Conformal Prediction for Time Series with Modern Hopfield Networks

Andreas Auer [1] **Martin Gauch** [1 2] **Daniel Klotz** [1] **Sepp Hochreiter** [1 3]

Abstract

To quantify uncertainty, conformal prediction methods are gaining continuously more interest and have already been successfully applied to various domains. However, they are difficult to apply to time series as the autocorrelative structure of time series violates basic assumptions required by conformal prediction. We propose HopCPT, a novel conformal prediction approach for time series that not only copes with temporal structures but leverages them. We show that our approach is theoretically well justified for time series where temporal dependencies are present. In experiments, we demonstrate that our new approach outperforms state-of-the-art conformal prediction methods on multiple real-world time series datasets from four different domains.

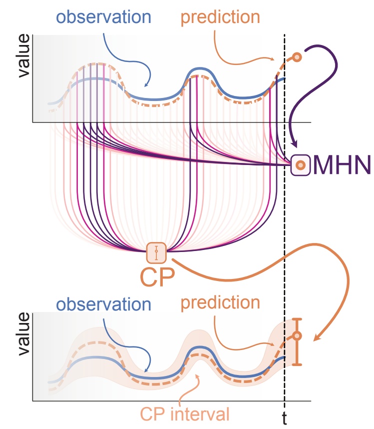

Figure 1. Schematic illustration of HopCPT. The Modern Hopfield Network (MHN) identifies regimes similar to the current one in the time series and weights them (indicated by the colored lines). The weighted information enriches the conformal prediction procedure so that prediction intervals can be derived.

1. Introduction

Uncertainty estimates are imperative to make actionable predictions for complex time-dependent systems (e.g., Gneiting & Katzfuss, 2014; Zhu & Laptev, 2017). This is particularly evident for environmental phenomena such as flood forecasting (e.g., Krzysztofowicz, 2001), since they exhibit pronounced seasonality. Conformal Prediction (CP, Vovk et al., 1999) provides uncertainty estimates based on prediction intervals. CP achieves finite-sample marginal coverage with almost no distributional assumptions, except that the data is exchangeable (Vovk et al., 2005; Vovk, 2012). However, CP for time series is not trivial because temporal dependencies generally violate the exchangeability assumption.

HopCPT. Time series models often exhibit dynamical errors with locally specific behavior. HopCPT uses continuous Modern Hopfield Networks (MHNs) to learn a weighting of similar situations for CP. The CP procedure is then able to use these situations to produce strong uncertainty estimates that exhibit approximately the desired coverage level, and achieve new state-of-the-art efficiency, even with non-exchangeable data. We designed our method for being applicable to large datasets.

Our main contributions are:

1. We propose HopCPT, a CP approach designed for time series prediction, a domain where CP methods struggle with non-exchangeable data.
2. HopCPT uses MHN to introduce the novel technique of similarity-based sample reweighting for time series CP. In contrast to existing approaches, HopCPT can learn from larger datasets and predict intervals at arbitrary coverage levels without retraining.
3. HopCPT achieves state-of-the-art results for conformal time series prediction tasks from various real-world applications.
4. We are the first to provide formal guarantees for uncertainty estimation in streamflow prediction — a domain where uncertainty plays a key role in tasks such as flood forecasting and hydropower management.

[1]ELLIS Unit Linz and LIT AI Lab, Institute for Machine Learning, Johannes Kepler University, Linz, Austria [2]Google Research, Linz, Austria [3]Institute of Advanced Research in Artificial Intelligence (IARAI), Vienna, Austria. Correspondence to: Andreas Auer <auer@ml.jku.at>.

Preprint - Under Review

1.1. Related Work

Regimes. In a world with non-linear dynamics, different environmental conditions lead to different error characteristics of models that predict based on these conditions. If we do not account for these different conditions, temporal changes may lead to unnecessarily large prediction intervals, i.e., to high uncertainty. For example, solar energy production is high and stable on a sunny day, fluctuates during cloudy days, and is zero at night. Often, the current environmental condition was already observed at previous time points. The error at these time points is therefore assumed to have the same distribution as the current error. Following Quandt (1958) and Hamilton (1990), we call the sets of time points with similar environmental condition *regimes*. Although conditional CP is in general impossible (Foygel Barber et al., 2021), we show that conditioning on such regimes can lead to better prediction intervals while preserving the specified coverage.

Applications of regimes. Hamilton (1990) models time series regimes as a discrete Markov process and conditions a classical autoregressive model on the regime states. Sanquer et al. (2012) use a smooth transition approach to model multi-regime time series. Tajeuna et al. (2021) propose an approach to discover and model regime shifts in an ecosystem that comprises multiple time series. Further, Masserano et al. (2022) handle distribution shifts by retraining a forecasting model with training data from a non-uniform adaptive sampling. Although these approaches are not in a CP setting, their work is similar in spirit, as they also follow the general idea to condition on parts of the time series with similar regimes.

CP and extensions. For thorough introductions to CP, we refer the reader to the foundational work of Vovk et al. (1999) and a recent introductory paper by Angelopoulos & Bates (2021). There exist a variety of extensions for CP that go "beyond exchangeability" (Vovk et al., 2005). For example, Papadopoulos & Haralambous (2011) apply CP to a nearest neighbor regression setting, Teng et al. (2022) apply CP to the feature space of models, Angelopoulos et al. (2020) use CP to generate uncertainty sets for image classification tasks, and Toccaceli et al. (2017) use a label-conditional variant to apply CP to biological activity prediction. Of specific interest to us is the research regarding non-exchangeable data of Tibshirani et al. (2019) and Foygel Barber et al. (2022). Both handle potential shifts between the calibration and test set by reweighting the data points. Tibshirani et al. (2019) restrict themselves to settings with full knowledge about the change in distribution; Foygel Barber et al. (2022) rely on fixed weights. In our work, we refrain from this assumption because such information is typically not available in time series prediction.

Another important research direction is the work on normalized conformity scores (see Fontana et al., 2023, and references therein). In this setting, the goal is to adapt the conformal bounds through a scaling factor in the nonconformity function. The work on normalized conformity scores does not explicitly tailor their approaches to time series.

CP for time series. Gibbs & Candes (2021) and Zaffran et al. (2022) account for shifts in sequential data by continuously adapting an internal coverage target. Adaption-based approaches like these are orthogonal to HopCPT and can serve as an enhancement. Stankeviciute et al. (2021) use CP in conjunction with recurrent neural networks in a multi-step prediction setting, assuming that the series of observations is independent. Thus, no weighting of the scores is required. Sun & Yu (2022) introduce CopulaCPTS which applies CP to time series with multivariate targets. They conformalize their prediction based on a copula of the target variables and adapt their calibration set in each step. Jensen et al. (2022) use a bootstrap ensemble to enable CP on time series. NexCP (Foygel Barber et al., 2022) uses exponential decay as the weighting method, arguing that the recent past is more likely to be of the same error distribution. HopCPT can learn this strategy, but does not a priori commit to it. Xu & Xie (2022a) propose EnbPI, which uses quantiles of the k most recent errors for the prediction interval. Additionally, they introduce a novel leave-one-out ensembling technique. This is specifically geared to settings with scarce data and difficult to use for larger datasets, which is why we do not apply it in our experiments. EnbPI is designed around the notion that near-term errors are often independent and identically distributed and therefore exchangeable. SPCI (Xu & Xie, 2022b) softens this requirement by exploiting the autocorrelative structure with a random forest. However, it re-calculates the random forest model at each time step, which is a computational burden that prohibits its application to large datasets. Our approach relaxes the requirement even further, as we do not assume that the data for the interval computations pertains to the k most recent errors.

Continuous Modern Hopfield Networks. MHN are energy-based associative memory networks. They advance conventional Hopfield Networks (Hopfield, 1982) by introducing continuous queries and states via a new energy function. The new energy function leads to exponential storage capacity, while retrieval is possible with a one-step update (Ramsauer et al., 2021). Examples for successful applications of MHN are Widrich et al. (2020); Fürst et al. (2022); Dong et al. (2022); Sanchez-Fernandez et al. (2022); Paischer et al. (2022); Schäfl et al. (2022) and Xu et al. (2022). For more details, we refer to Appendix F.

1.2. Setting

Our setting consists of a multivariate time series $\{(\boldsymbol{x}_t, y_t)\}$, $t = 1, \ldots, T$, with a feature vector $\boldsymbol{x}_t \in \mathbb{R}^m$, a target variable $y_t \in \mathbb{R}$, and a given black-box prediction model $\mu : \mathbb{R}^m \mapsto \mathbb{R}$ that generates a point prediction $\hat{y}_t = \mu(\boldsymbol{X}_t)$. The input feature matrix $\boldsymbol{X}_{t+1}$ can include all previous and current features $\{\boldsymbol{x}_i\}_{i=1}^{t+1}$, as well as all previous targets $\{y_i\}_{i=1}^{t}$. Given the features $\boldsymbol{Z}_{t+1}$, our goal is to construct a corresponding prediction interval $\hat{C}_\alpha(\boldsymbol{Z}_{t+1})$ — a set that includes y_{t+1} at the specified probability $1 - \alpha$. In its basic form, $\boldsymbol{Z}_{t+1}$ will only contain $\hat{y}_{t+1}$, but it can also inherit $\boldsymbol{X}_{t+1}$ or other useful features. Following Vovk et al. (2005), we define the *coverage* as

$$\Pr\left\{Y_{t+1} \in \hat{C}_\alpha\left(\boldsymbol{Z}_{t+1}\right)\right\} \geq 1 - \alpha, \tag{1}$$

where Y_{t+1} is the random variable of the prediction. An infinitely wide prediction interval is 100% reliable, but not informative of the uncertainty. Thus, CP aims to minimize the width of the prediction interval $\hat{C}_\alpha$, while preserving the coverage. A smaller prediction interval is called a more *efficient* interval (Vovk et al., 2005). We use the mean of the interval width over the prediction period (*PI-Width*) as a metric to evaluate the efficiency.

Standard split conformal prediction takes a calibration set of size n which has not been used to train the prediction model μ. For each data sample, it calculates the so-called conformal score (Vovk et al., 2005). In a regression setting, this score often simply corresponds to the absolute error of the prediction (e.g., Foygel Barber et al., 2022). The prediction interval is then calculated based on the empirical $1 - \alpha$ quantile $\mathcal{Q}_{1-\alpha}$ of the calibration scores:

$$\hat{C}_\alpha(\boldsymbol{Z}_{t+1}) = \mu(\boldsymbol{X}_{t+1}) \pm \mathcal{Q}_{1-\alpha}(\{|y_i - \mu(\boldsymbol{X}_i)|\}_{i=1}^n). \tag{2}$$

If the data is exchangeable and μ treats the data points symmetrically, the errors on the test set follow the distribution from the calibration. Hence, the empirical quantiles on the calibration and test set will be approximately equal and it is guaranteed that the interval provides the desired coverage.

If the specified *miscoverage* α differs from the actual marginal miscoverage $\alpha^\star$ in the evaluation, we denote the difference as the *coverage gap*

$$\Delta \text{ Cov} = \alpha - \alpha^\star. \tag{3}$$

The remainder of this manuscript is structured as follows: In Section 2, we present HopCPT alongside a theoretical motivation and a synthetic example that demonstrates the advantages of the approach. In Section 3, we evaluate the performance against state-of-the-art CP approaches and discuss the results. Section 4 gives our conclusions and provides an outlook on potential future work.

2. HopCPT

HopCPT combines the conformal-style quantile selection of errors with a learned similarity-based retrieval using MHN.

2.1. Theoretical Motivation

Foygel Barber et al. (2022) introduced CP with weighted quantiles. In the split conformal setting, the according prediction interval is calculated as

$$\hat{C}_\alpha(\boldsymbol{X}_{t+1}) = \mu(\boldsymbol{X}_{t+1}) \pm \mathcal{Q}_{1-\alpha}\left(\sum_{i=1}^{t} a_i \delta_{\epsilon_i} + a_{t+1}\delta_{+\infty}\right), \tag{4}$$

where μ represents an existing point prediction model, $\mathcal{Q}_\tau$ is the τ-quantile of a distribution, δ_{ϵ_i} is a point mass at $|\epsilon_i|$ (i.e., a probability distribution that has all its mass at $|\epsilon_i|$), where ϵ_i are the errors of the existing prediction model:

$$\epsilon_i = y_i - \mu(\boldsymbol{X}_i). \tag{5}$$

The normalized weight a_i of data sample i is defined as

$$a_i = \begin{cases} \frac{1}{\omega_1 + \ldots + \omega_t + 1} & \text{if } i = t+1, \\ \frac{\omega_i}{\omega_1 + \ldots + \omega_t + 1} & \text{else}, \end{cases} \tag{6}$$

where ω_i are the un-normalized weights of the samples. In the case of $\omega_1 = \ldots = \omega_t = 1$, this corresponds to standard split CP. Given this framework, Foygel Barber et al. (2022) show that Δ Cov can be bounded in a non-exchangeable data setting: Let $D = ((\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_{t+1}, y_{t+1}))$ be a dataset where the last entry represents the test sample, and D^i be a permutation of D which exchanges the test sample at $t + 1$ with the i-th sample. Then, Δ Cov can be bounded from below by the weighted sum of the total variation distances d_{TV} between these permutations in the following way (Foygel Barber et al., 2022):

$$\Delta \text{ Cov} \geq -\sum_{i=1}^{t} a_i \cdot d_{\text{TV}}(D, D^i) \tag{7}$$

If D is a composite of multiple regimes and the test sample is from the same regime as the calibration sample i, then the distance between D and D^i is small. Conversely, the distance might be big if the calibration sample is from a different regime. Note that lower values for ω_i would lead to a tighter bound for Δ Cov (Equations 6 and 7) but also to an increase in the interval width (Equation 4). In a similar but more flexible fashion, HopCPT does not fix the weights a priori. The MHN association resembles direct estimates of a_i — dynamically assigning high values to samples from similar regimes.

2.2. Associative Soft-selection for CP

We use a MHN to identify parts of the time series where the conditional error distribution is similar: For time step $t+1$, we query the memory of the past and look for matching patterns. The MHN then provides an association vector $\boldsymbol{a}_{t+1}$ that allows to soft-select the relevant periods of the memory. The selection procedure is analogous to a k-nearest neighbor classifier for hard selection, but it has the advantage that the similarity measure can be learned. Formally, the soft-selection is defined as:

$$\boldsymbol{a}_{t+1} = \mathrm{softmax}\big(\beta\, m(\boldsymbol{Z}_{t+1})\, \boldsymbol{W}_q\, \boldsymbol{W}_k\, m(\boldsymbol{Z}_{1:t})\big), \tag{8}$$

where m is an encoding network (Section 2.3) that transforms the raw time series features of the current step $\boldsymbol{Z}_{t+1}$ to the query pattern and the memory $\boldsymbol{Z}_{1:t}$ to the stored key patterns; $\boldsymbol{W}_q$ and $\boldsymbol{W}_k$ are the learned transformations which are applied before associating the query with the memory; β is a hyperparameter that controls the softmax temperature. As mentioned above, HopCPT uses the softmax to amplify the impact of the data samples that are likely to follow a similar error distribution and to reduce the impact of samples that follow different distributions (see Section 2.1). This error weighting leads not only to more efficient prediction intervals in our experiments, but can also reduce the miscoverage (Section 3.2).

With the soft-selected time steps we can derive the CP interval using the observed errors ϵ. The CP component of HopCPT follows Xu & Xie (2022a). Like them, we use individual quantiles for the upper and lower bound of the prediction interval, calculated from the errors themselves. HopCPT computes the prediction interval $\hat{C}_\alpha$ for time step $t+1$ in the following way:

$$\begin{aligned}\hat{C}_\alpha(\boldsymbol{Z}_{t+1}) = \Big[&\mu(\boldsymbol{X}_{t+1}) + q(\frac{\alpha}{2}, \boldsymbol{Z}_{t+1}),\\ &\mu(\boldsymbol{X}_{t+1}) + q(1-\frac{\alpha}{2}, \boldsymbol{Z}_{t+1})\Big],\end{aligned} \tag{9}$$

where $q(\tau, \boldsymbol{Z}_{t+1}) = \mathcal{Q}_\tau\left(E_{t+1}\right)$ and E_{t+1} is a multiset created by drawing n times from $[\epsilon_i]_{i=1}^{t}$ with corresponding probabilities $[a_{t+1,i}]_{i=1}^{t}$. Standard CP excludes the highest absolute errors from the set that defines the prediction interval (Vovk et al., 2005); Equation 9 instead removes the lowest and highest non-absolute errors from the prediction interval. This can produce more efficient prediction intervals when $\mathbb{E}(\epsilon_i) \neq 0$ in certain error distribution regimes.

2.3. Encoding Network

We embed the raw time series features using a 2-layer fully connected network m^L with ReLU activations and enhance the representation by temporal encoding features $z_{T,t}^{\text{time}}$. The full encoding network is:

$$m(\boldsymbol{Z}_t) = [m^L(\boldsymbol{Z}_t) \mid\mid z_{T,t}^{\text{time}}]. \tag{10}$$

We use a simple temporal encoding to make time dependent notions of similarity learnable, for example, the windowed approach of EnbPI or the exponentially decaying weighting scheme of NexCP:

$$\boldsymbol{z}_{T,t}^{\text{time}} = \frac{t}{T}. \tag{11}$$

2.4. Training Procedure

We partition the split conformal calibration data into training and validation sets. However, training MHN with quantiles is difficult, which is why we use an auxiliary task: Instead of applying the association mechanism from Equation 8 directly, we use the absolute errors as the value patterns of the MHN. This way, the MHN learns to align errors from time steps with similar regime properties. Intuitively, the observed errors from these time steps should work best to predict the current absolute error. We use the mean squared error as loss function (Equation 12). To allow for efficient training, we simultaneously calculate the association of all T time steps within a training split with each other. We mask the association from a time step to itself. The resulting association from each step to each step is $\boldsymbol{A}_{1:T,1:T}$, and the loss function $\mathcal{L}$ is

$$\mathcal{L} = \frac{1}{T} * \|(|\boldsymbol{\epsilon}_{1:T}| - \boldsymbol{A}_{1:T,1:T}|\boldsymbol{\epsilon}_{1:T}|)^2\|_1. \tag{12}$$

The network has the incentive to learn a representation such that the resulting soft-selection focuses on time steps with a similar error distribution. Alternatively, one could use a loss based on sampling from the softmax. This would correspond more closely to the inference procedure. However, it makes training less efficient because each training step only carries information about a single value of ϵ_i. In contrast, $\mathcal{L}$ leads to more sample-efficient training. Choosing $\mathcal{L}$ assumes that it leads to a retrieval of errors from appropriate regimes instead of mixing unrelated errors. Section 3.2 and Appendix A.3 provide evidence that this holds empirically.

