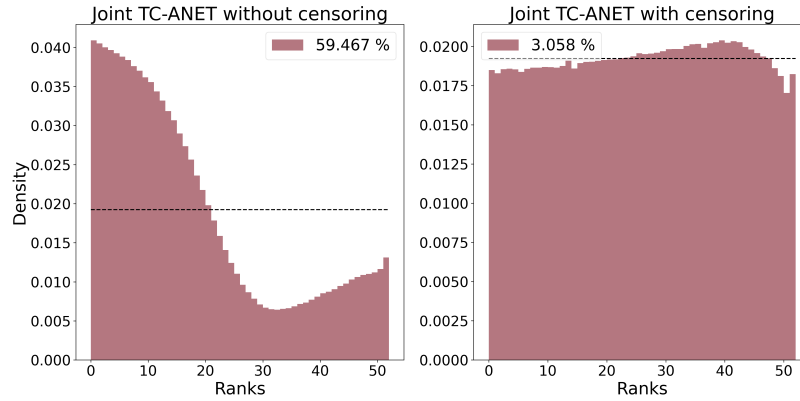


Figure 10. The marginal random rank histogram for the joint temporal TC-ANET forecast. (Left): The uncensored joint distribution where we treat zero precipitation events as regular realizations. (Right): The censored joint distribution using DFM. A uniform rank is preferable and is denoted by the black dashed line. The average discrepancy between expected and observed rank histogram heights relative to expected height is provided by the number in the label.

Temperature	BQN	MBM	TC-ANet	Precipitation	BQN	MBM	TC-ANet	Wind gusts	BQN	MBM	TC-ANet
BQN ↑	×	7.4	4.9	BQN ↑	×	×	3.3	BQN ↑	×	13.9	1.6
MBM ↑	3.3	×	4.9	MBM ↑	×	×	×	MBM ↑	11.5	×	8.2
TC-ANet ↑	9.0	16.4	×	TC-ANet ↑	56.6	×	×	TC-ANet ↑	6.6	13.9	×

Table 6. The Diebold-Mariano (Diebold and Mariano, 1995) test of predictive performance based on the ES. Each entry represents the percentage of stations for which the method in the row outperforms its counterpart in the column in a statistically significant manner. The percentage $100 - \mathcal{M}_{i,j} - \mathcal{M}_{j,i}$ provides the percentage of equal performance cases.

Additionally, the importance of accounting for censored values when modelling the joint distribution becomes apparent in Figure 10. The uncorrected joint forecast’s random rank histogram is highly skewed towards the lower ranks. This is mainly due to an excess of low precipitation forecasts. Many of these represent a no-precipitation event, but did not converge to zero. One could select a cut-off threshold to attempt to resolve these cases. This would require evaluating multiple cut-off thresholds, potentially introducing different biases if we cut legitimate low-value precipitation forecasts. The continuous-discrete flow matching framework resolves this issue by design, the model deciding on an appropriate threshold on a sample-by-sample basis. This produces a close-to-uniform marginal random rank histogram with a shape similar to the marginal model’s histogram. Finally, let’s investigate the impact that including information from multiple lead times has on the post-processed forecast. To gauge the impact, we propose the following experiment: consider the set of forecast fields $\{\mathbf{X}_d\}$. For each target weather variable, we create a sorted list $[\mathbf{X}_d]$ where the rank denotes the importance of each field for post-processing in a decreasing order. This ranking is determined by fitting a simple linear regression model with LASSO (Tibshirani, 1996) regularisation.

The ranking of each forecast field is given by the magnitude of the linear model's coefficients, with a larger absolute weight denoting a more impactful forecast field. Then, taking TC-ANet and BQN, we train D sets of ten-member ensembles for each
 535 model, where each set $i \in [1, D]$ contains i most impactful forecast fields.

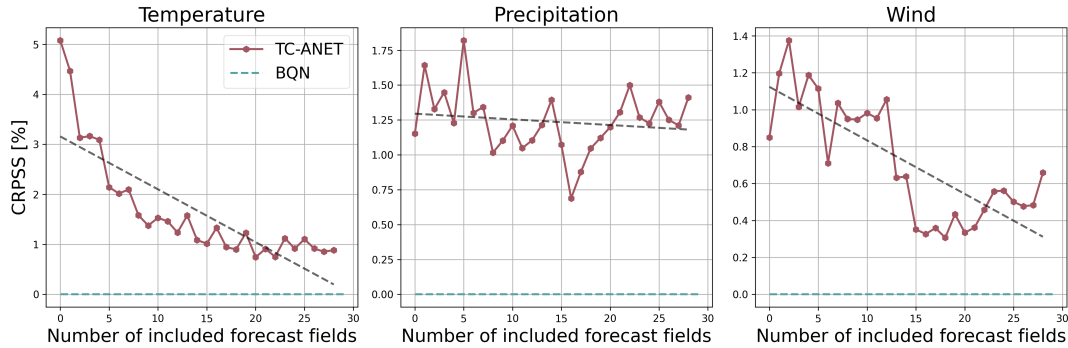


Figure 11. The CRPSS for TC-ANet and BQN, where the latter is used as the baseline. The forecast fields were ranked by importance using a linear regression model with LASSO regularization (Tibshirani, 1996) and ranking each field by the magnitude of the coefficient's absolute value. The black dashed line represents the linear trend of the CRPSS.

