Spatio-Temporal BLIS-Net: Wavelets for Space-Time Graph Signals

Chaz Okada
Department of Electrical & Computer Engineering
Yale University
New Haven, CT 06511
`chaz.okada@yale.edu`

Abstract

Spatio-temporal graphs can be useful to model space-time data and relationships in many different applications, including traffic analysis, pandemic forecasting, and motion tracking. There are many different architectures that try to capture this space-time relationship, including RNNs, attention mechanisms, and convolution. This project expands the Bi-Lipschitz Scattering Net (BLIS-Net) architecture to the temporal domain, inpired by existing methods described in the Spatio-Temporal Graph Scattering Transform. Separable space/time wavelet frames are constructed and diffused across the graph to extract spectral features. These features are aggregated and passed through an MLP for a regression task. This architecture is evaluated on two traffic datasets, METR-LA and PEMS-BAY, and it shows significant improvement over existing methods. However, after further study, it is shown that the superior performance is achieved by the MLP, while the Spatio-Temporal BLIS (STBLIS) module has an insignificant impact on the overall model performance.

1 Introduction

Spatio-temporal graphs are graphs that have both spatial and time-series elements in their structure. There are many natural applications for these types of graph structures, including solar PV prediction [1], pandemic forecasting [2], motion tracking [3] [4], and traffic prediction [5] [6] [7] [8] [9] [10] [11]. Currently, there are no standard benchmarks for spatio-temporal graph tasks [9], but traffic forecasting presents a unique challenge for regression across all graph nodes for multiple timesteps. This work proposes an extension of the Bi-Lipshitz Scattering Net (BLIS-Net) architecture [5], using techniques from the Spatio-Temporal Graph Scattering Transform (ST-GST) presented in [3]. Separable space/time wavelet frames are constructed and diffused across the graph to extract spectral features, and these features are aggregated and passed through an MLP for regression on the METR-LA and PEMS-BAY traffic datasets. The Spatio-Temporal BLIS-Net (STBLIS-Net) architecture significantly outperforms existing methods, but it is shown that the improvement is due to the MLP regression layers, rather than the STBLIS module. Despite the inconclusive results for the STBLIS module evaluation, this work highlights the fact that simple models can be much more performant and efficient than complex architectures on certain tasks.

2 Related Work

Given the prevalence of spatio-temporal graphs, there have been many proposed approaches to graph learning across space and time dependencies [9] [12]. Many of these approaches separate the temporal layer from the GNN, which allows for a variety of mix-and-match techniques to handle both the spatial and temporal dependencies. The time-based methods can be separated into three major categories: RNNs (LSTM/GRU), attention, and convolutional mechanisms. Some examples of the RNN category include Graph WaveNet [6], Diffusion Convolutional RNN (DCRNN) [7], and A3T-GCN [10]. Some examples of the attention category include Graph Multiattention Network (GMAN) [8], Attention-based Spatio-temporal GAT (ASTGAT) [11], and Temporal-Spatial Multi-Window GAT (TSM-GAT) [1]. Finally, some examples of convolutional approaches include Graph Wavelet NN (GWNN) [13], Dynamic Wavelets for Global Interactions (DEFT) [14], and ST-GST [3]. For the sake of illustration, this categorization is simplified, as many of these networks incorporate multiple mechanisms, such as the DCRNN, which combines spectral convolution with a GRU. Due to the interconnected structure of spatio-temporal graphs, both the spatial and temporal learning architecutre choices impact model performance for various tasks.

Spectral convolution techniques for graph signal processing have recently shown promising results. In particular, the BLIS-Net architecture proposed in [5] has demonstrated superior performance in the spatial domain. BLIS-Net uses wavelets to transform the spatial signals on the graph into the frequency domain which allows for signal localization, which is not possible with the Fourier Transform. However, as-proposed, BLIS-Net is only structured for spatial graphs. Thus, the ST-GST framework proposed in [3] will be considered to augment the BLIS-Net for temporal graph data. ST-GST is actually not a deep learning approach, unlike the aforementioned architectures. Instead, it shows that pre-designed wavelets can be used to process spatio-temporal graph signals without using neural networks. Similar to BLIS-Net, this is a pre-processing technique that encodes graph information in the spectral domain, and due to the fact that ST-GST does not require backpropogation (a random forest is used after the ST-GST process), it has performed much better in situations with limited data. However, [3] does not include mention of several key properties that are provided in BLIS-Net, including the bi-Lipschitz property and energy conservation. Thus, the STBLIS-Net architecture seeks to combine these two works.