2.5. Synthetic Example

The following synthetic example illustrates the advantages of the HopCPT association mechanism for CP. We model a bivariate time series $D = \{(x_t, y_t)\}_{t=1}^{T}$. While y_t serves as the target variable, x_t represents the feature in our prediction. The series is composed of two different regimes: target values are generated by $y_t = 10 + x_t + \mathcal{N}(0, \frac{x_t}{2})$ or $y_t = 10 + x_t + U(-x_t, x_t)$. x_t is constant within a regime (that is, $x = 3$ and $x = 21$). The regimes alternate. For each regime, we sample the number of time steps from the discrete uniform distribution $\mathcal{U}(1, 25)$. We create 1,000 time steps and split the data equally into training, calibration, and test sets. We use a ridge regression model as prediction model μ. HopCPT can identify the time steps of relevant regimes and therefore creates efficient prediction intervals while still preserving the coverage (Figure 2). EnbPI, SPCI, and NexCP focus only on the recent time steps and thus fail to base their intervals on information from the correct regime. Whenever the regime changes from small to high errors, EnbPI propagates the small error signal and therefore loses coverage. Similarly, its prediction intervals for the small error regime are inefficient (Figure 2, row 3). SPCI and NexCP cannot properly select the relevant time steps, either. They do not lose coverage, but produce wide intervals for all time steps (Figure 2, rows 4 and 5). Lastly, if we replace the MHN with a kNN, it can retrieve information from similar regimes. However, its naive retrieval mechanism fails to focus on the informative features because it cannot learn them (Figure 2, row 2).

3. Experiments

This section provides a comparative evaluation of HopCPT and a qualitative analysis of its association mechanism.

3.1. Setup

Datasets. We use datasets from four different domains: (a) Three solar radiation datasets from the US National Solar Radiation Database (Sengupta et al., 2018). The smallest one consists of 8 time series, each from a different location, over a period of 84 days. This dataset is also used in Xu & Xie (2022a;b). In addition, we evaluate on a 1-year and a 3-year dataset, with 50 time series each. (b) An air quality dataset from Beijing, China (Zhang et al., 2017). It consists of 12 time series, each from a different measurement station, over a period of 4 years. The dataset has two prediction targets, the PM10 (as in Xu & Xie, 2022a;b) and PM2.5 concentrations, which we evaluate separately. (c) Sap flow[1] measurements from the Sapfluxnet data project (Poyatos et al., 2021). Since the individual measurement series are considerably heterogeneous in length, we use a subset of 24 time series, each with between 15,000 and 20,000 data points and varying sampling rates. (d) Streamflow, a dataset of water flow measurements and corresponding meteorologic observations from 531 rivers across the continental United States (Newman et al., 2015; Addor et al., 2017). The measurements span 28 years at a daily time scale. For more detailed information about the datasets see Appendix B.

[1]Sap flow refers to the movement of water within a plant. In the environmental sciences, it is commonly used as a proxy for plant transpiration.

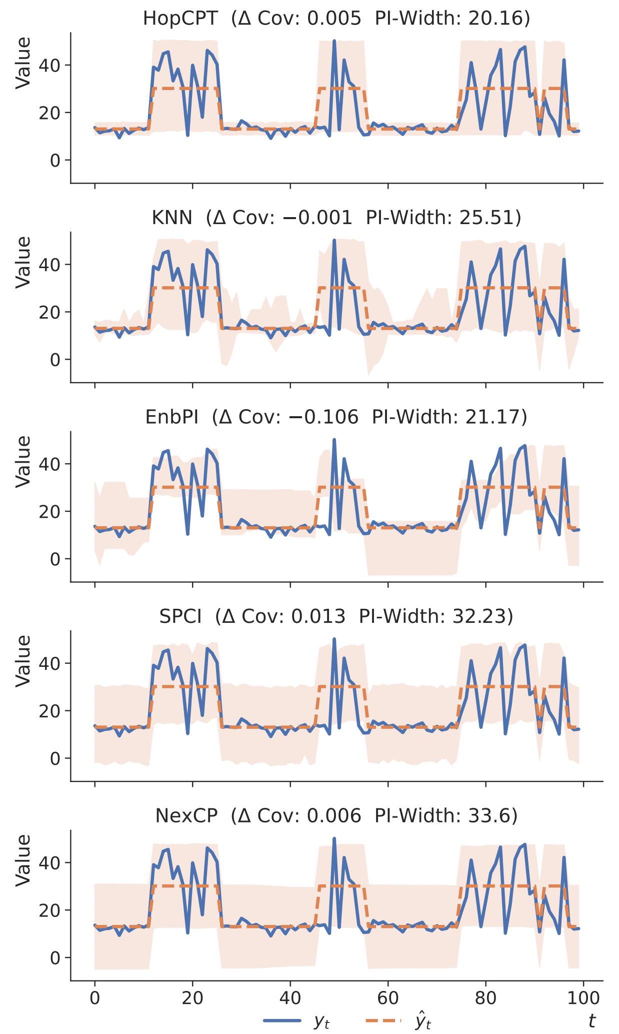

Figure 2. Different approaches for our synthetic example. HopCPT has the smallest width of the prediction intervals (PI-width), while maintaining a coverage close to the specified one (i.e., a Δ Cov that is positive and close to zero).

Prediction models. We use four prediction models for the solar radiation, the air quality, and the sap flux datasets to ensure that our results generalize: a random forest, a LightGBM, a ridge regression model, and a Long Short-Term Memory (LSTM) model. For the former three models we follow the related work (Xu & Xie, 2022a;b; Foygel Barber et al., 2022) and train a separate prediction model for each individual time series. The random forest and LightGBM models are implemented with the darts library (Herzen et al., 2022), the ridge regression model with sklearn (Pedregosa et al., 2011). For the LSTM model we instead train a global model on all time series of a dataset, as is standard for state of the art deep learning models (e.g., Oreshkin et al., 2020; Salinas et al., 2020; Smyl, 2020). The LSTM is implemented with PyTorch (Paszke et al., 2019). For the streamflow dataset, we deviate from this scheme and instead only use the state-of-the-art model, which is an LSTM network (Hochreiter & Schmidhuber, 1997; Kratzert et al., 2021, see Appendix B).

Compared approaches. We compare HopCPT to different state-of-the-art CP approaches for time series data: EnbPI (Xu & Xie, 2022a), SPCI (Xu & Xie, 2022b), NexCP (Foygel Barber et al., 2022), CopulaCPTS (Sun & Yu, 2022), and AdaptiveCI[2] (Gibbs & Candes, 2021). In addition, the results of standard split CP (CP) serve as a baseline, which, for the LSTM base predictor, corresponds to CF-RNN (Stankeviciute et al., 2021) in our setting (one-step, univariate target variable). Appendix A.1 describes the hyperparameter search that we conducted for each method. For SPCI, an adaption of the original algorithm was necessary to provide scalability to larger datasets. Appendix A.2 provides more details and an empirical justification. In Appendix E we additionally evaluate the addition of AdaptiveCI (Gibbs & Candes, 2021) as an enhancement to HopCPT and the other time series CP methods. Lastly, Appendix C presents a supplemental comparison to kNN that shows the superiority of the learned similarity representation in HopCPT.

Metrics. In our analyses, we compute Δ Cov, PI-Width (Section 1.2), and the Winkler score (Winkler, 1972) per time series and miscoverage level. The Winkler score jointly elicitates miscoverage and interval width in a single metric:

$$\mathrm{WS}_\alpha(\boldsymbol{Z}_t, y_t) = \begin{cases} \mathrm{IW}_\alpha(\boldsymbol{Z}_t) + \frac{2}{\alpha}(\hat{C}^l_\alpha(\boldsymbol{Z}_t) - y_t) & \text{if } y_t < \hat{C}^l_\alpha(\boldsymbol{Z}_t), \\ \mathrm{IW}_\alpha(\boldsymbol{Z}_t) + \frac{2}{\alpha}(y_t - \hat{C}^u_\alpha(\boldsymbol{Z}_t)) & \text{if } y_t > \hat{C}^u_\alpha(\boldsymbol{Z}_t), \\ \mathrm{IW}_\alpha(\boldsymbol{Z}_t) & \text{else.} \end{cases} \tag{13}$$

The score is calculated per time step t and miscoverage level α. It corresponds to the interval width $\mathrm{IW}_\alpha = \hat{C}^u_\alpha - \hat{C}^l_\alpha$ whenever the observed value y_t is between the upper bound $\hat{C}^u_\alpha(\boldsymbol{Z}_t)$ and the lower bound $\hat{C}^l_\alpha(\boldsymbol{Z}_t)$ of $\hat{C}_\alpha(\boldsymbol{Z}_t)$. If y_t is outside these bounds, a penalty is added to the interval width. We evaluate the mean Winkler score over all time steps.

We repeated each experiment with 12 different seeds. For brevity, we only show the mean performance of one dataset per domain for $\alpha = 0.1$ in the main paper (which is the most commonly reported value in the CP literature; e.g., Xu & Xie, 2022b; Foygel Barber et al., 2022; Gibbs & Candes, 2021). Appendix A.3 presents additional results for all datasets and more α levels.

[2] AdaptiveCI works on top of an existing quantile prediction method. Hence, we exclusively make comparisons based on the LightGBM models that can predict quantiles instead of point estimates. The approach is orthogonal to the remaining compared models and could be combined with HopCPT.

3.2. Results & Discussion

HopCPT has the most efficient prediction intervals for each domain — with only one exception (Table 1; significance tested with a Mann–Whitney U test at $p < 0.005$) for the evaluated miscoverage level ($\alpha = 0.1$). In multiple experiments (Solar (3Y), Solar (1Y)), the HopCPT prediction intervals are less than half as wide as those of the approach with the second-smallest PI-Width. The second-most efficient intervals are predicted most often by SPCI. This ranking also holds for the Winkler score, where HopCPT achieves the best (i.e., lowest) Winkler scores and SPCI ranks second in most experiments. Notably, these results reflect the increasing requirements posed by each method on the data (Section 1.1).

Further, HopCPT outperforms the other methods regarding both Winkler score and PI-Width when we evaluate over additional datasets and at different miscoverage levels (see Appendix A.3). HopCPT is the best-performing approach in all cases, except for the smallest of all datasets. The limited amount of training data appears to hinder the MHN from learning generalizable retrieval patterns. We argue that this is not a limitation of similarity-based CP but rather an artifact of dataset size. In fact, a simplified and almost parameter-free variation of HopCPT, which replaces the MHN with kNN retrieval, performs best on this small dataset, as we show in Appendix C.

All approaches report a delta coverage close to zero — in other words, they approximately achieve the specified marginal coverage level. This is also reflected by the fact that the ranking of Winkler scores and PI-Widths agree in most evaluations. Appendix A.4 provides a supplementary analysis of the local coverage.

Interestingly, standard CP also achieves good coverage for all experiments at the cost of inefficient prediction intervals. However, there is one notable exception: for the Sap flow dataset with ridge regression, we find Δ Cov $= -0.358$. In this case, we argue that the bad performance of standard CP is driven by a strong violation of the (required) exchangeability assumption. Specifically, a trend in the errors leads to a distribution shift over time (as illustrated in Appendix A.3, Figure 4). HopCPT, EnbPI, and NexCP handle this shift without substantial loss in coverage. The inferior coverage of SPCI is likely influenced by the modification to larger datasets (see Appendix A.2).

Performance gaps between the approaches differ considerably across datasets, while they are generally consistent across prediction models. The biggest differences between the best and worst methods exist in Solar (3Y) and Sap flow, likely due to the strongly distinctive regimes in these datasets. The smallest (but still significant) differences are visible in the Streamflow data. On this dataset, we also eval-

Table 1. Performance of the evaluated CP algorithms for the Solar (3Y), Air Quality (10PM), Sap flow, and Streamflow datasets. The specified miscoverage level is $\alpha = 0.1$ for all experiments. The column FC specifies the prediction algorithm used for the experiment (Forest: Random Forest, LGBM: LightGBM, Ridge: Ridge Regression, LSTM: LSTM neural network). Bold numbers correspond to the best result for the respective metric in the experiment (PI-Width and Winkler score). The error term represents the standard deviation over repeated runs with different seeds (results without an error term are from deterministic models).

Data	FC	UC	HopCPT	SPCI	EnbPI	NexCP	CopulaCPTS	CP/CF-RNN	AdaptiveCI
Solar 3Y	Forest	Δ Cov	$0.029^{\pm 0.012}$	$0.012^{\pm 0.000}$	-0.031	-0.002	0.005	0.004	
		PI-Width	**39.0**$^{\pm 6.2}$	$103.1^{\pm 0.1}$	131.1	166.6	174.9	174.6	
		Winkler	**0.73**$^{\pm 0.20}$	$1.74^{\pm 0.00}$	2.47	2.53	2.75	2.76	
	LGBM	Δ Cov	$0.001^{\pm 0.003}$	$0.014^{\pm 0.000}$	-0.023	-0.002	0.006	0.006	0.001
		PI-Width	**37.7**$^{\pm 0.7}$	$102.2^{\pm 0.1}$	133.6	159.9	169.8	170.2	67.1
		Winkler	**0.57**$^{\pm 0.01}$	$1.75^{\pm 0.00}$	2.52	2.55	2.80	2.81	1.19
	Ridge	Δ Cov	$0.040^{\pm 0.001}$	$0.002^{\pm 0.000}$	-0.074	-0.001	0.004	0.005	
		PI-Width	**44.9**$^{\pm 0.5}$	$108.2^{\pm 0.0}$	131.1	171.0	166.0	167.7	
		Winkler	**0.64**$^{\pm 0.00}$	$1.82^{\pm 0.00}$	2.49	2.66	2.73	2.74	
	LSTM	Δ Cov	$0.001^{\pm 0.006}$	$0.014^{\pm 0.000}$	-0.018	-0.001	0.007	0.007	
		PI-Width	**17.9**$^{\pm 0.6}$	$27.7^{\pm 0.0}$	24.6	28.2	31.9	33.0	
		Winkler	**0.30**$^{\pm 0.01}$	$0.62^{\pm 0.00}$	0.64	0.63	0.68	0.70	
Air 10 PM	Forest	Δ Cov	$0.028^{\pm 0.019}$	$0.008^{\pm 0.000}$	-0.066	-0.004	-0.019	-0.033	
		PI-Width	**93.9**$^{\pm 11.1}$	$118.5^{\pm 0.1}$	202.8	263.5	243.1	229.8	
		Winkler	**1.50**$^{\pm 0.09}$	$2.23^{\pm 0.00}$	4.16	4.03	4.94	4.98	
	LGBM	Δ Cov	$0.017^{\pm 0.016}$	$0.023^{\pm 0.000}$	-0.057	-0.004	-0.017	-0.028	-0.001
		PI-Width	**85.6**$^{\pm 7.4}$	$113.2^{\pm 0.1}$	178.3	224.8	206.7	196.5	186.4
		Winkler	**1.45**$^{\pm 0.06}$	$1.94^{\pm 0.00}$	3.69	3.64	4.33	4.36	3.00
	Ridge	Δ Cov	$0.010^{\pm 0.007}$	$0.024^{\pm 0.000}$	-0.045	-0.002	0.010	0.012	
		PI-Width	**79.9**$^{\pm 4.7}$	$93.9^{\pm 0.1}$	120.0	153.3	153.7	155.3	
		Winkler	**1.35**$^{\pm 0.06}$	$1.52^{\pm 0.00}$	2.60	2.68	2.95	2.96	
	LSTM	Δ Cov	$-0.002^{\pm 0.005}$	$0.010^{\pm 0.000}$	-0.025	-0.002	0.001	0.004	
		PI-Width	$62.7^{\pm 1.5}$	$62.3^{\pm 0.1}$	**58.1**	62.4	61.8	63.0	
		Winkler	$1.33^{\pm 0.01}$	**1.21**$^{\pm 0.00}$	1.32	1.29	1.34	1.34	
Sap flow	Forest	Δ Cov	$-0.027^{\pm 0.028}$	$0.007^{\pm 0.000}$	-0.042	0.000	0.014	0.005	
		PI-Width	**917.8**$^{\pm 57.4}$	$1741.8^{\pm 2.4}$	3671.6	6137.1	7131.1	7201.5	
		Winkler	**0.29**$^{\pm 0.01}$	$0.59^{\pm 0.00}$	1.24	1.56	1.76	1.80	
	LGBM	Δ Cov	$-0.062^{\pm 0.022}$	$0.003^{\pm 0.000}$	-0.040	-0.003	0.006	-0.007	0.010
		PI-Width	**801.2**$^{\pm 41.7}$	$1582.3^{\pm 1.0}$	2924.1	4805.3	5588.5	5614.8	6273.5
		Winkler	**0.28**$^{\pm 0.01}$	$0.49^{\pm 0.00}$	0.96	1.25	1.43	1.46	1.50
	Ridge	Δ Cov	$-0.034^{\pm 0.018}$	$-0.241^{\pm 0.000}$	-0.041	-0.015	-0.251	-0.358	
		PI-Width	**1486.1**$^{\pm 78.6}$	$2060.5^{\pm 1.6}$	3117.5	10628.9	8943.3	7148.7	
		Winkler	**0.41**$^{\pm 0.02}$	$2.52^{\pm 0.00}$	0.92	2.42	3.31	5.85	
	LSTM	Δ Cov	$0.004^{\pm 0.004}$	$0.004^{\pm 0.000}$	-0.019	-0.000	-0.022	-0.042	
		PI-Width	**594.3**$^{\pm 7.7}$	$628.6^{\pm 0.8}$	768.0	990.0	903.9	817.2	
		Winkler	**0.19**$^{\pm 0.01}$	$0.24^{\pm 0.00}$	0.28	0.32	0.35	0.36	
Streamflow	LSTM	Δ Cov	$0.001^{\pm 0.041}$	$0.027^{\pm 0.000}$	-0.054	-0.000	0.005	0.009	
		PI-Width	**1.39**$^{\pm 0.17}$	$1.58^{\pm 0.00}$	1.55	1.94	1.99	2.08	
		Winkler	**0.79**$^{\pm 0.03}$	$0.91^{\pm 0.00}$	1.27	1.21	1.28	1.29	

uated the state-of-the-art non-CP uncertainty model in the domain (Klotz et al., 2022, see Appendix A.3) and found that HopCPT outperforms it with respect to both PI-Width and Winkler score.

