Spatial-Temporal Bipartite Graph Attention Network for Traffic Forecasting

Dimuthu Lakmal*, Kushani Perera, Renata Borovica-Gajic, and Shanika Karunasekera

[1] University of Melbourne, VIC 3010, Australia
[2] {dkariyawasan@student, kushani.perera, renata.borovica, karus}@unimelb.edu.au

Abstract. Accurate traffic forecasting is pivotal for an efficient data-driven transportation system. The intricate nature of spatial-temporal dependencies and non-linearity present in traffic data has posed a significant challenge to the modeling of accurate traffic forecasting systems. Lately, there has been a significant effort to develop complex Spatial-Temporal Graph Neural Networks (STGNN) that predominantly utilize various Graph Neural Networks (GNN) and attention-based encoder-decoder architectures due to their ability to capture non-linear dependencies in spatial and temporal domains effectively. However, conventional GNNs limit explicit propagation of past information among nodes, while attention-based models such as transformers do not support finer-grained attention score distribution. In this study, we address the aforementioned issues and introduce a novel STGNN namely, Spatio-Temporal Bipartite Graph Attention Network (STBGAT) that allows explicit modeling of past information propagation among nodes. Further, we present a heterogeneous cross-attention mechanism in a transformer to compute finer-grained feature-wise attention distribution enabling the model to capture richer and more expressive temporal dependencies. Our experiments reveal that the proposed architecture outperforms the state-of-the-art approaches proposed in recent literature.

Keywords: Graph Attention Network · Traffic Forecasting · Transformers · Spatial Graph Attention Networks

1 Introduction

Transportation systems have become complex with the rapid growth of infrastructure and people's needs. Thus, relevant stakeholders continuously invest in implementing intelligent transportation systems (ITS) aiming for more efficient, accurate, and data-driven traffic management solutions . Accurate and real-time traffic condition forecasting is one of the core components of ITS. Traffic condition forecasting systems are designed to predict future traffic conditions given the historical traffic condition observations. Particularly, we focus on forecasting traffic flow, one of the main traffic condition measurements. Typically, traffic flow in a specific road section is influenced by not only its own historical traffic conditions, but also the traffic conditions in adjacent connected road sections. *Hence, it is crucial to consider the propagation of traffic information through the spatial structure of the road network when forecasting traffic flow. Moreover, intricate temporal dynamics in road networks have made long-term (30$\backsim$60 minutes) traffic forecasting even more challenging* [28].

To address the aforementioned challenges, researchers have formulated traffic flow forecasting as a spatial-temporal graph modeling problem and proposed various types of Spatial-Temporal Graph Neural Networks (STGNN) [29, 12]. Even though recent efforts have attained substantial improvements [9] in accuracy compared to early versions of STGNNs [27], they are not sophisticated enough to effectively discover and leverage intricate spatial and temporal dependencies. This study primarily focuses on two major deficiencies of existing traffic forecasting STGNNs. First, the effects of traffic conditions on roads take time to gradually propagate to their adjacent roads through the network. However, existing approaches fail to ascertain how the traffic flow of a specific road at a given time is impacted by previous traffic conditions on adjacent roads. Second, the majority of existing approaches relied on raw historical observations as input features and have not included and assessed alternative feature sequences, such as averaged traffic flow sequences which could reveal more temporal and spatial dependencies [29].

To address these shortcomings, we present a novel spatial-temporal graph neural network (STBGAT), that consists of a bipartite graph attention network and a transformer with a heterogeneous cross-attention mechanism (Source code is available here: https://github.com/DimuthuLakmal/STBGAT). We conducted a comprehensive set of experiments using five different traffic datasets to evaluate the performance of the proposed model. Those experiments revealed that STBGAT significantly outperforms the current state-of-the-art models. The main contributions of this study can be summarized as follows:

- We propose a novel Bipartite Graph Attention Network for past neighborhood information propagation towards center nodes. This mechanism ensures the impact of the previous traffic conditions on adjacent roads is explicitly considered.
- We introduce a heterogeneous cross-attention mechanism in the transformer model which enables the decoder to assign separate feature-wise attention scores to the encoder outputs. This architecture allows for the integration of multiple encoders, each handling different input sequences. It will alleviate the impact of noise and missing values in each feature sequence while revealing more temporal and spatial dependencies.