3 Method

This STBLIS-Net architecture will be an extension of the BLIS-Net architecture for the spatio-temporal domain. The overall BLIS-Net architecture consists of four layers: BLIS module, moment aggregation, embedding layer, and classification layer [5]. However, in the BLIS module layer, only spatial signals were considered. As discussed in Section 2, there are many options that have shown viable success for processing time-based signals on graph data. The proposed STBLIS-Net architecture veers off of the traditional deep learning route by using a similar architecture to [3], encoding time-based relations in the graph, rather than LSTM, GRU, or attention-based architectures. BLIS-Net and ST-GST are a natural pair, as they both utilize wavelets for spectral graph signal analysis. Specifically, the BLIS module will be augmented with separable graph waveletes for the time domain, as defined in [3]. The time-domain wavelets will be constructed using the BLIS module, retaining key properties including bi-Lipschitz and energy conservation. Similar to ST-GST, a line graph (and associated adjacency matrix) is constructed for the time-domain, which allows for the computation of the BLIS-style wavelet scattering.

More formally, for a spatio-temporal graph of N nodes and T timesteps, we can define a graph $x \in \mathbb{R}^{NxT}$. The wavelet frame $\mathcal{W}_J = \{\Psi_j\}_{j=0}^{J} \cup \{\Phi_J\}$, as well as the wavelets $\mathcal{W}_J^{(1)}$ and $\mathcal{W}_J^{(2)}$ are defined identically to [5]. If the spatial wavelet frame is defined as $\mathcal{H} = \{H_k\}_{k=0}^{K+1}$ and the temporal wavelet frame is defined as $\mathcal{G} = \{G_j\}_{j=0}^{J+1}$, the STBLIS module is be defined in layer $m+1$ as

$$B_{ST}^{m+1}(x) = \sigma_{i_m}(H_{k_m} B_{ST}^{m} G_{j_m}^{\top}) \tag{1}$$

where $B_{ST}^0 = x$, $i_m \in \{1, 2\}$. As shown in [3], this is equivalent to using a Kronecker product between H and G:

$$(\mathcal{H} \otimes \mathcal{G}) B_{ST}^{m} = H_{k_m} B_{ST}^{m} G_{j_m}^{\top} \tag{2}$$

After applying both the space and time wavelet frames, two activation functions are used to induce injectivity, identical to the BLIS-Net architecture:

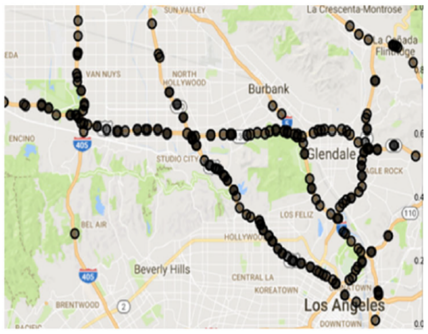
$$\sigma_1(x) := ReLU(x), \sigma_2(x) := ReLU(-x) \tag{3}$$

Thus, the STBLIS-Net module has output $b \in \mathbb{R}^{2 \times (K+2)N \times (J+2)T}$. Due to the added time domain scattering, the computational complexity increases exponentially as the number of layers increases.

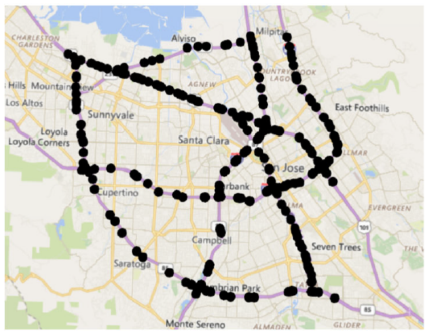
Therefore, the STBLIS-Net only uses a single layer of the STBLIS module before the moment aggregation module. The STBLIS-Net moment aggregation module uses mean aggregation across the last layer of the STBLIS module output such that the output shape follows $\mathbb{R}^{NxT}$. A residual sum of the original input x is added to the moment aggregation, as the mean aggregation compresses the input information considerably. The output is then flattened to an embedding layer, consisting of single-layer MLP. In the original BLIS-Net, the traffic task was formulated as classification, but this work uses a regression layer after the embedding layer, outputting all T timesteps in a single shot for all N sensors. This is different than an autoregressive or RNN architecture, where the model predicts a single output and uses the previous output to predict the next output. To accomplish this, the regression layer has $N \times T$ neurons at the output.