To assess whether HopCPT learns meaningful associations within regimes, we conducted a qualitative study on the Solar (3Y) dataset. Figure 3 shows that HopCPT retrieves the most highly weighted errors from time steps with similar regimes. The illustrated weighting at the time step with a low prediction value retrieves previous time steps which are also in low-valued regimes. Similarly, Figure 5 (Appendix A.3) suggests that the learned distinction corresponds to the error regimes, which is crucial for HopCPT.

4. Conclusions

We have introduced HopCPT, a novel CP approach for time series tasks. HopCPT uses continuous Modern Hopfield Networks to construct prediction intervals based on previously seen events with similar error distribution regimes. We exploit that similar features lead to similar errors. Associating features with errors identifies regimes with similar error distributions. HopCPT learns this association in the Modern Hopfield Network, which dynamically adjusts its focus on stored previous features according to the current regime.

Our experiments with established and novel datasets show that HopCPT achieves state of the art. It generates more efficient prediction intervals than existing CP methods and approximately preserves coverage, even in non-exchangeable scenarios like time series. HopCPT comes with formal guarantees within the CP framework for uncertainty estimation in real-world applications such as streamflow prediction. Furthermore, HopCPT scales well to large datasets, provides multiple coverage levels after calibration, and shares information across individual time series within a dataset.

Future work comprises: (a) Drawing from multiple time series during the inference phase. HopCPT is already trained on the whole dataset at once, but it might be advantageous to leverage more information during inference as well. (b) Investigating direct training objectives from which learning-based CP might benefit. (c) Using HopCPT beyond time series, as non-exchangeability is also an issue in other domains which limits the applicability of existing CP methods.

Acknowledgements

We would like to thank Angela Bitto-Nemling for discussions during the development of the method. The ELLIS Unit Linz, the LIT AI Lab, and the Institute for Machine Learning, are supported by the Federal State Upper Austria. IARAI is supported by Here Technologies. We thank the projects AI-MOTION (LIT-2018-6-YOU-212), DeepFlood (LIT-2019-8-YOU-213), Medical Cognitive Computing Center (MC3), INCONTROL-RL (FFG-881064), PRIMAL (FFG-873979), S3AI (FFG-872172), DL for GranularFlow (FFG-871302), EPILEPSIA (FFG-892171), AIRI FG 9-N (FWF-36284, FWF-36235), ELISE (H2020-ICT-2019-3 ID: 951847), Stars4Waters (HORIZON-CL6-2021-CLIMATE-01-01). We thank the Audi.JKU Deep Learning Center, TGW LOGISTICS GROUP GMBH, Silicon Austria Labs (SAL), FILL Gesellschaft mbH, Anyline GmbH, Google, ZF Friedrichshafen AG, Robert Bosch GmbH, UCB Biopharma SRL, Merck Healthcare KGaA, Verbund AG, GLS (Univ. Waterloo) Software Competence Center Hagenberg GmbH, TÜV Austria, Frauscher Sensonic, and the NVIDIA Corporation. The computational results presented have been achieved in part using the Vienna Scientific Cluster (VSC) and the In-Memory Supercomputer MACH-2 (operated by the Scientific Computing Administration of JKU Linz).

References

Abbott, L. F. and Arian, Y. Storage capacity of generalized networks. *Physical Review A*, 36:5091–5094, 1987. doi: 10.1103/PhysRevA.36.5091.

Addor, N., Newman, A. J., Mizukami, N., and Clark, M. P. The CAMELS data set: catchment attributes and meteorology for large-sample studies. *Hydrology and Earth System Sciences (HESS)*, 21(10):5293–5313, 2017.

Angelopoulos, A., Bates, S., Malik, J., and Jordan, M. I. Uncertainty sets for image classifiers using conformal prediction. *arXiv preprint arXiv:2009.14193*, 2020.

Angelopoulos, A. N. and Bates, S. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*, 2021.

Baldi, P. and Venkatesh, S. S. Number of stable points for spin-glasses and neural networks of higher orders. *Physical Review Letters*, 58:913–916, 1987. doi: 10.1103/PhysRevLett.58.913.

Bishop, C. M. Mixture density networks. Technical report, Neural Computing Research Group, 1994.

Caputo, B. and Niemann, H. Storage capacity of kernel associative memories. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*, pp. 51–56, Berlin, Heidelberg, 2002. Springer-Verlag.

Chen, H. H., Lee, Y. C., Sun, G. Z., Lee, H. Y., Maxwell, T., and Giles, C. L. High order correlation model for associative memory. *AIP Conference Proceedings*, 151 (1):86–99, 1986. doi: 10.1063/1.36224.

Dong, Z., Chen, Z., and Wang, Q. Retrosynthesis prediction based on graph relation network. In *2022 15th International Congress on Image and Signal Processing,*

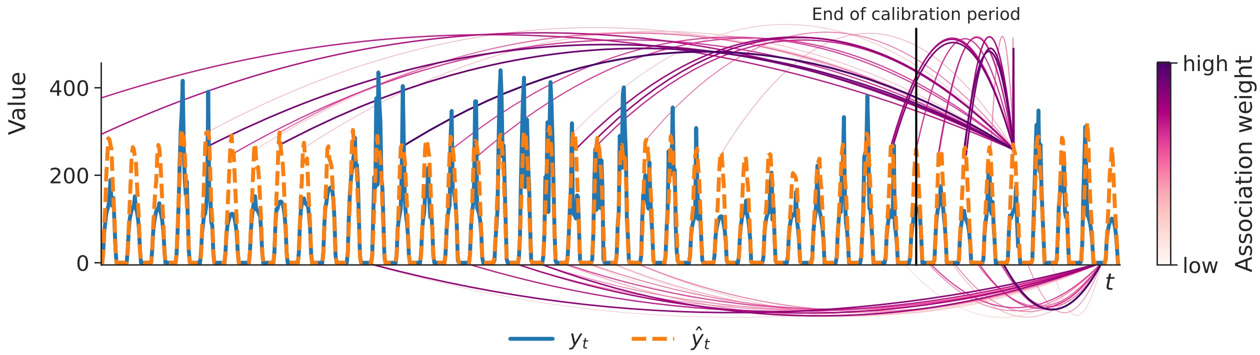

Figure 3. Exemplary visualization of the 30 highest association weights that the MHN places on previous time steps. HopCPT retrieves similar peak values when estimating at a peak, and it retrieves similar small values when estimating at a small value.

BioMedical Engineering and Informatics (CISP-BMEI), pp. 1–5. IEEE, 2022.

Fontana, M., Zeni, G., and Vantini, S. Conformal prediction: a unified review of theory and new challenges. *Bernoulli*, 29(1):1–23, 2023.

Fox, M. and Rubin, H. Admissibility of quantile estimates of a single location parameter. *The Annals of Mathematical Statistics*, pp. 1019–1030, 1964.

Foygel Barber, R., Candes, E. J., Ramdas, A., and Tibshirani, R. J. The limits of distribution-free conditional predictive inference. *Information and Inference: A Journal of the IMA*, 10(2):455–482, 2021.

Foygel Barber, R., Candes, E. J., Ramdas, A., and Tibshirani, R. J. Conformal prediction beyond exchangeability. *arXiv preprint arXiv:2202.13415*, 2022.

Fürst, A., Rumetshofer, E., Lehner, J., Tran, V. T., Tang, F., Ramsauer, H., Kreil, D. P., Kopp, M. K., Klambauer, G., Bitto-Nemling, A., and Hochreiter, S. CLOOB: Modern Hopfield Networks with infoLOOB outperform CLIP. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022.

Gardner, E. Multiconnected neural network models. *Journal of Physics A*, 20(11):3453–3464, 1987. doi: 10.1088/0305-4470/20/11/046.

Gibbs, I. and Candes, E. J. Adaptive conformal inference under distribution shift. *Advances in Neural Information Processing Systems*, 34:1660–1672, 2021.

Gneiting, T. and Katzfuss, M. Probabilistic forecasting. *Annual Review of Statistics and Its Application*, 1(1):125–151, 2014. doi: 10.1146/annurev-statistics-062713-085831.

Hamilton, J. D. Analysis of time series subject to changes in regime. *Journal of econometrics*, 45(1-2):39–70, 1990.

Herzen, J., Lässig, F., Piazzetta, S. G., Neuer, T., Tafti, L., Raille, G., Van Pottelbergh, T., Pasieka, M., Skrodzki, A., Huguenin, N., et al. Darts: User-friendly modern machine learning for time series. *Journal of Machine Learning Research*, 23(124):1–6, 2022.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, 1997.

Hopfield, J. J. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, 1982.

Hopfield, J. J. Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences*, 81(10): 3088–3092, 1984. doi: 10.1073/pnas.81.10.3088.

Horn, D. and Usher, M. Capacities of multiconnected memory models. *Journal of Physics France*, 49(3):389–395, 1988. doi: 10.1051/jphys:01988004903038900.

Jensen, V., Bianchi, F. M., and Anfinsen, S. N. Ensemble conformalized quantile regression for probabilistic time series forecasting. *arXiv preprint arXiv:2202.08756*, 2022.

Klotz, D., Kratzert, F., Gauch, M., Keefe Sampson, A., Brandstetter, J., Klambauer, G., Hochreiter, S., and Nearing, G. Uncertainty estimation with deep learning for rainfall–runoff modeling. *Hydrology and Earth System Sciences*, 26(6):1673–1693, 2022. doi: 10.5194/hess-26-1673-2022.

Kratzert, F., Klotz, D., Shalev, G., Klambauer, G., Hochreiter, S., and Nearing, G. Towards learning universal, regional, and local hydrological behaviors via machine learning applied to large-sample datasets. *Hydrology and Earth System Sciences*, 23(12):5089–5110, 2019. doi: 10.5194/hess-23-5089-2019.

Kratzert, F., Klotz, D., Hochreiter, S., and Nearing, G. S. A note on leveraging synergy in multiple meteorological data sets with deep learning for rainfall–runoff modeling. *Hydrology and Earth System Sciences*, 25(5):2685–2703, 2021.

Kratzert, F., Gauch, M., Nearing, G., and Klotz, D. NeuralHydrology — a Python library for deep learning research in hydrology. *Journal of Open Source Software*, 7(71): 4050, 2022. doi: 10.21105/joss.04050.

Krotov, D. and Hopfield, J. J. Dense associative memory for pattern recognition. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, pp. 1172–1180. Curran Associates, Inc., 2016.

Krzysztofowicz, R. The case for probabilistic forecasting in hydrology. *Journal of hydrology*, 249(1-4):2–9, 2001.

Lei, J. and Wasserman, L. A. Distribution-free prediction bands for non-parametric regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76, 2014.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Masserano, L., Rangapuram, S. S., Kapoor, S., Nirwan, R. S., Park, Y., and Bohlke-Schneider, M. Adaptive sampling for probabilistic forecasting under distribution shift. In *NeurIPS 2022 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2022.

Maurer, E. P., Wood, A. W., Adam, J. C., Lettenmaier, D. P., and Nijssen, B. A long-term hydrologically based dataset of land surface fluxes and states for the conterminous United States. *Journal of climate*, 15(22):3237–3251, 2002.

Newman, A. J., Clark, M. P., Sampson, K., Wood, A., Hay, L. E., Bock, A., Viger, R. J., Blodgett, D., Brekke, L., Arnold, J. R., Hopson, T., and Duan, Q. Development of a large-sample watershed-scale hydrometeorological data set for the contiguous USA: data set characteristics and assessment of regional variability in hydrologic model performance. *Hydrology and Earth System Sciences*, 19 (1):209–223, 2015. doi: 10.5194/hess-19-209-2015.

Newman, A. J., Mizukami, N., Clark, M. P., Wood, A. W., Nijssen, B., and Nearing, G. Benchmarking of a physically based hydrologic model. *Journal of Hydrometeorology*, 18(8):2215–2225, 2017. doi: 10.1175/JHM-D-16-0284.1.

Oreshkin, B. N., Carpov, D., Chapados, N., and Bengio, Y. N-beats: Neural basis expansion analysis for interpretable time series forecasting. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=r1ecqn4YwB.

Paischer, F., Adler, T., Patil, V., Bitto-Nemling, A., Holzleitner, M., Lehner, S., Eghbal-zadeh, H., and Hochreiter, S. History compression via language models in reinforcement learning. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 17156–17185. PMLR, 17–23 Jul 2022.

Papadopoulos, H. and Haralambous, H. Reliable prediction intervals with regression neural networks. *Neural Networks*, 24(8):842–851, 2011.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

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

Poyatos, R., Granda, V., Flo, V., Adams, M. A., Adorján, B., Aguadé, D., Aidar, M. P., Allen, S., Alvarado-Barrientos, M. S., Anderson-Teixeira, K. J., et al. Global transpiration data from sap flow measurements: the sapfluxnet database. *Earth system science data*, 13(6):2607–2649, 2021.

Psaltis, D. and Cheol, H. P. Nonlinear discriminant functions and associative memories. *AIP Conference Proceedings*, 151(1):370–375, 1986. doi: 10.1063/1.36241.

Quandt, R. E. The estimation of the parameters of a linear regression system obeying two separate regimes. *Journal of the american statistical association*, 53(284):873–880, 1958.

Ramsauer, H., Schäfl, B., Lehner, J., Seidl, P., Widrich, M., Gruber, L., Holzleitner, M., Pavlović, M., Sandve, G. K., Greiff, V., Kreil, D., Kopp, M., Klambauer, G., Brandstetter, J., and Hochreiter, S. Hopfield networks is all you need. In *9th International Conference on Learning Representations (ICLR)*, 2021.

Salinas, D., Flunkert, V., Gasthaus, J., and Januschowski, T. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191, 2020.

Sanchez-Fernandez, A., Rumetshofer, E., Hochreiter, S., and Klambauer, G. CLOOME: contrastive learning unlocks bioimaging databases for queries with chemical structures. In *NeurIPS 2022 Women in Machine Learning Workshop*, 2022.

Sanquer, M., Chatelain, F., El-Guedri, M., and Martin, N. A smooth transition model for multiple-regime time series. *IEEE transactions on signal processing*, 61(7):1835–1847, 2012.

Schäfl, B., Gruber, L., Bitto-Nemling, A., and Hochreiter, S. Hopular: Modern Hopfield Networks for tabular data. *arXiv preprint arXiv:2206.00664*, 2022.

Sengupta, M., Xie, Y., Lopez, A., Habte, A., Maclaurin, G., and Shelby, J. The national solar radiation data base (NSRDB). *Renewable and sustainable energy reviews*, 89:51–60, 2018.

Smyl, S. A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *International Journal of Forecasting*, 36(1):75–85, 2020.

Stankeviciute, K., M Alaa, A., and van der Schaar, M. Conformal time-series forecasting. *Advances in Neural Information Processing Systems*, 34:6216–6228, 2021.

Sun, S. and Yu, R. Copula conformal prediction for multi-step time series forecasting. *ArXiv*, abs/2212.03281, 2022.

Tagasovska, N. and Lopez-Paz, D. Single-model uncertainties for deep learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Tajeuna, E. G., Bouguessa, M., and Wang, S. Modeling regime shifts in multiple time series. *arXiv preprint arXiv:2109.09692*, 2021.

Teng, J., Wen, C., Zhang, D., Bengio, Y., Gao, Y., and Yuan, Y. Predictive inference with feature conformal prediction. *arXiv preprint arXiv:2210.00173*, 2022.

Thornton, P. E., Running, S. W., White, M. A., et al. Generating surfaces of daily meteorological variables over large regions of complex terrain. *Journal of hydrology*, 190(3-4):214–251, 1997.

Tibshirani, R. J., Foygel Barber, R., Candes, E. J., and Ramdas, A. Conformal prediction under covariate shift. *Advances in Neural Information Processing Systems*, 32, 2019.

Toccaceli, P., Nouretdinov, I., and Gammerman, A. Conformal prediction of biological activity of chemical compounds. *Annals of Mathematics and Artificial Intelligence*, 81(1):105–123, 2017.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, pp. 5998–6008. Curran Associates, Inc., 2017.

Vovk, V. Conditional validity of inductive conformal predictors. In *Asian conference on machine learning*, pp. 475–490. PMLR, 2012.

Vovk, V., Gammerman, A., and Saunders, C. Machine-learning applications of algorithmic randomness. In Bratko, I. and Dzeroski, S. (eds.), *Proceedings of the Sixteenth International Conference on Machine Learning (ICML 1999), Bled, Slovenia, June 27 - 30, 1999*, pp. 444–453. Morgan Kaufmann, 1999.

Vovk, V., Gammerman, A., and Shafer, G. *Algorithmic learning in a random world*. Springer Science & Business Media, 2005.

Widrich, M., Schäfl, B., Pavlović, M., Ramsauer, H., Gruber, L., Holzleitner, M., Brandstetter, J., Sandve, G. K., Greiff, V., Hochreiter, S., and Klambauer, G. Modern Hopfield Networks and attention for immune repertoire classification. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2020.

