RECOWNs: Probabilistic Circuits for Trustworthy Time Series Forecasting

Nils Thoma[1] **Zhongjie Yu**[1] **Fabrizio Ventola**[1] **Kristian Kersting**[1,2]

[1]Department of Computer Science, TU Darmstadt, Darmstadt, Germany
[2]Centre for Cognitive Science, TU Darmstadt, and Hessian Center for AI (hessian.AI)

Abstract

Time series forecasting is a relevant task that is performed in several real-world scenarios such as product sales analysis and prediction of energy demand. Given their accuracy performance, currently, Recurrent Neural Networks (RNNs) are the models of choice for this task. Despite their success in time series forecasting, less attention has been paid to make the RNNs trustworthy. For example, RNNs can not naturally provide an uncertainty measure to their predictions. This could be extremely useful in practice in several cases e.g. to detect when a prediction might be completely wrong due to an unusual pattern in the time series. Whittle Sum-Product Networks (WSPNs), prominent deep tractable probabilistic circuits (PCs) for time series, can assist an RNN with providing meaningful probabilities as uncertainty measure. With this aim, we propose RECOWN, a novel architecture that employs RNNs and a discriminant variant of WSPNs called Conditional WSPNs (CWSPNs). We also formulate a Log-Likelihood Ratio Score as better estimation of uncertainty that is tailored to time series and Whittle likelihoods. In our experiments, we show that RECOWNs are accurate and trustworthy time series predictors, able to "know when they do not know".

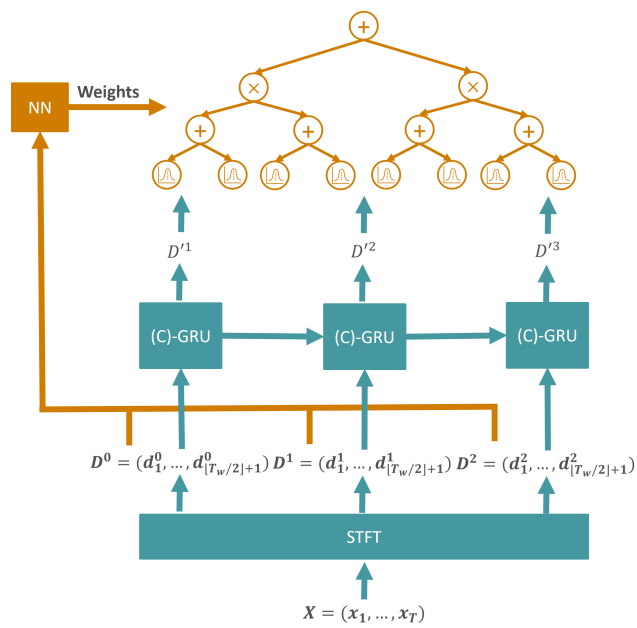

Figure 1: Overview of the RECOWN architecture. The context X is transformed using STFT with a window size of T_w, a) determining the weights of the CWSPN via the Neural Network (NN) and b) serving as input to the (complex)-Gated Recurrent Units (GRU) [Chung et al., 2014] of the RNN, resulting in the prediction of the Fourier coefficients D'^1, D'^2, D'^3. Those are then provided to the CWSPN, which computes the conditional Whittle log-likelihood $\ell(D'^1, D'^2, D'^3 | D^0, D^1, D^2)$ (see Section 2.3).

1 INTRODUCTION

Time series forecasting is the task to predict the future course (Y) of a time series given its past X, also known as context. Currently, Recurrent Neural Networks [Rumelhart et al., 1985] are models of choice when it comes to time series forecasting. Recent advancements in RNNs research fostered their adoption in practice surpassing established models as e.g. ARIMA (Autoregressive Integrated Moving Average) [Siami-Namini et al., 2018] in several scenarios [Fei and Yeung, 2015, Li et al., 2019, Li and Cao, 2018]. However, in complex real-world applications, time series are highly subject to several influence factors which are often hard to capture. For example, in the case of grocery demand, the demand for ice cream or BBQ-related products is expected to be higher as long as the weather is warmer than usual. Furthermore, exceptional events like a pandemic can significantly influence demands as well. In

Accepted for the 37th Conference on Uncertainty in Artificial Intelligence (UAI 2021).

