Appendix to Whittle Networks: A Deep Likelihood Model for Time Series

Zhongjie Yu Fabrizio Ventola Kristian Kersting

A. Stationary Time Series Datasets

Two datasets from financial market index are employed in order to show the capability of modeling stationary time series with WSPNs. The first stationary time series dataset is formed by the index values of 11 sectors from "Standard & Poor's" (*S&P*) from October 16, 2013 to May 24, 2019. The second is global stock index (*Stock*) from 17 markets extracted from June 2, 1997 to June 30, 1999. Before modeling the joint distribution in the spectral domain, the real-world market data is first converted to its log-return:

$$r_t = 100 \log(x(t)/x(t-1)). \tag{1}$$

Both *S&P* and *Stock* datasets are transformed with a sliding window of size 32. The *S&P* time series has length 1408 after the log-return transformation, and thus 44 time series are extracted by sliding window without overlap. The *Stock* series has length 522 after the log-return transformation. The sliding window is enabled with a step size of 10 in order to have more time series for training. In the end, 50 time series instances with length 32 are extracted. Tab. 1 lists all the names of the *Stock* index and the corresponding markets. More details of the *Stock* dataset can be found in Tank et al. (2015).

The *VAR* series is simulated from an order-1 vector autoregressive process, with $p = 7$ dimensions. Time series is simulated from the model:

$$x(t) = Ax(t-1) + \epsilon(t), \tag{2}$$

where $x(t) \in \mathbb{R}^p$, $A \in \mathbb{R}^{p \times p}$ and $\epsilon \sim \mathcal{N}(0, I_{p \times p})$. Following Tank et al. (2015) and Songsiri & Vandenberghe (2010), we first restrict A to be upper triangular, and set the diagonal elements to a constant $A_{ii} = 0.5$. Then, the upper diagonal elements A_{ij} are sampled from a Binomial distribution with $p = 0.2$. The corresponding inverse spectral density matrix of the process is:

$$S(\lambda)^{-1} = I + A^T A + e^{-i\lambda} A + e^{i\lambda} A^T. \tag{3}$$

The conditional independencies between time series are encoded by zeros in the inverse spectral density matrix $S(\lambda)^{-1}$. The matrix A is accepted when 1) the absolute values of all eigenvalues of A are less than one, making the series stationary, and 2) the graph G determined by A is decomposable. We generate the 7-D series from the following matrix $A =$

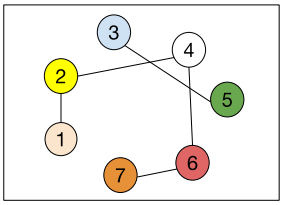

Figure 1. Graph visualization of the conditional independencies of the simulated *VAR* series.

$$\begin{bmatrix} 0.5 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.5 \end{bmatrix}. \tag{4}$$

An example of the inverse spectral density matrix at frequency $\lambda = \pi/10$ is $S(\lambda = \pi/10)^{-1} \approx$

$$\begin{bmatrix} 2.2 & \bar{k} & 0 & 0 & 0 & 0 & 0 \\ k & 3.2 & 0 & \bar{k} & 0 & 0 & 0 \\ 0 & 0 & 2.2 & 0 & \bar{k} & 0 & 0 \\ 0 & k & 0 & 3.2 & 0 & \bar{k} & 0 \\ 0 & 0 & k & 0 & 3.2 & 0 & 0 \\ 0 & 0 & 0 & k & 0 & 3.2 & \bar{k} \\ 0 & 0 & 0 & 0 & 0 & k & 3.2 \end{bmatrix}, \tag{5}$$

where $k = 1.5 + 0.3i$ and $\bar{k}$ being its conjugate.

This inverse spectral density matrix implies the conditional independencies shown in Fig. 1, which is also shown in Fig. 7 (Left) in Section 5 in the paper. From the above matrix A, a 7-D series with length 557056 is generated. With a sliding window of size 32, 17408 series instances are extracted, where 16384 form the training set and 1024 the test set.