Winkler, R. L. A decision-theoretic approach to interval estimation. *Journal of the American Statistical Association*, 67(337):187–191, 1972.

Xia, Y., Mitchell, K., Ek, M., Sheffield, J., Cosgrove, B., Wood, E., Luo, L., Alonge, C., Wei, H., Meng, J., et al. Continental-scale water and energy flux analysis and validation for the North American Land Data Assimilation System project phase 2 (NLDAS-2): 1. intercomparison and application of model products. *Journal of Geophysical Research: Atmospheres*, 117(D3), 2012.

Xu, C. and Xie, Y. Conformal prediction set for time-series. *arXiv preprint arXiv:2206.07851*, 2022a. doi: 10.48550/arXiv.2206.07851.

Xu, C. and Xie, Y. Sequential predictive conformal inference for time series. *arXiv preprint arXiv:2212.03463*, 2022b.

Xu, Y., Yu, W., Ghamisi, P., Kopp, M., and Hochreiter, S. Txt2Img-MHN: Remote sensing image generation from text using Modern Hopfield Networks. *arXiv preprint arXiv:2208.04441*, 2022.

Zaffran, M., Féron, O., Goude, Y., Josse, J., and Dieuleveut, A. Adaptive conformal predictions for time series. In *International Conference on Machine Learning*, pp. 25834–25866. PMLR, 2022.

Zhang, S., Guo, B., Dong, A., He, J., Xu, Z., and Chen, S. X. Cautionary tales on air-quality improvement in Beijing. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2205):20170457, 2017.

Zhu, L. and Laptev, N. Deep and confident prediction for time series at Uber. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 103–110. IEEE, 2017.

A. Extended Experimental Setup & Results

A.1. Hyperparameter Search

We conducted an individual hyperparameter grid search for each predictor–dataset combination. For methods that require calibration data (HopCPT, AdaptiveCI, NexCP), each calibration set was split in half: One part served as actual calibration data while the other half was used for validation.[3] As EnbPI requires only the k past points, we used the full calibration set minus these k points for validation, so that it could fully exploit the available data. Table 2 shows all sets of hyperparameters used for the search.

Table 2. Parameters used in the hyperparameter search.

Method	Parameter	Value
HopCPT	Learning Rate	0.01, 0.001
	Dropout	0, 0.25, 0.5
	Time Encode	yes/no
AdaptiveCI	Mode	simple, momentum
	γ	0.002, 0.005, 0.01, 0.02
EnbPI	Window Length	200, 150, 125, 100, 75, 50, 25, 10
NexCP	ρ	0.999, 0.995, 0.993, 0.99, 0.98, 0.95, 0.90
kNN	k-Top Share	0.025, 0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35

Model selection. Model selection was done uniformly for all algorithms: As a first step, all models with a negative Δ Cov on the validation set were excluded. From the remaining models, the model with the smallest PI-Width was selected. In cases where no model achieved non-negative Δ Cov, the model with the highest Δ Cov was selected.

HopCPT. HopCPT was trained for 3,000 epochs in each experiment. We chose this number so that the loss and model selection metric curves are already converged. Throughout training, we validated every 5 epochs and selected the model that best fulfilled the model selection criteria described above. AdamW (Loshchilov & Hutter, 2019) with standard parameters ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\delta = 0.01$) is used as optimizer. The learning rate was part of the hyperparameter search. Depending on the dataset size, the batch size was set to 2 or 4, where a batch size of n would mean that the full training part of the calibration set of n time series is used.

[3]Note that this is only the case in the hyperparameter search. In the evaluation, the full calibration set was used for calibration.

SPCI. The computational demand of SPCI did not allow to conduct a full hyperparameter search for the window length parameter (see more in Section A.2). Since SPCI applies a random forest on top of the window, one can assume that it is capable to find the relevant parts of the window. On top of that, a longer window has less risk than a window that is too short and (potentially) cuts off relevant information. Hence, we set the window length to 100 for all experiments, which corresponds to the longest setting in the original paper. To check our reasoning, we evaluated the performance of SPCI with window length 25 on all but the largest two datasets (Table 3). We found hardly any differences except for the smallest datasets Solar (3M). In that case we reason that the result is due to the limited calibration data in this setting.

A.2. SPCI Retraining

As our results show, SPCI is very competitive (see, for example, Section 3.2). Xu & Xie (2022b) did, however, design SPCI so that its random forest regressor is retrained after each prediction step. This design choice is computationally prohibitive for larger datasets. To nevertheless allow a comparison against SPCI on the larger dataset, we modified the algorithm so that the retraining is skipped. Experiments on the smallest dataset (Table 4) show only small performance decrease with the modified algorithm. One limitation of the adapted algorithm is, however, that a strong shift in the error distribution(s) would potentially require a retraining on the new distribution. A viable proxy to detect such a change in our setting is the coverage performance of standard CP. The reason for this is that standard CP predictions are solely based on the calibration data and can thus not account for shifts. In the experiments (Section 3 and Appendix A.3), standard CP achieves good coverage with only one exception (we look at this exception in detail in Section 3). Hence, we decided to include SPCI (in the modified version) to enable a comparison to it on larger datasets.

Table 3. Performance of SPCI with an input window of length 100 and length 25. The differences in performance are small compared to the differences across methods (Section 3.2). The error term represents the standard deviation over repeated runs.

Data	FC	Window	100	25
Solar 1Y	Forest	Δ Cov	$0.045^{\pm 0.000}$	$0.047^{\pm 0.000}$
		PI-Width	$97.2^{\pm 0.1}$	$97.8^{\pm 0.3}$
		Winkler	$1.26^{\pm 0.00}$	$1.25^{\pm 0.00}$
	LGBM	Δ Cov	$0.045^{\pm 0.000}$	$0.047^{\pm 0.000}$
		PI-Width	$96.6^{\pm 0.2}$	$97.6^{\pm 0.1}$
		Winkler	$1.26^{\pm 0.00}$	$1.26^{\pm 0.00}$
	Ridge	Δ Cov	$0.031^{\pm 0.000}$	$0.030^{\pm 0.000}$
		PI-Width	$112.9^{\pm 0.2}$	$113.7^{\pm 0.1}$
		Winkler	$1.40^{\pm 0.00}$	$1.42^{\pm 0.00}$
Solar Small	Forest	Δ Cov	$-0.064^{\pm 0.002}$	$-0.024^{\pm 0.000}$
		PI-Width	$38.8^{\pm 0.4}$	$43.2^{\pm 0.2}$
		Winkler	$1.82^{\pm 0.01}$	$1.53^{\pm 0.00}$
	LGBM	Δ Cov	$-0.052^{\pm 0.002}$	$-0.027^{\pm 0.001}$
		PI-Width	$37.4^{\pm 0.2}$	$43.3^{\pm 0.1}$
		Winkler	$1.84^{\pm 0.01}$	$1.63^{\pm 0.00}$
	Ridge	Δ Cov	$-0.055^{\pm 0.002}$	$-0.057^{\pm 0.001}$
		PI-Width	$51.9^{\pm 0.2}$	$52.8^{\pm 0.2}$
		Winkler	$1.91^{\pm 0.02}$	$1.83^{\pm 0.00}$
Air 10 PM	Forest	Δ Cov	$0.008^{\pm 0.000}$	$0.008^{\pm 0.000}$
		PI-Width	$118.5^{\pm 0.0}$	$118.5^{\pm 0.1}$
		Winkler	$2.23^{\pm 0.00}$	$2.23^{\pm 0.00}$
	LGBM	Δ Cov	$0.024^{\pm 0.000}$	$0.023^{\pm 0.000}$
		PI-Width	$113.2^{\pm 0.2}$	$113.2^{\pm 0.1}$
		Winkler	$1.94^{\pm 0.00}$	$1.94^{\pm 0.00}$
	Ridge	Δ Cov	$0.024^{\pm 0.000}$	$0.024^{\pm 0.000}$
		PI-Width	$93.9^{\pm 0.0}$	$93.8^{\pm 0.0}$
		Winkler	$1.52^{\pm 0.00}$	$1.52^{\pm 0.00}$
Air 25 PM	Forest	Δ Cov	$-0.009^{\pm 0.000}$	$-0.009^{\pm 0.000}$
		PI-Width	$81.5^{\pm 0.0}$	$81.4^{\pm 0.0}$
		Winkler	$2.02^{\pm 0.00}$	$2.02^{\pm 0.00}$
	LGBM	Δ Cov	$0.002^{\pm 0.001}$	$0.001^{\pm 0.000}$
		PI-Width	$73.3^{\pm 0.0}$	$73.3^{\pm 0.0}$
		Winkler	$1.84^{\pm 0.00}$	$1.83^{\pm 0.00}$
	Ridge	Δ Cov	$0.010^{\pm 0.000}$	$0.010^{\pm 0.000}$
		PI-Width	$65.5^{\pm 0.0}$	$65.4^{\pm 0.1}$
		Winkler	$1.40^{\pm 0.00}$	$1.40^{\pm 0.00}$
Sap flow	Forest	Δ Cov	$0.007^{\pm 0.000}$	$0.007^{\pm 0.000}$
		PI-Width	$1743.1^{\pm 1.4}$	$1742.7^{\pm 1.2}$
		Winkler	$0.59^{\pm 0.00}$	$0.59^{\pm 0.00}$
	LGBM	Δ Cov	$0.003^{\pm 0.000}$	$0.003^{\pm 0.000}$
		PI-Width	$1581.9^{\pm 0.7}$	$1583.6^{\pm 1.3}$
		Winkler	$0.49^{\pm 0.00}$	$0.49^{\pm 0.00}$
	Ridge	Δ Cov	$-0.241^{\pm 0.000}$	$-0.242^{\pm 0.000}$
		PI-Width	$2061.5^{\pm 1.4}$	$2066.8^{\pm 1.2}$
		Winkler	$2.52^{\pm 0.00}$	$2.52^{\pm 0.00}$

Table 4. Performance of the original SPCI algorithm (Retrain) and the modified version (No Retrain) on the Solar (3M) dataset. Performance in terms of the evaluation metrics is similar compared to the differences between methods (Section 3.2). The computational demand of the modified version is considerably lower. The error term represents the standard deviation over repeated runs.

FC	α	UC	No Retrain	Retrain
Forest	0.05	Δ Cov	$-0.074^{\pm 0.001}$	$-0.049^{\pm 0.001}$
		PI-Width	$58.0^{\pm 0.2}$	$62.4^{\pm 0.0}$
		Winkler	$2.62^{\pm 0.03}$	$2.28^{\pm 0.01}$
	0.10	Δ Cov	$-0.064^{\pm 0.002}$	$-0.045^{\pm 0.003}$
		PI-Width	$38.8^{\pm 0.4}$	$41.6^{\pm 0.0}$
		Winkler	$1.82^{\pm 0.01}$	$1.67^{\pm 0.00}$
	0.15	Δ Cov	$-0.050^{\pm 0.002}$	$-0.038^{\pm 0.003}$
		PI-Width	$26.8^{\pm 0.2}$	$28.9^{\pm 0.0}$
		Winkler	$1.44^{\pm 0.02}$	$1.35^{\pm 0.00}$
LGBM	0.05	Δ Cov	$-0.062^{\pm 0.001}$	$-0.061^{\pm 0.001}$
		PI-Width	$56.1^{\pm 0.4}$	$58.2^{\pm 0.0}$
		Winkler	$2.65^{\pm 0.03}$	$2.51^{\pm 0.01}$
	0.10	Δ Cov	$-0.052^{\pm 0.002}$	$-0.049^{\pm 0.001}$
		PI-Width	$37.4^{\pm 0.2}$	$39.8^{\pm 0.0}$
		Winkler	$1.84^{\pm 0.01}$	$1.78^{\pm 0.01}$
	0.15	Δ Cov	$-0.042^{\pm 0.001}$	$-0.037^{\pm 0.001}$
		PI-Width	$26.9^{\pm 0.3}$	$28.3^{\pm 0.0}$
		Winkler	$1.47^{\pm 0.00}$	$1.42^{\pm 0.00}$
Ridge	0.05	Δ Cov	$-0.063^{\pm 0.001}$	$-0.056^{\pm 0.002}$
		PI-Width	$67.3^{\pm 0.2}$	$68.7^{\pm 0.0}$
		Winkler	$2.47^{\pm 0.01}$	$2.25^{\pm 0.01}$
	0.10	Δ Cov	$-0.055^{\pm 0.002}$	$-0.048^{\pm 0.001}$
		PI-Width	$51.9^{\pm 0.2}$	$54.2^{\pm 0.0}$
		Winkler	$1.91^{\pm 0.02}$	$1.77^{\pm 0.01}$
	0.15	Δ Cov	$-0.039^{\pm 0.003}$	$-0.036^{\pm 0.000}$
		PI-Width	$43.1^{\pm 0.3}$	$45.0^{\pm 0.0}$
		Winkler	$1.59^{\pm 0.00}$	$1.50^{\pm 0.00}$

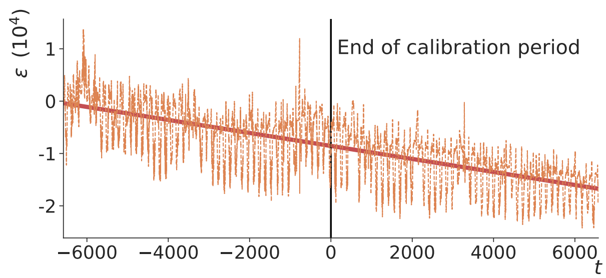

Figure 4. Time series of the prediction error ϵ for the ridge regression model on the Sap flow dataset. The time series spans the calibration ($t < 0$) and test ($t \geq 0$) data. The red line (fitted by the least squares method) shows a strong trend in the error distribution.

A.3. Additional Results

Tables 13–19 show the results for miscoverage levels $\alpha \in \{0.05, 0.10, 0.15\}$ for all evaluated combinations of datasets and predictors. Results for individual time series of the datasets are uploaded to the code repository (Appendix H).

CMAL. For the Streamflow dataset, we additionally compare to CMAL, which is the state of the art non-CP uncertainty estimation technique in the respective domain (Klotz et al., 2022). CMAL is a mixture density network (Bishop, 1994) based on an LSTM that predicts the parameters of asymmetric Laplacian distributions. As our experiments use the same dataset, we adopt the hyperparameters from Klotz et al. (2022) but lower the learning rate to 0.0001 because we train CMAL on more training samples (18 years, i.e., the training and calibration period combined, which allows for a fair comparison against the CP methods). Despite the fact that CMAL is not based on the CP paradigm, it achieves good coverage results, however, at the cost of wide prediction intervals and high Winkler scores (see Table 19).

Error trend. Figure 4 shows the prediction errors of the ridge regression model on the Sap flow dataset. The errors exhibit a strong trend and shift towards a negative expectation value.

Association weights. Figure 5 investigates the association patterns of HopCPT and shows its capabilities to focus on time steps from similar error regimes. The depicted time step in a regime with negative errors retrieves time steps with primarily negative errors and, likewise, the time step in a regime with positive errors retrieves time steps with primarily positive errors of the same magnitude.

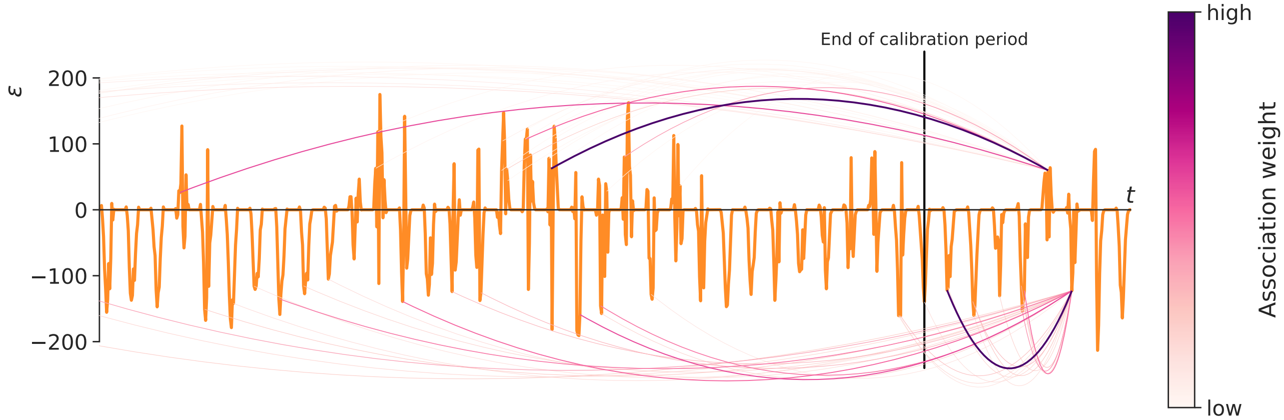

Figure 5. Visualization of the 30 highest association weights that the Hopfield network places on previous time steps. HopCPT retrieves similar error values when predicting at a time of high error, and it retrieves similar, previous small errors when predicting at a time step with small error.

A.4. Local Coverage

Standard CP only provides marginal coverage guarantees and a constant prediction interval width (Vovk et al., 2005). In time series prediction tasks, this can lead to bad local coverage (Lei & Wasserman, 2014). To evaluate whether the coverage approximately holds also locally, we evaluated Δ Cov on windows of size k. To avoid compensation of negative Δ Cov in some windows by other windows with positive Δ Cov, we upper-bounded each window's Δ Cov by zero before averaging over the bounded coverage gaps — i.e., we calculated $\frac{1}{W}\sum_{w=1}^{W} -\Delta\ \mathrm{Cov}_w^{\top 0}$, with

$$\Delta\ \mathrm{Cov}_w^{\top 0} = \begin{cases} \Delta\ \mathrm{Cov}_w & \text{if } \Delta\ \mathrm{Cov}_w \leq 0, \\ 0 & \text{else,} \end{cases} \tag{14}$$

where $\Delta\ \mathrm{Cov}_w$ is the Δ Cov within window w.

