Fast Temporal Wavelet Graph Neural Networks

Duc Thien Nguyen * † THIENND9@FPT.COM
Department of Information and Communications Engineering
Tokyo Institute of Technology, Tokyo, Japan

Manh Duc Tuan Nguyen *† TUANNMD@FPT.COM
Faculty of Information Networking for Innovation And Design
Toyo University, Tokyo, Japan

Truong Son Hy * ‡ TRUONGSON.HY@INDSTATE.EDU
Department of Mathematics and Computer Science
Indiana State University, Terre Haute, IN 47809, USA

Risi Kondor RISI@UCHICAGO.EDU
Department of Computer Science
University of Chicago, Chicago, IL 60637, USA

Editor: Sophia Sanborn, Christian Shewmake, Simone Azeglio, Nina Miolane.

Abstract

Spatio-temporal signals forecasting plays an important role in numerous domains, especially in neuroscience and transportation. The task is challenging due to the highly intricate spatial structure, as well as the non-linear temporal dynamics of the network. To facilitate reliable and timely forecast for the human brain and traffic networks, we propose the *Fast Temporal Wavelet Graph Neural Networks* (FTWGNN) that is both time- and memory-efficient for learning tasks on timeseries data with the underlying graph structure, thanks to the theories of *multiresolution analysis* and *wavelet theory* on discrete spaces. We employ *Multiresolution Matrix Factorization* (MMF) (Kondor et al., 2014) to factorize the highly dense graph structure and compute the corresponding sparse wavelet basis that allows us to construct fast wavelet convolution as the backbone of our novel architecture. Experimental results on real-world PEMS-BAY, METR-LA traffic datasets and AJILE12 ECoG dataset show that FTWGNN is competitive with the state-of-the-arts while maintaining a low computational footprint. Our PyTorch implementation is publicly available at https://github.com/HySonLab/TWGNN.

Keywords: Multiresolution Matrix Factorization, Temporal Graph Neural Networks, Wavelet Transform, Wavelet Convolution, Multivariate Timeseries.

1. Introduction

Time series modeling has been a quest in a wide range of academic fields and industrial applications, including neuroscience (Pourahmadi and Noorbaloochi, 2016) and traffic modeling (Li et al., 2018). Traditionally, model-based approaches such as autoregressive (AR) and Support Vector Regression (Smola and Schölkopf, 2004) require domain-knowledge

* Co-first authors
† Also affiliated with FPT Software AI Center, Hanoi, Vietnam
‡ Correspondent author

as well as stationary assumption, which are often violated by the complex and non-linear structure of neural and traffic data.

Recently, there has been intensive research with promising results on the traffic forecasting problem using deep learning such as Recurrent Neural Network (RNN) (Qin et al., 2017), LSTM (Koprinska et al., 2018), and graph-learning using Tranformer (Xu et al., 2020). On the other hand, forecasting in neuroscience has been focusing mainly on long-term evolution of brain network structure based on fMRI data, such as predicting brain connectivities of an Alzheimer's disease after several months (Bessadok et al., 2022), where existing methods are GCN-based (Göktaş et al., 2020) or GAN-based graph autoencoder (Gürler et al., 2020). Meanwhile, research on instantaneous time series forecasting of electroencephalogram (EEG) or electrocorticography (ECoG) remains untouched, even though EEG and ECoG are often cheaper and quicker to obtain than fMRI, while short-term forecasting may be beneficial for patients with strokes or epilepsy (Shoeibi et al., 2022).

In graph representation learning, a dense adjacency matrix expressing a densely connected graph can be a waste of computational resources, while physically, it may fail to capture the local "smoothness" of the network. To tackle such problems, a mathematical framework called Multiresolution Matrix Factorization (MMF) (Kondor et al., 2014) has been adopted to "sparsify" the adjacency and graph Laplacian matrices of highly dense graphs. MMF is unusual amongst fast matrix factorization algorithms in that it does not make a low rank assumption. Multiresolution matrix factorization (MMF) is an alternative paradigm that is designed to capture structure at multiple different scales. This makes MMF especially well suited to modeling certain types of graphs with complex multiscale or hierarchical structure (Hy and Kondor, 2022), compressing hierarchical matrices (e.g., kernel/gram matrices) (Teneva et al., 2016; Ding et al., 2017), and other applications in computer vision (Ithapu et al., 2017). One important aspect of MMF is its ability to construct wavelets on graphs and matrices during the factorization process (Kondor et al., 2014; Hy and Kondor, 2022). The wavelet basis inferred by MMF tends to be highly sparse, which allows the corresponding wavelet transform to be executed efficiently via sparse matrix multiplication. (Hy and Kondor, 2022) exploited this property to construct fast wavelet convolution and consequentially wavelet neural networks learning on graphs for graph classification and node classification tasks. In this work, we propose the incorporation of fast wavelet convolution based on MMF to build a time- and memory-efficient temporal architecture learning on timeseries data with the underlying graph structure.

From the aforementioned arguments, we propose the *Fast Temporal Wavelet Graph Neural Network* (FTWGNN) for graph time series forecasting, in which the MMF theory is utilized to describe the local smoothness of the network as well as to accelerate the calculations. Experiments on real-world traffic and ECoG datasets show competitive performance along with remarkably smaller computational footprint of FTWGNN. In summary:

- We model the spatial domain of the graph time series as a diffusion process, in which the theories of *multiresolution analysis* and *wavelet theory* are adopted. We employ *Multiresolution Matrix Factorization* (MMF) to factorize the underlying graph structure and derive its sparse wavelet basis.
- We propose the *Fast Temporal Wavelet Graph Neural Network* (FTWGNN), an end-to-end model capable of modeling spatiotemporal structures.

- We tested on two real-world traffic datasets and an ECoG dataset and achieved competitive results to state-of-the-art methods with remarkable reduction in computational time.

2. Related work

A spatial-temporal forecasting task utilizes spatial-temporal data information gathered from various sensors to predict their future states. Traditional approaches, such as the autoregressive integrated moving average (ARIMA), k-nearest neighbors algorithm (kNN), and support vector machine (SVM), can only take into account temporal information without considering spatial features (Van Lint and Van Hinsbergen, 2012; Jeong et al., 2013). Aside from traditional approaches, deep neural networks are proposed to model much more complex spatial-temporal relationships. Specifically, by using an extended fully-connected LSTM with embedded convolutional layers, FC-LSTM (Sutskever et al., 2014) combines CNN and LSTM to model spatial and temporal relations. When predicting traffic, ST-ResNet (Zhang et al., 2017) uses a deep residual CNN network, revealing the powerful capabilities of the residual network. Despite the impressive results obtained, traffic forecasting scenarios with graph-structured data is incompatible with all of the aforementioned methods because they are built for grid data. For learning tasks on graphs, node representations in GNNs (Kipf and Welling, 2016) uses a neighborhood aggregation scheme, which involves sampling and aggregating the features of nearby nodes. Since temporal-spatial data such as traffic data or brain network is a well-known type of non-Euclidean structured graph data, great efforts have been made to use graph convolution methods in traffic forecasting. As an illustration, DCRNN (Li et al., 2018) models traffic flow as a diffusion process and uses directed graph bidirectional random walks to model spatial dependency.

In the field of image and signal processing, processing is more efficient and simpler in a sparse representation where fewer coefficients reveal the information that we are searching for. Based on this motivation, Multiresolution Analysis (MRA) has been proposed by (Mallat, 1989) as a design for multiscale signal approximation in which the sparse representations can be constructed by decomposing signals over elementary waveforms chosen in a family called *wavelets*. Besides Fourier transforms, the discovery of wavelet orthogonal bases such as Haar (Haar, 1910) and Daubechies (Daubechies, 1988) has opened the door to new transforms such as continuous and discrete wavelet transforms and the fast wavelet transform algorithm that have become crucial for several computer applications (Mallat, 2008).

(Kondor et al., 2014) and (Hy and Kondor, 2022) have introduced Multiresolution Matrix Factorization (MMF) as a novel method for constructing sparse wavelet transforms of functions defined on the nodes of an arbitrary graph while giving a multiresolution approximation of hierarchical matrices. MMF is closely related to other works on constructing wavelet bases on discrete spaces, including wavelets defined based on diagonalizing the diffusion operator or the normalized graph Laplacian (Coifman and Maggioni, 2006) (Hammond et al., 2011) and multiresolution on trees (Gavish et al., 2010) (Bickel and Ritov, 2008).