4 Experiments

Despite the lack of standard spatio-temporal graph benchmarks [9], traffic prediction/forecasting is a common task in the current literature. Several studies, including [5] [6] [7] [8] and [11] were evaluated on PEMS-BAY (or similar CalTrans PEMS datasets), and [6] and [7] were evaluated on the METR-LA dataset. PEMS-BAY is a CalTrans dataset that consists of 325 sensors with time-series data of speed readings in 5-minute intervals. METR-LA consists of 207 sensors in LA County with time-series data of speed readings in 5-minute intervals [6] and [7]. These datasets are not graph datasets out-of-the-box, so they are processed into graph datasets as described in [6] and [7] to maintian continuity of evaluation across various architectures. To compute the graph, the pairwise road network distances between sensors are computed, and the adjacency matrix is built using a thresholded Gaussian kernel [6] [7]. To process the time-series data, the speed readings are Z-score normalized for each sensor, and samples of 12 timesteps are extracted into train/test/validation sets (0.7/0.2/0.1). Thus, the model will see the previous 12 timestamps (60 minutes) over all sensors and need to forecast the normalized speed of each sensor over the next 12 timesteps (60 minutes). For consistency, scripts and data from [6] and [7] were used during data preprocessing with minimal alterations. Figure 1 shows the physical locations of the sensors on a map.



(a) METR-LA Sensors



(b) PEMS-BAY Sensors

Figure 1: Location of sensors in METR-LA and PEMS-BAY datasets [7].

Table 1 shows a summary of both datasets. Figure 2 shows an example of the time-series data for a single PEMS-BAY sensor (METR-LA time-series data is similar).

Data	# Nodes	# Edges	Total Time Steps
METR-LA	207	1722	34,272
PEMS-BAY	325	2694	52,116

Table 1: METR-LA and PEMS-BAY summary statistics.

Before training the model embedding/regression MLP layers, the STBLIS module is precomputed using $K = 4$ and $J = 2$ for the space and time wavelet frames, respectively. Wavelet type $\mathcal{W}^{(2)}$ [5] is used for model evaluation. After the precomputation, the MLP is trained with an embedding