Figure 11 displays the CRPSS for the two models. A flat CRPSS indicates that adding multiple lead times as predictors fails to impact the forecast quality, with the margins remaining fixed for all forecast field counts. Looking at the CRPSS plots in Figure 11, we can observe some evidence to the contrary. All plots show a decreasing trend in CRPSS with increasing predictor counts, the drop being most pronounced for temperature and the least for precipitation. The relative distance between TC-ANet
 540 and BQN for precipitation remains nearly constant, indicating that additional lead times do not meaningfully impact the quality of the post-processed forecast. This is somewhat expected as precipitation is the least autocorrelated, followed by wind gusts, and temperature, the latter being the most autocorrelated of the three. Wind forecasts are a curious example of the CRPSS decreasing dramatically with the addition of two forecast fields, which linear regression failed to recognise as impactful, that being the fields u_{10} and v_{100} . These results suggest that using multiple lead times as predictors pays off when other forecast
 545 predictors are not available and the target variable is autocorrelated. In such settings, TC-ANet with a single forecast field as input, outperforms a BQN model trained with four forecast fields.

6 Conclusions

We presented a novel, flexible ensemble forecast post-processing method named TC-ANet, tailored for joint lead time forecasting. The method offers quick and efficient sampling of both the marginal and joint temporal distributions using two separate
 550 distribution models, the former based on normalising spline flows and the latter on flow matching. Additionally, TC-ANet can approximate different target distributions, censored and uncensored, in a marginal or joint capacity. For marginal distribution modelling, a censoring term is included in the log-likelihood loss, accounting for the amount of required censoring, while we combined continuous and discrete flow matching techniques to model joint temporal dependencies. We further improved



the marginal calibration of the joint forecast by applying the ensemble copula coupling correction using TC-ANet's joint and
555 marginal distribution samples in a single forecast. The resulting post-processed forecast retained the temporal correlation struc-
ture of the former and marginal calibration of the latter. Furthermore, TC-ANet's per-lead-time attention mechanism enables it
to extract information from multiple lead times to correct forecasts at other lead times. This increases its performance margin
against per-lead-time methods, especially in predictor-limited scenarios, temporally correlated target variables, such as tem-
perature, seeing the largest benefit. All of this combined enabled TC-ANet to compete favourably against recently proposed
560 state-of-the-art deep learning techniques and two neural-network-based baseline methods. Furthermore, we evaluated TC-ANet
for post-processing three different target variables: temperature, precipitation, and wind gusts, and found it competitive in all
three scenarios.

Still, many promising avenues for future research remain. For example, the performance at early lead times demonstrated by
Feik et al. (2024) suggests that TC-ANet could benefit from explicitly combining information from multiple stations, something
565 we did not explore in this work. Furthermore, the static, forecast-independent attention mechanism implies that our method
acts similarly to a graph neural network over lead time. This is analogous to Feik et al. (2024) where the graph neural network
spans stations instead. The attention weights depend solely on the target station, time of year, and lead time pairs. However,
one could conceive of scenarios where the forecast itself should also dictate the degree of dependence. While we had limited
success exploring forecast-conditioned attention weights, we still believe they might improve the model's post-processing per-
570 formance in specific weather scenarios.

Overall, we believe that TC-ANet constitutes a meaningful addition to the roster of post-processing methods. It can suc-
cessfully adapt to different post-processing scenarios, be it in a predictor-limited setting or when estimating highly skewed
distributions. The ability to provide exact marginal and efficient joint distribution forecasts facilitates a detailed and informed
decision-making process.

575 *Code availability.* The TC-ANet model implementation, forecast prediction, and evaluation functions used in this manuscript are publicly
available through a frozen release at <https://doi.org/10.5281/zenodo.23262191> (Mlakar, 2026a). The frozen release additionally includes the
model predictions for the test dataset and the model weights for each of the three considered weather variables. The project's source code is
also available on GitHub <https://github.com/petermlakar/TC-ANet> (Mlakar, 2026b).

Data availability. The EUPPBench dataset is openly available at <https://doi.org/10.5281/zenodo.7708362> (Bhend et al., 2023). Additional
580 information on the structure and contents of the dataset is also available on Github [https://eupp-benchmark.github.io/EUPPBench-doc/files/](https://eupp-benchmark.github.io/EUPPBench-doc/files/EUPPBench_datasets.html)
[EUPPBench_datasets.html](https://eupp-benchmark.github.io/EUPPBench-doc/files/EUPPBench_datasets.html) (EUPPBench, 2023).



A Ensemble forecast encoder

We define the lead time embedding by the following recursive relationship

$$\beta_1^{(T)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}); \beta_i^{(T)} = \frac{\beta_{i-1}^{(T)} + \alpha \epsilon}{\sqrt{1 + \alpha^2}}, \quad (\text{A1})$$

585 for $i \in [2, T]$, $\alpha = 0.5$, and $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. This ensures that the lead time embeddings slowly drift apart, with speed proportional to α , while remaining normally distributed at each lead time.