Table 5 shows the results of this evaluation for window sizes $k \in \{10, 20, 50\}$ and miscoverage level $\alpha = 0.1$. Depending on the dataset, most often either HopCPT or SPCI perform best. The overall rankings are only partly similar to the evaluation of the marginal coverage (Table 1 and Appendix A.3). Especially standard CP, which achieves competitive results in marginal coverage, falls short in this comparison. Only for Solar (3M), where the approach achieves a high marginal Δ Cov, it preserves the local coverage best. Note that this comes with the drawback of very wide prediction intervals, i.e., bad efficiency. Overall, the results show that HopCPT and other time series CP methods improve the local coverage in non-exchangeable time series settings, compared to standard CP.

A.5. Negative Results

While developing our approach we tested several configurations and adaptions. In the following, we briefly describe those that did not work (so that potential future research can avoid these paths):

- **Pinball Loss**. We tried to train the MHN with a pinball loss (the original use seems to stem from Fox & Rubin, 1964, we are however not aware of the naming origin) in many different variations (see points that follow), but consistently got worse results than with the mean squared error.
  - Inspired by the ideas lined out by Tagasovska & Lopez-Paz (2019) we tried to use the miscoverage level as an additional input to the network, while also parameterizing the loss with it.
  - We tried to use the pinball loss to predict multiple miscoverage values at the same time to get a more informed representation of the distribution we want to approximate.
- **Softmax Probabilities**. We tried to use the softmax output of the MHN to directly estimate probabilities and use a maximum likelihood procedure. This did train, but it did not produce good results.

A.6. Computational Resources

We used different hardware setups in our experiments, however, most of them were executed on a machine with an Nvidia P100 GPU and a Xeon E5-2698 CPU. The runtime differs greatly between different dataset sizes. We report the approximate run times for a single experiment (i.e., one

Table 5. Negative average of zero upper bounded Δ Cov of rolling windows with miscoverage level $\alpha = 0.1$. We evaluate each dataset–predictor combination at the windows sizes $k \in \{10, 20, 50\}$.

Data	FC	k	HopCPT	SPCI	EnbPI	NexCP	CP
Solar 3Y	Forest	10	**.030**	.040	.081	.059	.059
		20	**.020**	.022	.055	.034	.040
		50	**.012**	.016	.042	.021	.032
	LGBM	10	.051	**.040**	.076	.059	.058
		20	.040	**.022**	.052	.035	.039
		50	.031	**.015**	.038	.022	.031
	Ridge	10	**.023**	.049	.099	.055	.056
		20	**.014**	.035	.083	.039	.040
		50	**.007**	.028	.075	.027	.030
Solar 1Y	Forest	10	.023	.023	.070	.055	**.019**
		20	.015	.012	.043	.031	**.010**
		50	.010	.008	.031	.017	**.005**
	LGBM	10	.059	.023	.071	.056	**.018**
		20	.049	.012	.046	.032	**.010**
		50	.041	.007	.033	.018	**.005**
	Ridge	10	.063	**.030**	.070	.055	.066
		20	.053	**.021**	.055	.038	.052
		50	.045	**.016**	.043	.027	.045
Solar Small	Forest	10	**.045**	.108	.074	.075	.083
		20	**.026**	.074	.046	.045	.055
		50	**.015**	.067	.034	.030	.042
	LGBM	10	**.048**	.093	.074	.074	.078
		20	**.032**	.061	.044	.044	.048
		50	**.020**	.051	.030	.029	.035
	Ridge	10	**.051**	.100	.065	.058	.064
		20	**.039**	.073	.049	**.039**	.045
		50	**.032**	.066	.041	.033	.038
Air 10 PM	Forest	10	**.033**	.049	.124	.075	.106
		20	**.025**	.042	.114	.067	.100
		50	**.018**	.035	.099	.053	.089
	LGBM	10	.038	**.037**	.115	.074	.100
		20	**.030**	**.030**	.104	.065	.093
		50	**.021**	.023	.089	.052	.083
	Ridge	10	.041	**.035**	.099	.068	.064
		20	.032	**.028**	.089	.059	.057
		50	.024	**.020**	.074	.045	.048
Air 25 PM	Forest	10	.076	**.061**	.139	.080	.114
		20	.067	**.053**	.129	.071	.108
		50	.056	**.044**	.113	.056	.096
	LGBM	10	.073	**.051**	.124	.080	.113
		20	.064	**.044**	.114	.072	.105
		50	.053	**.035**	.097	.059	.093
	Ridge	10	.057	**.043**	.107	.075	.079
		20	.048	**.035**	.096	.066	.072
		50	.039	**.027**	.081	.053	.061
Sap flow	Forest	10	.070	**.058**	.107	.076	.073
		20	.062	**.050**	.097	.068	.067
		50	.051	**.042**	.082	.056	.059
	LGBM	10	.107	**.060**	.105	.078	.082
		20	.098	**.052**	.094	.069	.075
		50	.086	**.045**	.078	.056	.067
	Ridge	10	.097	.286	.102	**.086**	.404
		20	.088	.279	.089	**.077**	.397
		50	.075	.271	.071	**.063**	.390

seed, all evaluated coverage levels, not including training the prediction model μ):

1. **HopCPT**: Solar (3Y) 5h, Solar (1Y) 1h, Solar (3M) 7min, Air Quality (10PM) 3h, Air Quality (25PM) 3h, Sap flow 2.5h, Streamflow 12–20h
2. **SPCI**: (adapted): Solar (3Y) 5–10 days, Solar (1Y) 10–30h, Solar (3M) 45min, Air Quality (10PM) 13–30h, Air Quality (25PM) 13–30h, Sap flow 20–50h, Streamflow 12–17 days
3. **EnbPI**: all datasets under 6h
4. **NexCP**: all datasets under 45min
5. **AdaptiveCI**: all datasets under 45min
6. **Standard CP**: all datasets under 45min

B. Dataset Details

Datsets from four different domains were used in the experiments. The details are summarized in the following paragraphs. A quantitative overview is given in Table 6.

Solar. We conducted experiments on three solar radiation datasets: Solar (3M), Solar (1Y), and Solar (3Y). All three datasets are based on data from the US National Solar Radiation Database (NSDB; Sengupta et al., 2018). Besides solar radiation as the target variable, the datasets include 8 other environmental features. Solar (3M) is the same dataset as used by Xu & Xie (2022a;b) and focuses on data from 8 cities in California. To show the scalability of HopCPT we additionally generated two larger datasets Solar (1Y) and Solar (3Y), which include data from 50 cities from different parts of the US. We selected the cities by ordering the areas provided by NSDB according to the population density and picked the top 50 under the condition that no city appears twice in the dataset.

Air quality. The datasets Air Quality (10PM) and Air Quality (25PM) are air quality measurements from 12 monitoring sites in Beijing, China (Zhang et al., 2017). The datasets provide two target variables (10PM and 25PM measurements) which we use separately in our experiments (the variables are used mutually exclusively in the datasets, e.g., when predicting 10PM we do not use 25PM as a feature). We encode the wind direction, given as a categorical variable, by mapping the north–south and the east–west components each to a scalar from -1 to 1.

Sap flow. The Sap flow dataset is a subset of the Sapflux (Poyatos et al., 2021) data project. This dataset includes sap flow measurements which we use as a target variable, as well as a set of environmental variables. The available features, the length of the measurements, and the sampling vary between the individual time series. To get a set of comparable time series we processed the data as follows: (a) We removed all time series where not all of the 10 environmental features are available. (b) If there was a single missing value between two existing ones, we filled the value with the value before (forward fill). (c) We cut out all sequences without any missing target or feature values (after step b). (d) From the resulting set of sequences we remove all that have less than 15,000 or more than 20,000 time steps.

Streamflow. For our experiments on the streamflow data, we use the Catchment Attributes and Meteorology for Large-sample Studies (CAMELS) dataset (Newman et al., 2015; Addor et al., 2017). It provides meteorological time series from different data products, corresponding streamflow measurements, and static catchment attributes for catchments across the continental United States. We ran our experiments on the subset of 531 catchments that were used in previous hydrological benchmarking efforts (e.g., Newman et al., 2017; Kratzert et al., 2019; 2021; Klotz et al., 2022). Specifically, we trained the LSTM prediction model with the NeuralHydrology Python library (Kratzert et al., 2022) on precipitation, solar radiation, minimum and maximum temperature, and vapor pressure from the NLDAS (Xia et al., 2012), Maurer (Maurer et al., 2002), and Daymet (Thornton et al., 1997) meteorological data products. Further, the LSTM received 26 static catchment attributes at each time step that identify the catchment properties. Table 4 in Kratzert et al. (2019) provides a full list of these attributes. We trained the prediction model on data from the period Oct 1981 – Sep 1990 (note that for some catchments, this period starts later due to missing data in the beginning), calibrated the uncertainty models on Oct 1990 – Sep 1999, and tested them on the period Oct 1999 – Sep 2008.

C. kNN vs. Learned Representation

As mentioned in the main paper, we designed HopCPT with large datasets in mind. It is only in such a setting that the learned part of our approach can truly play to its strengths and take advantage of nuanced interrelationships in the data. kNN provides us with a natural "fallback" option for settings where not enough data is available to infer these relationships. Our comparisons in Table 7 and Table 8 substantiate this argument: kNN provides competitive results for the small dataset Solar (3M) (this can also be seen by contrasting the performance from Table 7 to the respective results in Appendix A.3), but is outperformed by HopCPT for the larger datasets.

D. Quantile of Sample vs. Quantile of Weighted ECDF

HopCPT constructs prediction intervals by calculating a quantile on the set of weight-sampled errors (see Equation 9). An alternative approach is to calculate the quantile over a weighted empirical CDF of the errors. This approach would define $q(\tau, \boldsymbol{Z}_{t+1})$ in Equation 9 as

$$q(\tau, \boldsymbol{Z}_{t+1}) = \mathcal{Q}_\tau \left(\sum_{i=1}^{t} a_{t+1,i} \delta_{\epsilon_i} \right), \tag{15}$$

where $\delta\epsilon_i$ is a point mass at $|\epsilon_i|$.

Empirically, we find little differences in the performance when comparing the two approaches (Tables 9 and 10).

E. AdaptiveCI and HopCPT

As noted in section 3, AdaptiveCI is orthogonal to HopCPT, SPCI, EnbPI, and NexCP. We therefore also evaluated the combined application of the models. Nevertheless, AdaptiveCI is an independent model on top of an existing model, which is reflected in the way we select the hyperparameters of the combined model: First, the hyperparameters are selected for each model without adaption through AdaptiveCI (see Appendix A.1) — hence we use the same hyperparameters as in the main evaluation. Second, we conduct another hyperparameter search, given the model parameters from the first search, where we only search for the parameter of the adaptive component.

Tables 11 and 12 show the results of these experiments. Overall, the results are slightly better, as the Winkler score (which considers both width and coverage) slightly increases in most experiments. The ranking between the different models stays similar to the non-adaptive comparison (see Section 3.2) with HopCPT performing best on all but the smallest dataset.

F. Details on Continuous Modern Hopfield Networks

The following arguments are adopted from Fürst et al. (2022) and Ramsauer et al. (2021). Associative memory networks have been designed to store and retrieve samples. Hopfield networks are energy-based, binary associative memories, which were popularized as artificial neural network architectures in the 1980s (Hopfield, 1982; 1984). Their storage capacity can be considerably increased by polynomial terms in the energy function (Chen et al., 1986; Psaltis & Cheol, 1986; Baldi & Venkatesh, 1987; Gardner, 1987; Abbott & Arian, 1987; Horn & Usher, 1988; Caputo & Niemann, 2002; Krotov & Hopfield, 2016). In contrast to these binary memory networks, we use continuous associative memory

Table 6. Details of the evaluated datasets.

	Number of Series	Time Steps per Series	Period	Sampling	Number of Features	Data Split [%]
Solar (3M)	8	2,000	01–03 2018	60m	8	60/15/25
Solar (1Y)	50	8,760	2019	60m	8	60/15/25
Solar (3Y)	50	26,304	2018–20	60m	8	34/33/33
Air Quality (10PM)	12	35,064	2013–17	60m	11	34/33/33
Air Quality (25PM)	12	35,064	2013–17	60m	11	34/33/33
Sap flow	24	15,000–20,000	2008–16	varying	10	34/33/33
Streamflow	531	9,862	1981–2008	24h	41	34/33/33

Table 7. Performance of kNN compared to HopCPT for the miscoverage $\alpha = 0.10$ on the solar datasets. The error term represents the standard deviation over repeated runs.

Data	FC	UC	kNN	HopCPT
Solar 3Y	Forest	Δ Cov	0.015	$0.029^{\pm 0.012}$
		PI-Width	97.2	**39.0**$^{\pm 6.2}$
		Winkler	1.59	**0.73**$^{\pm 0.20}$
	LGBM	Δ Cov	0.012	$0.001^{\pm 0.003}$
		PI-Width	108.4	**37.7**$^{\pm 0.7}$
		Winkler	1.74	**0.57**$^{\pm 0.01}$
	Ridge	Δ Cov	-0.009	$0.040^{\pm 0.001}$
		PI-Width	136.9	**44.9**$^{\pm 0.5}$
		Winkler	1.99	**0.64**$^{\pm 0.00}$
Solar 1Y	Forest	Δ Cov	0.026	$0.047^{\pm 0.004}$
		PI-Width	66.9	**28.6**$^{\pm 1.0}$
		Winkler	1.00	**0.40**$^{\pm 0.04}$
	LGBM	Δ Cov	0.003	$-0.003^{\pm 0.009}$
		PI-Width	72.2	**40.7**$^{\pm 2.8}$
		Winkler	1.08	**0.57**$^{\pm 0.08}$
	Ridge	Δ Cov	-0.007	$-0.011^{\pm 0.016}$
		PI-Width	128.8	**59.9**$^{\pm 5.6}$
		Winkler	1.63	**0.92**$^{\pm 0.12}$
Solar Small	Forest	Δ Cov	0.034	$0.008^{\pm 0.006}$
		PI-Width	**35.0**	$38.4^{\pm 3.4}$
		Winkler	**0.93**	$1.09^{\pm 0.08}$
	LGBM	Δ Cov	-0.022	$0.007^{\pm 0.012}$
		PI-Width	**47.2**	$48.2^{\pm 1.7}$
		Winkler	**1.17**	$1.24^{\pm 0.08}$
	Ridge	Δ Cov	0.004	$-0.016^{\pm 0.022}$
		PI-Width	**49.1**	$78.2^{\pm 24.5}$
		Winkler	**1.08**	$1.78^{\pm 0.59}$

Table 8. Performance of kNN compared to HopCPT for the miscoverage $\alpha = 0.10$ on the different non-solar datasets. The error term represents the standard deviation over repeated runs.

Data	FC	UC	kNN	HopCPT
Air 10 PM	Forest	Δ Cov	-0.071	$0.028^{\pm 0.019}$
		PI-Width	186.3	**93.9**$^{\pm 11.1}$
		Winkler	3.81	**1.50**$^{\pm 0.09}$
	LGBM	Δ Cov	-0.064	$0.017^{\pm 0.016}$
		PI-Width	164.6	**85.6**$^{\pm 7.4}$
		Winkler	3.42	**1.45**$^{\pm 0.06}$
	Ridge	Δ Cov	-0.054	$0.010^{\pm 0.007}$
		PI-Width	114.8	**79.9**$^{\pm 4.7}$
		Winkler	2.39	**1.35**$^{\pm 0.06}$
Air 25 PM	Forest	Δ Cov	-0.081	$-0.024^{\pm 0.017}$
		PI-Width	158.4	**48.1**$^{\pm 5.6}$
		Winkler	3.77	**1.12**$^{\pm 0.05}$
	LGBM	Δ Cov	-0.073	$-0.021^{\pm 0.019}$
		PI-Width	130.4	**46.8**$^{\pm 6.0}$
		Winkler	3.10	**1.10**$^{\pm 0.05}$
	Ridge	Δ Cov	-0.059	$-0.005^{\pm 0.026}$
		PI-Width	97.5	**49.3**$^{\pm 10.6}$
		Winkler	2.30	**1.04**$^{\pm 0.11}$
Sap flow	Forest	Δ Cov	-0.056	$-0.027^{\pm 0.028}$
		PI-Width	3246.1	**917.8**$^{\pm 57.4}$
		Winkler	1.00	**0.29**$^{\pm 0.01}$
	LGBM	Δ Cov	-0.063	$-0.062^{\pm 0.022}$
		PI-Width	2647.5	**801.2**$^{\pm 41.7}$
		Winkler	0.88	**0.28**$^{\pm 0.01}$
	Ridge	Δ Cov	-0.052	$-0.034^{\pm 0.018}$
		PI-Width	2808.4	**1486.1**$^{\pm 78.6}$
		Winkler	0.82	**0.41**$^{\pm 0.02}$
Streamflow	LSTM	Δ Cov	-0.108	$0.001^{\pm 0.041}$
		PI-Width	1.70	**1.39**$^{\pm 0.17}$
		Winkler	1.39	**0.79**$^{\pm 0.03}$

Table 9. Performance of the weighted sample quantile (Sample) and the weighted empirical CDF (ECDF) quantile strategies that HopCPT uses for the miscoverage $\alpha = 0.10$ on the solar datasets. The error term represents the standard deviation over repeated runs.