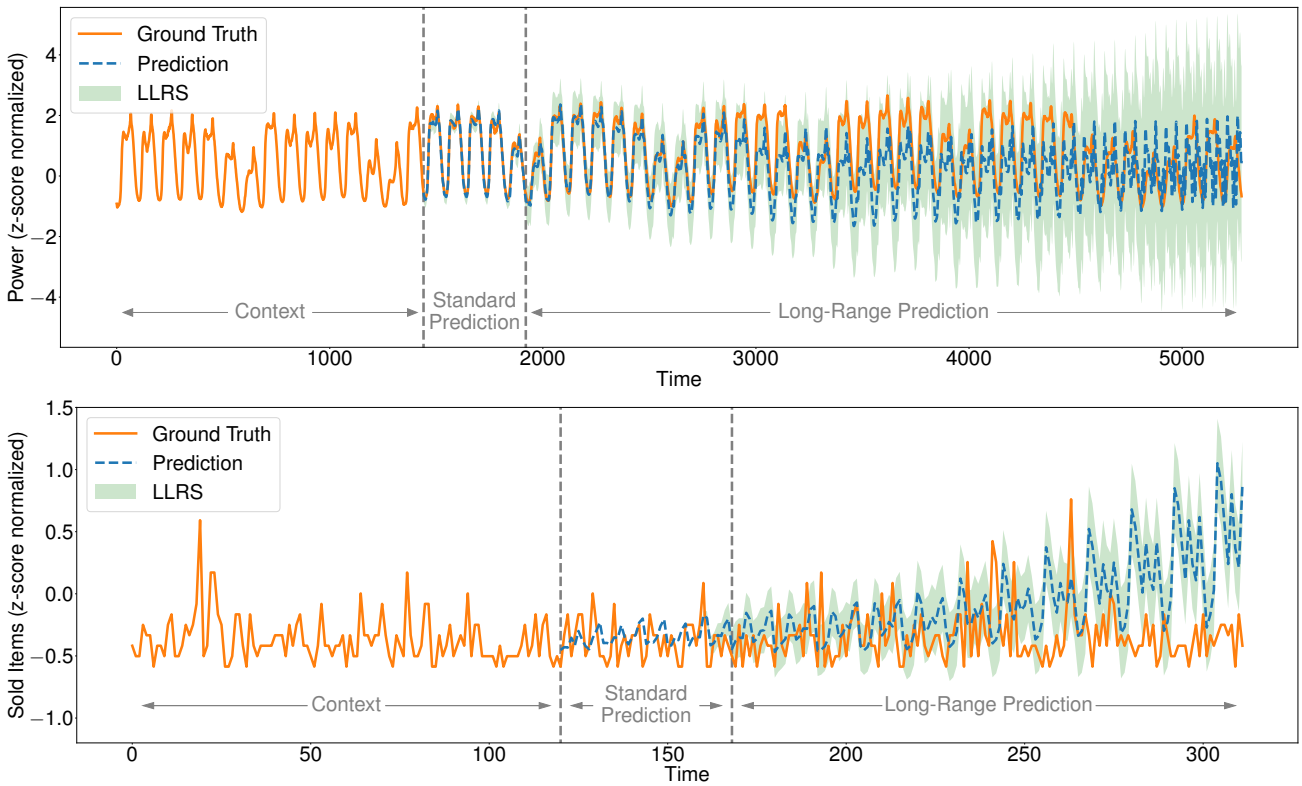

Figure 2: A long-term prediction on *Power* (top) and *Retail* datasets (bottom), showing how RECOWN correctly indicate a increasingly worse prediction in the long run. RECOWNs are trustworthy models that can provide predictions with an uncertainty measure estimated by the Log-Likelihood Ratio Score (LLRS, in green, see section 3.3). This can be provided at any time point of the prediction. Thus, the increasing of uncertainty (LLRS) after the standard prediction length is meaningful. This additional information makes the model trustworthy and can support users in decision-making processes.

such cases, the prediction of a model will likely be less accurate compared to usual circumstances. To properly detect such cases, a measure of uncertainty about the prediction is valuable [Guo et al., 2017, Laptev et al., 2017, Gal and Ghahramani, 2016]. It would make the predictions trustworthy and it would better support the users in decision-making processes. A relevant approach to achieve this is provided by Gaussian Processes (GPs) [Rasmussen and Williams, 2006]. However, GPs are comparably computationally expensive and therefore not suited for large datasets [Seeger, 2004]. Several methods have been proposed to scale GPs on large datasets, e.g. by modeling the mixture of subspaces using GP experts with Sum-Product-Networks (SPNs) [Trapp et al., 2020, Bruinsma et al., 2020, Yu et al., 2021b]. Still, for time series forecasting, RNN architectures [Alpay et al., 2016, Wolter et al., 2020, Koutnik et al., 2014] have shown to be superior in terms of prediction accuracy.

Therefore, we introduce REcurrent COnditional Whittle Networks (RECOWNs), a deep architecture that makes use of Short Time Fourier Transform (STFT) and integrates a Spectral RNN [Wolter et al., 2020] and a conditional Whittle SPN (CWSPN) [Yu et al., 2021a] i.e. a discriminative probabilistic circuits tailored for time series. The overall architecture is depicted in Figure 1. In this way, RECOWN can keep the good forecasting performance of RNNs while, thanks to the CWSPN, providing meaningful probabilities as a measure of uncertainty of the predictions. A showcase of its potential can be found in Figure 2, showing how the CWSPN correctly predicts when the forecast of the Spectral RNN increasingly diverges from the ground truth. RECOWNs can be trained end-to-end by gradient descent and have modular nature, in fact, one can employ any differentiable density estimator instead of a CWSPN. However, in our experiments, CWSPNs resulted the best choice compared to another state-of-the-art density estimators, namely MAF [Papamakarios et al., 2017], especially when the model capacity is reduced. Furthermore, CWSPNs are also more flexible given that they can answer to a wider range of exact probabilistic queries in a tractable way.

Similar to Yu et al. [2021a], RECOWNs model time series in the spectral domain, but employ a different kind of Fourier Transform – the STFT – to account for changes in the frequency domain as the signal changes over time. While in Yu et al. [2021a] the authors focused on modeling the joint distribution of the multivariate time series, in this work we shift our focus on predictions, providing them

with probabilities as a useful measure of uncertainty. As successfully done in Yu et al. [2021a] and other previous methods [Tank et al., 2015], modeling in the spectral domain enables us to make use of the so-called Whittle Approximation Assumption [Whittle, 1953] which facilitates the modeling of Fourier coefficients of a time series. Additionally, since RECOWNs model the time series in the spectral domain, we propose Log-Likelihood Ratio Score (LLRS) which enables to compute the confidence intervals back in the time domain. Experiments show that, compared to state-of-the-art, RECOWNs are more accurate and trustworthy. Our contributions are the following:

- We introduce RECOWN, the first deep tractable model for time series forecasting that is accurate and that provides an uncertainty measure for its predictions, making them more trustworthy.
- We introduce a data sample weighting strategy based on the Mean Squared Error.
- We formulate the (Whittle) Log-Likelihood Ratio Score, tailored to better estimate the uncertainty of time series predictions based on the Whittle likelihood.