The lead time embeddings play a crucial role in the information flow inside the attention blocks. Consider the individual i -th attention block and let $\mathbf{V}_i$ denote its value matrix. The matrix $\mathbf{V}$ is defined by

$$\mathbf{V}_i^{(s,r)} = \mathbf{W}_v^{(i)} f_{j+i-1}(\bar{\mathbf{X}}^{(s,r)})$$

590 where $\mathbf{W}_v^{(i)}$ denotes the value weight matrix of block i , with dimensions $[F, F]$, and $f_j(\bar{\mathbf{X}}^{(s,r)})$ the augmented input tensor, transformed by the encoder neural network up to layer j . We can consider the tensor $\mathbf{W}_v^{(i)}$ of size $[B, T, F]$ acting as a matrix of shape $[T, F]$ in this context, since we broadcast all operations over the first, batch dimension. Attention can be computed in parallel for multiple attention heads (Vaswani et al., 2017). However, since we use a single attention head only, we economise on the notation. We define the attention between the t_1 -th and the t_2 -th lead time as

$$595 \mathbf{A}_{t_1, t_2}^{(s,r)} = g(\beta^{(S)}, \beta^{(Y)}, \beta_{t_1}^{(T)}, \beta_{t_2}^{(T)}). \quad (\text{A2})$$

The function $g: \mathbb{R}^{20} \times \mathbb{R}^2 \times \mathbb{R}^{10} \times \mathbb{R}^{10} \rightarrow \mathbb{R}$ is a shallow two-layer fully connected neural network, with hidden dimension size of 32, which provides the forecast-independent importance between the lead times t_1 and t_2 . The vectors $\beta_{t_1}^{(T)}$ and $\beta_{t_2}^{(T)}$ denote the t_1 -th and t_2 -th lead time predictors. The resulting attention matrix is forecast independent, conditioned only on the time of year, target station, and the lead times of interest. We compute the i -th attention output following

$$600 f_{j+i}(f_{j+i-1}(\bar{\mathbf{X}}^{(s,r)})) = f_{j+i-1}(\bar{\mathbf{X}}^{(s,r)}) + \mathbf{W}_i \mathbb{S}(\mathbb{D}(\mathbf{A}_i^{(s,r)})) \mathbf{V}_i^{(s,r)}, \quad (\text{A3})$$

where $\mathbb{S}$ represents the softmax function, $\mathbb{D}$ the dropout regularization operation, and the matrix $\mathbf{W}_i$ applies a final linear re-projection after attention is applied. The dropout layer, with probability $p = 0.1$, sets a random subset of attention values to $-\infty$, which, after applying $\mathbb{S}$, results in the corresponding input-output lead time combinations not influence each other. The output of the attention operation defined by Equation (A3) is passed through a residual layer, the output of which is considered
 605 as the output of the i -th attention block.

B Marginal distribution model

The log likelihood function of the marginal model can be defined as

$$\mathcal{L}_t^{(s,r)} = \ln(\phi(g_\theta(o_t^{(s,r)}))) + \ln \left(\left| \frac{\partial g_\theta}{\partial o_t^{(s,r)}}(o_t^{(s,r)}) \right| \right),$$



where ϕ represents the density function of the base distribution, $o_t^{(s,r)}$ the weather variable observation at lead time t , and g_θ the rational quadratic spline transformation, parametrised by the marginal decoder through θ . Since g_θ is a diffeomorphism, it admits the estimation of the quantiles of the transformed univariate distribution by

$$q_\tau = g_\theta^{-1}(\Phi(\tau)),$$

where Φ denotes the cumulative distribution function of the base distribution and $\tau \in (0, 1)$ the desired percentile. To model the diffeomorphic transformation, we employ a single spline with ten knots. We found that increasing the number of splines did not improve performance. Our base distribution of choice is the normal distribution. We have evaluated different base distributions, such as the logistic and student distributions, but found no significant impact on the final performance.

The censoring-adapted log-likelihood loss is defined as

$$f(o_t^{(s,r)}) = \begin{cases} \Phi(g_\theta(c)), & o_t^{(s,r)} \leq c \\ \phi(g_\theta(o_t^{(s,r)})) \left| \frac{\partial g_\theta}{\partial o_t^{(s,r)}}(o_t^{(s,r)}) \right|, & \text{otherwise,} \end{cases} \quad (\text{B1})$$

where c equals the censoring threshold which, in case of precipitation, equals $c = 0$. The cumulative probability of the distribution in the interval $(-\infty, 0]$ (first case of Equation A2) is assigned to the censoring threshold, resulting in a point-mass at zero as required. The training can be conducted by minimizing the negative log-likelihood function of the above density.

C Joint distribution model

Let $\phi_l : [0, 1] \times \mathbb{R}^T \rightarrow \mathbb{R}^T$ denote the time l dependent flow function. As l progresses from zero to one, the flow ϕ_l transforms the noisy sample into a sample from the target distribution. The main insight of Lipman et al. (2023) is that one can define ϕ_l in terms of a vector field, related to ϕ_l through the following ordinary differential equation (ODE)

$$\frac{\partial}{\partial l} \phi_l(x) = v_l(\phi_l(x)),$$

where v_l denotes the time dependent vector field *generating* the flow. In short, if we can somehow model the vector field, its corresponding flow can be recovered through the ODE, enabling sample generation.

Lipman et al. (2023) state that a suitable candidate for a vector field can be the simple linear interpolant between $o^{(s,r)}$ and o_0 . This is the main crux behind the success of FM, replacing expensive integration during training of a CNF with a simple regression task. By parametrising v_l with a neural network, we can estimate the flow by minimising the following loss function

$$\begin{aligned} \mathcal{L}_{\text{FM}}(\theta) &= \mathbb{E}_{l, o_0, o^{(s,r)}} \|v_l^\theta(o_l^{(s,r)}, \mathbf{E}^{(s,r)}) - (o^{(s,r)} - o_0)\|^2, \\ o_l^{(s,r)} &= (1 - l)o_0 + lo^{(s,r)}, \end{aligned}$$

where

$$l \sim \sigma(N(0, 1)), o_0 \sim \mathcal{N}(0, \mathcal{I}), o^{(s,r)} \sim O^{(s,r)}.$$



The value $\mathbf{o}^{(s,r)}$ denotes the weather variable observation vector, distributed by the unknown distribution $O^{(s,r)}$ which we wish to model, v_l^θ denotes the output of our neural network, parametrised by θ , given $\mathbf{o}_l^{(s,r)}$ and l as input, and σ represents the sigmoid function.