Data	FC	UC	Sample	ECDF
Solar 3Y	Forest	Δ Cov	$0.029^{\pm 0.012}$	$0.021^{\pm 0.008}$
		PI-Width	$39.0^{\pm 6.2}$	$\mathbf{33.1}^{\pm 1.0}$
		Winkler	$0.73^{\pm 0.20}$	$\mathbf{0.62}^{\pm 0.03}$
	LGBM	Δ Cov	$0.001^{\pm 0.003}$	$-0.006^{\pm 0.006}$
		PI-Width	$37.7^{\pm 0.7}$	$\mathbf{37.1}^{\pm 0.9}$
		Winkler	$\mathbf{0.57}^{\pm 0.01}$	$0.58^{\pm 0.01}$
	Ridge	Δ Cov	$0.040^{\pm 0.001}$	$0.041^{\pm 0.001}$
		PI-Width	$\mathbf{44.9}^{\pm 0.5}$	$45.0^{\pm 0.5}$
		Winkler	$\mathbf{0.64}^{\pm 0.00}$	$\mathbf{0.64}^{\pm 0.00}$
Solar 1Y	Forest	Δ Cov	$0.047^{\pm 0.004}$	$0.048^{\pm 0.005}$
		PI-Width	$\mathbf{28.6}^{\pm 1.0}$	$28.9^{\pm 1.0}$
		Winkler	$\mathbf{0.40}^{\pm 0.04}$	$\mathbf{0.40}^{\pm 0.04}$
	LGBM	Δ Cov	$-0.003^{\pm 0.009}$	$0.000^{\pm 0.007}$
		PI-Width	$40.7^{\pm 2.8}$	$\mathbf{40.5}^{\pm 2.9}$
		Winkler	$0.57^{\pm 0.08}$	$\mathbf{0.56}^{\pm 0.08}$
	Ridge	Δ Cov	$-0.011^{\pm 0.016}$	$-0.006^{\pm 0.013}$
		PI-Width	$59.9^{\pm 5.6}$	$\mathbf{57.5}^{\pm 2.5}$
		Winkler	$0.92^{\pm 0.12}$	$\mathbf{0.88}^{\pm 0.10}$
Solar Small	Forest	Δ Cov	$0.008^{\pm 0.006}$	$0.026^{\pm 0.023}$
		PI-Width	$38.4^{\pm 3.4}$	$\mathbf{37.1}^{\pm 5.7}$
		Winkler	$1.09^{\pm 0.08}$	$\mathbf{0.99}^{\pm 0.06}$
	LGBM	Δ Cov	$0.007^{\pm 0.012}$	$0.011^{\pm 0.013}$
		PI-Width	$48.2^{\pm 1.7}$	$\mathbf{46.3}^{\pm 1.3}$
		Winkler	$1.24^{\pm 0.08}$	$\mathbf{1.16}^{\pm 0.04}$
	Ridge	Δ Cov	$-0.016^{\pm 0.022}$	$0.012^{\pm 0.011}$
		PI-Width	$\mathbf{78.2}^{\pm 24.5}$	$109.0^{\pm 18.6}$
		Winkler	$\mathbf{1.78}^{\pm 0.59}$	$2.55^{\pm 0.38}$

Table 10. Performance of the weighted sample quantile (Sample) and the weighted empirical CDF (ECDF) quantile strategies that HopCPT uses for the miscoverage $\alpha = 0.10$ on the different non-solar datasets. The error term represents the standard deviation over repeated runs.

Data	FC	UC	Sample	ECDF
Air 10 PM	Forest	Δ Cov	$0.028^{\pm 0.019}$	$0.027^{\pm 0.021}$
		PI-Width	$93.9^{\pm 11.1}$	$\mathbf{93.4}^{\pm 11.0}$
		Winkler	$\mathbf{1.50}^{\pm 0.09}$	$\mathbf{1.50}^{\pm 0.09}$
	LGBM	Δ Cov	$0.017^{\pm 0.016}$	$0.014^{\pm 0.015}$
		PI-Width	$85.6^{\pm 7.4}$	$\mathbf{84.5}^{\pm 8.3}$
		Winkler	$1.45^{\pm 0.06}$	$\mathbf{1.43}^{\pm 0.09}$
	Ridge	Δ Cov	$0.010^{\pm 0.007}$	$0.013^{\pm 0.010}$
		PI-Width	$79.9^{\pm 4.7}$	$\mathbf{79.3}^{\pm 4.6}$
		Winkler	$1.35^{\pm 0.06}$	$\mathbf{1.34}^{\pm 0.05}$
Air 25 PM	Forest	Δ Cov	$-0.024^{\pm 0.017}$	$-0.024^{\pm 0.019}$
		PI-Width	$48.1^{\pm 5.6}$	$\mathbf{47.7}^{\pm 4.5}$
		Winkler	$1.12^{\pm 0.05}$	$\mathbf{1.11}^{\pm 0.01}$
	LGBM	Δ Cov	$-0.021^{\pm 0.019}$	$-0.023^{\pm 0.019}$
		PI-Width	$46.8^{\pm 6.0}$	$\mathbf{46.6}^{\pm 6.1}$
		Winkler	$\mathbf{1.10}^{\pm 0.05}$	$\mathbf{1.10}^{\pm 0.05}$
	Ridge	Δ Cov	$-0.005^{\pm 0.026}$	$-0.007^{\pm 0.027}$
		PI-Width	$49.3^{\pm 10.6}$	$\mathbf{49.1}^{\pm 10.7}$
		Winkler	$\mathbf{1.04}^{\pm 0.11}$	$\mathbf{1.04}^{\pm 0.11}$
Sap flow	Forest	Δ Cov	$-0.027^{\pm 0.028}$	$-0.030^{\pm 0.028}$
		PI-Width	$917.8^{\pm 57.4}$	$\mathbf{908.6}^{\pm 54.1}$
		Winkler	$\mathbf{0.29}^{\pm 0.01}$	$\mathbf{0.29}^{\pm 0.01}$
	LGBM	Δ Cov	$-0.062^{\pm 0.022}$	$-0.015^{\pm 0.015}$
		PI-Width	$\mathbf{801.2}^{\pm 41.7}$	$879.6^{\pm 54.0}$
		Winkler	$0.28^{\pm 0.01}$	$\mathbf{0.27}^{\pm 0.01}$
	Ridge	Δ Cov	$-0.034^{\pm 0.018}$	$-0.035^{\pm 0.018}$
		PI-Width	$\mathbf{1486.1}^{\pm 78.6}$	$1486.3^{\pm 92.8}$
		Winkler	$\mathbf{0.41}^{\pm 0.02}$	$\mathbf{0.41}^{\pm 0.02}$
Streamflow	LSTM	Δ Cov	$0.001^{\pm 0.041}$	$0.028^{\pm 0.002}$
		PI-Width	$\mathbf{1.39}^{\pm 0.17}$	$1.49^{\pm 0.02}$
		Winkler	$0.79^{\pm 0.03}$	$\mathbf{0.77}^{\pm 0.01}$

Table 11. Performance of the CP algorithms HopCPT, SPCI, EnbPI, and NexCP, each combined with AdaptiveCI, for the miscoverage $\alpha = 0.10$ on the solar datasets. Bold numbers correspond to the best result for the respective metric in the experiment (PI-Width and Winkler score). The error term represents the standard deviation over repeated runs. † combined with AdaptiveCI.

Data	FC	UC	HopCPT †	SPCI †	EnbPI †	NexCP †
Solar 3Y	Forest	Δ Cov	$0.004^{\pm 0.001}$	$0.002^{\pm 0.000}$	-0.001	-0.000
		PI-Width	$\mathbf{29.8}^{\pm 0.7}$	$97.6^{\pm 0.1}$	155.7	170.8
		Winkler	$\mathbf{0.59}^{\pm 0.03}$	$1.69^{\pm 0.00}$	2.43	2.54
	LGBM	Δ Cov	$-0.001^{\pm 0.000}$	$0.002^{\pm 0.000}$	-0.001	-0.000
		PI-Width	$\mathbf{39.9}^{\pm 7.0}$	$95.8^{\pm 0.1}$	155.0	164.8
		Winkler	$\mathbf{0.62}^{\pm 0.10}$	$1.70^{\pm 0.00}$	2.49	2.57
	Ridge	Δ Cov	$0.002^{\pm 0.000}$	$-0.000^{\pm 0.000}$	-0.006	0.000
		PI-Width	$\mathbf{36.0}^{\pm 0.3}$	$110.6^{\pm 0.1}$	166.3	177.3
		Winkler	$\mathbf{0.62}^{\pm 0.00}$	$1.77^{\pm 0.00}$	2.28	2.63
Solar 1Y	Forest	Δ Cov	$0.007^{\pm 0.001}$	$0.011^{\pm 0.000}$	-0.002	-0.001
		PI-Width	$\mathbf{21.5}^{\pm 1.2}$	$72.6^{\pm 0.1}$	109.5	126.7
		Winkler	$\mathbf{0.40}^{\pm 0.02}$	$1.14^{\pm 0.00}$	1.63	1.85
	LGBM	Δ Cov	$0.000^{\pm 0.001}$	$0.005^{\pm 0.000}$	-0.001	-0.000
		PI-Width	$\mathbf{41.6}^{\pm 1.9}$	$69.2^{\pm 0.2}$	106.2	117.1
		Winkler	$\mathbf{0.56}^{\pm 0.04}$	$1.15^{\pm 0.00}$	1.63	1.78
	Ridge	Δ Cov	$-0.003^{\pm 0.005}$	$0.004^{\pm 0.000}$	-0.002	-0.000
		PI-Width	$\mathbf{61.5}^{\pm 9.9}$	$98.9^{\pm 0.2}$	167.8	174.8
		Winkler	$\mathbf{0.87}^{\pm 0.11}$	$1.32^{\pm 0.00}$	2.01	2.08
Solar Small	Forest	Δ Cov	$0.006^{\pm 0.003}$	$-0.023^{\pm 0.002}$	-0.003	-0.014
		PI-Width	$\mathbf{33.6}^{\pm 2.9}$	$64.7^{\pm 0.7}$	92.9	127.3
		Winkler	$\mathbf{1.01}^{\pm 0.04}$	$1.95^{\pm 0.01}$	2.38	3.27
	LGBM	Δ Cov	$-0.001^{\pm 0.002}$	$-0.015^{\pm 0.001}$	-0.005	-0.005
		PI-Width	$\mathbf{44.8}^{\pm 1.0}$	$60.9^{\pm 0.6}$	96.4	135.7
		Winkler	$\mathbf{1.18}^{\pm 0.04}$	$1.95^{\pm 0.01}$	2.54	3.32
	Ridge	Δ Cov	$0.008^{\pm 0.004}$	$-0.015^{\pm 0.001}$	-0.003	0.008
		PI-Width	$104.7^{\pm 19.8}$	$\mathbf{69.3}^{\pm 0.7}$	102.5	97.7
		Winkler	$2.44^{\pm 0.42}$	$\mathbf{1.92}^{\pm 0.01}$	2.56	2.75

Table 12. Performance of the CP algorithms HopCPT, SPCI, EnbPI, and NexCP, each combined with AdaptiveCI, for the miscoverage $\alpha = 0.10$ on the different non-solar datasets. Bold numbers correspond to the best result for the respective metric in the experiment (PI-Width and Winkler score). The error term represents the standard deviation over repeated runs. † combined with AdaptiveCI.

Data	FC	UC	HopCPT †	SPCI †	EnbPI †	NexCP †
Air 10 PM	Forest	Δ Cov	$0.002^{\pm 0.001}$	$-0.002^{\pm 0.000}$	-0.002	0.000
		PI-Width	**80.7**$^{\pm 2.6}$	$112.6^{\pm 0.1}$	250.4	270.5
		Winkler	**1.45**$^{\pm 0.06}$	$1.96^{\pm 0.00}$	3.70	3.92
	LGBM	Δ Cov	$0.001^{\pm 0.001}$	$0.001^{\pm 0.000}$	-0.001	0.000
		PI-Width	**78.5**$^{\pm 2.2}$	$102.5^{\pm 0.2}$	220.0	233.2
		Winkler	**1.38**$^{\pm 0.04}$	$1.83^{\pm 0.00}$	3.38	3.54
	Ridge	Δ Cov	$0.001^{\pm 0.001}$	$0.000^{\pm 0.000}$	-0.001	0.001
		PI-Width	**75.3**$^{\pm 0.8}$	$83.7^{\pm 0.1}$	147.2	159.5
		Winkler	**1.30**$^{\pm 0.02}$	$1.46^{\pm 0.00}$	2.46	2.67
Air 25 PM	Forest	Δ Cov	$-0.001^{\pm 0.001}$	$-0.002^{\pm 0.000}$	-0.003	-0.001
		PI-Width	**53.4**$^{\pm 1.1}$	$86.5^{\pm 0.1}$	221.7	245.1
		Winkler	**1.10**$^{\pm 0.02}$	$1.76^{\pm 0.00}$	3.68	3.97
	LGBM	Δ Cov	$-0.000^{\pm 0.001}$	$0.000^{\pm 0.000}$	-0.001	0.000
		PI-Width	**52.5**$^{\pm 1.7}$	$76.1^{\pm 0.1}$	180.6	191.7
		Winkler	**1.10**$^{\pm 0.05}$	$1.69^{\pm 0.00}$	3.17	3.38
	Ridge	Δ Cov	$-0.000^{\pm 0.001}$	$-0.001^{\pm 0.000}$	-0.001	0.001
		PI-Width	**50.4**$^{\pm 3.9}$	$63.2^{\pm 0.1}$	130.0	143.6
		Winkler	**1.04**$^{\pm 0.10}$	$1.33^{\pm 0.00}$	2.36	2.64
Sap flow	Forest	Δ Cov	$-0.004^{\pm 0.007}$	$0.004^{\pm 0.000}$	-0.006	0.000
		PI-Width	**1006.9**$^{\pm 44.5}$	$1651.0^{\pm 1.9}$	4142.8	6151.7
		Winkler	**0.29**$^{\pm 0.01}$	$0.51^{\pm 0.00}$	1.11	1.54
	LGBM	Δ Cov	$-0.001^{\pm 0.003}$	$0.007^{\pm 0.000}$	-0.005	-0.000
		PI-Width	**919.1**$^{\pm 25.8}$	$1601.7^{\pm 3.0}$	3327.0	4845.1
		Winkler	**0.26**$^{\pm 0.01}$	$0.45^{\pm 0.00}$	0.88	1.23
	Ridge	Δ Cov	$-0.006^{\pm 0.002}$	$-0.121^{\pm 0.000}$	-0.003	-0.000
		PI-Width	**1492.5**$^{\pm 29.9}$	$2974.7^{\pm 6.9}$	3431.4	10665.6
		Winkler	**0.39**$^{\pm 0.01}$	$1.53^{\pm 0.00}$	0.87	2.39

networks with far higher storage capacity. These networks are continuous and differentiable, retrieve with a single update, and have exponential storage capacity (and are therefore scalable, i.e., able tackle large problems; Ramsauer et al., 2021).

Formally, we denote a set of patterns $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\} \subset \mathbb{R}^d$ that are stacked as columns to the matrix $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$ and a state pattern (query) $\boldsymbol{\xi} \in \mathbb{R}^d$ that represents the current state. The largest norm of a stored pattern is $M = \max_i \|\boldsymbol{x}_i\|$. Then, the energy E of continuous Modern Hopfield Networks with state $\boldsymbol{\xi}$ is defined as (Ramsauer et al., 2021)

$$\mathrm{E} \ = \ - \beta^{-1} \ \log \left(\sum_{i=1}^{N} \exp(\beta \boldsymbol{x}_i^T \boldsymbol{\xi}) \right) + \frac{1}{2} \, \boldsymbol{\xi}^T \boldsymbol{\xi} + \mathrm{C}, \quad (16)$$

where $\mathrm{C} = \beta^{-1} \log N \ + \ \frac{1}{2} \, M^2$. For energy E and state $\boldsymbol{\xi}$, Ramsauer et al. (2021) proved that the update rule

$$\boldsymbol{\xi}^{\mathrm{new}} \ = \ \boldsymbol{X} \ \mathrm{softmax}(\beta \boldsymbol{X}^T \boldsymbol{\xi}) \quad (17)$$

converges globally to stationary points of the energy E and coincides with the attention mechanisms of Transformers (Vaswani et al., 2017; Ramsauer et al., 2021).

The *separation* Δ_i of a pattern $\boldsymbol{x}_i$ is its minimal dot product difference to any of the other patterns:

$$\Delta_i = \min_{j, j \neq i} \left(\boldsymbol{x}_i^T \boldsymbol{x}_i - \boldsymbol{x}_i^T \boldsymbol{x}_j \right) . \quad (18)$$

A pattern is *well-separated* from the data if Δ_i is above a given threshold (specified in Ramsauer et al., 2021). If the patterns $\boldsymbol{x}_i$ are well-separated, the update rule Equation 17 converges to a fixed point close to a stored pattern. If some patterns are similar to one another and, therefore, not well-separated, the update rule converges to a fixed point close to the mean of the similar patterns.

The update rule of a Hopfield network thus identifies sample–sample relations between stored patterns. This enables similarity-based learning methods like nearest neighbor search (see Schäfl et al., 2022), which HopCPT leverages to learn a retrieval of samples from similar error regimes.

G. Potential Social Impact

Reliable uncertainty estimates are crucial, especially for complex time-dependent environmental phenomena. However, overreliance on these estimates can be dangerous. For example, unseen regimes might not be properly predicted. A changing climate that evolves the environment beyond already seen conditions can cause new forms of error regimes which cannot be predicted reliably. As most machine learning approaches, our method requires accurately labeled training data. Incorrect labels may lead to unexpected biases and prediction errors.

H. Code and Data

The code and data to reproduce all of our experiments are available at `https://github.com/ml-jku/HopCPT`.

Table 13. Performance of the evaluated CP algorithms on the Solar (3Y) datasets for the miscoverage levels $\alpha \in \{0.05, 0.10, 0.15\}$. The column FC specifies the prediction algorithm used for the experiment (Forest: Random Forest, LGBM: LightGBM, Ridge: Ridge Regression, LSTM: LSTM neural network). Bold numbers correspond to the best result for the respective metric in the experiment (PI-Width and Winkler score). The error term represents the standard deviation over repeated runs (results without an error term are from deterministic models).