3. Background

3.1. Multiresolution Matrix Factorization

Most commonly used matrix factorization algorithms, such as principal component analysis (PCA), singular value decomposition (SVD), or non-negative matrix factorization (NMF) are inherently single-level algorithms. Saying that a symmetric matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is of rank $r \ll n$ means that it can be expressed in terms of a dictionary of r mutually orthogonal unit vectors $\{u_1, u_2, \dots, u_r\}$ in the form

$$\boldsymbol{A} = \sum_{i=1}^{r} \lambda_i u_i u_i^T,$$

where $u_1, \dots, u_r$ are the normalized eigenvectors of A and $\lambda_1, \dots, \lambda_r$ are the corresponding eigenvalues. This is the decomposition that PCA finds, and it corresponds to factorizing $\boldsymbol{A}$ in the form

$$\boldsymbol{A} = \boldsymbol{U}^T \boldsymbol{H} \boldsymbol{U}, \tag{1}$$

where $\boldsymbol{U}$ is an orthogonal matrix and $\boldsymbol{H}$ is a diagonal matrix with the eigenvalues of $\boldsymbol{A}$ on its diagonal. The drawback of PCA is that eigenvectors are almost always dense, while matrices occuring in learning problems, especially those related to graphs, often have strong locality properties, in the sense that they tend to closely couple certain clusters of nearby coordinates, rather than those farther apart with respect to the underlying topology. In such cases, modeling A in terms of a basis of global eigenfunctions is both computationally wasteful and conceptually unreasonable: a localized dictionary would be more appropriate. In contrast to PCA, (Kondor et al., 2014) proposed *Multiresolution Matrix Factorization*, or MMF for short, to construct a sparse hierarchical system of L-level dictionaries. The corresponding matrix factorization is of the form

$$\boldsymbol{A} = \boldsymbol{U}_1^T \boldsymbol{U}_2^T \dots \boldsymbol{U}_L^T \boldsymbol{H} \boldsymbol{U}_L \dots \boldsymbol{U}_2 \boldsymbol{U}_1,$$

where $\boldsymbol{H}$ is close to diagonal and $\boldsymbol{U}_1, \dots, \boldsymbol{U}_L$ are sparse orthogonal matrices with the following constraints:

- Each $\boldsymbol{U}_\ell$ is k-point rotation (i.e. Givens rotation) for some small k, meaning that it only rotates k coordinates at a time. Formally, Def. 1 defines the k-point rotation matrix.
- There is a nested sequence of sets $\mathbb{S}_L \subseteq \dots \subseteq \mathbb{S}_1 \subseteq \mathbb{S}_0 = [n]$ such that the coordinates rotated by $\boldsymbol{U}_\ell$ are a subset of $\mathbb{S}_\ell$.
- $\boldsymbol{H}$ is an $\mathbb{S}_L$-core-diagonal matrix that is formally defined in Def. 2.

We formally define MMF in Defs. 3 and 4. A special case of MMF is the Jacobi eigenvalue algorithm (Jacobi, 1846) in which each $\boldsymbol{U}_\ell$ is a 2-point rotation (i.e. $k = 2$).

4. Method

4.1. Wavelet basis and convolution on graph

Section A.2 introduces the theory of multiresolution analysis behind MMF as well as the construction of a *sparse* wavelet basis for a symmetric matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$. Without the loss of

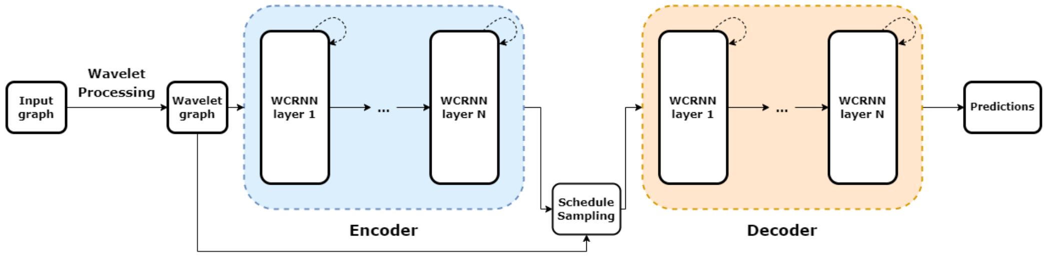

Figure 1: Architecture of Fast Temporal Wavelet Neural Network. **WC:** graph wavelet convolution given MMF's wavelet basis.

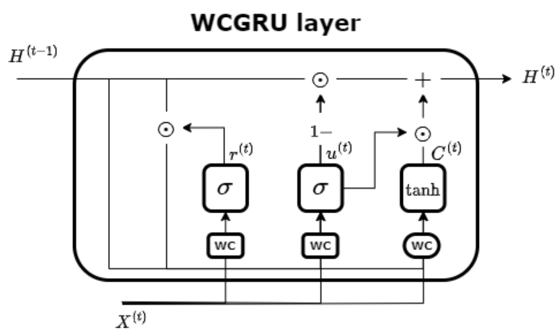

Figure 2: Architecture for the Wavelet Convolutional Gated Recurrent Unit. **WC**: graph wavelet convolution given MMF's wavelet basis

generality, we assume that $\boldsymbol{A}$ is a weight matrix of a weighted undirected graph $\mathcal{G} = (V, E)$ in which $V = \{v_1, .., v_n\}$ is the set of vertices and $E = \{(v_i, v_j)\}$ is the set of edges with the weight of edge (v_i, v_j) is given by $\boldsymbol{A}_{i,j}$. Given a graph signal $\boldsymbol{f} \in \mathbb{R}^n$ that is understood as a function $f : V \to \mathbb{R}$ defined on the vertices of the graph, the wavelet transform (up to level L) expresses this graph signal, without loss of generality $f \in \mathbb{V}_0$, as:

$$f(v) = \sum_{\ell=1}^{L} \sum_{m} \alpha_m^\ell \psi_m^\ell(v) + \sum_{m} \beta_m \phi_m^L(v), \quad \text{for each} \quad v \in V,$$

where $\alpha_m^\ell = \langle f, \psi_m^\ell \rangle$ and $\beta_m = \langle f, \phi_m^L \rangle$ are the wavelet coefficients. Based on the wavelet basis construction via MMF detailed in (Hy and Kondor, 2022):

- For L levels of resolution, we get exactly L mother wavelets $\overline{\psi} = \{\psi^1, \psi^2, \dots, \psi^L\}$, each corresponds to a resolution.
- The rows of $\boldsymbol{H} = \boldsymbol{A}_L$ make exactly $n - L$ father wavelets $\overline{\phi} = \{\phi_m^L = \boldsymbol{H}_{m,:}\}_{m \in \mathbb{S}_L}$. In total, a graph of n vertices has exactly n wavelets, both mothers and fathers.

Analogous to the convolution based on Graph Fourier Transform (Bruna et al., 2014), each convolution layer $k \in \{1, .., K\}$ of wavelet neural network transforms an input vector $\boldsymbol{f}^{(k-1)}$ of size $|V| \times F_{k-1}$ into an output $\boldsymbol{f}^{(k)}$ of size $|V| \times F_k$ as

$$\boldsymbol{f}_{:,j}^{(k)} = \sigma\left(\boldsymbol{W} \sum_{i=1}^{F_{k-1}} \boldsymbol{g}_{i,j}^{(k)} \boldsymbol{W}^T \boldsymbol{f}_{:,i}^{(k-1)}\right) \quad \text{for} \quad j = 1, \dots, F_k, \tag{2}$$

where $\boldsymbol{W}$ is our wavelet basis matrix as we concatenate $\overline{\phi}$ and $\overline{\psi}$ column-by-column, $\boldsymbol{g}_{i,j}^{(k)}$ is a parameter/filter in the form of a diagonal matrix learned in spectral domain, and σ is an element-wise non-linearity (e.g., ReLU, sigmoid, etc.). In Eq.(2), first we employ the wavelet transform of a graph signal $\boldsymbol{f}$ into the spectral domain (i.e. $\hat{\boldsymbol{f}} = \boldsymbol{W}^T \boldsymbol{f}$ is the forward transform and $\boldsymbol{f} = \boldsymbol{W}\hat{\boldsymbol{f}}$ is the inverse transform), then a learnable filter $\boldsymbol{g}$ to the wavelet coefficients, the inverse transform back to the spatial domain and let everything through a non-linearity σ. Since the wavelet basis matrix $\boldsymbol{W}$ is *sparse*, both the wavelet transform and its inverse transform can be implemented efficiently via sparse matrix multiplication.

