
Spatial-Temporal Data Modeling with Graph Neural Networks

*Thesis submitted in fulfilment of the requirements
for the degree of*

Doctor of Philosophy

by

Zonghan Wu

under the supervision of

Guodong Long
Jing Jiang
Shirui Pan

to

University of Technology Sydney
Faculty of Engineering and Information Technology

August 2021

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, *Zonghan Wu* declare that this thesis, submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the *Australian Artificial Intelligence Institute, Faculty of Engineering and Information Technology* at the University of Technology Sydney, Australia. This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis. This document has not been submitted for qualifications at any other academic institution. This research is supported by the Australian Government Research Training Program.

Production Note:

SIGNATURE: Signature removed prior to publication.

DATE: 19th August, 2021

PLACE: Sydney, Australia

ABSTRACT

Spatial-temporal graph modeling is an important task to analyze the spatial relations and temporal trends of components in a system. It aims to model the dynamic node-level inputs by assuming inter-dependency between connected nodes. A basic assumption behind spatial-temporal graph modeling is that a node's future information is conditioned on its historical information as well as its neighbors' historical information. Therefore how to capture spatial and temporal dependencies simultaneously becomes a primary challenge. Current studies on spatial-temporal graph modeling face four major shortcomings: 1) Most graph neural networks only focus on the low frequency band of graph signals; 2) Current studies assume the graph structure of data reflects the genuine dependency relationships among nodes; 3) Existing studies on spatial-temporal graph neural networks are not applicable to pure multivariate time series data due to the absence of a predefined graph and lack of a general framework; 4) Existing approaches either model spatial-temporal dependencies locally or model spatial correlations and temporal correlations separately. The aim of this thesis is to study spatial-temporal data from the perspective of deep learning on graphs. I have studied the research objective in deep depth with four research questions: (1) How to coordinate the low, middle, and high frequency band of graph signals in graph convolution networks. (2) How to model spatial-temporal graph data effectively and efficiently; (3) How to handle spatial dependencies when a graph is totally missing, incomplete or inaccurate in spatial-temporal graph modeling; (4) In contrast to traditional spatial-temporal graph neural networks that handle spatial dependencies and temporal dependencies in separate, how to unify space and time as a whole in message passing. To address the aforementioned four research problems, I proposed four algorithms or models that can achieve satisfactory results. Specifically, I proposed an Automatic Graph Convolutional Network to learn graph frequency bands for graph convolution filters automatically; I introduced an efficient and effective framework that integrates diffusion graph convolution and dilated temporal convolution to capture spatial-temporal dependencies simultaneously. I developed a novel joint-learning algorithm that can capture spatial-temporal dependencies and learn latent graph structures at the same time; I designed a unified graph neural network that captures the inner spatial-temporal dependencies without compromising space-time integrity. To validate the proposed methods, I have conducted experiments on real-world datasets with a range of tasks including node classification, graph classification, and spatial-temporal graph forecasting. Experimental results demonstrate the effectiveness of the proposed methods.

ACKNOWLEDGMENTS

I would like to express my deepest appreciation to my supervisors Dr. Shirui Pan, Dr. Guodong Long, Dr. Jing Jiang, and Prof. Chengqi Zhang for being my knowledgeable academic teachers and firm life supporters during my Ph.D study. Their unparalleled knowledge, practical suggestions, and constructive criticism built a solid foundation for my academic career.

A very special gratitude goes to Dr Huan Huo, who highly stimulated my motivation to go through higher research studies. Without Dr. Huo's encouragement and support, I could not stand where I am today.

I am also grateful to the University of Technology Sydney, who has provided me full research scholarships. Thanks to all staff members working behind the graduate research school. Their smooth and hard work provides me a comfortable working environment.

Finally, I would like to extend my deep gratitude to my family: my father Mr. Runwen Huang who always protect my mother and me when we are in bad situations, my mother Mrs. Jiangjing Wu who provides me endless loves in my life, and my wife Mrs. Yelu Yu who gives me full confidence to take difficult challenges.