2 Related Work

Prior to recent advancements in Graph Neural Networks (GNN), researchers have widely adopted classic statistical time series algorithms and machine learning models to make predictions [25, 21]. However, these models are only capable of analyzing temporal dynamics in traffic data leaving spatial correlations unused. In contrast, Spatial-Temporal Graph Neural Networks (STGNN) have significantly improved accuracy by efficiently capturing and modeling both temporal and spatial dynamics in road networks [27, 24]. Since the introduction of STGCN architecture by Yu et al. [27], various highly complex STGNN architectures have been proposed in the literature attaining substantially improved accuracy [9, 29].

Various graph neural network architectures have been adopted as the spatial module in STGNNs, ranging from recurrent GNN to Graph Attention Neural Networks (GAT) [17, 7, 12]. GCN [11] is one of the prominent works in the Graph Neural Network domain that led to rapid success in STGNNs [26]. The superiority in efficiency, flexibility, and accuracy of GCN and its variants over prior GNN architectures have resulted in wide adoption of GCNs in STGNN models. On the other hand, Graph Attention Network (GAT) [23] outperforms GCNs as it uses an attention mechanism in the data propagation process within the graph. In this study, we develop the proposed bipartite graph by extending the default GAT implementation.

Further, various architectures have been proposed for the temporal module in STGNNs. There are three frequently used neural network architectures in the temporal module: 1) RNN-based, 2) CNN-based, and 3) Attention-based [1, 14, 16]. Compared to the other two, the attention mechanism has emerged as a highly compelling approach in temporal sequence modeling. We develop the proposed temporal module based on a transformer which is an encoder-decoder architecture relying on an attention mechanism [22]. None of the recent STGNN approaches have proposed explicit modeling of past information propagation from neighbors due to the additional complexity it imposes on these models. Further, only a few attempted to incorporate features beyond raw traffic flow values as inputs [5], and these attempts were insufficient to distinctly discern the significance of each feature in making predictions.

3 Definitions and Problem Statement

3.1 Definitions

Traffic Road Network We represent a traffic road network with a directed graph $G = (V, E, A)$ where $V = v_1, ..., v_N$ is a set of N nodes representing traffic sensors; E is a set of edges among nodes; $A \in R_{N\times N}$ is a weighted adjacency matrix representing connectivity among nodes and edge weights between any of two connected nodes.

Traffic Flow Matrix $X_t \in N \times C$ denotes the traffic condition feature matrix at time step t. N represents the number of nodes in the network and C represents the number of traffic condition-related features including traffic flow value associated with each individual node. $X_t^{'} \in N \times 1$ denotes the traffic flow matrix at time step t.

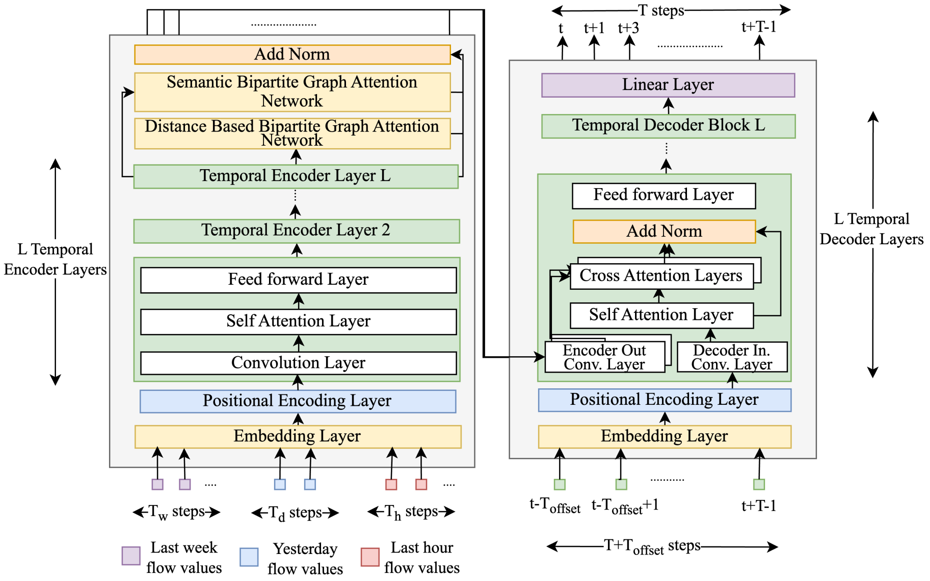

Fig. 1. Overall architecture of STBGAT

3.2 Problem Statement

Given the historical observation spanning a specific time frame, the problem is to establish a mapping function that can output the sequence of future traffic flow values having a predefined length. Let us assume the number of historical observations ending at the current step t is T_1 and the length of the prediction sequence is T_2. Then the mapping function f can be formally expressed as, $[X_{(t-T_1+1)}, X_{(t-T_1+2)}, ..., X_t; G] \xrightarrow{f} [X'_{(t+1)}, X'_{(t+2)}, ..., X'_{(t+T_2)}; G]$

4 Methodology

4.1 Data Inputs and Data Preprocessing

This section focuses on briefing data inputs and the non-trivial data preprocessing steps we followed in this study.