4.2. Temporal Wavelet Neural Networks

Capturing spatiotemporal dependencies among time series in various spatiotemporal forecasting problems demands both spatial and temporal models. We build our novel *Fast Temporal Wavelet Graph Neural Network* with the architectural backbone from *Diffusion Convolutional Recurrent Neural Network* (DCRNN) (Li et al., 2018), that combines both spatial and temporal models to solve these tasks.

Spatial Dependency Model The spatial dynamic in the network is captured by diffusion process. Let $G = (\boldsymbol{X}, \boldsymbol{A})$ represent an undirected graph, where $\boldsymbol{X} = [\boldsymbol{x}_1^T, \dots, \boldsymbol{x}_N^T]^T \in \mathbb{R}^{N \times D}$ denotes signals of N nodes, each has D features. Define further the right-stochastic edge weights matrix $\tilde{\boldsymbol{A}} \in \mathbb{R}^{N \times N}$ in which $\sum_j \tilde{\boldsymbol{A}}_{ij} = 1 \forall i$. In the simplest case, when $\tilde{\boldsymbol{L}} = \boldsymbol{I} - \tilde{\boldsymbol{A}}$ is the nomalized random walk matrix, the diffusion process on graph is governed by the following equation (Chamberlain et al., 2021):

$$\frac{\mathrm{d}\boldsymbol{X}(t)}{\mathrm{d}t} = (\tilde{\boldsymbol{A}} - \boldsymbol{I})\boldsymbol{X}(t) \tag{3}$$

where $\boldsymbol{X}(t) = [\boldsymbol{x}_1^T(t), \dots, [\boldsymbol{x}_N^T(t)]^T \in \mathbb{R}^{N \times D}$ and $\boldsymbol{X}(0) = \boldsymbol{X}$. Applying forward Euler discretization with step size 1, gives:

$$\begin{aligned}\boldsymbol{X}(k) &= \boldsymbol{X}(k-1) + (\tilde{\boldsymbol{A}} - \boldsymbol{I})\boldsymbol{X}(k-1)\\ &= \boldsymbol{X}(k-1) - \tilde{\boldsymbol{L}}\boldsymbol{X}(k-1)\\ &= \tilde{\boldsymbol{A}}\boldsymbol{X}(k-1)\\ &= \tilde{\boldsymbol{A}}^k\boldsymbol{X}(0)\end{aligned} \tag{4}$$

Eq.4 is similar to the well-established GCN architecture propose in (Kipf and Welling, 2016). Then, the diffusion convolution operation over a graph signal $\mathbb{R}^{N \times D}$ and filter $f_{\boldsymbol{\theta}}$ is defined as:

$$\boldsymbol{X}_{:,d} \star_{\mathcal{G}} f_{\boldsymbol{\theta}} = \sum_{k=0}^{K-1} \theta_k \tilde{\boldsymbol{A}}^k \boldsymbol{X}_{:,d} \quad \forall d \in \{1, \dots, D\} \tag{5}$$

where $\boldsymbol{\Theta} \in \mathbb{R}^{K \times 2}$ are the parameters for the filter.

Temporal Dependency Model The DCRNN is leveraged from the recurrent neural networks (RNNs) to model the temporal dependency. In particular, the matrix multiplications in GRU is replaced with the diffusion convolution, which is called *Diffusion Convolutional*

Gated Recurrent Unit (DCGRU).

$$\begin{aligned}
\boldsymbol{r}^{(t)} &= \sigma(\boldsymbol{\Theta}_r \star_{\mathcal{G}} [\boldsymbol{X}^{(t)}, \boldsymbol{H}^{(t-1)}] + \boldsymbol{b}) \\
\boldsymbol{u}^{(t)} &= \sigma(\boldsymbol{\Theta}_r \star_{\mathcal{G}} [\boldsymbol{X}^{(t)}, \boldsymbol{H}^{(t-1)}] + \boldsymbol{b_u}) \\
\boldsymbol{C}^{(t)} &= \tanh\Big(\boldsymbol{\Theta}_r \star_{\mathcal{G}} [\boldsymbol{X}^{(t)}, (\boldsymbol{r} \odot \boldsymbol{H}^{(t-1)})] + \boldsymbol{b_c}\Big) \\
\boldsymbol{H}^{(t)} &= \boldsymbol{u}^{(t)} \odot \boldsymbol{H}^{(t-1)} + (1 - \boldsymbol{u}^{(t)}) \odot \boldsymbol{C}^{(t)}
\end{aligned}$$

where $\boldsymbol{X}(t), \boldsymbol{H}(t)$ denote the input and output of at time t, while $\boldsymbol{r}^{(t)}, \boldsymbol{u}^{(t)}$ are reset gate and update gate at time t, respectively.

Both the encoder and the decoder are recurrent neural networks with DCGRU following *Sequence-to-Sequence* style. To mitigate the distribution differences between training and testing data, scheduled sampling technique (Bengio et al., 2015) is used, where the model is fed with either the ground truth with probability ϵ_i or the prediction by the model with probability $1 - \epsilon_i$.

For our novel *Fast Temporal Wavelet Graph Neural Network* (FTWGNN), the fundamental difference is that instead of using temporal traffic graph as the input of DCRNN, we use the sparse wavelet basis matrix $\boldsymbol{W}$ which is extracted via MMF (see Section A.2) and replace the diffusion convolution by our fast *wavelet convolution*. Given the sparsity of our wavelet basis, we significantly reduce the overall computational time and memory usage. Each Givens rotation matrix $\boldsymbol{U}_\ell$ (see Def. 1) is a highly-sparse orthogonal matrix with a non-zero core of size $K \times K$. The number of non-zeros in MMF's wavelet basis $\boldsymbol{W}$, that can be computed as product $\boldsymbol{U}_1\boldsymbol{U}_2 \cdots \boldsymbol{U}_L$, is $O(LK^2)$ where L is the number of resolutions (i.e. number of Givens rotation matrices) and K is the number of columns in a Givens rotation matrix. (Kondor et al., 2014) and (Hy and Kondor, 2022) have shown in both theory and practice that L only needs to be in $O(n)$ where n is the number of columns and K small (e.g., 2, 4, 8) to get a decent approximation/compression for a symmetric hierarchical matrix. Technically, MMF is able to compress a symmetric hierararchical matrix from the original quadratic size $n \times n$ to a linear number of non-zero elements $O(n)$. Practically, all the Givens rotation matrices $\{\boldsymbol{U}_\ell\}_{\ell=1}^{L}$ and the wavelet basis $\boldsymbol{W}$ can be stored in Coordinate Format (COO), and the wavelet transform and its inverse in wavelet convolution (see Eq. 2) can be implemented efficiently by sparse matrix multiplication in PyTorch's sparse library (Paszke et al., 2019). The architecture of our model is shown in Figures 1 and 2.

5. Experiments

Our PyTorch implementation is publicly available at `https://github.com/HySonLab/TWGNN`. The implementation of multiresolution matrix factorization and graph wavelet computation (Hy and Kondor, 2022) is publicly available at `https://github.com/risilab/Learnable_MMF`.

To showcase the competitive performance and remarkable acceleration of FTWGNN, we conducted experiments on two well-known traffic forecasting benchmarks METR-LA and PEMS-BAY, and one challenging ECoG dataset AJILE12. We compare our model with widely used time series models (details included in the Appendix, Section B). Methods are evaluated on three metrics: (i) Mean Absolute Error (MAE); (ii) Mean Absolute Percentage

Error (MAPE); and (iii) Root Mean Squared Error (RMSE). FTWGNN and DCRNN are implemented using PyTorch (Paszke et al., 2019) on an NVIDIA A100-SXM4-80GB GPU. Below is our detail setting of FTWGNN.