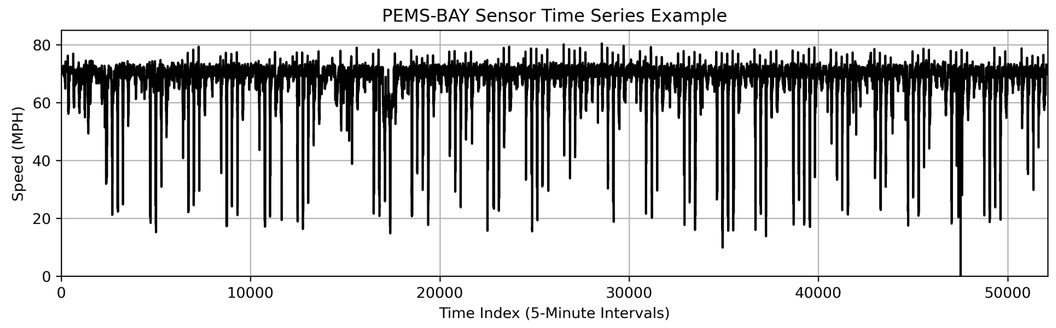


Figure 2: Sample time-series plot of a single PEMS-BAY sensor

Data	Models	15 min			30 min			60 min		
		MAE	RMSE	MAPE[†]	MAE	RMSE	MAPE[†]	MAE	RMSE	MAPE[†]
METR-LA	ARIMA [7]	3.99	8.21	9.60	5.15	10.45	12.70	6.90	13.23	17.40
	FC-LSTM [7]	3.44	6.30	9.60	3.77	7.23	10.90	4.37	8.69	13.20
	WaveNet [15]	2.99	5.89	8.04	3.59	7.28	10.25	4.45	8.93	13.62
	DCRNN[‡] [7]	2.67	5.17	6.84	3.08	6.30	8.38	3.56	7.52	10.30
	STGCN [16]	2.88	5.74	7.62	3.47	7.24	9.57	4.59	9.40	12.70
	Graph WaveNet [6]	2.69	5.15	6.90	3.07	6.22	8.37	3.53	7.37	10.01
	KDE Random Sample	0.98 ± .00	1.50 ± .00	4.11 ± .12	0.98 ± .00	1.50 ± .00	4.11 ± .12	0.98 ± .00	1.50 ± .00	4.11 ± .12
	STBLIS-Net	**0.29 ± .00**	**0.62 ± .01**	**1.65 ± .01**	**0.32 ± .00**	**0.71 ± .00**	**1.84 ± .02**	**0.38 ± .00**	**0.81 ± .00**	**1.94 ± .02**
PEMS-BAY	ARIMA [7]	1.62	3.30	3.50	2.33	4.76	5.40	3.38	6.50	8.30
	FC-LSTM [7]	2.05	4.19	4.80	2.20	4.55	5.20	2.37	4.96	5.70
	WaveNet [15]	1.39	3.01	2.91	1.83	4.21	4.16	2.35	5.43	5.87
	DCRNN[‡] [7]	1.31	2.76	2.74	1.66	3.78	3.76	1.98	4.62	4.74
	STGCN [16]	1.36	2.96	2.90	1.81	4.27	4.17	2.49	5.69	5.79
	Graph WaveNet [6]	1.30	2.74	2.73	1.63	3.70	3.67	1.95	4.52	4.63
	KDE Random Sample	0.87 ± .00	1.41 ± .00	5.86 ± .12	0.87 ± .00	1.41 ± .00	5.86 ± .12	0.87 ± .00	1.41 ± .00	5.86 ± .12
	STBLIS-Net	**0.30 ± .00**	**0.60 ± .00**	**2.13 ± .06**	**0.30 ± .00**	**0.62 ± .00**	**2.17 ± .07**	**0.32 ± .00**	**0.65 ± .00**	**2.23 ± .06**

Table 2: Comparison of STBLIS-Net with other baselines for 15, 30, and 60 minute forecasts. STBLIS metrics are calculated across three trials. Baseline results (except KDE random sample) are referenced from [6].
[†]MAPE is calculated according to [7] for consistency, not as a true percentage.
[‡]DCRNN values are updated to reflect bug fix after original publication.

dimension of $(N \times T)/4$. Batch normalization and a dropout of $p = 0.5$ is included at the embedding layer. A batch size of 64 is used, and the model is trained over 50 epochs using early stopping with a patience value of 10 and learning rate of 0.001. MAE is the loss function used during training. The model is trained using an AMD Ryzen Threadripper PRO 5975WX with four Nvidia RTX A6000 GPUs using PyTorch DataParallel. For consistency with [6] and [7], mean absolute error (MAE), root mean squared error (RMSE), and mean absolute percentage error (MAPE) are calculated and averaged over three trials, shown in Table 2. These metrics were calculated exactly as described in [7].

Table 2 shows that the STBLIS-Net significantly outperforms all of the other baseline models. To truly understand the model performance, a kernel density estimator (KDE) is trained on the training dataset, before being randomly sampled to generate data with a similar distribution to the real data. These random samples are evaluated against the testing dataset, and the results are shown in Table 2. These results show that the STBLIS-Net performs better than random samples, while the other baseline models actually perform worse than the random KDE samples (with exception of MAPE for some evaluation points). The baseline models show a trend of increasing error as the forecasting target is farther in the future. While this is also seen in the STBLIS-Net results, the magnitude of error is much smaller as prediction targets are farther away. Figure 3 shows that the model can accurately predict drastic changes in the future, despite the fact that the input series is stable. Thus, the model is successfully learning from the graph structure of the data.

