Fast Temporal Wavelet Graph Neural Networks

Duc Thien Nguyen*[1], Manh Duc Tuan Nguyen *[2] Truong Son Hy*†[3], and Risi Kondor[4]

Tokyo Institute of Technology[1], Toyo University[2], Indiana State University[3], University of Chicago[4]

*: Equal contribution. †: Correspondence to TruongSon.Hy@indstate.edu

Motivation & Main Results

- Spatiotemporal forecast is crucial but challenging (medical, traffic, weather, etc.)
- Bottlenecks:
  - High-dimensional due to large network and long time series
  - Strong and dynamic hidden spatio-temporal dependencies
  - Possibly noisy and corrupted signals
- Graph neural networks (GNNs) have seen success in modeling spatiotemporal signals
- For dense graphs, adjacency matrix may waste resource and fail to capture locality
- We propose:
  - **Memory & time efficient** end-to-end model for spatiotemporal forecast.
  - **Multiresolution analysis & wavelet theory** to represent graph structure.
  - **Traffic & brain signals prediction** with competitive performance

Prior Arts

Traditional methods

- Historical Average
- ARIMA with Kalman filter
- Vector Auto-regressive VAR
- Linear Support Vector Regression SVR

Deep learning

- Feed-forward neural network FNN
- Fully-connected LSTM
- Spatio-Temporal Graph Convolutional Networks (STGCN)
- GWaveNet
- Diffusion Convolutional RNN (DCRNN)

Wavelet Neural Networks

Based on Graph Fourier Transform (GFT) (Bruna et al., 2014), each convolution layer $k = 1, .., K$ transforms an input vector $\boldsymbol{f}^{(k-1)}$ into an output $\boldsymbol{f}^{(k)}$ as

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

where $\boldsymbol{W} = [\overline{\phi}, \overline{\psi}]$ is our wavelet basis matrix of a total of N wavelets:

- L mother wavelets $\overline{\psi} = \{\psi^1, .., \psi^L\}$,
- $N - L$ father wavelets $\overline{\phi} = \{\phi^L_m = \boldsymbol{H}_{m,:}\}_{m \in \mathbb{S}_L}$;

and

- $\boldsymbol{g}^{(k)}_{i,j}$ is a parameter/filter,
- σ is a non-linear activation function,
- **Sparse wavelet bases ⟶ Efficient sparse wavelet transform**

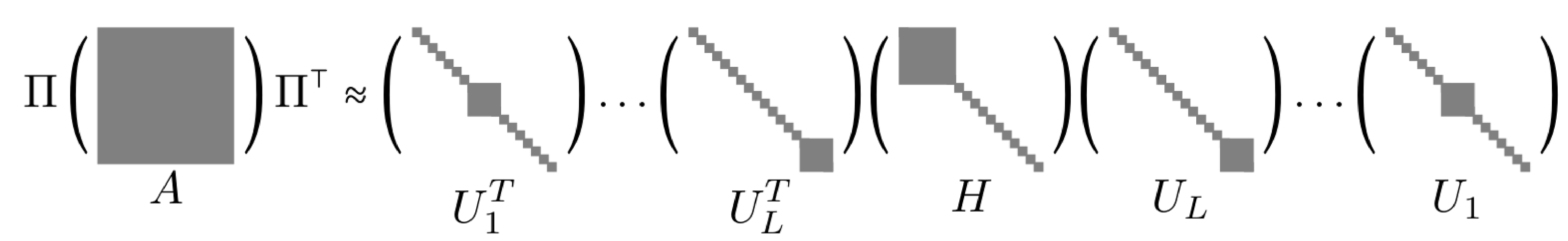
Multiresolution Matrix Factorization

MMF of a symmetric adjacency matrix $\boldsymbol{L} \in \mathbb{R}^{n \times n}$ (Kondor et al., 2014) is:

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

where:

- Each $\boldsymbol{U}_\ell$ is an orthogonal matrix that is a k-point rotation (small k),
- 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 meaning that is diagonal with a an additional small $\mathbb{S}_L \times \mathbb{S}_L$ dimensional "core".