Dataset	T	Metric	HA	ARIMA$_{kal}$	VAR	SVR	FNN	FC-LSTM	STGCN	GWaveNet	DCRNN	FTWGNN
METR-LA	15 min	MAE	4.16	3.99	4.42	3.99	3.99	3.44	2.88	**2.69**	2.77	2.70
		RMSE	7.80	8.21	7.89	8.45	7.94	6.30	5.74	**5.15**	5.38	**5.15**
		MAPE	13.0%	9.6%	10.2%	9.3%	9.9%	9.6%	7.6%	6.9%	7.3%	**6.8%**
	30 min	MAE	4.16	5.15	5.41	5.05	4.23	3.77	3.47	3.07	3.15	**3.02**
		RMSE	7.80	10.45	9.13	10.87	8.17	7.23	7.24	6.22	6.45	**5.95**
		MAPE	13.0%	12.7%	12.7%	12.1%	12.9%	10.9%	9.6%	8.4%	8.8%	**8.0%**
	60 min	MAE	4.16	6.90	6.52	6.72	4.49	4.37	4.59	3.53	3.60	**3.42**
		RMSE	7.80	13.23	10.11	13.76	8.69	8.69	9.40	7.37	7.59	**6.92**
		MAPE	13.0%	17.4%	15.8%	16.7%	14.0%	13.2%	12.7%	10.0%	10.5%	**9.8%**
PEMS-BAY	15 min	MAE	2.88	1.62	1.74	1.85	2.20	2.05	1.36	1.3	1.38	**1.14**
		RMSE	5.59	3.30	3.16	3.59	4.42	4.19	2.96	2.74	2.95	**2.40**
		MAPE	6.8%	3.5%	3.6%	3.8%	5.2%	4.8%	2.9%	2.7%	2.9%	**2.3%**
	30 min	MAE	2.88	2.33	2.32	2.48	2.30	2.20	1.81	1.63	1.74	**1.50**
		RMSE	5.59	4.76	4.25	5.18	4.63	4.55	4.27	3.70	3.97	**3.27**
		MAPE	6.8%	5.4%	5.0%	5.5%	5.43%	5.2%	4.2%	3.7%	3.9%	**3.2%**
	60 min	MAE	2.88	3.38	2.93	3.28	2.46	2.37	2.49	1.95	2.07	**1.79**
		RMSE	5.59	6.5	5.44	7.08	4.98	4.96	5.69	4.52	4.74	**3.99**
		MAPE	6.8%	8.3%	6.5%	8.0%	5.89%	5.7%	5.8%	4.6%	4.9%	**4.1%**

Table 1: Performance comparison of different models for traffic speed forecasting.

Data preparation For all datasets, the train/validation/test ratio is 0.7/0.2/0.1, divided into batch size 64.

Adjacency matrix The k-neighborhood of the traffic network in Eq. 16 is thresholded by $k = 0.01$, while for the brain network, the parameter λ_A is set to 10^{-5} in Task (17).

Wavelet basis For the traffic datasets, 100 mother wavelets are extracted, *i.e.*, $L = 100$, while for the AJILE12 dataset, $L = 10$ was used.

Model architecture For the RNN wavelet convolution, both encoder and decoder contain two recurrent layers, each with 64 units. The initial learning rate is 10^{-2}, decaying by $\frac{1}{10}$ per 20 epochs; the dropout ratio is 0.1; and the maximum diffusion step, *i.e.*, K, is set to 2. In addition, the optimizer is the Adam optimizer (Kingma and Ba, 2014).

The sparsity of wavelet bases is reported in Table 3, which demonstrate a remarkable compression of wavelet bases compared to that of Fourier bases.

Dataset	T	DCRNN	FTWGNN	Speedup
METR-LA	15 min	350s	**217s**	1.61x
	30 min	620s	**163s**	3.80x
	60 min	1800s	**136s**	13.23x
PEMS-BAY	15 min	427s	**150s**	2.84x
	30 min	900s	**173s**	5.20x
	60 min	1800s	**304s**	5.92x
AJILE12	1 sec	80s	**35s**	2.28x
	5 sec	180s	**80s**	2.25x
	15 sec	350s	**160s**	2.18x

Table 2: Training time/epoch between DCRNN and FTWGNN.

Dataset	Fourier basis	Wavelet basis
METR-LA	99.04%	**1.11%**
PEMS-BAY	96.35%	**0.63%**
AJILE12	100%	**1.81%**

Table 3: Sparsity bases (i.e. percentage of non-zeros).

5.1. Traffic prediction

Two real-world large-scale traffic datasets are considered:

- **METR-LA** Data of 207 sensors in the highway of Los Angeles County (Jagadish et al., 2014) over the 4-month period from Mar 1st 2012 to Jun 30th 2012.
- **PEMS-BAY** Data of 325 sensors in the Bay Area over the 6-month period from Jan 1st 2017 to May 31th 2017 from the California Transportation Agencies (CalTrans) Performance Measurement System (PeMS).

Dataset	T	Metric	HA	VAR	LR	SVR	LSTM	DCRNN	FTWGNN
AJILE12	1 sec	MAE	0.88	0.16	0.27	0.27	0.07	0.05	**0.03**
		RMSE	1.23	0.25	0.37	0.41	**0.09**	0.45	0.35
		MAPE	320%	58%	136%	140%	38%	7.84%	**5.27%**
	5 sec	MAE	0.88	0.66	0.69	0.69	0.39	0.16	**0.11**
		RMSE	1.23	0.96	0.92	0.93	0.52	0.24	**0.15**
		MAPE	320%	221%	376%	339%	147%	64%	**57%**
	15 sec	MAE	0.88	0.82	0.86	0.86	0.87	0.78	**0.70**
		RMSE	1.23	1.15	1.13	1.13	1.14	1.01	**0.93**
		MAPE	320%	320%	448%	479%	330%	294%	**254%**

Table 4: Performance comparison on ECoG signals forecast.

The distance function $\text{dist}(v_i, v_j)$ in (16) represents the road network distance from sensor v_i to sensor v_j, producing an asymmetric adjacency matrix for a directed graph. Therefore, the symmetrized matrix $\hat{\boldsymbol{A}} \coloneqq \frac{1}{2}(\boldsymbol{A} + \boldsymbol{A}^T)$ is taken to compute the wavelet basis matrix $\boldsymbol{W}$ following Sec. A.2.

Table 1 shows the evaluation of different approaches on the two traffic datasets, while Table 2 reports the training time per epoch of FTWGNN and DCRNN. Overall, FTWGNN performs better than DCRNN by about 10%, while it is significantly faster by about 5 times on average.

5.2. Brain networks

Annotated Joints in Long-term Electrocorticography for 12 human participants (AJILE12), publicly available at (Peterson et al., 2022b), records intracranial neural activity via the invasive ECoG, which involves implanting electrodes directly under the skull (Peterson et al., 2022a). For each participant, ECoG recordings are sporadically sampled at 500Hz in 7.4 ± 2.2 days (mean±std) from at least 64 electrodes, each of which is encoded with an unique set of Montreal Neurological Institute (MNI) x, y, z coordinates.

The proposed model is tested on the first one hour of recordings of subject number 5 with 116 good-quality electrodes. Subject 5 was chosen because he/she has the highest number of validated electrodes. Signals are downsampled to 1Hz, thus producing a network of 116 nodes, each with 3,600 data points. Furthermore, the signals are augmented by applying the spline interpolation to get the upper and lower envelopes along with an average curve (Melia et al., 2014) (see Figure 4). The adjacency matrix $\boldsymbol{A}$ is obtained by solving task (17), then the wavelet basis matrix $\boldsymbol{W}$ is constructed based on Sec. A.2.

Table 4 reports the performance of different methods on the AJILE12 dataset for 1-, 5-, and 15-second prediction. Generally, errors are much higher than those in the traffic forecasting problem, since the connections within the brain network are much more complicated and ambiguous (Breakspear, 2017). High errors using HA and VAR methods show that the AJILE12 data follows no particular pattern or periodicity, making long-step prediction extremely challenging. Despite having a decent performance quantitatively, Figure 5 demonstrates the superior performance of FTWGNN, in which DCRNN fails to approximate the trend and the magnitude of the signals. Even though FTWGNN performs well at 1-second prediction, it produces unstable and erroneous forecast at longer steps of 5 or 15 seconds. Meanwhile, similar to traffic prediction case, FTWGNN also sees a remarkable improvement in computation time by around 2 times on average (see Table 2).

6. Conclusion

We propose a new class of spatial-temporal graph neural networks based on the theories of multiresolution analysis and wavelet theory on discrete spaces with RNN backbone, coined *Fast Temporal Wavelet Graph Neural Network* (FTWGNN). Fundamentally, we employ *Multiresolution Matrix Factorization* to factorize the underlying graph structure and extract its corresponding sparse wavelet basis that consequentially allows us to construct efficient wavelet transform and convolution on graph. Experiments on real-world large-scale datasets show promising results and computational efficiency of FTGWNN in network time series modeling including traffic prediction and brain networks. Some future directions are: (i) investigating synchronization phenomena in brain networks (Honda, 2018); (ii) developing a robust model against outliers/missing data that appear frequently in practice; etc.

References

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. *Advances in neural information processing systems*, 28, 2015.

Alaa Bessadok, Mohamed Ali Mahjoub, and Islem Rekik. Graph neural networks in network neuroscience. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

Peter J. Bickel and Ya'acov Ritov. Discussion of: Treelets—an adaptive multi-scale basis for sparse unordered data. *The Annals of Applied Statistics*, 2(2):474–477, 2008. ISSN 19326157. URL http://www.jstor.org/stable/30244209.