Additionally, an MLP baseline is trained with the same hyperparameters and setup as described with the STBLIS-Net. The only difference is that the MLP uses the normalized raw data, not the precomputed data. After training and evaluation, the MLP demonstrated results that are not statistically different from the STBLIS-Net results shown in Figure 2. This proves that the STBLIS module precomputation is not the driving factor behind the improvement, but rather the MLP architecture is capturing the key features of the graph (for these particular datasets and tasks). In general, the training of the MLP converges within 50 epochs, and the early stopping mechanism helps to prevent overfitting. Furthermore, the STBLIS-Net/MLP are not sensitive to hyperparameter changes. It is observed, however, that the STBLIS module will oversmooth when the frames (K and J hyperparameters) are larger. This is caused by the mean agreggation, as more low-frequency features are incorporated into the aggregation when the frame size increases. Lastly, there are two downsides to the STBLIS-Net architecture. First, the computational cost grows exponentially as more layers are added, due to the Kronecker product with the spatial and temporal wavelet frames. Thus, only one layer of the STBLIS module is utilized in this architecture. Second, the model is not autoregressive, so it has a fixed output size, deccreasing flexibility during model inference. Overall, the STBLIS module performance itself is shown to be inconclusive for these datasets and tasks, but significant results are achieved by the MLP for traffic forecasting on the METR-LA and PEMS-BAY datasets.

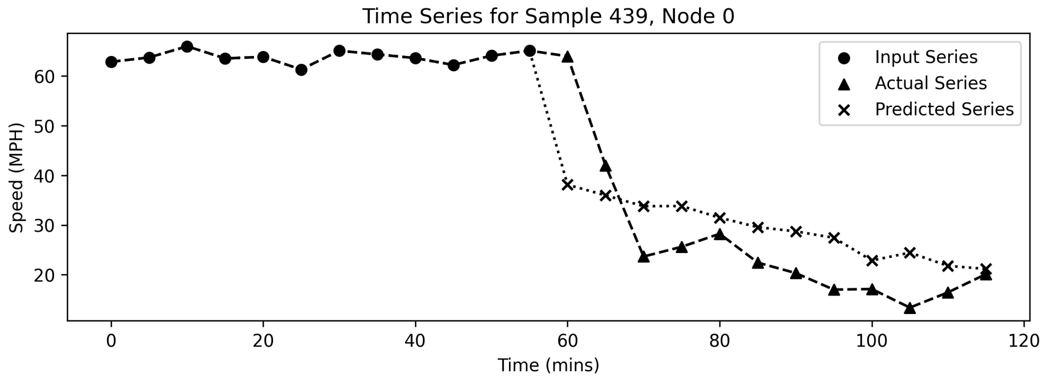


Figure 3: Sample predicted output of STBLIS-Net from the METR-LA dataset.

5 Conclusion

This work evaluates an extension of the BLIS-Net [5] architecture to the temporal domain, utilizing the spatio-temporal graph scattering transform described in ST-GST [3]. Two traffic datasets, METR-LA and PEMS-BAY, are evaluated for traffic forecasting, and the results are compared with the baselines presented in [6] and [7]. The Spatio-Temporal BLIS-Net (STBLIS-Net) architecture showed a significant performance increase over existing baselines. This study also finds that the current published results for the baselines perform worse than random samples from a kernel density estimate of the training dataset. After further evaluation of the STBLIS-Net architecture, it is found that the STBLIS module is insignificant to the superior results achieved in this work. The simple MLP regression layers are shown to provide the same results on the raw normalized training dataset. For future work, other spatio-temporal datasets/tasks may be more appropriate to evaluate the STBLIS module itself. These results support the idea that simple architectures may be the best for certain tasks, even when the tasks can be complex, such as traffic forecasting on a graph dataset. The associated code to reproduce the results in this work is available at `https://github.com/chazokada/STBLIS-NET`.

References

[1] Jelena Simeunović, Baptiste Schubnel, Pierre-Jean Alet, Rafael E. Carrillo, and Pascal Frossard. Interpretable temporal-spatial graph attention network for multi-site PV power forecasting. *Applied Energy*, 327:120127, December 2022. ISSN 03062619. doi: 10.1016/j.apenergy.2022.120127. URL `https://linkinghub.elsevier.com/retrieve/pii/S0306261922013848`.