If the target sequence is the traffic flow in the next hour from the current time, the most recent available sequence will be the traffic flow from the last hour, and it is incorporated as a part of the input sequence to the encoder. Several past studies have utilized additional sequences of historical observations as inputs, where these sequences may closely resemble either the pattern of the target sequence or the latest historical observation sequence [19]. We identified two effective periodic sequence patterns that tend to yield better results across all the datasets that we experimented with. The first is the sequence of traffic flow from the same hour as the hour just preceding the target hour, but on the previous date. The second effective periodic sequence pattern is from the same hour as the hour just preceding the target hour, but on the same day in last week.

We concatenate these three shorter sequences into one single sequence and use as an input sequence to the model. Additionally, we use another input sequence to the model, referred to as a representative input sequence, representing averaged behavior corresponding to the time duration of the raw input sequence described earlier. Incorporating repetitive and representative sequence patterns could benefit the model in two ways. First, it helps the model to identify long-term and short-term trends. Second, it helps to mitigate the impact of missing values in shorter sequences. The total length of each encoder sequence can be defined as $T_e = T_{last\_week} + T_{last\_day} + T_{last\_hour}$.

To construct the representative input sequence, it is required to determine the average behavior at each weekly time index. All traffic flow datasets we tested in this study consist of 12 traffic flow values per hour, totaling 2016 per week. Therefore, we can assign a weekly time index for each traffic flow record. When calculating the average traffic flow at a particular weekly time index, we judiciously apply a rule-based filter to remove traffic flow records with noises.

We redefined the connectivity within the sensor network in certain datasets for more efficient and accurate flow of information among nodes. We introduced two types of connectivity producing two different connectivity graphs namely: distance-based bipartite graph and semantic bipartite graph. The distance-based graph is defined based on the assumption that two sensors in close geographic proximity to each other exhibit significant correlations between their recorded traffic flow values. To accommodate this assumption, we calculated edge attributes based on the shortest distance between nodes using Dijkstra's algorithm [10]. Then we dropped edges that exceeded a predefined distance threshold.

We defined a second graph based on the time series semantics among nodes. This helps to identify nodes that have similar behavior, but are not connected in the geographically connected graph defined above. For instance, in a scenario where two sensor nodes are located near two different schools, but are physically distant from each other, it may be important to propagate information between those two sensor nodes to identify common short-term temporal behaviors. For each weekly index described above, we picked a certain number of most similar nodes for every node in the graph. Then, a single global semantic graph is constructed assigning the set of nodes as neighbors of each node in the graph, which have the highest number of short-term similar behaviors with each center node. This calculation is based on the semantic distances among representative time series of nodes, each of which consists of 12 time steps. Dynamic Time Warping (DTW) algorithm is used to measure the similarity between two sequences [3].

4.2 Encoder Decoder Architecture

In this section, we brief the overall architecture of STBGAT model depicted in Fig.1. The model follows transformer encoder-decoder architecture with some optimization done focusing on the traffic forecasting problem. The model can accommodate multiple encoders at once to facilitate feature extraction from multiple types of input sequences. In this study, we employ two encoders to process the two input sequences described in Section 4.1. The embedding layer

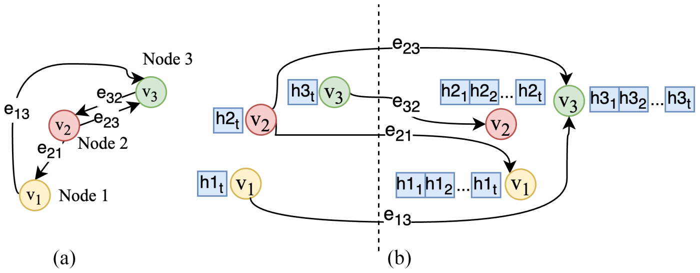

Fig. 2. Formation of bipartite graph. Fig.2a shows the example base graph with three nodes for the bipartite graph depicted in Fig.2b.

maps the inputs to a higher dimensional vector space. The positional embedding layer fuses time step information and makes sure that the self-attention module is aware of the positional information.