Michael Breakspear. Dynamic models of large-scale brain activity. *Nature neuroscience*, 20 (3):340–352, 2017.

Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann Lecun. Spectral networks and locally connected networks on graphs. In *International Conference on Learning Representations (ICLR2014), CBLS, April 2014*, 2014.

Ben Chamberlain, James Rowbottom, Maria I Gorinova, Michael Bronstein, Stefan Webb, and Emanuele Rossi. Grand: Graph neural diffusion. In *International Conference on Machine Learning*, pages 1407–1418. PMLR, 2021.

Ronald R. Coifman and Mauro Maggioni. Diffusion wavelets. *Applied and Computational Harmonic Analysis*, 21(1):53–94, 2006. ISSN 1063-5203. doi: https://doi.org/10.1016/j.acha.2006.04.004. URL https://www.sciencedirect.com/science/article/pii/S106352030600056X. Special Issue: Diffusion Maps and Wavelets.

Ingrid Daubechies. Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 41:909–996, 1988.

Yi Ding, Risi Kondor, and Jonathan Eskreis-Winkler. Multiresolution kernel approximation for gaussian process regression. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/850af92f8d9903e7a4e0559a98ecc857-Paper.pdf.

Matan Gavish, Boaz Nadler, and Ronald R. Coifman. Multiscale wavelets on trees, graphs and high dimensional data: Theory and applications to semi supervised learning. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 367–374, Madison, WI, USA, 2010. Omnipress. ISBN 9781605589077.

Ahmet Serkan Göktaş, Alaa Bessadok, and Islem Rekik. Residual embedding similarity-based network selection for predicting brain network evolution trajectory from a single observation. In *International Workshop on PRedictive Intelligence In MEdicine*, pages 12–23. Springer, 2020.

Zeynep Gürler, Ahmed Nebli, and Islem Rekik. Foreseeing brain graph evolution over time using deep adversarial network normalizer. In *International Workshop on Predictive Intelligence In Medicine*, pages 111–122. Springer, 2020.

Alfred Haar. Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen*, 69: 331–371, 1910.

James Douglas Hamilton. *Time series analysis.* Princeton university press, 2020.

David K. Hammond, Pierre Vandergheynst, and Rémi Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150, 2011. ISSN 1063-5203. doi: https://doi.org/10.1016/j.acha.2010.04.005. URL https://www.sciencedirect.com/science/article/pii/S1063520310000552.

Haoyu Han, Mengdi Zhang, Min Hou, Fuzheng Zhang, Zhongyuan Wang, Enhong Chen, Hongwei Wang, Jianhui Ma, and Qi Liu. Stgcn: a spatial-temporal aware graph learning method for poi recommendation. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 1052–1057. IEEE, 2020.

Hirotada Honda. On mathematical modeling and analysis of brain network. In *Mathematical Analysis of Continuum Mechanics and Industrial Applications II: Proceedings of the International Conference CoMFoS16 16*, pages 169–180. Springer, 2018.

Truong Son Hy and Risi Kondor. Multiresolution matrix factorization and wavelet networks on graphs. In Alexander Cloninger, Timothy Doster, Tegan Emerson, Manohar Kaul, Ira Ktena, Henry Kvinge, Nina Miolane, Bastian Rice, Sarah Tymochko, and Guy Wolf, editors, *Proceedings of Topological, Algebraic, and Geometric Learning Workshops 2022*, volume 196 of *Proceedings of Machine Learning Research*, pages 172–182. PMLR, 25 Feb–22 Jul 2022. URL https://proceedings.mlr.press/v196/hy22a.html.

Vamsi K. Ithapu, Risi Kondor, Sterling C. Johnson, and Vikas Singh. The incremental multiresolution matrix factorization algorithm. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 692–701, 2017. doi: 10.1109/CVPR.2017.81.

C.G.J. Jacobi. Über ein leichtes verfahren die in der theorie der säcularstörungen vorkommenden gleichungen numerisch aufzulösen*). 1846(30):51–94, 1846. doi: doi: 10.1515/crll.1846.30.51. URL https://doi.org/10.1515/crll.1846.30.51.

H. V. Jagadish, Johannes Gehrke, Alexandros Labrinidis, Yannis Papakonstantinou, Jignesh M. Patel, Raghu Ramakrishnan, and Cyrus Shahabi. Big data and its technical challenges. *Commun. ACM*, 57(7):86–94, 2014. URL http://dblp.uni-trier.de/db/journals/cacm/cacm57.html#JagadishGLPPRS14.

Young-Seon Jeong, Young-Ji Byon, Manoel Mendonca Castro-Neto, and Said M Easa. Supervised weighting-online learning algorithm for short-term traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems*, 14(4):1700–1707, 2013.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Risi Kondor, Nedelina Teneva, and Vikas Garg. Multiresolution matrix factorization. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1620–1628, Bejing, China, 22–24 Jun 2014. PMLR. URL https://proceedings.mlr.press/v32/kondor14.html.

Risi Kondor, Nedelina Teneva, and Pramod Kaushik Mudrakarta. Parallel MMF: a multiresolution approach to matrix computation. *CoRR*, abs/1507.04396, 2015. URL http://arxiv.org/abs/1507.04396.

Irena Koprinska, Dengsong Wu, and Zheng Wang. Convolutional neural networks for energy time series forecasting. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2018. doi: 10.1109/IJCNN.2018.8489399.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=SJiHXGWAZ.

S.G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989. doi: 10.1109/34.192463.

Stphane Mallat. *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*. Academic Press, Inc., USA, 3rd edition, 2008. ISBN 0123743702.

Umberto Melia, Francesc Clariá, Montserrat Vallverdú, and Pere Caminal. Filtering and thresholding the analytic signal envelope in order to improve peak and spike noise reduction in eeg signals. *Medical engineering & physics*, 36(4):547–553, 2014.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Steven M. Peterson, Satpreet H. Singh, Benjamin Dichter, Michael Scheid, Rajesh P. N. Rao, and Bingni W. Brunton. Ajile12: Long-term naturalistic human intracranial neural recordings and pose. *Scientific Data*, 9(1):184, Apr 2022a. ISSN 2052-4463. doi: 10.1038/s41597-022-01280-y. URL https://doi.org/10.1038/s41597-022-01280-y.

Steven M. Peterson, Satpreet H. Singh, Benjamin Dichter, Michael Scheid, Rajesh P. N. Rao, and Bingni W. Brunton. Ajile12: Long-term naturalistic human intracranial neural recordings and pose (Version 0.220127.0436) [Data set]. *DANDI archive*, 2022b. URL https://doi.org/10.48324/dandi.000055/0.220127.0436.

Mohsen Pourahmadi and Siamak Noorbaloochi. Multivariate time series analysis of neuroscience data: some challenges and opportunities. *Current Opinion in Neurobiology*, 37:12–15, 2016. ISSN 0959-4388. doi: https://doi.org/10.1016/j.conb.2015.12.006. URL https://www.sciencedirect.com/science/article/pii/S0959438815001841. Neurobiology of cognitive behavior.

Yao Qin, Dongjin Song, Haifeng Cheng, Wei Cheng, Guofei Jiang, and Garrison W. Cottrell. A dual-stage attention-based recurrent neural network for time series prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, page 2627–2633. AAAI Press, 2017. ISBN 9780999241103.

Lawrence K Saul and Sam T Roweis. Think globally, fit locally: unsupervised learning of low dimensional manifolds. *Journal of machine learning research*, 4(Jun):119–155, 2003.

Afshin Shoeibi, Parisa Moridian, Marjane Khodatars, Navid Ghassemi, Mahboobeh Jafari, Roohallah Alizadehsani, Yinan Kong, Juan Manuel Gorriz, Javier Ramírez, Abbas Khosravi, et al. An overview of deep learning techniques for epileptic seizures detection and prediction based on neuroimaging modalities: Methods, challenges, and future works. *Computers in Biology and Medicine*, page 106053, 2022.

David I Shuman, Sunil K. Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, 30(3):83–98, 2013. doi: 10.1109/MSP.2012.2235192.

Konstantinos Slavakis and Isao Yamada. Fejér-monotone hybrid steepest descent method for affinely constrained and composite convex minimization tasks. *Optimization*, 67(11): 1963–2001, 2018.

Alex J. Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, aug 2004. ISSN 0960-3174. doi: 10.1023/B:STCO.0000035301.49549.88. URL https://doi.org/10.1023/B:STCO.0000035301.49549.88.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.