[2] George Panagopoulos, Giannis Nikolentzos, and Michalis Vazirgiannis. Transfer Graph Neural Networks for Pandemic Forecasting, April 2021. URL `http://arxiv.org/abs/2009.08388`. arXiv:2009.08388 [cs, stat].

[3] Chao Pan, Siheng Chen, and Antonio Ortega. Spatio-Temporal Graph Scattering Transform, February 2021. URL `http://arxiv.org/abs/2012.03363`. arXiv:2012.03363 [cs, eess].

[4] Maosen Li, Siheng Chen, Yangheng Zhao, Ya Zhang, Yanfeng Wang, and Qi Tian. Multiscale Spatio-Temporal Graph Neural Networks for 3D Skeleton-Based Motion Prediction. *IEEE Transactions on Image Processing*, 30:7760–7775, 2021. ISSN 1057-7149, 1941-0042. doi: 10.1109/TIP.2021.3108708. URL `http://arxiv.org/abs/2108.11244`. arXiv:2108.11244 [cs].

[5] Charles Xu, Laney Goldman, Valentina Guo, Benjamin Hollander-Bodie, Maedee Trank-Greene, Ian Adelstein, Edward De Brouwer, Rex Ying, Smita Krishnaswamy, and Michael Perlmutter. BLIS-Net: Classifying and Analyzing Signals on Graphs, October 2023. URL `http://arxiv.org/abs/2310.17579`. arXiv:2310.17579 [cs, eess].

[6] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph WaveNet for Deep Spatial-Temporal Graph Modeling, May 2019. URL `http://arxiv.org/abs/1906.00121`. arXiv:1906.00121 [cs, stat].

[7] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting, February 2018. URL `http://arxiv.org/abs/1707.01926`. arXiv:1707.01926 [cs, stat].

[8] Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. GMAN: A Graph Multi-Attention Network for Traffic Prediction, November 2019. URL `http://arxiv.org/abs/1911.08415`. arXiv:1911.08415 [cs, eess].

[9] Zahraa Al Sahili. Spatio-Temporal Graph Neural Networks: A Survey.

[10] Jiawei Zhu, Yujiao Song, Ling Zhao, and Haifeng Li. A3T-GCN: Attention Temporal Graph Convolutional Network for Traffic Forecasting, June 2020. URL `http://arxiv.org/abs/2006.11583`. arXiv:2006.11583 [cs, stat].

[11] Yi Wang, Changfeng Jing, Shishuo Xu, and Tao Guo. Attention based spatiotemporal graph attention networks for traffic flow forecasting. *Information Sciences*, 607:869–883, August 2022. ISSN 00200255. doi: 10.1016/j.ins.2022.05.127. URL `https://linkinghub.elsevier.com/retrieve/pii/S0020025522005679`.

[12] Benedek Rozemberczki, Paul Scherer, Yixuan He, George Panagopoulos, Alexander Riedel, Maria Astefanoaei, Oliver Kiss, Ferenc Beres, Guzmán López, Nicolas Collignon, and Rik Sarkar. PyTorch Geometric Temporal: Spatiotemporal Signal Processing with Neural Machine Learning Models, June 2021. URL `http://arxiv.org/abs/2104.07788`. arXiv:2104.07788 [cs].

[13] Bingbing Xu, Huawei Shen, Qi Cao, Yunqi Qiu, and Xueqi Cheng. Graph Wavelet Neural Network, April 2019. URL `http://arxiv.org/abs/1904.07785`. arXiv:1904.07785 [cs, stat].

[14] Anson Bastos, Abhishek Nadgeri, Kuldeep Singh, Toyotaro Suzumura, and Manish Singh. Learnable Spectral Wavelets on Dynamic Graphs to Capture Global Interactions. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(6):6779–6787, June 2023. ISSN 2374-3468, 2159-5399. doi: 10.1609/aaai.v37i6.25831. URL `https://ojs.aaai.org/index.php/AAAI/article/view/25831`.

[15] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. WaveNet: A Generative Model for Raw Audio, September 2016. URL `http://arxiv.org/abs/1609.03499`. arXiv:1609.03499 [cs].

[16] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pages 3634–3640, July 2018. doi: 10.24963/ijcai.2018/505. URL `http://arxiv.org/abs/1709.04875`. arXiv:1709.04875 [cs].