The paper is structured as follows: We start by introducing STFT, CWSPNs as well as Spectral RNNs, i.e. the main components of RECOWNs in Section 2. Then, in Section 3, we describe the experimental setting and analyze the results showing that RECOWNs are both accurate and trustworthy and that the LLRS is a valid tool to assess the uncertainty and the quality of the predictions. We conclude in Section 4 where we also point out future directions.

2 RECOWN: RECURRENT CONDITIONAL WHITTLE NETWORKS

2.1 SHORT TIME FOURIER TRANSFORM

Using discrete Fourier Transformation (DFT), a signal can be mapped from the time to the spectral domain using a decomposition into linear combinations of sinus functions. For a multivariate time series $\mathscr{X} = x_1, ..., x_T$ with $x_t \in \mathbb{R}^p$ and length T, we can define the discrete Fourier coefficients $d_k \in \mathbb{C}^p$ at frequency $\lambda_k = \frac{2\pi k}{T}$, $k = 0, \cdots, T-1$, using the Fourier transformation $\mathscr{F}$ as follows [Tank et al., 2015]:

$$\mathscr{F}(X)_k = d_k = \sum_{t=0}^{T-1} x_t e^{-i\lambda_k t}. \tag{1}$$

Moreover, given Fourier coefficients $D = (d_0, ..., d_{T-1})$, we can apply the inverse DFT to regain a time-domain representation:

$$\mathscr{F}^{-1}(D)_t = x_t = \frac{1}{T} \sum_{k=0}^{T-1} d_k e^{i\lambda_k t}. \tag{2}$$

However, based on the standard definition, infinite support of the input signal is required in order to apply DFT. To overcome this limitation, we apply STFT: Instead of approximating the whole signal at once, the input is divided into overlapping segments and each segment is then approximated separately. Mathematically speaking, considering segments of length T_w, extracted every S time steps, STFT is introduced by Griffin and Lim [1984]:

$$STFT(\mathscr{X})_k^m = \mathscr{F}(w(Sm-t)x_t)_k = \sum_{t=1}^{T_w} w(Sm-t)x_t e^{-i\lambda_k t}, \tag{3}$$

with w being a window function, m denoting a respective shift and the remaining defined analogues to Equation 1. This results in $n_s = \frac{T}{S}$ windows in total. As with the regular Fourier transform, STFT can also be inverted:

$$\begin{aligned} iSTFT(D)_t &= \mathscr{F}^{-1}(D(Sm))_t = x_t \\ &= \frac{\sum_{m=-\inf}^{\inf} w(Sm-t)\mathscr{F}^{-1}(D(Sm)_t)}{\sum_{m=-\inf}^{\inf} w^2(Sm-t)}. \end{aligned} \tag{4}$$

Due to Hermitian symmetry, for a real-valued time series, the negative Fourier coefficients are redundant and we only need to model $\mathscr{T} = \lfloor \frac{T_w}{2} \rfloor + 1$ Fourier coefficients for a window size of T_w. Furthermore, we apply a low-pass filter, to filter out noise and further reduce the number of parameters in the model. Details on the extent of filtering are described in the Experimental Evaluation section.

The window function can be selected by hand, but it may also be learned by an optimizer as part of a machine learning model, as e.g. described by Wolter et al. [2020]. Analogues to their approach, we apply a truncated Gaussian window:

$$w(n) = exp(-\frac{1}{2}(\frac{n - T_w/2}{\sigma T/2})^2). \tag{5}$$

The standard deviation σ is learned by the optimizer: By increasing σ, the window approaches a more rectangular shape, while it gets narrower with decreasing σ.

2.2 WHITTLE LIKELIHOOD

The Whittle likelihood models the multivariate time series in the spectral domain. With $\mathscr{X}$, x_t and T given as in Section 2.1, x_t is Gaussian stationary for $t \in \mathbb{Z}$, if:

$$E(x_t) = \mu \quad \forall t \in \mathbb{Z} \tag{6}$$

$$\mathrm{Cov}(x_t, x_{t+h}) = \Gamma(h) \quad \forall t, h \in \mathbb{Z}. \tag{7}$$

With $\mathscr{X}_{1:N} = \{\mathscr{X}^1, \ldots, \mathscr{X}^N\}$ being N independent realizations of $\mathscr{X}$, $d_{n,k} \in \mathbb{C}^p$ denotes the discrete Fourier coefficient of the nth sequence at frequency λ_k:

$$d_{n,k} = \mathscr{F}(\mathscr{X}^n)_k. \tag{8}$$

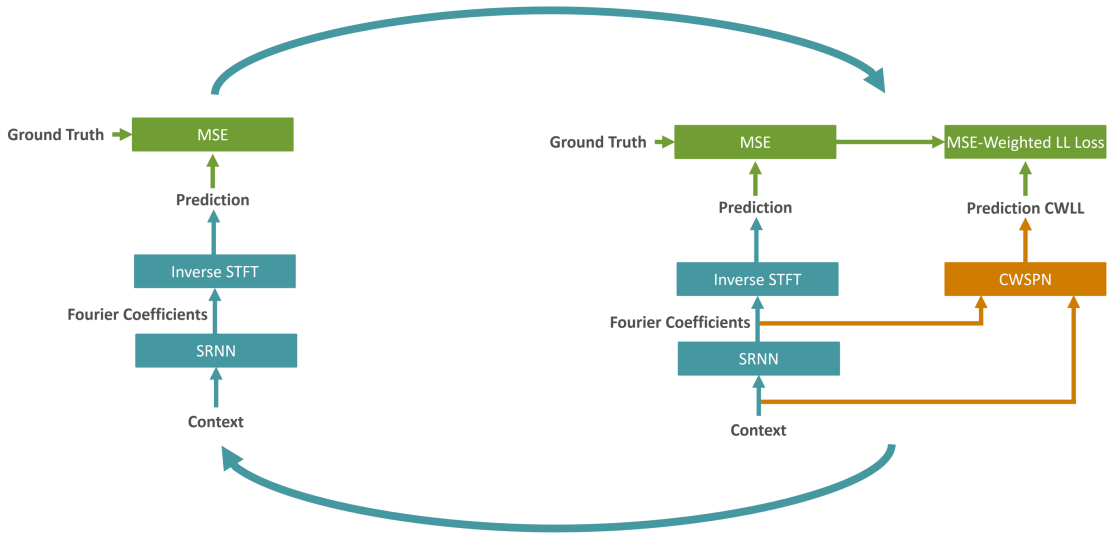

Figure 3: Depiction of the optimization process for RECOWN in co-ordinate descent fashion. First, SRNN weights are optimized (Left), then CWSPN weights are optimized using the MSE together with the CWLL, which is derived from the SRNN predictions (Right). These steps are iterated until convergence.

Based on the Whittle approximation assumption [Whittle, 1953], the Fourier coefficients are independent complex normal RVs with mean zero:

$$d_{n,k} \sim \mathcal{N}(0, S_k), \quad k = 0, \ldots, T-1, \tag{9}$$

with $S_k \in \mathbb{C}^{p \times p}$ being the *spectral density matrix*. For a stationary time series, it is defined as:

$$S_k = \sum_{h=-\infty}^{\infty} \Gamma(h) e^{-i\lambda_k h}. \tag{10}$$

Finally, the Whittle likelihood of the $\mathcal{X}_{1:N}$ is given by: $p(\mathcal{X}_{1:N} \mid S_{0:T-1}) \approx$

$$\prod_{n=1}^{N} \prod_{k=0}^{T-1} \frac{1}{\pi^p |S_k|} e^{-d_{n,k}^* S_k^{-1} d_{n,k}}. \tag{11}$$

The Whittle approximation holds asymptotically with large T and has been used in the Bayesian context. We will make use of it and place a Complex Conditional SPN over the frequencies, resulting in CWSPNs, modeling the conditional Whittle Log-Likelihood (CWLL).

2.3 COMPLEX CONDITIONAL SPN

Originally introduced by Poon and Domingos [2011], SPNs have gained recent popularity, showing also promising results on univariate time series [Melibari et al., 2016]. Recent approaches build upon randomly generated structures, RAT-SPNs, introduced by Peharz et al. [2020b], enabling scalability as well as training in an end-to-end fashion combined with NNs

In order to provide a measure of how good a prediction (Y) is with respect to a context (X), we aim for modeling the conditional likelihood $P(Y|X)$. Although any SPNs modeling the joint could be employed for this task (since the conditional can be derived from the joint using $P(Y|X) = \frac{P(X,Y)}{P(X)}$), the Conditial SPN (CSPN) is a natural choice. Originally proposed by Shao et al. [2020], the CSPN parameters are not learned directly. Instead, a neural network is employed to provide parameters to the SPN based on the input X, while only Y is provided as input to the SPN on the leaf layer. Training this architecture – using gradient descent – results in modeling $P(Y|X)$. The structure of CSPN is randomly generated based on RAT-SPNs [Peharz et al., 2020b].

We alter this approach to account for the complex values of the Fourier coefficients analogues to the Complex-Valued SPNs (CoSPNs) [Yu et al., 2021a]. Input to the leaves of the CoCSPN will be the Fourier coefficients of Y at frequency k and shift m, i.e. $d_k^m = STFT(Y)_k^m$. Based on the Whittle assumption, we know that the Fourier coefficients d_k^m are normal distributed. Therefore, also their real and imaginary parts will be jointly normal distributed. To account for the correlations between the two parts, they will be jointly modeled by a single pairwise Gaussian leaf node, parameterized by a vector of means $\mu_{d_k^m} \in \mathbb{R}^2$ and a covariance matrix $\Sigma_{d_k^m} \in \mathbb{R}^{2\times 2}$. In total, the CoCSPN therefore encodes the conditional density

$$\begin{aligned} &p([Re(d_1^0), Im(d_1^0)], \ldots, [Re(d_p^0), Im(d_p^0)], \ldots, \\ &\quad [Re(d_p^{n_s}), Im(d_p^{n_s})] | STFT(X)), \end{aligned} \tag{12}$$

resulting in the CWLL

$$\begin{aligned} &\ell([Re(d_1^0), Im(d_1^0)], \ldots, [Re(d_p^0), Im(d_p^0)], \ldots, \\ &\quad [Re(d_p^{n_s}), Im(d_p^{n_s})] | STFT(X)). \end{aligned} \tag{13}$$

The structural constraints of completeness and decomposability for SPNs still hold, see Yu et al. [2021a].

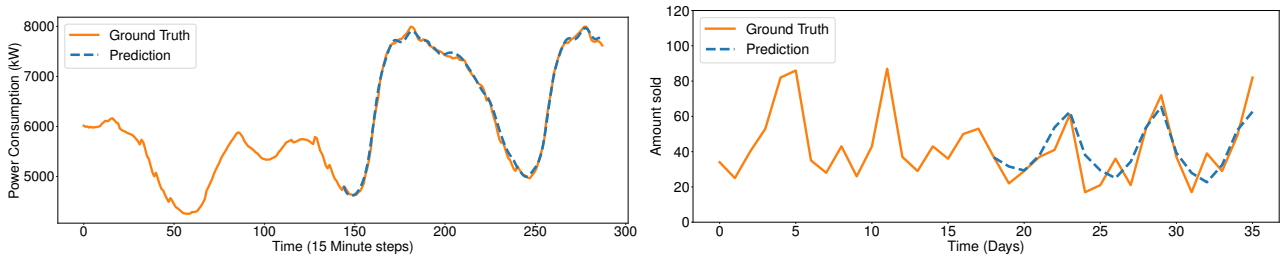

Figure 4: Illustration of exemplary predictions on the *Power* (left) and *Retail* dataset (right), the context has been cut for visual purposes. In comparison, we can observe much bigger variance between time steps for the *Retail* data, causing the SRNN prediction to be less accurate.