Hemant D. Tagare. Notes on optimization on stiefel manifolds. 2011.

Nedelina Teneva, Pramod Kaushik Mudrakarta, and Risi Kondor. Multiresolution matrix compression. In Arthur Gretton and Christian C. Robert, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 1441–1449, Cadiz, Spain, 09–11 May 2016. PMLR. URL https://proceedings.mlr.press/v51/teneva16.html.

JWC Van Lint and CPIJ Van Hinsbergen. Short-term traffic and travel time prediction models. *Artificial Intelligence Applications to Critical Transportation Issues*, 22(1):22–41, 2012.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. *arXiv preprint arXiv:1906.00121*, 2019.

Mingxing Xu, Wenrui Dai, Chunmiao Liu, Xing Gao, Weiyao Lin, Guo-Jun Qi, and Hongkai Xiong. Spatial-temporal transformer networks for traffic flow forecasting. *arXiv preprint arXiv:2001.02908*, 2020.

Junbo Zhang, Yu Zheng, and Dekang Qi. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

Appendix A. Multiresolution Matrix Factorization

A.1. Formal definitions

Definition 1 *We say that* $\boldsymbol{U} \in \mathbb{R}^{n\times n}$ *is an* ***elementary rotation of order*** k *(also called as a k-point rotation) if it is an orthogonal matrix of the form*

$$\boldsymbol{U} = \boldsymbol{I}_{n-k} \oplus_{(i_1,\cdots,i_k)} \boldsymbol{O}$$

for some $\mathbb{I} = \{i_1,\cdots,i_k\} \subseteq [n]$ *and* $\boldsymbol{O} \in \mathbb{SO}(k)$. *We denote the set of all such matrices as* $\mathbb{SO}_k(n)$.

Definition 2 *Given a set* $\mathbb{S} \subseteq [n]$, *we say that a matrix* $\boldsymbol{H} \in \mathbb{R}^{n\times n}$ *is* $\mathbb{S}$*-core-diagonal if* $\boldsymbol{H}_{i,j} = 0$ *unless* $i,j \in \mathbb{S}$ *or* $i = j$. *Equivalently,* $\boldsymbol{H}$ *is* $\mathbb{S}$*-core-diagonal if it can be written in the form* $\boldsymbol{H} = \boldsymbol{D} \oplus_{\mathbb{S}} \overline{\boldsymbol{H}}$, *for some* $\overline{H} \in \mathbb{R}^{|\mathbb{S}|\times|\mathbb{S}|}$ *and* $\boldsymbol{D}$ *is diagonal. We denote the set of all* $\mathbb{S}$*-core-diagonal symmetric matrices of dimension* n *as* $\mathbb{H}_n^{\mathbb{S}}$.

Definition 3 *Given an appropriate subset* $\mathbb{O}$ *of the group* $\mathbb{SO}(n)$ *of* n*-dimensional rotation matrices, a depth parameter* $L \in \mathbb{N}$, *and a sequence of integers* $n = d_0 \geq d_1 \geq d_2 \geq \cdots \geq d_L \geq 1$, *a* ***Multiresolution Matrix Factorization (MMF)*** *of a symmetric matrix* $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ *over* $\mathbb{O}$ *is a factorization of the form*

$$\boldsymbol{A} = \boldsymbol{U}_1^T \boldsymbol{U}_2^T \dots \boldsymbol{U}_L^T \boldsymbol{H} \boldsymbol{U}_L \dots \boldsymbol{U}_2 \boldsymbol{U}_1, \tag{6}$$

where each $\boldsymbol{U}_\ell \in \mathbb{O}$ *satisfies* $[\boldsymbol{U}_\ell]_{[n]\setminus\mathbb{S}_{\ell-1},[n]\setminus\mathbb{S}_{\ell-1}} = \boldsymbol{I}_{n-d_\ell}$ *for some nested sequence of sets* $\mathbb{S}_L \subseteq \cdots \subseteq \mathbb{S}_1 \subseteq \mathbb{S}_0 = [n]$ *with* $|\mathbb{S}_\ell| = d_\ell$, *and* $\boldsymbol{H} \in \mathbb{H}_n^{\mathbb{S}_L}$ *is an* $\mathbb{S}_L$*-core-diagonal matrix.*

Definition 4 *We say that a symmetric matrix* $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ *is* ***fully multiresolution factorizable*** *over* $\mathbb{O} \subset \mathbb{SO}(n)$ *with* $(d_1,\dots,d_L)$ *if it has a decomposition of the form described in Def. 3.*

A.2. Multiresolution analysis

(Kondor et al., 2014) has shown that MMF mirrors the classical theory of multiresolution analysis (MRA) on the real line (Mallat, 1989) to discrete spaces. The functional analytic view of wavelets is provided by MRA, which, similarly to Fourier analysis, is a way of filtering some function space into a sequence of subspaces

$$\cdots \subset \mathbb{V}_{-1} \subset \mathbb{V}_0 \subset \mathbb{V}_1 \subset \mathbb{V}_2 \subset \dots \tag{7}$$

However, it is best to conceptualize (7) as an iterative process of splitting each $\mathbb{V}_\ell$ into the orthogonal sum $\mathbb{V}_\ell = \mathbb{V}_{\ell+1} \oplus \mathbb{W}_{\ell+1}$ of a smoother part $\mathbb{V}_{\ell+1}$, called the *approximation space*; and a rougher part $\mathbb{W}_{\ell+1}$, called the *detail space* (see Fig. 3). Each $\mathbb{V}_\ell$ has an orthonormal basis $\Phi_\ell \triangleq \{\phi_m^\ell\}_m$ in which each ϕ is called a *father* wavelet. Each complementary space $\mathbb{W}_\ell$ is also spanned by an orthonormal basis $\Psi_\ell \triangleq \{\psi_m^\ell\}_m$ in which each ψ is called a *mother* wavelet. In MMF, each individual rotation $\boldsymbol{U}_\ell : \mathbb{V}_{\ell-1} \rightarrow \mathbb{V}_\ell \oplus \mathbb{W}_\ell$ is a sparse basis transform that expresses $\Phi_\ell \cup \Psi_\ell$ in the previous basis $\Phi_{\ell-1}$ such that:

$$\phi_m^\ell = \sum_{i=1}^{\dim(\mathbb{V}_{\ell-1})} [\boldsymbol{U}_\ell]_{m,i} \phi_i^{\ell-1}, \qquad \psi_m^\ell = \sum_{i=1}^{\dim(\mathbb{V}_{\ell-1})} [\boldsymbol{U}_\ell]_{m+\dim(\mathbb{V}_{\ell-1}),i} \phi_i^{\ell-1},$$

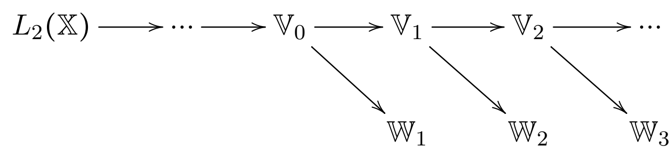

Figure 3: Multiresolution analysis splits each function space $\mathbb{V}_0, \mathbb{V}_1, \dots$ into the direct sum of a smoother part $\mathbb{V}_{\ell+1}$ and a rougher part $\mathbb{W}_{\ell+1}$.

in which Φ_0 is the standard basis, i.e. $\phi_m^0 = e_m$; and $\dim(\mathbb{V}_\ell) = d_\ell = |\mathbb{S}_\ell|$. In the $\Phi_1 \cup \Psi_1$ basis, $\boldsymbol{A}$ compresses into $\boldsymbol{A}_1 = \boldsymbol{U}_1 \boldsymbol{A} \boldsymbol{U}_1^T$. In the $\Phi_2 \cup \Psi_2 \cup \Psi_1$ basis, it becomes $\boldsymbol{A}_2 = \boldsymbol{U}_2 \boldsymbol{U}_1 \boldsymbol{A} \boldsymbol{U}_1^T \boldsymbol{U}_2^T$, and so on. Finally, in the $\Phi_L \cup \Psi_L \cup \dots \cup \Psi_1$ basis, it takes on the form $\boldsymbol{A}_L = \boldsymbol{H} = \boldsymbol{U}_L \dots \boldsymbol{U}_2 \boldsymbol{U}_1 \boldsymbol{A} \boldsymbol{U}_1^T \boldsymbol{U}_2^T \dots \boldsymbol{U}_L^T$ that consists of four distinct blocks (supposingly that we permute the rows/columns accordingly):