B. General Time Series Datasets

In this paper, we investigated five non-stationary time series datasets. The synthetic *Sine* data consists of 6 components: 3 trigonometric sines with same frequency while different

Market Code	Market	Ticker	Index Name
AT	Austria	ATX	Austrian Traded Index
AU	Australia	AORD	All Ordinary Composite
BE	Belgium	BFX	BEL 20
CA	Canada	GSPTSE	Toronto Stock Exchange 300
CH	Switzerland	SSMI	Swiss Market Index
FN	Finland	OMXH25	OMX Helsinki 25
FR	France	FCHI	CAC 40
GE	Germany	GDAX	DAX 30
HK	Hong Kong	HSI	Hang Seng Composite
IR	Ireland	ISEQ	Irish Stock Exchange Index
IT	Italy	FTMIB	FTSEMIB
JP	Japan	N225	Nikkei 225
NE	Netherlands	AEX	Amsterdam Exchange Index
PO	Portugal	PSI20	Portugal Stock Index
SP	Spain	IBEX	IBEX 35
UK	United Kingdom	FTSE	FTSE 100
US	United States	SPX	S&P 500

Table 1. The *Stock* dataset information.

phases (Sine11, Sine12, and Sine13) plus Gaussian noise; 2 sine series with another frequency with different phases (Sine21 and Sine22) plus Gaussian noise; and one series of pure Gaussian noise (Gauss1). The *training* and *test* sets have the 6 components in the order of: "Sine11, Sine12, Sine13, Sine21, Sine22, Gauss1", while the *ood* set has the following order: "Sine21, Sine22, Gauss1, Sine11, Sine12, Sine13". Each time series instance has length 32 with 6 components. In total, 16384 samples are generated for *training*, 1024 for *test* and 1024 as *ood* samples.

The synthetic *Billiards* data contains simulations of trajectories of three balls. The balls perform elastic collisions with each other or against the walls of the environment. The horizontal and vertical locations of the 3 balls form a 6-dimensional state vector at each time step. The *ood* data is generated by keeping the movement of one direction and replacing the movement of the other direction with Gaussian noise. Additionally, in the ood set the balls pass through each other instead of colliding. Therefore, the above behaviour makes the trajectory of ood set unrealistic and unnatural. Each trajectory has the locations of 100 time steps, which forms a multivariate time series with 6 components and a length of 100. We generate 9700 samples for *training*, 300 for *test* and 300 as *ood*.

The synthetic *Mackey-Glass* series is simulated from:

$$\frac{dx}{dt} = \frac{\beta x_\tau}{1 + x_\tau^n} - \gamma x, \tag{6}$$

where $\gamma = 0.1$, $\beta = 0.2$, $n = 10$. We simulate two series independently, one with a delay of $\tau = 17$, and another with a delay of $\tau = 17/3$, to form a 2-D multivariate time series. In total, 3000 series instances with length 1024 are generated, with the first 544 steps for training and last 480 steps for test.

Regarding MNIST, the *training* set consists of all the original training samples with labels "0-4". The *test* set consists of original test samples with labels "0-4" and our *ood* set consists of original test samples with labels "5-9", namely **outlier1**. The test set from Fashion-MNIST is also used as another outlier set for the Whittle Networks experiment. In order to have a similar number of samples in the second outlier set, images labeled "Sandal, Shirt, Sneaker, Bag, and Ankle boot" are used to form **outlier2**. The images are down-sampled to 14×14, with each row as one component of a 14-dimensional multivariate time series.

Finally, we used hyperspectral images of *plants* for a qualitative analysis of anomaly detection. The images were taken from leaves of sugar beet either healthy or inoculated with a disease named *Cercospora beticola*, with 328 wavelengths from 380nm to 1010nm. Each 328-D vector from one pixel can be viewed as a single univariate time series. There are 3 classes of pixels in one image, healthy, inoculated, and background. Tab. 2 summarizes the statistics of the introduced datasets.

C. Forecasting of *Mackey-Glass* Dataset

Forecasting is performed window by window. That is, given a window of series X_{t-1}, we try to predict the value of the next window X_t at once. Conditional distribution of the two windows of series is modeled in order to do forecasting.

	L	p	$\lvert T_{train}\rvert$	$\lvert T_{test}\rvert$	$\lvert T_{ood}\rvert$
S&P	32	11	44	-	-
Stock	32	17	50	-	-
VAR	32	7	16384	1024	-
Sine	32	6	16384	1024	1024
Mackey-Glass	32/64	2	48000	45000	-
MNIST	14	14	30596	5139	4861
Billiards	100	6	9700	300	300

Table 2. Datasets statistics. L is the length of a sample, p is the number of components in multivariate time series.

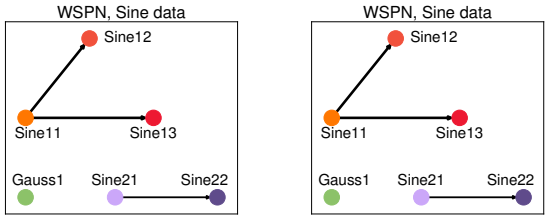

Figure 2. Directed independence structure among the 6 components discovered by Whittle sum-product network (Left) and non-Bayesian TGM (Right).