640 The above approach does not consider censored random variables. To that end, we extend the above method with DFM. As with FM, DFM also requires a suitable target variable. The target vector for DFM is, in our case, binary $\mathbf{d}^{(s,r)} = [\dots, \mathbb{1}_{\mathbf{o}_t^{(s,r)} > c}, \dots]$, with c corresponding to the censoring threshold. Then, using the continuous noise vector $\mathbf{o}_0$, we define the discrete noise vector as $\mathbf{d}_0 = [\dots, \mathbb{1}_{\mathbf{o}_0 > c}, \dots]$. This implies that $\mathbf{d}_0$ depends on $\mathbf{o}_0$ and we found that sampling an independently distributed $\mathbf{d}_0$, irrespective of $\mathbf{o}_0$, degrades performance. We designate an additional output of the joint decoder module to regress the target

645 vector directly by minimising the cross-entropy loss

$$\mathcal{L}_{\text{DFM}} = -\mathbb{E}_{l, \mathbf{d}_0, \mathbf{d}^{(s,r)}} \left[\sum_{t=1}^T \mathbf{d}_t^{(s,r)} \ln(\mathbf{p}_t^\theta) + (1 - \mathbf{d}_t^{(s,r)}) \ln(1 - \mathbf{p}_t^\theta) \right],$$

$$\mathbf{p}_t^\theta := \mathbf{p}_t^\theta(\mathbf{d}_t^{(s,r)}, \mathbf{E}^{(s,r)}),$$

$$\mathbf{d}_{l;t}^{(s,r)} = \begin{cases} \mathbf{d}_{0;t}, & \text{with probability } 1 - l, \\ \mathbf{d}_t^{(s,r)}, & \text{with probability } l. \end{cases}$$

Here $\mathbf{p}_i^\theta$ denotes the forecast conditioned DFM output from the joint decoder, corresponding to a vector containing probabilities

650 of the t -th lead time being uncensored, and $\mathbf{d}_l^{(s,r)}$ denotes the interpolated sample given time step l . The subscript t denotes taking the t -th element of the subscripted vector. We can recover the appropriate vector field at step $l \in [0, 1)$ during inference as

$$\mathbf{u}_l = \frac{\mathbf{p}^\theta(\bar{\mathbf{d}}_{l-\Delta}, \mathbf{E}^{(s,r)}) - \bar{\mathbf{d}}_{l-\Delta}}{1 - l}, \quad \bar{\mathbf{d}}_0 = \mathbf{d}_0,$$

where $\bar{\mathbf{d}}_{l-\Delta}$ denotes the estimated discrete sample at the previous integration step $l - \Delta$ with step size Δ . The t -th element of

655 the discrete sample at time step l can then be computed as

$$\bar{\mathbf{d}}_{l;t} = \begin{cases} 1, & \text{with probability } \bar{\mathbf{d}}_{l-\Delta;t} + \Delta \mathbf{u}_{l;t}, \\ 0, & \text{otherwise.} \end{cases}$$

D Evaluation metrics

D1 Continuous ranked probability skill score

The CRPS is defined by the following integral

$$660 \quad \text{CRPS}(F, o) = \int_{-\infty}^{\infty} (F(x) - \mathbb{1}_{x > o})^2 dx, \quad (\text{D1})$$



where F represents the distribution function of the issued forecast and $\mathbb{1}_{x>o}$ a step function at the threshold corresponding to the observation o . While analytic formulations of the CRPS exist for certain distribution families, the expression often remains intractable for complex distribution models. In such cases, one can resort to the following estimator of the CRPS

$$\text{CRPS}(p, o) = \mathbb{E}_p |p - o| - \frac{1}{2} \mathbb{E}_{p,p'} |p - p'|, \quad (\text{D2})$$

665 where p and p' represent independent samples from the issued forecast distribution F (Gneiting and Raftery, 2007). The CRPSS at lead time t is defined as

$$\text{CRPSS}_t^{\text{model}} = \left(1 - \frac{\langle \text{CRPS}_t \rangle^{\text{model}}}{\langle \text{CRPS}_t \rangle^{\text{baseline}}} \right) 100,$$

where the CRPS is averaged over all stations and run times.

D2 Threshold-weighted continuous ranked probability score

670 The twCRPS is defined as

$$\text{twCRPS}(F, o) = \int w_c(x) (F(x) - \mathbb{1}_{x>o})^2 dx, \quad (\text{D3})$$

$$w_c(x) = \begin{cases} 1, & x > c, \\ 0, & \text{otherwise,} \end{cases} \quad (\text{D4})$$

675 where $w_c(x)$ denotes a weighting function, emphasising certain regions of the distribution. We specifically consider half-open intervals delimited by the threshold c , resulting in this type of weighting function. Different weighting functions exist, and we refer the reader to Gneiting and Ranjan (2011); Allen (2024) for more information regarding this topic. The above expression for the twCRPS is, again, difficult to evaluate analytically, so we resort to the expectation formula for a tractable alternative

$$\text{twCRPS}(p, o) = \mathbb{E}_p |v_c(p) - v_c(o)| - \frac{1}{2} \mathbb{E}_{p,p'} |v_c(p) - v_c(p')|,$$

$$v(x) = \max(x, c).$$

680 The choice of chaining function $v_c(x)$ (Allen, 2024) is dependent on $w_c(x)$. We consider the twCRPSS at a specific threshold c

$$\text{twCRPSS}_c^{\text{model}} = \left(1 - \frac{\langle \text{twCRPS}_c \rangle^{\text{model}}}{\langle \text{twCRPS}_c \rangle^{\text{baseline}}} \right) 100,$$

where the twCRPS is averaged over all stations, run times, and lead times.