$$\boldsymbol{H} = \begin{pmatrix} \boldsymbol{H}_{\Phi,\Phi} & \boldsymbol{H}_{\Phi,\Psi} \\ \boldsymbol{H}_{\Psi,\Phi} & \boldsymbol{H}_{\Psi,\Psi} \end{pmatrix},$$

where $\boldsymbol{H}_{\Phi,\Phi} \in \mathbb{R}^{\dim(\mathbb{V}_L) \times \dim(\mathbb{V}_L)}$ is effectively $\boldsymbol{A}$ compressed to $\mathbb{V}_L$, $\boldsymbol{H}_{\Phi,\Psi} = \boldsymbol{H}_{\Psi,\Phi}^T = 0$ and $\boldsymbol{H}_{\Psi,\Psi}$ is diagonal. MMF approximates $\boldsymbol{A}$ in the form

$$\boldsymbol{A} \approx \sum_{i,j=1}^{d_L} h_{i,j} \phi_i^L \phi_j^{L\,T} + \sum_{\ell=1}^{L} \sum_{m=1}^{d_\ell} c_m^\ell \psi_m^\ell \psi_m^{\ell\,T},$$

where $h_{i,j}$ coefficients are the entries of the $\boldsymbol{H}_{\Phi,\Phi}$ block, and $c_m^\ell = \langle \psi_m^\ell, \boldsymbol{A} \psi_m^\ell \rangle$ wavelet frequencies are the diagonal elements of the $\boldsymbol{H}_{\Psi,\Psi}$ block.

In particular, the dictionary vectors corresponding to certain rows of $\boldsymbol{U}_1$ are interpreted as level one wavelets, the dictionary vectors corresponding to certain rows of $\boldsymbol{U}_2 \boldsymbol{U}_1$ are interpreted as level two wavelets, and so on. One thing that is immediately clear is that whereas Eq. (1) diagonalizes $\boldsymbol{A}$ in a single step, multiresolution analysis will involve a sequence of basis transforms $\boldsymbol{U}_1, \boldsymbol{U}_2, \dots, \boldsymbol{U}_L$, transforming $\boldsymbol{A}$ step by step as

$$\boldsymbol{A} \to \boldsymbol{U}_1 \boldsymbol{A} \boldsymbol{U}_1^T \to \dots \to \boldsymbol{U}_L \dots \boldsymbol{U}_1 \boldsymbol{A} \boldsymbol{U}_1^T \dots \boldsymbol{U}_L^T \triangleq \boldsymbol{H}, \tag{8}$$

so the corresponding matrix factorization must be a multilevel factorization

$$\boldsymbol{A} \approx \boldsymbol{U}_1^T \boldsymbol{U}_2^T \dots \boldsymbol{U}_\ell^T \boldsymbol{H} \boldsymbol{U}_\ell \dots \boldsymbol{U}_2 \boldsymbol{U}_1. \tag{9}$$

A.3. MMF optimization problem

Finding the best MMF factorization to a symmetric matrix $\boldsymbol{A}$ involves solving

$$\min_{\substack{\mathbb{S}_L \subseteq \dots \subseteq \mathbb{S}_1 \subseteq \mathbb{S}_0 = [n] \\ \boldsymbol{H} \in \mathbb{H}_n^{\mathbb{S}_L};\, \boldsymbol{U}_1, \dots, \boldsymbol{U}_L \in \mathbb{O}}} \| \boldsymbol{A} - \boldsymbol{U}_1^T \dots \boldsymbol{U}_L^T \boldsymbol{H} \boldsymbol{U}_L \dots \boldsymbol{U}_1 \|. \tag{10}$$

Assuming that we measure error in the Frobenius norm, (10) is equivalent to

$$\min_{\substack{\mathbb{S}_L \subseteq \dots \subseteq \mathbb{S}_1 \subseteq \mathbb{S}_0 = [n] \\ \boldsymbol{U}_1, \dots, \boldsymbol{U}_L \in \mathbb{O}}} \| \boldsymbol{U}_L \dots \boldsymbol{U}_1 \boldsymbol{A} \boldsymbol{U}_1^T \dots \boldsymbol{U}_L^T \|_{\mathrm{resi}}^2, \tag{11}$$

where $\|\cdot\|^2_{\text{resi}}$ is the squared residual norm $\|\boldsymbol{H}\|^2_{\text{resi}} = \sum_{i\neq j;\,(i,j)\notin\mathbb{S}_L\times\mathbb{S}_L}|\boldsymbol{H}_{i,j}|^2$. The optimization problem in (10) and (11) is equivalent to the following 2-level one:

$$\min_{\mathbb{S}_L\subseteq\cdots\subseteq\mathbb{S}_1\subseteq\mathbb{S}_0=[n]}\ \min_{\boldsymbol{U}_1,\ldots,\boldsymbol{U}_L\in\mathbb{O}} \|\boldsymbol{U}_L\ldots\boldsymbol{U}_1\boldsymbol{A}\boldsymbol{U}_1^T\ldots\boldsymbol{U}_L^T\|^2_{\text{resi}}. \tag{12}$$

There are two fundamental problems in solving this 2-level optimization:

- For the inner optimization, the variables (i.e. Givens rotations $\boldsymbol{U}_1,\ldots,\boldsymbol{U}_L$) must satisfy the orthogonality constraints.
- For the outer optimization, finding the optimal nested sequence of indices $\mathbb{S}_L\subseteq\cdots\subseteq\mathbb{S}_1\subseteq\mathbb{S}_0=[n]$ is a combinatorics problem, given an exponential search space.

In order to address these above problems, (Hy and Kondor, 2022) proposes a learning algorithm combining Stiefel manifold optimization and Reinforcement Learning (RL) for the inner and outer optimization, respectively. In this paper, we assume that a nested sequence of indices $\mathbb{S}_L\subseteq\cdots\subseteq\mathbb{S}_1\subseteq\mathbb{S}_0=[n]$ is given by a fast heuristics instead of computationally expensive RL. There are several heuristics to find the nested sequence, for example: clustering based on similarity between rows (Kondor et al., 2014) (Kondor et al., 2015). In the next section, we introduce the solution for the inner problem.

A.4. Stiefel manifold optimization

In order to solve the inner optimization problem of (12), we consider the following generic optimization with orthogonality constraints:

$$\min_{\boldsymbol{X}\in\mathbb{R}^{n\times p}}\mathcal{F}(\boldsymbol{X}),\quad \text{s.t.}\quad \boldsymbol{X}^T\boldsymbol{X}=\boldsymbol{I}_p, \tag{13}$$

where $\boldsymbol{I}_p$ is the identity matrix and $\mathcal{F}(\boldsymbol{X}):\mathbb{R}^{n\times p}\to\mathbb{R}$ is a differentiable function. The feasible set $\mathcal{V}_p(\mathbb{R}^n)=\{\boldsymbol{X}\in\mathbb{R}^{n\times p}:\boldsymbol{X}^T\boldsymbol{X}=\boldsymbol{I}_p\}$ is referred to as the Stiefel manifold of p orthonormal vectors in $\mathbb{R}^n$. We will view $\mathcal{V}_p(\mathbb{R}^n)$ as an embedded submanifold of $\mathbb{R}^{n\times p}$. In the case there are more than one orthogonal constraints, (13) is written as

$$\min_{\boldsymbol{X}_1\in\mathcal{V}_{p_1}(\mathbb{R}^{n_1}),\ldots,\boldsymbol{X}_q\in\mathcal{V}_{p_q}(\mathbb{R}^{n_q})}\mathcal{F}(\boldsymbol{X}_1,\ldots,\boldsymbol{X}_q) \tag{14}$$

where there are q variables with corresponding q orthogonal constraints. In the MMF optimization problem (12), suppose we are already given $\mathbb{S}_L\subseteq\cdots\subseteq\mathbb{S}_1\subseteq\mathbb{S}_0=[n]$ meaning that the indices of active rows/columns at each resolution were already determined, for simplicity. In this case, we have $q=L$ number of variables such that each variable $\boldsymbol{X}_\ell=\boldsymbol{O}_\ell\in\mathbb{R}^{k\times k}$, where $\boldsymbol{U}_\ell=\boldsymbol{I}_{n-k}\oplus_{\mathbb{I}_\ell}\boldsymbol{O}_\ell\in\mathbb{R}^{n\times n}$ in which $\mathbb{I}_\ell$ is a subset of k indices from $\mathbb{S}_\ell$, must satisfy the orthogonality constraint. The corresponding objective function is

$$\mathcal{F}(\boldsymbol{O}_1,\ldots,\boldsymbol{O}_L)=\|\boldsymbol{U}_L\ldots\boldsymbol{U}_1\boldsymbol{A}\boldsymbol{U}_1^T\ldots\boldsymbol{U}_L^T\|^2_{\text{resi}}. \tag{15}$$

Therefore, we can cast the inner problem of (12) as an optimization problem on the Stiefel manifold, and solve it by the specialized steepest gradient descent (Tagare, 2011).