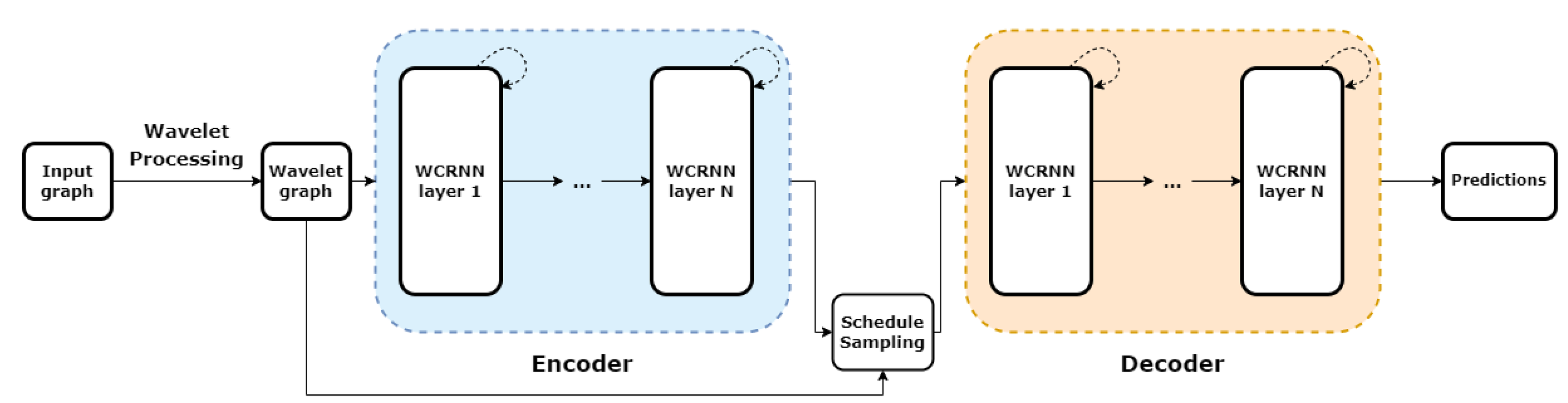
Not based on the low-rank assumption

Fast Temporal Wavelet Graph Neural Networks

1. **Spatial Dependency Model:** Captures spatial dynamics using a diffusion process on an undirected graph $G = (\boldsymbol{X}, \boldsymbol{A})$. The diffusion equation is given by

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

2. **Temporal Dependency Model:** Utilizes the *Diffusion Convolutional Gated Recurrent Unit* (DCGRU) to model temporal dependencies. Key equations include reset gate $\boldsymbol{r}^{(t)}$, update gate $\boldsymbol{u}^{(t)}$, cell state $\boldsymbol{C}^{(t)}$, and hidden state $\boldsymbol{H}^{(t)}$.
3. **FTWGNN:** Differs from DCRNN by using a sparse wavelet basis matrix $\boldsymbol{W}$ extracted via MMF and employing fast *wavelet convolution* in place of diffusion convolution. This reduces computational time and memory usage.



Software

Our PyTorch implementation is publicly available at:

https://github.com/HySonLab/TWGNN

We utilize MMF implementation at:

https://github.com/risilab/Learnable_MMF

Experiments

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%**

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%**

FTWGNN outperforms others by roughly 10%.

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

FTWGNN's training time is faster than DCRNN's by 5 times on average.

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

FTWGNN provides a remarkable compression of wavelet bases compared to Fourier bases.

Reference

[1] Kondor et al., Multiresolution Matrix Factorization, ICML 2014

[2] Truong Son Hy and Risi Kondor, Multiresolution Matrix Factorization and Wavelet Networks on Graphs, PMLR 196:172-182