With a sliding window of size 64 with step size 32, 16 windows are extracted from each training instance with length 544. In order to model the conditional distribution of either $p(X_t \mid X_{t-1})$ in the time domain, or $p(d_t \mid d_{t-1})$ in the Fourier domain, the first 32 steps in one window form X_{t-1}, and the last 32 steps form X_t.

The hyperparameters of both CSPN and conditional WSPN are: $C = 1$, as we want to model the conditional distribution of data without class labels, and depth $D = 2$, number of splittings $R = 8$, number of sum nodes in regions $S = 8$, input distributions per leaf region $I = 4$. Both models are trained with Adam optimizer for 10 epochs, with a learning rate of 0.001 and a batch size of 64. Details of the CSPN settings and hyperparameters can be found in Shao et al. (2020).

In the test phase with conditional distributions, the true value of X_{t-1} is given, and the MPE of either $p(X_t \mid X_{t-1})$ or $p(d_t \mid d_{t-1})$ is estimated as the prediction. Note that in the basic LSTM experiment, the initial hidden state is set to a 0 vector at the beginning of the test. This is the reason why the prediction of LSTM from the 32^{nd} step (start of prediction) is far from the true value.

Regarding LSTM architecture, it is composed by stacking one LSTM recurrent layer having 32 hidden units and one linear layer that transforms the 32-dimensional vectors in 2-dimensional vectors. We trained the network for 100 epochs with Adam on min-max scaled data. We employed a learning rate of 0.05, batch size of 512 samples, and MSE as loss function. The model has been implemented in PyTorch 1.7.1 using the default values for the other hyperparameters.

D. Independence Structure of *Sine* Dataset

In order to explore the ability of discovering conditional independencies of non-stationary time series, we apply WSPNs to the *Sine* dataset and extract independence structures (DAGs) from it. The resulting structures are shown in Fig. 2. One can clearly see that the three sine components ("Sine11, Sine12 and Sine13") that have the same frequency are highly correlated. The other two components ("Sine21 and Sine22") are also highly correlated while the Gaussian noise component is independent of the others. As a comparison, non-Bayesian TGM produces the same directed graph structure since the synthetic sine dataset is relatively simple.

E. RAT-SPN Hyperparameters

The RAT-SPN hyperparameters, see Peharz et al. (2020) for details, were set as follows: $C = 1$, as we want to model the joint distribution of data without class labels, and depth $D = 7$, number of splittings $R = 2$, number of sum nodes in regions $S = 2$, input distributions per leaf region $I = 2$. We use the Adam optimizer with a learning rate of 0.003 on *plants* dataset, and a learning rate of 0.004 on MNIST dataset.

F. Whittle Network Results

Additional results from Whittle Network, instantiated as Whittle AE, on MNIST are shown in Fig. 3. The results support our claim that Whittle Networks indeed provide meaningful probabilities to neural networks. Both **outlier1** (images of digits "5-9") and **outlier2** (images from Fashion-MNIST) have a lower average likelihood compared to the average likelihood of our MNIST test data which is composed of in-domain samples, in other words, unseen samples of the same "0-4" digits (i.e. labels) as the training set. Furthermore, **outlier2** (clothes images) has even lower likelihood than **outlier1** (handwritten digits images) given it is from a very different domain. This shows that Whittle Network is also able to clearly distinguish very different domains.

G. Computational Complexity and Running Times

The following running times were estimated on a workstation with AMD Ryzen Threadripper 1950X 16-Core Processor with 128GB of RAM. The deep neural network experiments were executed on a GPU NVIDIA GeForce GTX 1080 Ti with 11GB of RAM. Learning both the structure