FC	α	UC	HopCPT	SPCI	EnbPI	NexCP	CopulaCPTS	CP/CF-RNN	AdaptiveCI
Forest	0.05	Δ Cov	$0.006^{\pm 0.006}$	$0.007^{\pm 0.000}$	-0.030	-0.002	0.002	0.002	
		PI-Width	**47.8**$^{\pm 9.5}$	$149.2^{\pm 0.1}$	173.0	216.5	237.2	236.7	
		Winkler	**0.99**$^{\pm 0.35}$	$2.22^{\pm 0.00}$	3.01	2.95	3.30	3.30	
	0.10	Δ Cov	$0.029^{\pm 0.012}$	$0.012^{\pm 0.000}$	-0.031	-0.002	0.005	0.004	
		PI-Width	**39.0**$^{\pm 6.2}$	$103.1^{\pm 0.1}$	131.1	166.6	174.9	174.6	
		Winkler	**0.73**$^{\pm 0.20}$	$1.74^{\pm 0.00}$	2.47	2.53	2.75	2.76	
	0.15	Δ Cov	$0.052^{\pm 0.018}$	$0.014^{\pm 0.000}$	-0.027	-0.002	0.006	0.006	
		PI-Width	**33.6**$^{\pm 4.6}$	$75.3^{\pm 0.1}$	101.4	129.1	132.2	132.0	
		Winkler	**0.61**$^{\pm 0.15}$	$1.46^{\pm 0.00}$	2.12	2.23	2.38	2.39	
LGBM	0.05	Δ Cov	$-0.008^{\pm 0.002}$	$0.007^{\pm 0.000}$	-0.022	-0.002	0.002	0.002	0.001
		PI-Width	**45.6**$^{\pm 0.8}$	$148.0^{\pm 0.1}$	183.0	215.9	237.9	237.8	88.1
		Winkler	**0.72**$^{\pm 0.02}$	$2.25^{\pm 0.00}$	3.09	3.06	3.45	3.46	1.36
	0.10	Δ Cov	$0.001^{\pm 0.003}$	$0.014^{\pm 0.000}$	-0.023	-0.002	0.006	0.006	0.001
		PI-Width	**37.7**$^{\pm 0.7}$	$102.2^{\pm 0.1}$	133.6	159.9	169.8	170.2	67.1
		Winkler	**0.57**$^{\pm 0.01}$	$1.75^{\pm 0.00}$	2.52	2.55	2.80	2.81	1.19
	0.15	Δ Cov	$0.009^{\pm 0.003}$	$0.017^{\pm 0.000}$	-0.022	-0.002	0.008	0.009	0.001
		PI-Width	**32.7**$^{\pm 0.7}$	$75.5^{\pm 0.1}$	100.6	121.3	126.6	127.2	55.1
		Winkler	**0.50**$^{\pm 0.01}$	$1.46^{\pm 0.00}$	2.15	2.21	2.39	2.40	1.11
Ridge	0.05	Δ Cov	$0.010^{\pm 0.001}$	$-0.003^{\pm 0.000}$	-0.080	-0.002	0.003	0.003	
		PI-Width	**52.7**$^{\pm 0.6}$	$146.3^{\pm 0.0}$	151.0	226.2	223.8	224.8	
		Winkler	**0.78**$^{\pm 0.00}$	$2.31^{\pm 0.00}$	3.04	3.20	3.44	3.46	
	0.10	Δ Cov	$0.040^{\pm 0.001}$	$0.002^{\pm 0.000}$	-0.074	-0.001	0.004	0.005	
		PI-Width	**44.9**$^{\pm 0.5}$	$108.2^{\pm 0.0}$	131.1	171.0	166.0	167.7	
		Winkler	**0.64**$^{\pm 0.00}$	$1.82^{\pm 0.00}$	2.49	2.66	2.73	2.74	
	0.15	Δ Cov	$0.070^{\pm 0.001}$	$0.009^{\pm 0.000}$	-0.069	-0.000	0.004	0.006	
		PI-Width	**39.6**$^{\pm 0.5}$	$89.4^{\pm 0.0}$	114.7	142.2	141.2	142.7	
		Winkler	**0.56**$^{\pm 0.00}$	$1.56^{\pm 0.00}$	2.21	2.34	2.37	2.37	
LSTM	0.05	Δ Cov	$-0.003^{\pm 0.005}$	$0.004^{\pm 0.000}$	-0.018	-0.001	0.002	0.001	
		PI-Width	**22.7**$^{\pm 0.8}$	$47.7^{\pm 0.1}$	41.0	47.3	54.1	54.7	
		Winkler	**0.38**$^{\pm 0.01}$	$0.87^{\pm 0.00}$	0.90	0.87	0.96	0.97	
	0.10	Δ Cov	$0.001^{\pm 0.006}$	$0.014^{\pm 0.000}$	-0.018	-0.001	0.007	0.007	
		PI-Width	**17.9**$^{\pm 0.6}$	$27.7^{\pm 0.0}$	24.6	28.2	31.9	33.0	
		Winkler	**0.30**$^{\pm 0.01}$	$0.62^{\pm 0.00}$	0.64	0.63	0.68	0.70	
	0.15	Δ Cov	$0.002^{\pm 0.007}$	$0.024^{\pm 0.000}$	-0.017	-0.001	0.010	0.009	
		PI-Width	**15.0**$^{\pm 0.4}$	$18.1^{\pm 0.0}$	16.2	18.6	20.5	21.3	
		Winkler	**0.26**$^{\pm 0.00}$	$0.48^{\pm 0.00}$	0.50	0.50	0.54	0.55	

Table 14. Performance of the evaluated CP algorithms on the Solar (1Y) dataset for the miscoverage levels $\alpha \in \{0.05, 0.10, 0.15\}$. The column FC specifies the prediction algorithm used for the experiment (Forest: Random Forest, LGBM: LightGBM, Ridge: Ridge Regression, LSTM: LSTM neural network). Bold numbers correspond to the best result for the respective metric in the experiment (PI-Width and Winkler score). The error term represents the standard deviation over repeated runs (results without an error term are from deterministic models).

FC	α	UC	HopCPT	SPCI	EnbPI	NexCP	CopulaCPTS	CP/CF-RNN	AdaptiveCI
Forest	0.05	Δ Cov	$0.016^{\pm 0.002}$	$0.029^{\pm 0.000}$	-0.017	0.002	0.034	0.035	
		PI-Width	**35.0**$^{\pm 1.2}$	$147.8^{\pm 0.3}$	133.4	173.9	241.8	262.9	
		Winkler	**0.50**$^{\pm 0.05}$	$1.71^{\pm 0.00}$	1.97	2.18	2.61	2.82	
	0.10	Δ Cov	$0.047^{\pm 0.004}$	$0.045^{\pm 0.000}$	-0.018	0.002	0.056	0.063	
		PI-Width	**28.6**$^{\pm 1.0}$	$97.1^{\pm 0.2}$	98.8	127.8	182.4	204.9	
		Winkler	**0.40**$^{\pm 0.04}$	$1.26^{\pm 0.00}$	1.65	1.84	2.10	2.30	
	0.15	Δ Cov	$0.078^{\pm 0.006}$	$0.052^{\pm 0.000}$	-0.016	0.002	0.070	0.086	
		PI-Width	**24.6**$^{\pm 0.8}$	$67.7^{\pm 0.1}$	73.1	93.6	138.5	161.2	
		Winkler	**0.35**$^{\pm 0.03}$	$1.00^{\pm 0.00}$	1.42	1.59	1.76	1.93	
LGBM	0.05	Δ Cov	$-0.010^{\pm 0.006}$	$0.029^{\pm 0.000}$	-0.019	0.001	0.034	0.036	-0.000
		PI-Width	**51.0**$^{\pm 3.4}$	$147.8^{\pm 0.2}$	131.5	165.0	237.9	265.2	62.2
		Winkler	**0.71**$^{\pm 0.10}$	$1.72^{\pm 0.00}$	2.05	2.16	2.60	2.86	0.78
	0.10	Δ Cov	$-0.003^{\pm 0.009}$	$0.045^{\pm 0.000}$	-0.017	0.002	0.056	0.065	0.000
		PI-Width	**40.7**$^{\pm 2.8}$	$96.5^{\pm 0.2}$	93.9	118.0	171.1	196.6	49.5
		Winkler	**0.57**$^{\pm 0.08}$	$1.26^{\pm 0.00}$	1.65	1.78	2.02	2.24	0.69
	0.15	Δ Cov	$0.001^{\pm 0.012}$	$0.053^{\pm 0.000}$	-0.014	0.002	0.068	0.086	0.000
		PI-Width	**34.5**$^{\pm 2.5}$	$68.0^{\pm 0.1}$	68.2	85.7	125.5	149.1	42.5
		Winkler	**0.50**$^{\pm 0.07}$	$1.01^{\pm 0.00}$	1.40	1.52	1.66	1.83	0.67
Ridge	0.05	Δ Cov	$-0.025^{\pm 0.012}$	$0.018^{\pm 0.000}$	-0.025	-0.004	0.008	0.011	
		PI-Width	**70.3**$^{\pm 6.5}$	$151.0^{\pm 0.2}$	178.0	196.3	207.9	219.8	
		Winkler	**1.15**$^{\pm 0.18}$	$1.78^{\pm 0.00}$	2.26	2.29	2.35	2.49	
	0.10	Δ Cov	$-0.011^{\pm 0.016}$	$0.031^{\pm 0.000}$	-0.031	-0.006	-0.010	-0.015	
		PI-Width	**59.9**$^{\pm 5.6}$	$112.8^{\pm 0.1}$	158.4	171.7	171.7	172.0	
		Winkler	**0.92**$^{\pm 0.12}$	$1.40^{\pm 0.00}$	2.06	2.09	2.12	2.17	
	0.15	Δ Cov	$0.000^{\pm 0.020}$	$0.035^{\pm 0.001}$	-0.034	-0.008	-0.024	-0.040	
		PI-Width	**52.9**$^{\pm 4.9}$	$92.5^{\pm 0.1}$	144.1	154.6	150.2	146.6	
		Winkler	**0.81**$^{\pm 0.10}$	$1.21^{\pm 0.00}$	1.92	1.96	1.98	2.02	
LSTM	0.05	Δ Cov	$0.019^{\pm 0.003}$	$0.006^{\pm 0.000}$	-0.018	-0.001	0.010	0.013	
		PI-Width	**21.4**$^{\pm 0.7}$	$37.0^{\pm 0.1}$	29.5	33.6	39.4	42.2	
		Winkler	**0.27**$^{\pm 0.01}$	$0.58^{\pm 0.00}$	0.60	0.57	0.59	0.61	
	0.10	Δ Cov	$0.028^{\pm 0.010}$	$0.018^{\pm 0.000}$	-0.018	-0.001	0.018	0.025	
		PI-Width	**16.0**$^{\pm 0.6}$	$22.5^{\pm 0.0}$	17.4	19.5	23.1	25.0	
		Winkler	**0.22**$^{\pm 0.01}$	$0.41^{\pm 0.00}$	0.42	0.41	0.43	0.43	
	0.15	Δ Cov	$0.029^{\pm 0.017}$	$0.032^{\pm 0.000}$	-0.014	0.001	0.030	0.040	
		PI-Width	$13.1^{\pm 0.5}$	$15.3^{\pm 0.0}$	**11.1**	12.6	15.2	16.9	
		Winkler	**0.19**$^{\pm 0.00}$	$0.32^{\pm 0.00}$	0.33	0.33	0.33	0.34	

Table 15. Performance of the evaluated CP algorithms on the Solar (3M) dataset for the miscoverage levels $\alpha \in \{0.05, 0.10, 0.15\}$. The column FC specifies the prediction algorithm used for the experiment (Forest: Random Forest, LGBM: LightGBM, Ridge: Ridge Regression). Bold numbers correspond to the best result for the respective metric in the experiment (PI-Width and Winkler score). The error term represents the standard deviation over repeated runs (results without an error term are from deterministic models).

FC	α	UC	HopCPT	SPCI	EnbPI	NexCP	CopulaCPTS	CP	AdaptiveCI
Forest	0.05	Δ Cov	$-0.002^{\pm 0.005}$	$-0.074^{\pm 0.001}$	-0.022	-0.013	-0.022	-0.028	
		PI-Width	**47.5**$^{\pm 5.0}$	$57.9^{\pm 0.3}$	110.7	160.2	162.0	155.0	
		Winkler	**1.29**$^{\pm 0.11}$	$2.62^{\pm 0.02}$	2.92	3.70	3.95	4.04	
	0.10	Δ Cov	$0.008^{\pm 0.006}$	$-0.064^{\pm 0.002}$	-0.022	-0.021	-0.027	-0.025	
		PI-Width	**38.4**$^{\pm 3.4}$	$38.8^{\pm 0.3}$	86.0	122.9	110.4	111.4	
		Winkler	**1.09**$^{\pm 0.08}$	$1.82^{\pm 0.01}$	2.54	3.28	3.47	3.48	
	0.15	Δ Cov	$0.020^{\pm 0.010}$	$-0.052^{\pm 0.002}$	-0.028	-0.019	-0.020	-0.017	
		PI-Width	$32.6^{\pm 2.8}$	**26.7**$^{\pm 0.3}$	64.8	91.0	77.5	79.6	
		Winkler	**0.99**$^{\pm 0.07}$	$1.45^{\pm 0.02}$	2.40	2.91	2.99	2.98	
LGBM	0.05	Δ Cov	$0.002^{\pm 0.008}$	$-0.063^{\pm 0.001}$	-0.024	-0.014	-0.018	-0.019	-0.000
		PI-Width	$59.7^{\pm 3.4}$	**56.3**$^{\pm 0.3}$	111.7	161.7	164.5	162.1	61.8
		Winkler	$1.49^{\pm 0.15}$	$2.66^{\pm 0.02}$	3.10	3.87	4.13	4.18	**1.45**
	0.10	Δ Cov	$0.007^{\pm 0.012}$	$-0.052^{\pm 0.002}$	-0.022	-0.020	-0.022	-0.021	-0.004
		PI-Width	$48.2^{\pm 1.7}$	**37.5**$^{\pm 0.3}$	84.3	119.9	107.6	106.9	49.2
		Winkler	**1.24**$^{\pm 0.08}$	$1.84^{\pm 0.01}$	2.67	3.35	3.52	3.53	1.26
	0.15	Δ Cov	$0.005^{\pm 0.016}$	$-0.043^{\pm 0.002}$	-0.024	-0.019	-0.013	-0.011	-0.007
		PI-Width	$41.4^{\pm 1.3}$	**26.9**$^{\pm 0.2}$	60.8	85.1	75.8	77.5	44.1
		Winkler	**1.11**$^{\pm 0.07}$	$1.47^{\pm 0.01}$	2.48	2.93	2.98	2.97	1.22
Ridge	0.05	Δ Cov	$-0.017^{\pm 0.014}$	$-0.064^{\pm 0.001}$	-0.021	0.004	0.009	-0.002	
		PI-Width	$92.7^{\pm 32.4}$	**67.2**$^{\pm 0.3}$	110.1	144.8	156.1	133.6	
		Winkler	**1.99**$^{\pm 0.69}$	$2.49^{\pm 0.02}$	3.07	3.82	3.78	3.75	
	0.10	Δ Cov	$-0.016^{\pm 0.022}$	$-0.056^{\pm 0.002}$	-0.020	0.007	0.014	-0.003	
		PI-Width	$78.2^{\pm 24.5}$	**51.8**$^{\pm 0.3}$	84.9	85.2	88.6	78.1	
		Winkler	**1.78**$^{\pm 0.59}$	$1.91^{\pm 0.01}$	2.66	2.85	2.83	2.81	
	0.15	Δ Cov	$-0.009^{\pm 0.021}$	$-0.039^{\pm 0.002}$	-0.017	0.003	0.014	0.003	
		PI-Width	$67.4^{\pm 19.0}$	**43.2**$^{\pm 0.2}$	71.6	70.9	71.9	70.7	
		Winkler	$1.61^{\pm 0.51}$	**1.59**$^{\pm 0.00}$	2.29	2.34	2.34	2.33	

Table 16. Performance of the evaluated CP algorithms on the Air Quality (10PM) dataset for the miscoverage levels $\alpha \in \{0.05, 0.10, 0.15\}$. The column FC specifies the prediction algorithm used for the experiment (Forest: Random Forest, LGBM: LightGBM, Ridge: Ridge Regression, LSTM: LSTM neural network). Bold numbers correspond to the best result for the respective metric in the experiment (PI-Width and Winkler score). The error term represents the standard deviation over repeated runs (results without an error term are from deterministic models).