2.4 SPECTRAL RNN

We employ Spectral RNN [Wolter et al., 2020] for univariate time series forecasting. Compared to a standard RNN, the recurrent steps are performed over the windows retrieved from STFT. Therefore, for a window with width T_w and a step size S, it only has to perform n_s instead of T time steps for an input of length T. Mathematically, the SRNN is defined as follows:

$$\mathbf{X}_\tau = STFT(X)^\tau \tag{14}$$

$$\mathbf{z}_\tau = \mathbf{W}_c \mathbf{h}_{\tau-1} + \mathbf{V}_c \mathbf{X}_\tau + \mathbf{b}_c \tag{15}$$

$$\mathbf{h}_\tau = f_a(\mathbf{z}_\tau) \tag{16}$$

$$\mathbf{y}_\tau = iSTFT(\mathbf{W}_{pc}\mathbf{h}_0, \ldots, \mathbf{W}_{pc}\mathbf{h}_\tau), \tag{17}$$

with $\tau = [0, n_s]$ enumerating the total number of segments n_s. Since $\mathbf{X}_\tau \in \mathbb{C}^{\mathscr{T}} \times 1$ is a complex signal, the RNN cell either needs to operate in the complex space or needs to provide projections $\mathscr{I} : \mathbb{C}^{\mathscr{T}} \mapsto \mathbb{R}^{n_i}$, $\mathscr{O} : \mathbb{R}^{n_o} \mapsto \mathbb{C}^{\mathscr{T}}$ for in- and outputs respectively. Since complex units have not shown superior in our experiments (similar to findings in Wolter et al. [2020]), we employ standard Gated Recurrent Units (GRU) [Chung et al., 2014]. As projections, we employ concatenation and splitting respectively, i.e. $\mathscr{I}(X_\tau) = (Re(X_\tau), Im(X_\tau))$, $\mathscr{O}(h_\tau) = h_\tau^{1,\ldots,\mathscr{T}} + h_\tau^{\mathscr{T}+1,\ldots,2\mathscr{T}} i$ and $n_i = n_o = 2\mathscr{T}$. Therefore, the weights have the form of $\mathbf{h}_\tau \in \mathbb{R}^{n_h \times 1}, \mathbf{W}_c \in \mathbb{R}^{n_h \times n_h}, \mathbf{V}_c \in \mathbb{R}^{n_h \times 2\mathscr{T}}, \mathbf{b}_c \in \mathbb{R}^{n_h \times 1}$ and $\mathbf{W}_{pc} \in \mathbb{R}^{n_h \times 2\mathscr{T}}$, where n_h is the size of the hidden state and $\mathscr{T}$ the reduced amount of frequencies in the STFT. Further details on SRNN may be found in Wolter et al. [2020].

2.5 TRAINING RECOWNS

Training SRNN and CWSPN can be achieved jointly in a co-ordinate descent fashion, as shown in Figure 3, resulting in RECOWN. The overall network structure of RECOWN is shown in Figure 1. In each optimization step, the weights of the SRNN are updated first by minimizing the Mean Squared Error (MSE-loss), i.e. $MSE(Y_{GT}, Y_{Pred})$. Afterwards, the SRNN weights are fixed and the CWSPN is to be optimized w.r.t. minimizing an MSE-weighted negative Log-Likelihood (LL) loss on M samples of data:

$$-\frac{1}{M}\sum_{i=0}^{M} \frac{\ell(Y^i_{Pred}|X^i)}{SE(Y^i_{GT}, Y^i_{Pred})^2}, \tag{18}$$

where SE denotes the Squared Error. Therefore, the CWSPN learns a skewed conditional distribution of predictions given context. By using the inverse weighting by the SE, we simulate that less accurate predictions (high SE) occur less frequently in the distribution, while better predictions (low SE) are simulated to appear more often in the distribution. This helps the CWSPN to learn a more usable likelihood, as we will show in section 3. Furthermore, this allows for any gradient-descent-based density estimator to be placed in the RECOWN structure. Our experiments, however, as we will see in the next section, show CWSPNs to be especially suited to be used in the RECWON structure.

3 EXPERIMENTAL EVALUATION

To show the benefits of using RECOWNs, we investigate the following research questions:

(Q1) Quality of Predictions Can RECOWNs differentiate "bad" from "good" predictions on a sequence level?

(Q2) Uncertainty of Predictions Using the proposed LLRS, can RECOWNs estimate the uncertainty of predictions in the time domain?

3.1 DATASETS

For our experiments we evaluate the performance on two different datasets. We apply z-score normalization to normalize the data for all of our experiments.