LIST OF PUBLICATIONS

Journal Papers

- J-1. Wu Z., Pan S., Chen F., Long G., Zhang C. and Philip S.Y., 2020. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*. [CORE A*, Google Citations: 1,969]
- J-2. Wu Z., Pan S., Long G., Jiang J. and Zhang C., 2021. Beyond low-pass filtering: graph convolutional networks with automatic filtering. Submitted to *IEEE Transactions on Knowledge and Data Engineering*. [CORE A*, Under Review]
- J-3. Wu Z., Zheng D., Pan S., Gan G., Long G. and Karypis G., 2021. TraverseNet: Unifying Space and Time in Message Passing. Submitted to *IEEE Transactions on Neural Networks and Learning Systems*. [CORE A*, Under Review]

Conference Papers

- C-1. Wu Z., Pan S., Long G., Jiang J. and Zhang C., 2019. Graph wavenet for deep spatial-temporal graph modeling. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence* (pp. 1907-1913). [CORE A*, Google Citations: 190]
- C-2. Wu Z., Pan S., Long G., Jiang J., Chang X. and Zhang C., 2020, August. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 753-763). [CORE A*, Google Citations: 53]

TABLE OF CONTENTS

List of Publications	vii
List of Figures	xiii
List of Tables	xv
1 Introduction	1
1.1 Background	1
1.2 Research Objectives	4
1.3 Content and Organization	6
2 Literature Review	7
2.1 Background	8
2.2 Definition	10
2.3 Categorization and Frameworks	11
2.3.1 Taxonomy of Graph Neural Networks (GNNs)	11
2.3.2 Frameworks	14
2.4 Recurrent Graph Neural Networks	15
2.5 Convolutional Graph Neural Networks	18
2.5.1 Spectral-based ConvGNNs	18
2.5.2 Spatial-based ConvGNNs	21
2.5.3 Graph Pooling Modules	28
2.5.4 Discussion of Theoretical Aspects	30
2.6 Graph autoencoders	32
2.6.1 Network Embedding	32
2.6.2 Graph Generation	35
2.7 Spatial-temporal Graph Neural Networks	37
2.8 Applications	39

TABLE OF CONTENTS

2.8.1	Data Sets	39
2.8.2	Evaluation & Open-source Implementations	40
2.8.3	Practical Applications	41
3	Beyond Low-pass Filtering: Graph Convolutional Networks with Automatic Filtering	45
3.1	Motivation	45
3.2	Related Work to AutoGCN	48
3.3	Problem Formulation	49
3.4	Spectral-Rooted Spatial Graph Convolution	50
3.5	Automatic Graph Convolution	51
3.6	Experiments	56
3.6.1	Node Classification	56
3.6.2	Graph Prediction	58
3.6.3	Ablation Study	62
3.6.4	Limitation	62
3.6.5	Analysis of Model Depth.	62
3.7	Summary	63
4	Graph WaveNet for Deep Spatial-Temporal Graph Modeling	65
4.1	Motivation	65
4.2	Related Works to Graph WaveNet	67
4.3	Methodology	68
4.3.1	Problem Definition	68
4.3.2	Graph Convolution Layer	68
4.3.3	Temporal Convolution Layer	70
4.3.4	Framework of Graph WaveNet	71
4.4	Experiments	72
4.4.1	Baselines	73
4.4.2	Experimental Setups	74
4.4.3	Experimental Results	74
4.5	Summary	77
5	Connecting the Dots: Multivariate Time Series Forecasting with Graph Neural Networks	79
5.1	Motivation	79