D3 Quantile loss

QL is defined as

$$685 \quad \text{QL}(F^{-1}, o) = \sum_{k=1}^K \rho(F^{-1}(\tau_k), o), \quad (\text{D5})$$

$$\rho(q_\tau, o) = \begin{cases} (o - q_\tau)(\tau - 1), & o - q_\tau \leq 0 \\ (o - q_\tau)\tau, & \text{otherwise,} \end{cases} \quad (\text{D6})$$

for K quantiles $\tau_k \in (0, 1)$. Analogously to the CRPSS and twCRPSS, the QSS is defined as

$$\text{QSS}_{\tau_i}^{\text{model}} = \left(1 - \frac{\langle \text{QL}_{\tau_i} \rangle^{\text{model}}}{\langle \text{QL}_{\tau_i} \rangle^{\text{baseline}}} \right) 100,$$

The QL is averaged over all stations, run times, and lead times.

690 D4 Brier score

The BS is defined as

$$\text{BS}(F, o) = (1 - F(c) - \mathbb{1}_{o > c})^2, \quad (\text{D7})$$

it measures the ability of the forecaster to predict events where observations exceed a specified threshold c . For example, the BS for $c = 0$ would evaluate the performance of the model to forecast events where precipitation actually occurs. Again, we

695 investigate BSS for a specific threshold

$$\text{BSS}_c^{\text{model}} = \left(1 - \frac{\langle \text{BS}_c \rangle^{\text{model}}}{\langle \text{BS}_c \rangle^{\text{baseline}}} \right) 100,$$

where the BS is averaged over all stations, run times, and lead times.

D5 Rank histogram

The define the discrepancy between the predicted forecast density at each rank and the uniform histogram density as

$$700 \quad \mathcal{R} = \sum_{m=1}^{M+1} \left(1 - \left| r_m - \frac{1}{M+1} \right| (M+1) \right) 100, \quad (\text{D8})$$

where r_m denotes the density at rank m .

D6 Energy score

The ES is defined as

$$\text{ES}(\mathbf{p}, \mathbf{o}) = \mathbb{E}_{\mathbf{p}} \|\mathbf{p} - \mathbf{o}\|_2 - \frac{1}{2} \mathbb{E}_{\mathbf{p}, \mathbf{p}'} \|\mathbf{p} - \mathbf{p}'\|_2, \quad (\text{D9})$$



705 where $\mathbf{p}$ and $\mathbf{p}'$ represent identically distributed independent samples from the post-processed joint distribution and $\|\cdot\|_2$ the n -dimensional Euclidean norm. Using the above formulation, we define the energy skill score as

$$\text{ESS}^{\text{model}} = \left(1 - \frac{\langle \text{ES}^{\text{model}} \rangle}{\langle \text{ES}^{\text{baseline}} \rangle}\right) 100,$$

averaged over all stations and run times.

D7 Diebold-Mariano test

710 Given two sets of forecast scores for two models h and g at station s , the individual set is denoted as $\{\mathcal{S}_i^{(f)}\}_{i=1}^{R \times T}$. The i th score differential is thus defined by $\delta_i = \mathcal{S}_i^{(f)} - \mathcal{S}_i^{(g)}$. The DM test statistic can then be computed by

$$z = \sqrt{R \times T} \frac{\bar{\delta}}{\hat{\sigma}}, \quad \bar{\delta} = \frac{1}{R \times T} \sum_{i=1}^{R \times T} \delta_i, \quad (\text{D10})$$

and is approximately standard normal in case of equal predictive performance. The denominator holds the heteroscedastic and autocorrelation consistent estimator (Amisano and Giacomini, 2007) of the asymptotic variance, defined by

$$715 \quad \hat{\sigma}^2 = \hat{\gamma}_0 + 2 \sum_{j=1}^{k-1} \hat{\gamma}_j, \quad \hat{\gamma}_j = \frac{1}{R \times T} \sum_{i=j+1}^{R \times T} (\delta_i - \bar{\delta})(\delta_{i-j} - \bar{\delta}).$$

The time lag window at which we consider autocorrelation is denoted by k . Since temporal correlation is present in our signal and consecutive daily run times overlap in target lead times, we set the lag time window to equal $k = 5T$, which covers the maximal extent of overlap between a forecast at run time r and $r + 5$, with the lead time spanning 5 days. The abrupt truncation of the lag window did not introduce any negative variance estimates and therefore does not apply Bartlett weights at each lag j . In accordance with Feik et al. (2024), we apply the Benjamini-Hochberg correction (Benjamini and Hochberg, 1995) for selecting the appropriate p-value at each station separately. We report the ratio of rejected and accepted null hypothesis outcomes for each model pair based on the CRPS.