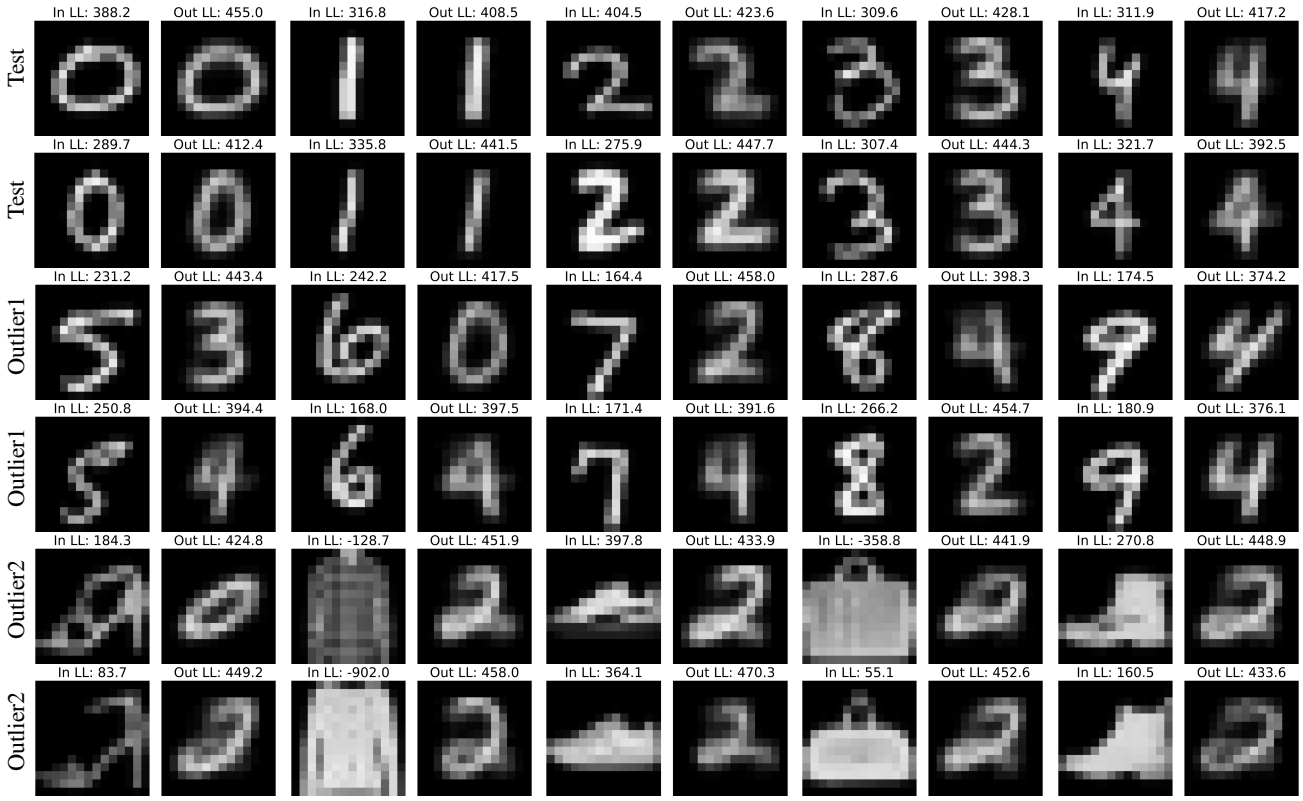

Figure 3. Additional qualitative results of Whittle Network, instantiated as Whittle AE. Visualization and likelihood (log-scale) of input (In) and output (Out) from (1st & 2nd Row) MNIST *test* set (digits "0-4"), (3rd & 4th Row) **outlier1** (MNIST *test* set digits "5-9"), and (5th & 6th Row) **outlier2** (Fashion-MNIST *test* set). Whittle Network provides higher likelihood to in-domain samples of "0-4" digits (same digits of training set samples) and lower likelihood to both outlier sets. Moreover, Whittle Network attributes lower likelihood to **outlier2** w.r.t. **outlier1** showing that is also able to clearly distinguish very different domains (clothes and handwritten digits).

and the parameters of the WSPN on CPU takes 15min on *Sine* dataset, 106min on MNIST, 23min on *S&P*, 76min on *Stock*, and 65min on *Billiards*. Computing the Whittle likelihood for all training, test, and OOD data takes 8min in total on MNIST and less than 1min on the other datasets. As a comparison, MADE takes 14min on *Sine*, 12min on MNIST, 15.4s on *S&P*, 18s on *Stock*, and 7min on *Billiards*. ResSPN takes 194min on *Sine*, 45h on MNIST, 35min on *S&P*, 64min on *Stock*, and 248min on *Billiards*.

We summarize the running times (in minutes) and the number of generated edges from the conditional independencies extraction procedure performed on CPU in Tab. 3. The results show that, when applying BIC, the complexity of the generated graphs can be reduced, resulting also in shorter running times. When BIC is not employed, the computational complexity of the graph generation is linear in the data size N and quadratic in the number of graph nodes k, i.e. $O(N \cdot k^2)$.

In the forecasting scenario, both CSPN and WSPN are trained on GPU, taking 3min 24s for training 10 epochs, and 16s for test. Training Whittle AE takes about 146min for 200 epochs on MNIST on GPU. Testing the Whittle AE on MNIST takes 20s in total.

		with BIC		**without BIC**	
	p	# edges	time	# edges	time
VAR (undir.)	7	5	16	7	19
VAR	7	5	44	8	53
Sine	6	3	26	5	31
Stock	17	19	288	28	308
S&P	11	11	40	17	42

Table 3. Running times (in minutes) and number of generated edges from the independencies extraction procedure on various datasets. *VAR (undir.)* refers to the generation of undirected graph while the others are DAGs.