5.2	Backgrounds	82
5.2.1	Multivariate Time Series Forecasting	82
5.2.2	Graph Neural Networks	83
5.3	Problem Formulation	83
5.4	Framework of MTGNN	84
5.4.1	Model Architecture	84
5.4.2	Graph Learning Layer	86
5.4.3	Graph Convolution Module	87
5.4.4	Temporal Convolution Module	89
5.4.5	Skip Connection Layer & Output Module	91
5.4.6	Proposed Learning Algorithm	91
5.5	Experimental Studies	92
5.5.1	Experimental Setting	93
5.5.2	Baseline Methods for Comparison	95
5.5.3	Main Results	96
5.5.4	Parameter Study	98
5.5.5	Ablation Study	99
5.5.6	Study of the Graph Learning Layer	101
5.5.7	Complexity Analysis	102
5.6	Summary	104
6	TraverseNet: Unifying Space and Time in Message Passing	105
6.1	Motivation	105
6.2	Background and Related Work to TraverseNet	108
6.2.1	Definitions and Notations	108
6.2.2	Related work	109
6.3	TraverseNet	110
6.3.1	Message Traverse Layer	111
6.3.2	Model Framework	113
6.3.3	Optimization & Implementation	114
6.4	Experimental Studies	114
6.4.1	Dataset	114
6.4.2	Baseline Methods	116
6.4.3	Experimental Setting	117
6.4.4	Overall Results	117

TABLE OF CONTENTS

6.4.5	Ablation Study	118
6.4.6	Hyperparameter Study	119
6.4.7	Case Study	120
6.4.8	Computation Time	120
6.5	Summary	122
7	Future Work	123
8	Conclusion	125
A	Appendix A	127
A.1	Data Set	127
A.2	Reported Experimental Results for Node Classification	128
A.3	Open-source Implementations	129
B	Appendix B	131
B.1	Derivation of the graph convolutional kernels of low-pass, high-pass, and middle filter functions.	131
	Bibliography	133

LIST OF FIGURES

FIGURE	Page
1.1 2D Convolution vs. Graph Convolution.	3
2.2 RecGNNs v.s. ConvGNNs	18
2.3 Differences between GCN [1] and GAT [2]	25
3.1 Illustration of low-pass, high-pass, and middle-pass filters.	46
3.2 Frequency profiles of low-pass, high-pass, and middle-pass filters with three different settings.	52
3.3 Comparison analysis of model depth between AutoGCN and GCN.	63
4.1 Spatial-temporal graph modeling. Each circle represents a node. Each node has dynamic input features. The aim is to model each node’s dynamic features given the graph structure.	66
4.2 Dilated casual convolution with kernel size 2. With a dilation factor k , it picks inputs every k step and applies the standard 1D convolution to the selected inputs.	71
4.3 The framework of Graph WaveNet. It consists of K spatial-temporal layers on the left and an output layer on the right. The inputs are first transformed by a linear layer and then passed to the gated temporal convolution module (Gated TCN) followed by the graph convolution layer (GCN). Each spatial-temporal layer has residual connections and is skip-connected to the output layer. . . .	73
4.4 Comparison of prediction curves between WaveNet and Graph WaveNet for 60 minutes ahead prediction on a snapshot of the test data of METR-LA. . . .	75
4.5 The learned self-adaptive adjacency matrix.	76
5.1 A concept map of my proposed framework.	82