E Hyperparameters

All compared methods, TC-ANet, BQN, and MBM, use the same training procedure hyperparameters. All methods use the AdamW (Loshchilov and Hutter, 2019) solver with a learning rate of 10^{-3} , weight decay of 10^{-8} , and tolerance of 10. The tolerance is increased to 20 for the training of TC-ANet's joint decoder model, since we observed that it requires longer to converge than its marginal counterpart. The feature sizes of individual layers for TC-ANet are recorded in Table E1. The number of parameters for the BQN and MBM backbone is recorded in Table E2.



Attention block embedding size	Marginal decoder embedding size	Joint decoder channel size	Joint decoder kernel size	Joint decoder embedding size	Total parameter count
32	128	32	3	32	36659

Table E1. The individual module parameter counts for the TC-ANet model. These values were determined by a grid search over multiple parameter combinations.

Number of fully connected layers	Per layer feature count	Total parameter count
3	128	31961

Table E2. Number of fully connected layers and features per layer for the neural network backbone of the BQN and MBM methods.

Author contributions. PM has developed and implemented TC-ANet, implemented the other baseline methods, and evaluation metrics. PM wrote the manuscript. JFP provided advice and guidance and helped improve the manuscript.

Competing interests. The authors declare no competing interests.

Acknowledgements. The authors would like to thank the Academic and Research Network of Slovenia (ARNES) and the Slovenian National Supercomputing Network (SLING) consortium for the ARNES computing cluster which was used for all computations. Furthermore, we would like to thank our colleagues at the Slovenian Environment Agency for their time and suggestions which improved the quality of the final work. This work was supported by the Slovenian Research and Innovation Agency (ARIS) research core funding P2-0209 (Jana Faganeli Pucer).



References

- Allen, S.: Weighted scoringRules: Emphasizing Particular Outcomes When Evaluating Probabilistic Forecasts, *Journal of Statistical Software*, 110, 1–26, <https://doi.org/10.18637/jss.v110.i08>, 2024.
- 740 Amisano, G. and Giacomini, R.: Comparing Density Forecasts via Weighted Likelihood Ratio Tests, *Journal of Business & Economic Statistics*, 25, 177–190, <https://doi.org/10.1198/073500106000000332>, 2007.
- Bauer, P., Thorpe, A., and Brunet, G.: The quiet revolution of numerical weather prediction, *Nature*, 525, 47–55, <https://doi.org/10.1038/nature14956>, 2015.
- Benjamini, Y. and Hochberg, Y.: Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing, *Journal of the Royal Statistical Society. Series B (Methodological)*, 57, 289–300, <http://www.jstor.org/stable/2346101>, 1995.
- 745 Berrocal, V. J., Raftery, A. E., and Gneiting, T.: Probabilistic quantitative precipitation field forecasting using a two-stage spatial model, *The Annals of Applied Statistics*, 2, 1170 – 1193, <https://doi.org/10.1214/08-AOAS203>, 2008.
- Bhend, J., Dabernig, M., Demaeyer, J., Mestre, O., and Taillardat, M.: EUPPBench postprocessing benchmark dataset - station data, <https://doi.org/10.5281/zenodo.7708362>, 2023.
- 750 Bouallègue, Z. B., Weyn, J. A., Clare, M. C. A., Dramsch, J., Dueben, P., and Chantry, M.: Improving Medium-Range Ensemble Weather Forecasts with Hierarchical Ensemble Transformers, *Artificial Intelligence for the Earth Systems*, 3, e230027, <https://doi.org/10.1175/AIES-D-23-0027.1>, 2024.
- Bremnes, J. B.: Constrained Quantile Regression Splines for Ensemble Postprocessing, *Monthly Weather Review*, 147, 1769 – 1780, <https://doi.org/10.1175/MWR-D-18-0420.1>, 2019.
- 755 Bremnes, J. B.: Ensemble Postprocessing Using Quantile Function Regression Based on Neural Networks and Bernstein Polynomials, *Monthly Weather Review*, 148, 403 – 414, <https://doi.org/10.1175/MWR-D-19-0227.1>, 2020.
- Bremnes, J. B.: Censored Bernstein Quantile Networks for Probabilistic Precipitation Forecasting, in: EGU General Assembly 2023, Copernicus Meetings, Vienna, Austria, <https://doi.org/10.5194/egusphere-egu23-13849>, eGU23-13849, 2023.
- Bremnes, J. B., Nipen, T. N., and Seierstad, I. A.: Evaluation of forecasts by a global data-driven weather model with and without probabilistic
- 760 post-processing at Norwegian stations, *Nonlinear Processes in Geophysics*, 31, 247–257, 2024.
- Brier, G. W.: Verification of forecasts expressed in terms of probability, *Monthly Weather Review*, 78, 1 – 3, [https://doi.org/10.1175/1520-0493\(1950\)078<0001:VOFEIT>2.0.CO;2](https://doi.org/10.1175/1520-0493(1950)078<0001:VOFEIT>2.0.CO;2), 1950.
- Bülte, C., Maskey, S., Scholl, P., von Berg, J., and Kutyniok, G.: Graph Neural Networks for Enhancing Ensemble Forecasts of Extreme Rainfall, in: ICLR 2025 Workshop on Tackling Climate Change with Machine Learning, <https://www.climatechange.ai/papers/iclr2025/>
- 765 38, 2025.
- Campbell, A., Yim, J., Barzilay, R., Rainforth, T., and Jaakkola, T.: Generative flows on discrete state-spaces: enabling multimodal flows with applications to protein co-design, in: Proceedings of the 41st International Conference on Machine Learning, ICML’24, JMLR.org, 2024.
- Chen, J., Janke, T., Steinke, F., and Lerch, S.: Generative machine learning methods for multivariate ensemble postprocessing, *The Annals of Applied Statistics*, 18, 159–183, 2024.
- 770 Chen, R. T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D.: Neural Ordinary Differential Equations, <https://arxiv.org/abs/1806.07366>, 2019.