Firstly, the problem of forecasting *Power* consumption, using the 15 minute-frequency data from the European Net-

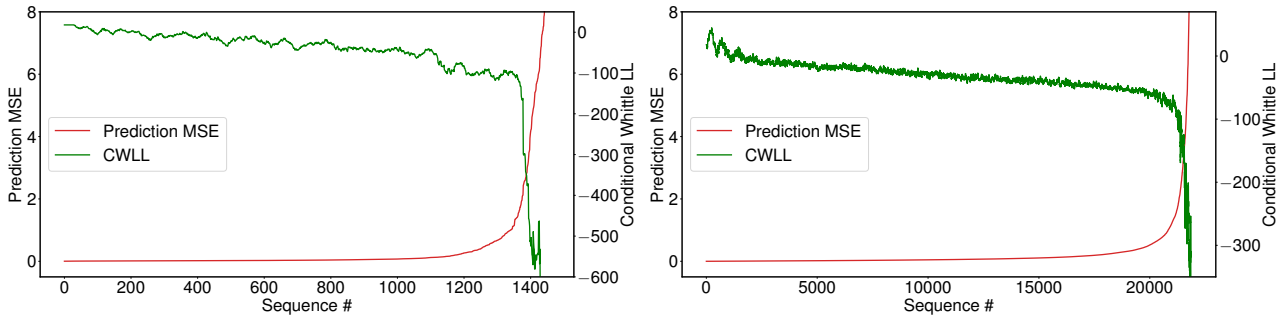

Figure 5: RECOWNs can correctly separate high-from low-error predictions, illustrated by the correlation of CWLL and MSE on *Power* (left) and *Retail* dataset (right). On the x-axis, the enumeration of all test sequences (in- and output) is denoted, ordered ascending by MSE. We see a clear correlation between a sinking CWLL and a rising MSE. For visual purposes, the CWLL is smoothed by a moving average of 12.

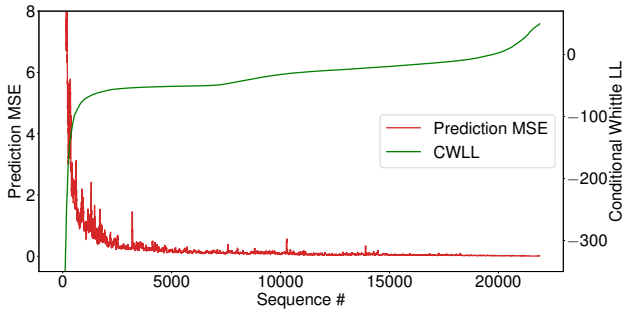

Figure 6: RECOWNs CWLL can be successfully employed to sort sequences w.r.t. their expected error, even when ground truth is not available yet. It represents the application use case, in which possibly "bad" need to be detected beforehand. This is illustrated by the correlation of the MSE with the CWLL on the *Retail* dataset. On the x-axis, the enumeration of all test sequences (in- and output) is denoted, ordered ascending by CWLL. For visual purposes, the MSE is smoothed by a moving average of 12.

work of Transmission System Operators for Electricity [1]. Given 14 days of context, the network needs to predict the power load from noon to midnight the following day (i.e. 1.5 days). As window size, we choose 96, which corresponds to a full day given the 15-minute sampling rate.

Secondly, the task of forecasting the *Retail* demand, using data from a retail location of a big german retailer, spanning 2 years and roughly 4000 different products with a daily sampling rate. Here, the task is to predict three weeks of demand given half a year of context. Since no sales data of Sundays is present, we filter them out, making a window size of 12 a reasonable choice, spanning 2 weeks of data. We deliberately use a smaller window size here as compared to the *Power* dataset, in order to verify that the approach works for different window sizes. Examples for sequences of both datasets can be found in Figure 4, together with exemplary predictions of the SRNN. Regarding the low-pass filter of STFT, we apply it with a factor of four to the *Power* and with a factor of two to the *Retail* data.

[1] `https://transparency.entsoe.eu/` - we used the crawled version made available by Wolter et al. [2020]

3.2 CLASSIFICATION ACCURACY

As the target of our approach is to use the likelihood of the CWSPN as an indicator for the quality of predictions, we will focus our evaluation on whether this correlation is true for a significant part of our test sequences or not. To do so, we will first have a look at the correlation using plots, while accessing it afterwards mathematically using a correlation error.

3.2.1 Correlation Plots

To visually access the quality of the CWLL as an indicator for the prediction quality, we first calculate the CWLL and MSE for all test sequences of each dataset and plot those sorted ascending by MSE. As shown in Figure 5, a clear correlation between CWLL and MSE exists: The higher the MSE, the lower the conditional Whittle likelihood. The same applies for the magnitude, i.e. the exponential-shaped curve of the MSE is reflected in the CWLL. Based on this observation, we assume that the CWLL can be a good indicator for the prediction quality, and may therefore fulfill the research question (Q1). A hypothesis, we want to mathematically analyze further ins section 3.2.2.

As in the application case, the ground truth is not (yet) known and the MSE is therefore missing, one could use the CWLL to sort sequences from high risk (aka low CWLL) to low risk (aka high CWLL) and further analyze those high-risk frequencies. This use case is depicted in Figure 6, showing again a good quality correlation between a rising CWLL and a lower MSE of prediction. To access it mathematically, we provide the following application example: Selecting the 5% sequences with the lowest CWLL

	Power			*Retail*		
	Small	Medium	Large	Small	Medium	Large
#Parameters	300k	900k	3M	30K	70K	200K
RECOWN	**0.019**	**0.016**	**0.011**	**0.036**	0.035	0.027
RECOWN (Time)	0.023	0.019	0.017	0.042	**0.031**	0.030
SRNN + MAF	0.045	0.026	**0.011**	0.044	0.033	**0.023**
SRNN + MAF (time)	0.093	0.058	0.051	0.047	0.045	0.029
Random		0.400			0.455	

Table 1: RECOWNs are trustworthy, shown by the test Correlation Error (lower is better) for all experimental configurations. The results indicate, that modeling in the spectral domain generally outperforms modeling in time domain w.r.t. the Correlation Error, in particular for MAF, significantly improving parameter efficiency. Furthermore, for smaller model sizes, RECOWN achieves the best scores, while MAF is better for larger model sizes.

from CWSPN on *Power*, we find that 75% of all sequences with the worst 5% of MSEs are included. Looking at the 10% sequences with the lowest CWLL covers 98.5% of all sequences with the worst 5% of MSEs, displaying the effectiveness of RECOWN.