FC	α	UC	HopCPT	SPCI	EnbPI	NexCP	CopulaCPTS	CP/CF-RNN	AdaptiveCI
Forest	0.05	Δ Cov	$0.006^{\pm 0.013}$	$-0.001^{\pm 0.000}$	-0.054	-0.005	-0.016	-0.027	
		PI-Width	**116.7**$^{\pm 14.4}$	$152.7^{\pm 0.1}$	242.9	321.5	322.5	297.1	
		Winkler	**1.94**$^{\pm 0.10}$	$2.86^{\pm 0.00}$	4.97	4.82	6.47	6.61	
	0.10	Δ Cov	$0.028^{\pm 0.019}$	$0.008^{\pm 0.000}$	-0.066	-0.004	-0.019	-0.033	
		PI-Width	**93.9**$^{\pm 11.1}$	$118.5^{\pm 0.1}$	202.8	263.5	243.1	229.8	
		Winkler	**1.50**$^{\pm 0.09}$	$2.23^{\pm 0.00}$	4.16	4.03	4.94	4.98	
	0.15	Δ Cov	$0.050^{\pm 0.025}$	$0.018^{\pm 0.000}$	-0.073	-0.002	-0.021	-0.039	
		PI-Width	**80.5**$^{\pm 9.4}$	$99.8^{\pm 0.1}$	175.7	229.4	207.1	198.2	
		Winkler	**1.28**$^{\pm 0.08}$	$1.92^{\pm 0.00}$	3.72	3.61	4.18	4.20	
LGBM	0.05	Δ Cov	$-0.000^{\pm 0.011}$	$0.008^{\pm 0.000}$	-0.047	-0.005	-0.014	-0.022	-0.000
		PI-Width	**106.3**$^{\pm 9.9}$	$146.1^{\pm 0.1}$	219.5	285.7	281.1	264.6	228.0
		Winkler	**1.87**$^{\pm 0.07}$	$2.49^{\pm 0.00}$	4.56	4.46	5.72	5.79	3.61
	0.10	Δ Cov	$0.017^{\pm 0.016}$	$0.023^{\pm 0.000}$	-0.057	-0.004	-0.017	-0.028	-0.001
		PI-Width	**85.6**$^{\pm 7.4}$	$113.2^{\pm 0.1}$	178.3	224.8	206.7	196.5	186.4
		Winkler	**1.45**$^{\pm 0.06}$	$1.94^{\pm 0.00}$	3.69	3.64	4.33	4.36	3.00
	0.15	Δ Cov	$0.033^{\pm 0.019}$	$0.038^{\pm 0.000}$	-0.061	-0.004	-0.017	-0.029	-0.001
		PI-Width	**73.4**$^{\pm 6.2}$	$94.9^{\pm 0.1}$	151.4	188.9	168.2	161.9	161.1
		Winkler	**1.24**$^{\pm 0.06}$	$1.66^{\pm 0.00}$	3.23	3.20	3.63	3.65	2.71
Ridge	0.05	Δ Cov	$0.001^{\pm 0.003}$	$0.010^{\pm 0.000}$	-0.037	-0.003	0.005	0.006	
		PI-Width	**100.6**$^{\pm 6.2}$	$120.2^{\pm 0.1}$	152.6	202.3	215.7	219.3	
		Winkler	**1.70**$^{\pm 0.08}$	$1.93^{\pm 0.00}$	3.32	3.42	3.96	3.98	
	0.10	Δ Cov	$0.010^{\pm 0.007}$	$0.024^{\pm 0.000}$	-0.045	-0.002	0.010	0.012	
		PI-Width	**79.9**$^{\pm 4.7}$	$93.9^{\pm 0.1}$	120.0	153.3	153.7	155.3	
		Winkler	**1.35**$^{\pm 0.06}$	$1.52^{\pm 0.00}$	2.60	2.68	2.95	2.96	
	0.15	Δ Cov	$0.016^{\pm 0.011}$	$0.038^{\pm 0.001}$	-0.049	-0.003	0.015	0.018	
		PI-Width	**68.1**$^{\pm 4.1}$	$79.4^{\pm 0.1}$	100.5	126.9	125.7	127.0	
		Winkler	**1.17**$^{\pm 0.05}$	$1.31^{\pm 0.00}$	2.23	2.31	2.46	2.46	
LSTM	0.05	Δ Cov	$-0.001^{\pm 0.001}$	$0.003^{\pm 0.000}$	-0.021	-0.002	-0.002	-0.001	
		PI-Width	$90.7^{\pm 1.9}$	$86.1^{\pm 0.0}$	**80.8**	88.8	86.8	88.1	
		Winkler	$1.83^{\pm 0.03}$	**1.63**$^{\pm 0.00}$	1.81	1.77	1.86	1.86	
	0.10	Δ Cov	$-0.002^{\pm 0.005}$	$0.010^{\pm 0.000}$	-0.025	-0.002	0.001	0.004	
		PI-Width	$62.7^{\pm 1.5}$	$62.3^{\pm 0.1}$	**58.1**	62.4	61.8	63.0	
		Winkler	$1.33^{\pm 0.01}$	**1.21**$^{\pm 0.00}$	1.32	1.29	1.34	1.34	
	0.15	Δ Cov	$-0.002^{\pm 0.010}$	$0.017^{\pm 0.000}$	-0.028	-0.002	0.005	0.009	
		PI-Width	$49.4^{\pm 1.7}$	$50.2^{\pm 0.0}$	**46.5**	49.6	49.6	50.8	
		Winkler	$1.09^{\pm 0.01}$	**1.01**$^{\pm 0.00}$	1.08	1.07	1.10	1.10	

Table 17. Performance of the evaluated CP algorithms on the Air Quality (25PM) dataset for the miscoverage levels $\alpha \in \{0.05, 0.10, 0.15\}$. The column FC specifies the prediction algorithm used for the experiment (Forest: Random Forest, LGBM: LightGBM, Ridge: Ridge Regression, LSTM: LSTM neural network). Bold numbers correspond to the best result for the respective metric in the experiment (PI-Width and Winkler score). The error term represents the standard deviation over repeated runs (results without an error term are from deterministic models).

FC	α	UC	HopCPT	SPCI	EnbPI	NexCP	CopulaCPTS	CP/CF-RNN	AdaptiveCI
Forest	0.05	Δ Cov	$-0.033^{\pm 0.013}$	$-0.015^{\pm 0.000}$	-0.065	-0.009	-0.018	-0.031	
		PI-Width	**58.7**$^{\pm 6.9}$	$102.6^{\pm 0.1}$	211.3	283.9	271.4	249.8	
		Winkler	**1.51**$^{\pm 0.04}$	$2.67^{\pm 0.00}$	5.01	4.71	6.47	6.59	
	0.10	Δ Cov	$-0.024^{\pm 0.017}$	$-0.009^{\pm 0.000}$	-0.079	-0.007	-0.025	-0.042	
		PI-Width	**48.1**$^{\pm 5.6}$	$81.5^{\pm 0.0}$	177.3	235.6	212.6	203.5	
		Winkler	**1.12**$^{\pm 0.05}$	$2.02^{\pm 0.00}$	4.31	4.05	4.94	4.98	
	0.15	Δ Cov	$-0.014^{\pm 0.019}$	$-0.000^{\pm 0.001}$	-0.084	-0.005	-0.029	-0.045	
		PI-Width	**41.4**$^{\pm 4.9}$	$70.8^{\pm 0.0}$	153.0	206.9	185.5	179.0	
		Winkler	**0.94**$^{\pm 0.05}$	$1.71^{\pm 0.00}$	3.88	3.67	4.23	4.25	
LGBM	0.05	Δ Cov	$-0.031^{\pm 0.014}$	$-0.007^{\pm 0.000}$	-0.054	-0.006	-0.020	-0.032	-0.001
		PI-Width	**57.2**$^{\pm 7.8}$	$93.5^{\pm 0.1}$	176.4	239.5	212.7	196.0	208.8
		Winkler	**1.48**$^{\pm 0.05}$	$2.45^{\pm 0.00}$	4.30	4.35	5.47	5.58	3.99
	0.10	Δ Cov	$-0.021^{\pm 0.019}$	$0.002^{\pm 0.000}$	-0.063	-0.007	-0.027	-0.042	-0.001
		PI-Width	**46.8**$^{\pm 6.0}$	$73.3^{\pm 0.0}$	142.1	182.2	154.5	143.9	163.3
		Winkler	**1.10**$^{\pm 0.05}$	$1.83^{\pm 0.00}$	3.54	3.55	4.09	4.14	3.35
	0.15	Δ Cov	$-0.010^{\pm 0.022}$	$0.010^{\pm 0.001}$	-0.066	-0.006	-0.029	-0.045	-0.001
		PI-Width	**40.2**$^{\pm 5.1}$	$62.4^{\pm 0.0}$	118.8	147.6	124.2	117.1	133.3
		Winkler	**0.92**$^{\pm 0.05}$	$1.55^{\pm 0.00}$	3.07	3.08	3.41	3.43	2.99
Ridge	0.05	Δ Cov	$-0.018^{\pm 0.018}$	$-0.001^{\pm 0.000}$	-0.043	-0.005	-0.002	-0.004	
		PI-Width	**60.2**$^{\pm 13.1}$	$84.0^{\pm 0.1}$	132.5	177.6	180.5	177.1	
		Winkler	**1.35**$^{\pm 0.12}$	$1.82^{\pm 0.00}$	3.11	3.31	3.82	3.85	
	0.10	Δ Cov	$-0.005^{\pm 0.026}$	$0.011^{\pm 0.000}$	-0.051	-0.006	-0.002	-0.004	
		PI-Width	**49.3**$^{\pm 10.6}$	$65.5^{\pm 0.0}$	106.2	134.3	127.4	125.3	
		Winkler	**1.04**$^{\pm 0.11}$	$1.40^{\pm 0.00}$	2.54	2.66	2.92	2.93	
	0.15	Δ Cov	$0.008^{\pm 0.032}$	$0.021^{\pm 0.000}$	-0.053	-0.007	-0.001	-0.003	
		PI-Width	**42.6**$^{\pm 9.3}$	$55.6^{\pm 0.0}$	88.7	109.7	102.5	101.6	
		Winkler	**0.89**$^{\pm 0.10}$	$1.20^{\pm 0.00}$	2.21	2.30	2.46	2.46	
LSTM	0.05	Δ Cov	$0.005^{\pm 0.005}$	$-0.015^{\pm 0.000}$	-0.023	-0.003	-0.015	-0.021	
		PI-Width	$57.1^{\pm 5.6}$	**45.0**$^{\pm 0.0}$	50.8	56.0	48.4	46.1	
		Winkler	**1.19**$^{\pm 0.08}$	$1.29^{\pm 0.00}$	1.33	1.27	1.39	1.40	
	0.10	Δ Cov	$0.007^{\pm 0.008}$	$-0.017^{\pm 0.000}$	-0.028	-0.003	-0.019	-0.025	
		PI-Width	$40.7^{\pm 4.3}$	**32.4**$^{\pm 0.0}$	35.9	38.6	34.0	32.8	
		Winkler	**0.88**$^{\pm 0.05}$	$0.93^{\pm 0.00}$	0.97	0.94	0.99	0.99	
	0.15	Δ Cov	$0.005^{\pm 0.011}$	$-0.016^{\pm 0.000}$	-0.029	-0.003	-0.019	-0.025	
		PI-Width	$32.5^{\pm 3.7}$	**26.1**$^{\pm 0.0}$	28.4	30.2	26.8	26.1	
		Winkler	**0.74**$^{\pm 0.04}$	$0.76^{\pm 0.00}$	0.79	0.77	0.80	0.81	

Table 18. Performance of the evaluated CP algorithms on the Sap flow dataset for the miscoverage levels $\alpha \in \{0.05, 0.10, 0.15\}$. The column FC specifies the prediction algorithm used for the experiment (Forest: Random Forest, LGBM: LightGBM, Ridge: Ridge Regression, LSTM: LSTM neural network). Bold numbers correspond to the best result for the respective metric in the experiment (PI-Width and Winkler score). The error term represents the standard deviation over repeated runs (results without an error term are from deterministic models).

FC	α	UC	HopCPT	SPCI	EnbPI	NexCP	CopulaCPTS	CP/CF-RNN	AdaptiveCI
Forest	0.05	Δ Cov	$-0.035^{\pm 0.025}$	$-0.002^{\pm 0.000}$	-0.047	-0.003	0.010	0.003	
		PI-Width	**1125.1**$^{\pm 69.9}$	$2193.8^{\pm 2.1}$	4264.1	7290.8	8805.8	8970.9	
		Winkler	**0.37**$^{\pm 0.02}$	$0.74^{\pm 0.00}$	1.42	1.75	2.04	2.12	
	0.10	Δ Cov	$-0.027^{\pm 0.028}$	$0.007^{\pm 0.000}$	-0.042	0.000	0.014	0.005	
		PI-Width	**917.8**$^{\pm 57.4}$	$1741.8^{\pm 2.4}$	3671.6	6137.1	7131.1	7201.5	
		Winkler	**0.29**$^{\pm 0.01}$	$0.59^{\pm 0.00}$	1.24	1.56	1.76	1.80	
	0.15	Δ Cov	$-0.019^{\pm 0.029}$	$0.014^{\pm 0.001}$	-0.034	0.002	0.017	0.009	
		PI-Width	**788.0**$^{\pm 49.1}$	$1456.7^{\pm 2.2}$	3200.5	5261.8	5980.4	6062.0	
		Winkler	**0.25**$^{\pm 0.01}$	$0.50^{\pm 0.00}$	1.12	1.43	1.57	1.60	
LGBM	0.05	Δ Cov	$-0.066^{\pm 0.021}$	$-0.006^{\pm 0.000}$	-0.043	-0.006	0.006	-0.002	0.008
		PI-Width	**984.8**$^{\pm 51.8}$	$1988.0^{\pm 1.0}$	3453.3	5737.6	6929.6	7087.5	7160.4
		Winkler	**0.37**$^{\pm 0.02}$	$0.62^{\pm 0.00}$	1.12	1.41	1.66	1.72	1.64
	0.10	Δ Cov	$-0.062^{\pm 0.022}$	$0.003^{\pm 0.000}$	-0.040	-0.003	0.006	-0.007	0.010
		PI-Width	**801.2**$^{\pm 41.7}$	$1582.3^{\pm 1.0}$	2924.1	4805.3	5588.5	5614.8	6273.5
		Winkler	**0.28**$^{\pm 0.01}$	$0.49^{\pm 0.00}$	0.96	1.25	1.43	1.46	1.50
	0.15	Δ Cov	$-0.055^{\pm 0.023}$	$0.010^{\pm 0.001}$	-0.036	-0.002	0.005	-0.012	0.010
		PI-Width	**686.7**$^{\pm 36.1}$	$1347.1^{\pm 1.3}$	2531.2	4147.4	4586.3	4544.3	5564.7
		Winkler	**0.24**$^{\pm 0.01}$	$0.42^{\pm 0.00}$	0.87	1.14	1.27	1.30	1.40
Ridge	0.05	Δ Cov	$-0.035^{\pm 0.015}$	$-0.231^{\pm 0.000}$	-0.040	-0.012	-0.174	-0.331	
		PI-Width	**1785.9**$^{\pm 92.7}$	$2539.3^{\pm 1.6}$	3595.3	11318.6	10230.3	8462.1	
		Winkler	**0.49**$^{\pm 0.02}$	$3.97^{\pm 0.01}$	1.04	2.54	3.53	8.20	
	0.10	Δ Cov	$-0.034^{\pm 0.018}$	$-0.241^{\pm 0.000}$	-0.041	-0.015	-0.251	-0.358	
		PI-Width	**1486.1**$^{\pm 78.6}$	$2060.5^{\pm 1.6}$	3117.5	10628.9	8943.3	7148.7	
		Winkler	**0.41**$^{\pm 0.02}$	$2.52^{\pm 0.00}$	0.92	2.42	3.31	5.85	
	0.15	Δ Cov	$-0.032^{\pm 0.021}$	$-0.235^{\pm 0.001}$	-0.042	-0.016	-0.292	-0.375	
		PI-Width	**1292.1**$^{\pm 69.4}$	$1792.0^{\pm 1.8}$	2775.8	10073.3	7959.3	6155.9	
		Winkler	**0.37**$^{\pm 0.02}$	$1.93^{\pm 0.00}$	0.85	2.34	3.17	4.89	
LSTM	0.05	Δ Cov	$0.001^{\pm 0.002}$	$0.000^{\pm 0.000}$	-0.018	-0.001	-0.012	-0.024	
		PI-Width	**783.5**$^{\pm 9.2}$	$898.3^{\pm 0.8}$	1020.5	1338.9	1300.1	1194.5	
		Winkler	**0.25**$^{\pm 0.01}$	$0.33^{\pm 0.00}$	0.36	0.40	0.45	0.47	
	0.10	Δ Cov	$0.004^{\pm 0.004}$	$0.004^{\pm 0.000}$	-0.019	-0.000	-0.022	-0.042	
		PI-Width	**594.3**$^{\pm 7.7}$	$628.6^{\pm 0.8}$	768.0	990.0	903.9	817.2	
		Winkler	**0.19**$^{\pm 0.01}$	$0.24^{\pm 0.00}$	0.28	0.32	0.35	0.36	
	0.15	Δ Cov	$0.005^{\pm 0.005}$	$0.007^{\pm 0.000}$	-0.019	0.002	-0.026	-0.048	
		PI-Width	**489.9**$^{\pm 7.0}$	$493.4^{\pm 0.5}$	618.2	780.6	681.5	620.4	
		Winkler	**0.17**$^{\pm 0.01}$	$0.20^{\pm 0.00}$	0.24	0.27	0.29	0.30	

Table 19. Performance of the evaluated CP algorithms and the CMAL baseline (see Appendix A.3) on the Streamflow dataset for the miscoverage levels $\alpha \in \{0.05, 0.10, 0.15\}$. Bold numbers correspond to the best result for the respective metric in the experiment (PI-Width and Winkler score). The error term represents the standard deviation over repeated runs (results without an error term are from deterministic models).

α	UC	HopCPT	SPCI	EnbPI	NexCP	CopulaCPTS	CP/CF-RNN	CMAL
0.05	Δ Cov	$-0.002^{\pm 0.022}$	$0.013^{\pm 0.000}$	-0.042	-0.001	0.003	0.006	$-0.003^{\pm 0.003}$
	PI-Width	$\mathbf{1.91}^{\pm 0.20}$	$2.57^{\pm 0.00}$	2.53	3.23	3.44	3.63	$2.4^{\pm 0.08}$
	Winkler	$\mathbf{1.05}^{\pm 0.03}$	$1.38^{\pm 0.00}$	1.91	1.80	1.93	1.94	$3.04^{\pm 0.04}$
0.10	Δ Cov	$0.001^{\pm 0.041}$	$0.027^{\pm 0.000}$	-0.054	-0.000	0.005	0.009	$-0.004^{\pm 0.005}$
	PI-Width	$\mathbf{1.39}^{\pm 0.17}$	$1.58^{\pm 0.00}$	1.55	1.94	1.99	2.08	$1.90^{\pm 0.06}$
	Winkler	$\mathbf{0.79}^{\pm 0.03}$	$0.91^{\pm 0.00}$	1.27	1.21	1.28	1.29	$2.46^{\pm 0.03}$
0.15	Δ Cov	$0.003^{\pm 0.056}$	$0.038^{\pm 0.000}$	-0.061	0.001	0.005	0.009	$-0.004^{\pm 0.007}$
	PI-Width	$\mathbf{1.11}^{\pm 0.15}$	$1.17^{\pm 0.00}$	1.12	1.39	1.39	1.45	$1.60^{\pm 0.05}$
	Winkler	$\mathbf{0.66}^{\pm 0.03}$	$0.71^{\pm 0.00}$	0.98	0.95	0.99	1.00	$2.15^{\pm 0.03}$