- Demaeyer, J., Bhend, J., Lerch, S., Primo, C., Van Schaeybroeck, B., Atencia, A., Ben Bouallègue, Z., Chen, J., Dabernig, M., Evans, G., Faganelli Pucer, J., Hooper, B., Horat, N., Jobst, D., Merše, J., Mlakar, P., Möller, A., Mestre, O., Taillardat, M., and Vannitsem, S.: The EUPPBench postprocessing benchmark dataset v1.0, *Earth System Science Data*, 15, 2635–2653, <https://doi.org/10.5194/essd-15-2635-2023>, 2023.
- Diebold, F. X. and Mariano, R. S.: Comparing Predictive Accuracy, *Journal of Business & Economic Statistics*, 13, 253–263, <http://www.jstor.org/stable/1392185>, 1995.
- Durkan, C., Bekasov, A., Murray, I., and Papamakarios, G.: Neural Spline Flows, <https://arxiv.org/abs/1906.04032>, 2019.
- EUPPBench: EUPPBench dataset, https://eupp-benchmark.github.io/EUPPBench-doc/files/EUPPBench_datasets.html, 2023.
- Feik, M., Lerch, S., and Stühmer, J.: Graph Neural Networks and Spatial Information Learning for Post-Processing Ensemble Weather Forecasts, <https://arxiv.org/abs/2407.11050>, 2024.
- Gneiting, T. and Raftery, A. E.: Strictly Proper Scoring Rules, Prediction, and Estimation, *Journal of the American Statistical Association*, 102, 359–378, <https://doi.org/10.1198/016214506000001437>, 2007.
- Gneiting, T. and Ranjan, R.: Comparing Density Forecasts Using Threshold- and Quantile-Weighted Scoring Rules, *Journal of Business & Economic Statistics*, 29, 411–422, <https://doi.org/10.1198/jbes.2010.08110>, 2011.
- Gneiting, T., Raftery, A. E., Westveld, A. H., and Goldman, T.: Calibrated Probabilistic Forecasting Using Ensemble Model Output Statistics and Minimum CRPS Estimation, *Monthly Weather Review*, 133, 1098 – 1118, <https://doi.org/10.1175/MWR2904.1>, 2005.
- Gneiting, T., Balabdaoui, F., and Raftery, A. E.: Probabilistic forecasts, calibration and sharpness, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69, 243–268, <https://doi.org/https://doi.org/10.1111/j.1467-9868.2007.00587.x>, 2007.
- Grathwohl, W., Chen, R. T. Q., Bettencourt, J., Sutskever, I., and Duvenaud, D.: FFIORD: Free-form Continuous Dynamics for Scalable Reversible Generative Models, <https://arxiv.org/abs/1810.01367>, 2018.
- Hamill, T. M.: Reliability Diagrams for Multicategory Probabilistic Forecasts, *Weather and Forecasting*, 12, 736 – 741, [https://doi.org/10.1175/1520-0434\(1997\)012<0736:RDFMPF>2.0.CO;2](https://doi.org/10.1175/1520-0434(1997)012<0736:RDFMPF>2.0.CO;2), 1997.
- Hamill, T. M.: Interpretation of Rank Histograms for Verifying Ensemble Forecasts, *Monthly Weather Review*, 129, 550 – 560, [https://doi.org/10.1175/1520-0493\(2001\)129<0550:IORHFV>2.0.CO;2](https://doi.org/10.1175/1520-0493(2001)129<0550:IORHFV>2.0.CO;2), 2001.
- Hendrycks, D. and Gimpel, K.: Gaussian Error Linear Units (GELUs), <https://arxiv.org/abs/1606.08415>, 2023.
- Hersbach, H.: Decomposition of the Continuous Ranked Probability Score for Ensemble Prediction Systems, *Weather and Forecasting*, 15, 559 – 570, [https://doi.org/10.1175/1520-0434\(2000\)015<0559:DOTCRP>2.0.CO;2](https://doi.org/10.1175/1520-0434(2000)015<0559:DOTCRP>2.0.CO;2), 2000.
- Höhlein, K., Schulz, B., Westermann, R., and Lerch, S.: Postprocessing of Ensemble Weather Forecasts Using Permutation-Invariant Neural Networks, *Artificial Intelligence for the Earth Systems*, 3, e230 070, <https://doi.org/10.1175/AIES-D-23-0070.1>, 2024.
- Koenker, R. and Bassett, G.: Regression Quantiles, *Econometrica*, 46, 33–50, <http://www.jstor.org/stable/1913643>, 1978.
- Landry, D., Monteleoni, C., and Charantonis, A.: Generating ensembles of spatially coherent in situ forecasts using flow matching, *Quarterly Journal of the Royal Meteorological Society*, 152, e70 055, <https://doi.org/https://doi.org/10.1002/qj.70055>, 2026.
- Lerch, S., Thorarindottir, T. L., Ravazzolo, F., and Gneiting, T.: Forecaster’s Dilemma: Extreme Events and Forecast Evaluation, *Statistical Science*, 32, 106–127, <http://www.jstor.org/stable/26408123>, 2017.
- Lerch, S., Freytag, J., Muschinski, T., and Allen, S.: Enhancing member-by-member post-processing with neural networks, in: *Proceedings of the Copernicus Meetings EGU24, European Geosciences Union Munich, Germany*, 2024.
- Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M.: Flow Matching for Generative Modeling, <https://arxiv.org/abs/2210.02747>, 2023.