3.2.2 Correlation Error (CE)

In order to provide a correlation error for each test sequence n, we calculate a relative prediction error (S^n_{Pred}) and a likelihood score (S^n_ℓ) respectively:

$$S^n_{Pred} = \sqrt{\frac{SE(Y^n_{Pred}, Y^n_{GT}) - \min_m SE(Y^m_{Pred}, Y^m_{GT})}{\max_m SE(Y^m_{Pred}, Y^m_{GT}) - \min_m SE(Y^m_{Pred}, Y^m_{GT})}}, \tag{19}$$

$$S^n_\ell = \sqrt{\frac{\ell(Y^n_{Pred}|X^n) - \max_m \ell(Y^m_{Pred}|X^m)}{\min_m \ell(Y^m_{Pred}|X^m) - \max_m \ell(Y^m_{Pred}|X^m)}}. \tag{20}$$

The square root is taken to account for the exponential shape of the conditional CWLL, as can be seen in Figure 5. Note that the calculation of S^n_ℓ corresponds to the likelihood ratios introduced in section 3.3.

Given that the MSE reflects the gold standard on where a sequence should be placed on the range from "bad" to "good", we define the Correlation Error (CE) for the CWLL as the quadratic distance of the scores:

$$CE^n = (S^n_{Pred} - S^n_\ell)^2. \tag{21}$$

Since $S^n_{Pred}, S^n_\ell \in [0,1]$ by definition, we have $CE^n \in [0,1]$. In order to enable a better assessment of this new proposed error, we provide a random baseline, which draws likelihood scores randomly from a uniform distribution, i.e. $S^n_{\ell_{random}} \sim \mathbf{U}(0,1)$.

For the evaluations of the correlation error, we compare the CWSPN with Masked Autoregressive Flow (MAF) [Papamakarios et al., 2017], a state-of-the-art neural density estimator. MAF is integrated into the joint training just as the CWSPN, therefore also follows the same training objective as given in Equation 18. For each system, we report scores on the Fourier coefficients as well as on the original time series (for CWSPN, modeling the time series in the time domain results in a standard CSPN). Furthermore, we evaluate three different model sizes, *Small, Medium, and Large*. Their respective amount of parameters is given in Table 1.

In general, we see that modeling in the spectral domain (i.e. using STFT) is mostly superior to operating in the time domain, seeming to improve parameter efficiency. This is especially true for MAF. Furthermore, while MAF achieves the best scores on bigger model sizes, CWSPN is especially good at modeling the CWLL with a smaller model size. In general, the error differences between the models are quite small, making especially *CWSPN* and *MAF* in all cases a good choice, enabling us to successfully answer the research question (Q1) – RECOWN can differentiate "bad" from "good" predictions on a sequence level. Note that SPN-architectures offer more inference options – like computing the MPE, all marginals in linear time, and similar – than MAF. Additionally, our experiments showed that the CWSPN is more stable w.r.t. changes in the hyperparameters compared to MAF. Finally, the predictions on the *Retail* dataset seem to be a bit harder to access overall compared to the *Power* data.

3.3 PROVIDING UNCERTAINTY TO PREDICTIONS

As an additional benefit, the conditional Whittle Likelihood can also be used to access the uncertainty for a prediction in the time domain, allowing to take insight into how it changes at different steps in time. To allow for this, we use the notion of likelihood ratios, leveraging the window function w to project the likelihood to the time domain at time step n:

$$\lambda_{LR}(n) = -2(w(n)\ell(Y_{pred}|X) - \max_{Y,X \in D_{Train}} \ell(Y|X)). \tag{22}$$

Furthermore, we define the maximum likelihood ratio occurring in the training data:

$$\lambda_{LRMax} = -2(\min_{Y,X \in D_{Train}} \ell(Y|X) - \max_{Y,X \in D_{Train}} \ell(Y|X)), \quad (23)$$

Using λ_{LRMax} as normalization for λ_{LR}, we can estimate the uncertainty of the prediction via the Log-Likelihood Ratio Score (LLRS):

$$LLRS(n) = \sqrt{\frac{\lambda_{LR}(n)}{\lambda_{LRMax}}}. \quad (24)$$

As with the correlation error, we take the square root to account for the exponential shape of the CWLL, as shown in Figure 5. Using this technique, a likelihood equally worse as the worst training likelihood (i.e. $\ell(Y_{pred}|X) = \min_{Y,X \in D_{Train}} \ell(Y|X)$) results in an LLRS of one: $LLRS = 1$. Bigger likelihoods (i.e. $\ell(Y_{pred}|X) > \min_{Y,X \in D_{Train}} \ell(Y|X)$) result in scores smaller one ($LLRS < 1$), smaller likelihoods (i.e. $\ell(Y_{pred}|X) < \min_{Y,X \in D_{Train}} \ell(Y|X)$) in larger scores ($LLRS > 1$). With z-score normalization, LLRS can be applied to any magnitude of the original data.

To show its potential, we ran a long-time forecasting experiment on the *Power* dataset over 40 days, performed by RECOWN trained for 5 days of prediction. We choose this setting, since we noticed an increasingly worsening prediction by the SRNN in the long run and wanted to make sure, that this unwanted behaviour is captured by the uncertainty estimated with CWLL. As it can be seen in Figure 2, the uncertainty grows together with an increasingly divergent prediction. This shows, that the uncertainty can be a good warning sign for users to detect such cases, even though the ground truth may not be available yet. Therefore, (Q2) can be answered affirmatively: RECOWNs can estimate the uncertainly of predictions in the time domain.