5.2	The framework of MTGNN. A 1×1 standard convolution first projects the inputs into a latent space. Afterward, temporal convolution modules and graph convolution modules are interleaved with each other to capture temporal and spatial dependencies respectively. The hyper-parameter, dilation factor d , which controls the receptive field size of a temporal convolution module, is increased at an exponential rate of q . The graph learning layer learns the hidden graph adjacency matrix, which is used by graph convolution modules. Residual connections and skip connections are added to the model to avoid the problem of gradient vanishing. The output module projects hidden features to the desired dimension to get the final results.	85
5.3	A demonstration of how a temporal convolution module and a graph convolution module collaborate with each other. A temporal convolution module filters the inputs by sliding a 1D window over the time and node axes, as denoted by the red. A graph convolution module filters the inputs at each step, denoted by the blue.	85
5.4	Graph convolution module and mix-hop propagation layer	87
5.5	Temporal convolution module and dilated inception layer.	89
5.6	Parameter Study. X-axis is the parameter to be studied. Y-axis is the MAE score.	100
5.7	Case study	102
6.1	Message Passing Diagrams	106
6.2	A demonstration of message traverse layer. The node v has two neighbors u_1 and u_2 . Each node has four consecutive states at t_1, t_2, t_3, t_4 . The message traverse layer allows the node v at a certain time step to receive information from its own previous time steps as well as its neighbors' previous time steps.	111
6.3	The model framework of TraverseNet. The TraverseNet mainly consists of three parts, the pre-processing layer, the message traverse layer, and the post-processing layer. The inputs is a sequence of node feature matrix $\mathbf{X}_{t_1}, \mathbf{X}_{t_2}, \dots, \mathbf{X}_{t_p}$. The pre-processing MLP layer projects the input feature matrices to a latent feature space. The message traverse layer propagates information across space and time. The post-processing layer projects the nodes hidden states to the output space.	111
6.4	Cross-correlations	116
6.5	Peak point	116
6.6	Parameter Study	120
6.7	Case study.	121

LIST OF TABLES

TABLE	Page
2.1 Commonly used notations.	10
2.2 Taxonomy and representative publications of Graph Neural Networks (GNNs)	13
2.3 Summary of RecGNNs and ConvGNNs. Missing values (“-”) in pooling and readout layers indicate that the method only experiments on node-level/edge-level tasks.	16
2.4 Time and memory complexity comparison for ConvGNN training algorithms (summarized by [3]). n is the total number of nodes. m is the total number of edges. K is the number of layers. s is the batch size. r is the number of neighbors being sampled for each node. For simplicity, the dimensions of the node hidden features remain constant, denoted by d	26
2.5 Main characteristics of selected GAEs	31
2.6 Summary of selected benchmark data sets.	39
3.1 Summary of node classification datasets. The number of training graphs, validation graphs, and test graphs for PUBMED, Arxiv-year, YelphChi, and Squirrel are missing because they only contain a single graph.	57
3.2 Performance on node classification tasks. Results on test sets are averaged over 5 runs with 5 different seeds. OOM stands for out of memory.	59
3.3 Summary of graph prediction datasets.	59
3.4 Performance on graph prediction tasks. Results on test sets are averaged over 5 runs with 5 different seeds.	60
3.5 Hyperparameter settings for all experiments. L is the number of layers; $hidden$ is the dimension of hidden features; $init\ lr$ is the initial learning rate, $patience$ is the decay patience, $min\ lr$ is the stopping lr, $weight\ decay$ is the weight decay rate, all experiments have $lr\ reduce\ factor\ 0.5$	61
3.6 Ablation study.	62

LIST OF TABLES

4.1	Summary statistics of METR-LA and PEMS-BAY.	72
4.2	Performance comparison of Graph WaveNet and other baseline models. Graph WaveNet achieves the best results on both datasets.	74
4.3	Experimental results of different adjacency matrix configurations. The forward-backward-adaptive model achieves the best results on both datasets. The adaptive-only model achieves nearly the same performance with the forward-only model.	75
4.4	The computation cost on the METR-LA dataset.	77
5.1	Dataset statistics.	93
5.2	Baseline comparison under single-step forecasting for multivariate time series methods.	97
5.3	Baseline comparison under multi-step forecasting for spatial-temporal graph neural networks.	98
5.4	Ablation study.	101
5.5	Comparison of different graph learning methods.	103
5.6	Time Complexity Analysis	103
6.1	Dataset statistics.	115
6.2	Performance comparison.	116
6.3	Ablation study.	118
6.4	Comparison of running time.	122
A.1	Reported experimental results for node classification on five frequently used data sets. Cora, Citeseer, and Pubmed are evaluated by classification accuracy. PPI and Reddit are evaluated by micro-averaged F1 score.	129
A.2	A Summary of Open-source Implementations	130