- Lipman, Y., Havasi, M., Holderrieth, P., Shaul, N., Le, M., Karrer, B., Chen, R. T. Q., Lopez-Paz, D., Ben-Hamu, H., and Gat, I.: Flow Matching Guide and Code, <https://arxiv.org/abs/2412.06264>, 2024.
- Loshchilov, I. and Hutter, F.: Decoupled Weight Decay Regularization, <https://arxiv.org/abs/1711.05101>, 2019.
- Matheson, J. E. and Winkler, R. L.: Scoring Rules for Continuous Probability Distributions, *Management Science*, 22, 1087–1096, <https://doi.org/10.1287/mnsc.22.10.1087>, 1976.
- 815 Mlakar, P.: petermlakar/TC-ANet: TC-ANet, <https://doi.org/10.5281/zenodo.23262191>, 2026a.
- Mlakar, P.: TC-ANet GitHub repository, <https://github.com/petermlakar/TC-ANet.git>, 2026b.
- Mlakar, P., Merše, J., and Faganeli Pucer, J.: Ensemble weather forecast post-processing with a flexible probabilistic neural network approach, *Quarterly Journal of the Royal Meteorological Society*, 150, 4156–4177, <https://doi.org/10.1002/qj.4809>, 2024.
- 820 Rasp, S. and Lerch, S.: Neural Networks for Postprocessing Ensemble Weather Forecasts, *Monthly Weather Review*, 146, 3885 – 3900, <https://doi.org/10.1175/MWR-D-18-0187.1>, 2018.
- Ravuri, S., Lenc, K., Willson, M., Kangin, D., Lam, R., Mirowski, P., Fitzsimons, M., Athanassiadou, M., Kashem, S., Madge, S., Prudden, R., Mandhane, A., Clark, A., Brock, A., Simonyan, K., Hadsell, R., Robinson, N., Clancy, E., Arribas, A., and Mohamed, S.: Skilful precipitation nowcasting using deep generative models of radar, *Nature*, 597, 672–677, <https://doi.org/10.1038/s41586-021-03854-z>, 2021.
- 825 Schefzik, R.: Ensemble calibration with preserved correlations: unifying and comparing ensemble copula coupling and member-by-member postprocessing, *Quarterly Journal of the Royal Meteorological Society*, 143, 999–1008, 2017.
- Schefzik, R., Thorarinsdottir, T. L., and Gneiting, T.: Uncertainty Quantification in Complex Simulation Models Using Ensemble Copula Coupling, *Statistical Science*, 28, 616 – 640, <https://doi.org/10.1214/13-STS443>, 2013.
- Scheuerer, M. and Hamill, T. M.: Statistical Postprocessing of Ensemble Precipitation Forecasts by Fitting Censored, Shifted Gamma Dis-
 tributions, *Monthly Weather Review*, 143, 4578 – 4596, <https://doi.org/10.1175/MWR-D-15-0061.1>, 2015.
- 830 Scheuerer, M., Switanek, M. B., Worsnop, R. P., and Hamill, T. M.: Using Artificial Neural Networks for Generating Probabilistic Subseasonal Precipitation Forecasts over California, *Monthly Weather Review*, 148, 3489 – 3506, <https://doi.org/10.1175/MWR-D-20-0096.1>, 2020.
- Schulz, B. and Lerch, S.: Machine Learning Methods for Postprocessing Ensemble Forecasts of Wind Gusts: A Systematic Comparison, *Monthly Weather Review*, 150, 235 – 257, <https://doi.org/10.1175/MWR-D-21-0150.1>, 2022.
- 835 Tibshirani, R.: Regression Shrinkage and Selection Via the Lasso, *Journal of the Royal Statistical Society: Series B (Methodological)*, 58, 267–288, <https://doi.org/10.1111/j.2517-6161.1996.tb02080.x>, 1996.
- Van Schaeybroeck, B. and Vannitsem, S.: Ensemble post-processing using member-by-member approaches: theoretical aspects, *Quarterly Journal of the Royal Meteorological Society*, 141, 807–818, <https://doi.org/10.1002/qj.2397>, 2015.
- 840 Vannitsem, S., Bremnes, J. B., Demayer, J., Evans, G. R., Flowerdew, J., Hemri, S., Lerch, S., Roberts, N., Theis, S., Atencia, A., Bouallégue, Z. B., Bhend, J., Dabernig, M., Cruz, L. D., Hieta, L., Mestre, O., Moret, L., Plenković, I. O., Schmeits, M., Taillardat, M., den Bergh, J. V., Schaeybroeck, B. V., Whan, K., and Ylhäisi, J.: Statistical Postprocessing for Weather Forecasts: Review, Challenges, and Avenues in a Big Data World, *Bulletin of the American Meteorological Society*, 102, E681 – E699, <https://doi.org/10.1175/BAMS-D-19-0308.1>, 2021.
- 845 Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I.: Attention is All you Need, in: *Advances in Neural Information Processing Systems*, edited by Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., vol. 30, Curran Associates, Inc., https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf, 2017.



Wessel, J. B., Ferro, C. A. T., and Kwasniok, F.: Lead-time-continuous statistical postprocessing of ensemble weather forecasts, Quarterly
850 Journal of the Royal Meteorological Society, 150, 2147–2167, <https://doi.org/https://doi.org/10.1002/qj.4701>, 2024.