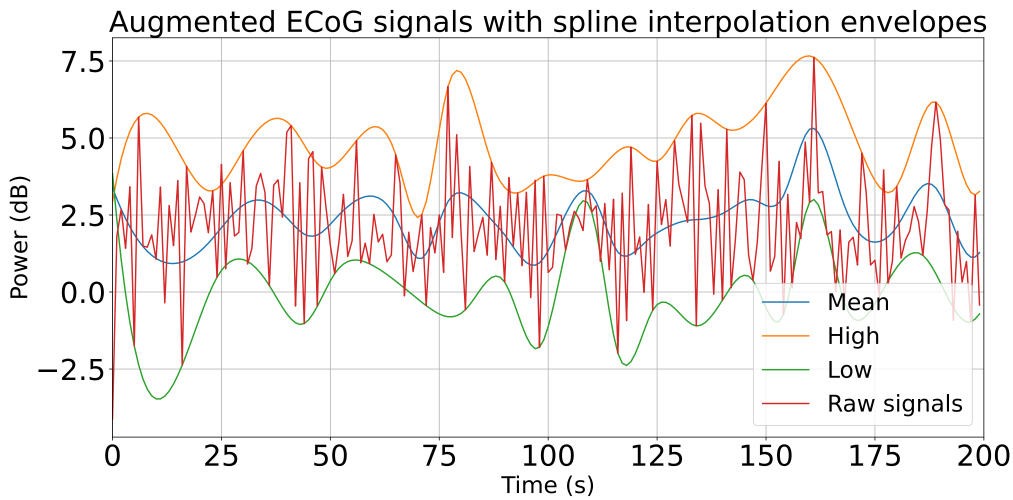

Figure 4: Augmented ECoG signals by spline interpolation envelopes.

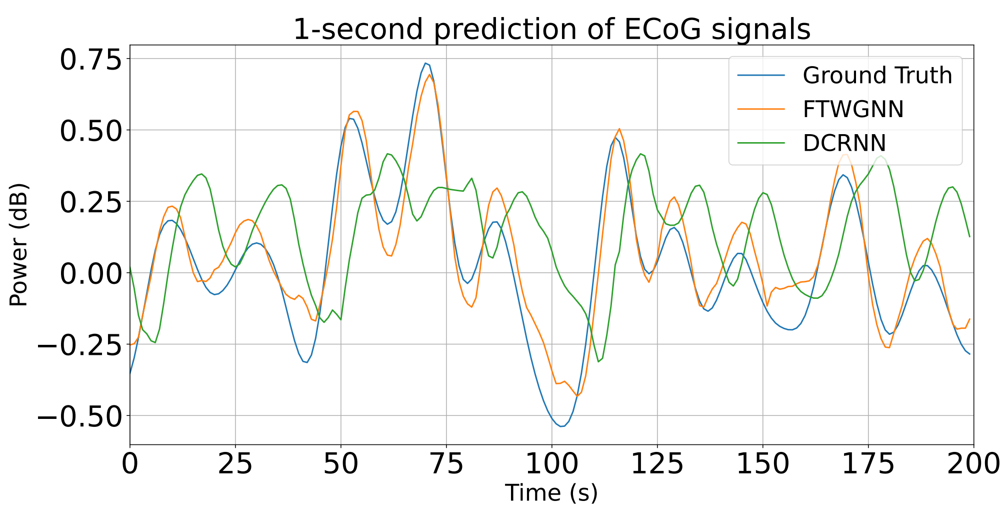

Figure 5: 1-second prediction of ECoG signals.

Appendix B. Experimental details

We compare our model with widely used time series models, including:

1. HA: Historical Average, which models traffic flow as a seasonal process and uses the weighted average of previous seasons as the prediction;

2. ARIMA$_{kal}$: Auto-Regressive Integrated Moving Average model with Kalman filter, implemented by the *statsmodel* package in Python;

3. VAR: Vector Auto-regressive model (Hamilton, 2020) with orders $(3, 0, 1)$, implemented by the Python *statsmodel* package;

4. SVR: Linear Support Vector Regression (Smola and Schölkopf, 2004) with 5 historical observations;

5. FNN: Feed forward neural network with two hidden layers, each with 256 units. The initial learning rate is 10^{-3}, and the decay rate is 10^{-1} per 20 epochs. In addition, for all hidden layers, dropout with ratio 0.5 and ℓ_2 weight decay 10^{-2} is used. The model is trained to minimize the MAE with batch size 64;

6. FC-LSTM: The encoder-decoder framework using LSTM with peephole (Sutskever et al., 2014). The encoder and decoder contain two recurrent layers, each of which consists of 256 LSTM units, with an ℓ_1 weight decay rate 2×10^{-5} and an ℓ_2 weight decay rate 5×10^{-4}. The initial learning rate is 10^{-4} and the decay rate is 10^{-1} per 20 epochs;

7. Spatio-Temporal Graph Convolutional Networks (STGCN) (Han et al., 2020) and GWaveNet (Wu et al., 2019).

8. **DCRNN** Settings of the Diffusion Convolutional Recurrent Neural Network follow its original work (Li et al., 2018).

Adjacency matrix According to DCRNN (Li et al., 2018), the traffic sensor network is expressed by an adjacency matrix which is constructed using the Gaussian kernel thresholded (Shuman et al., 2013). Specifically, for each pair of sensors v_i and v_j, the edge weight of v_i to v_j, denoted by A_{ij}, is defined as

$$A_{ij} \coloneqq \begin{cases} \exp\left(-\frac{\text{dist}(v_i, v_j)}{\sigma^2}\right), & \text{dist}(v_i, v_j) \le k \\ 0, & \text{otherwise} \end{cases}, \tag{16}$$

where $\text{dist}(v_i, v_j)$ denotes the spatial distance from v_i to v_j, σ is the standard deviation of the distances and k is the distance threshold.

Nevertheless, such a user-defined adjacency matrix requires expert knowledge, and thus may not work on other domains, *e.g.*, brain networks. In the ECoG time series forecasting case, the adjacency matrix is computed based on the popular Local Linear Embedding (LLE) (Saul and Roweis, 2003). In particular, for the matrix data $\boldsymbol{X} = [\boldsymbol{x}_1, \dots, \boldsymbol{x}_N] \in \mathbb{R}^{T \times N}$ where $\boldsymbol{x}_i$ denotes the time series data of node i for $i \in \{1, \dots, N\}$, an adjacency matrix $\boldsymbol{A}$ is identified to gather all the coefficients of the affine dependencies among $\{\boldsymbol{x}_i\}_{i=1}^N$ by solving the following optimization problem.

$$\begin{aligned} \boldsymbol{A} &\coloneqq \arg \min_{\hat{\boldsymbol{A}} \in \mathbb{R}^{N\times N}} \left\| \boldsymbol{X} - \boldsymbol{X}\hat{\boldsymbol{A}}^T \right\|_{\mathrm{F}}^2 + \lambda_A \left\| \hat{\boldsymbol{A}} \right\|_1 \\ &\quad \text{s.to} \quad \mathbf{1}_N^T \hat{\boldsymbol{A}} = \mathbf{1}_N^T, \quad \operatorname{diag}(\hat{\boldsymbol{A}}) = \mathbf{0}, \end{aligned} \tag{17}$$

where the constraint $\mathbf{1}_N^T \hat{\boldsymbol{A}} = \mathbf{1}_N^T$ realizes the affine combinations, while $\operatorname{diag}(\hat{\boldsymbol{A}}) = \mathbf{0}$ excludes the trivial solution $\hat{\boldsymbol{A}} = \boldsymbol{I}_N$. Furthermore, to promote the local smoothness of the graph, each data point x_i is assumed to be approximated by a few neighbors $\{x_{j_1}, x_{j_2}, \dots, x_{j_k}\}$, thus $\hat{A}$ is regularized by the l_1-norm loss $\left\| \hat{\boldsymbol{A}} \right\|_1$ to be sparse. Task (17) is a composite convex minimization problem with affine constraints, which can therefore be solved by (Slavakis and Yamada, 2018).

Construction of wavelet bases For traffic data sets, 100 mother wavelets are extracted, *i.e.*, $L = 100$, while for the AJILE12 dataset, $L = 10$ was used.