4 CONCLUSION

In this paper, we proposed RECOWNs, a novel architecture that jointly trains an SRNN and a probabilistic circuit in order to provide the SRNN with a measure of uncertainty of its predictions. This yields a trustworthy model for time series forecasting being able to inform the users when it is not confident of its predictions. We leveraged the conditional likelihood of in- and output together with the Whittle approximation to introduce CWSPNs, which can provide a likelihood measure of the prediction given the context in the spectral domain. Furthermore, we introduced LLRS, an effective score to evaluate the uncertainty interval for any time point of a prediction. Our experiments on real-world datasets show that RECOWNs are both accurate and trustworthy. In this context, a probabilistic circuit tailored for time series based on the Whittle approximation showed to be superior to MAF, a state-of-the-art density estimator. We hope our results will inspire further research on trustworthy models for time series forecasting.

Future work may extend our contributions in several ways. Since RECOWN can be trained in an end-to-end fashion, we envision more sophisticated strategies for joint training. For example, gradients from the CWSPN could be used to improve the predictions of the neural predictor. Moreover, it could be explored whether RECOWNs are robust against adversarial attacks. Besides, tractable and general inference of CWSPNs could be exploited to gain knowledge about factors of influence for the prediction, e.g., similarly to what done in Yu et al. [2021a], by constructing a Bayesian Network using the CWSPN. In the same direction, modeling the joint distribution of the coefficients instead of the conditional, thus, being able to compute any marginal, could open new opportunities. For this scope, employing Einsum Networks [Peharz et al., 2020a], fast and scalable probabilistic circuits, could be an effective solution.

References

Tayfun Alpay, Stefan Heinrich, and Stefan Wermter. Learning multiple timescales in recurrent neural networks. In *International conference on artificial neural networks*, pages 132–139. Springer, 2016.

Wessel Bruinsma, Eric Perim, William Tebbutt, J. Scott Hosking, Arno Solin, and Richard E. Turner. Scalable exact inference in multi-output gaussian processes. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119, pages 1190–1201, 2020.

Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*, 2014.

Mi Fei and Dit-Yan Yeung. Temporal models for predicting student dropout in massive open online courses. In *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, pages 256–263. IEEE, 2015.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pages 1050–1059, 2016.

Daniel Griffin and Jae Lim. Signal estimation from modified short-time fourier transform. *IEEE Transactions on acoustics, speech, and signal processing*, 32(2):236–243, 1984.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *Proceed-*

ings of the 34th International Conference on Machine Learning (ICML), pages 1321–1330, 2017.

Jan Koutnik, Klaus Greff, Faustino Gomez, and Juergen Schmidhuber. A clockwork rnn. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, pages 1863–1871, 2014.

Nikolay Laptev, Jason Yosinski, Li Erran Li, and Slawek Smyl. Time-series extreme event forecasting with neural networks at uber. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 34, pages 1–5, 2017.

Kezhi Li, John Daniels, Chengyuan Liu, Pau Herrero, and Pantelis Georgiou. Convolutional recurrent neural networks for glucose prediction. *IEEE journal of biomedical and health informatics*, 24(2):603–613, 2019.

YiFei Li and Han Cao. Prediction for tourism flow based on lstm neural network. *Procedia Computer Science*, 129: 277–283, 2018.

Mazen Melibari, Pascal Poupart, Prashant Doshi, and George Trimponias. Dynamic sum product networks for tractable inference on sequence data. In *Proceedings of International Conference on Probabilistic Graphical Models (PGM)*, pages 345–355, 2016.

George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*, pages 2335–2344, 2017.

Robert Peharz, Steven Lang, Antonio Vergari, Karl Stelzner, Alejandro Molina, Martin Trapp, Guy Van den Broeck, Kristian Kersting, and Zoubin Ghahramani. Einsum networks: Fast and scalable learning of tractable probabilistic circuits. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020a.

Robert Peharz, Antonio Vergari, Karl Stelzner, Alejandro Molina, Xiaoting Shao, Martin Trapp, Kristian Kersting, and Zoubin Ghahramani. Random sum-product networks: A simple and effective approach to probabilistic deep learning. In *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 334–344, 2020b.

Hoifung Poon and Pedro Domingos. Sum-product networks: a new deep architecture. In *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 337–346, 2011.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. MIT Press, 2006.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

Matthias Seeger. Gaussian processes for machine learning. *International journal of neural systems*, 14(02):69–106, 2004.

Xiaoting Shao, Alejandro Molina, Antonio Vergari, Karl Stelzner, Robert Peharz, Thomas Liebig, and Kristian Kersting. Conditional sum-product networks: Imposing structure on deep probabilistic architectures. In *International Conference on Probabilistic Graphical Models (PGM)*, pages 401–412, 2020.

Sima Siami-Namini, Neda Tavakoli, and Akbar Siami Namin. A comparison of arima and lstm in forecasting time series. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1394–1401. IEEE, 2018.

Alex Tank, Nicholas J Foti, and Emily B Fox. Bayesian structure learning for stationary time series. In *Proceedings of the 31st Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 872–881, 2015.

Martin Trapp, Robert Peharz, Franz Pernkopf, and Carl Edward Rasmussen. Deep structured mixtures of gaussian processes. In *Proceedings of International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 2251–2261, 2020.

Peter Whittle. The analysis of multiple stationary time series. *Journal of the Royal Statistical Society: Series B (Methodological)*, 15(1):125–139, 1953.

Moritz Wolter, Jürgen Gall, and Angela Yao. Sequence prediction using spectral rnns. In *International Conference on Artificial Neural Networks*, pages 825–837. Springer, 2020.

Zhongjie Yu, Fabrizio Ventola, and Kristian Kersting. Whittle networks: A deep likelihood model for time series. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021a.

Zhongjie Yu, Mingye Zhu, Martin Trapp, Arseny Skryagin, and Kristian Kersting. Leveraging probabilistic circuits for nonparametric multi-output regression. In *Proceedings of the 37th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2021b.