The output of the positional encoding layer is then processed through a temporal encoder layer stack comprised of L number of layers. Each encoder layer includes a convolution layer similar to the one used in the temporal trend-aware multi-head self-attention layer described in [4]. The processed output of the convolution layer is then fed into a self-attention layer followed by a feed-forward layer. The output of the last temporal encoder layer is subsequently passed into the spatial module that consists of two bipartite graph attention networks enabling information propagation among nodes. Bipartite graphs are constructed as explained in Section 4.1. Finally, the output from bipartite GATs and the output from the final temporal encoder layer are summed together to produce the final encoder output. The decoder uses the output from the encoder as the historical context in the process of cross-attention calculation.

The embedding layer, positional encoding layer, self-attention layer, and feed-forward layer of the decoder are similar to the ones used in the encoder. Further, the decoder input convolution layer is also similar to the convolution layer used in the encoder layer. We introduce a heterogeneous feature-wise cross-attention mechanism as described in Section 4.4. Latent vector output given by the last decoder layer L is projected to an output with a single dimension by the linear layer of the decoder.

4.3 Bipartite Graph Attention Layer

In this section, we explain one of the main contributions of this paper that solves past information propagation problem described in Section 1. Traffic flow recorded at a sensor at a specific time step is influenced not only by the traffic flow on neighboring roads at the exact time step but also by previous traffic conditions on neighboring roads. The dependency on traffic conditions during previous time steps arises due to the propagation delay between different parts of the road network [9]. For instance, if a road accident occurs on a road section, then the traffic conditions of neighboring road segments will gradually adjust over time to accommodate the impact of the accident.

It can be argued that as the input sequences pass through a set of temporal encoder layers before reaching the spatial module, the latent vector output at each time step comprises a certain level of information about its preceding temporal context. Nonetheless, our study reveals that this implicit representation of the temporal context alone is not sufficient to effectively address the propagation delay in road networks. Instead, we propose to use a bipartite graph G_t that consists of two sets of nodes, (u_t, v_{T_e}) where u_t represents nodes consisting of outputs of temporal encoder module at time step t, and each node in v_{T_e} consists of concatenated output of temporal encoder from time step $t = 1$ to $t = T_e$. The edge attributes of the bipartite graph are equal to the corresponding edge attributes in the regular graph. The formulation of the bipartite graph from a regular graph is shown in Fig.2. The implementation of Bipartite GAT can be outlined in three equations from Eq.1 to Eq.3.

$$e_{ij} = LeakyReLU(a^T(Wh_i||Wh_j||W_E E_{ij})) \tag{1}$$

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k \in N_i} exp(e_{ik})} \tag{2}$$

$$h_i^{imm} = \|_{k=1}^{K} \sigma(\sum_{j \in N_i} \alpha_{ij}^k W^k h_j) \tag{3}$$

In Eq.1, W, W_E and a denote learnable weights while E_{ij} denotes edge attribute associated with the edge connecting nodes i and j. Node i belongs to u_t and node j belongs to v_{T_e}. h_i represents the value of the central node i while h_j represents the value of neighbor node j. α_{ij} in Eq.2 is the attention score calculated for neighbor node j. The final output for central node i is derived using Eq.3. A single GAT layer consists of K heads. Thus, the final output h_i^{imm} is formed by either a concatenating or averaging of the outputs generated by K heads. Eq.3 shows only the concatenation operation over K heads.

4.4 Heterogeneous Cross Attention Layers

In this section, we present the second contribution of this paper that enables integrating multiple feature sequences. The transformer decoder consists of a cross-attention component that calculates attention values for elements in the encoder output sequence with respect to the decoder input sequence. This mechanism allows the decoder to focus on relevant information in the encoder output when generating the decoder output. The naive implementation of cross-attention only accepts the encoder output sequence as a single sequence and is not granular enough to calculate feature-wise attention. In contrast, we propose a heterogeneous cross-attention mechanism that is capable of calculating distinct attention distributions for different feature sequences. This mechanism allows for more precise modeling of temporal dynamics and relationships present in different feature sequences. STBGAT produces two separate encoder output sequences in parallel using the two input sequences described in Section 4.1.

Table 1. Prediction accuracy results (PEMS04-08)

		VAR	SVR	LSTM	DC-RNN	ST-GCN	GMAN	AST-GNN	PDF-ormer	PDF-ormer(L)	**ST-BGAT**
PEMS04	MAE	23.75	28.66	26.81	23.65	22.27	19.14	18.60	18.39	18.40	**18.17**
	RMSE	36.66	44.59	40.74	37.12	35.02	31.60	31.03	30.01	30.25	**28.23**
	MAPE	18.10	19.15	22.33	14.75	13.87	13.19	12.63	12.13	12.23	**12.02**
PEMS07	MAE	101.20	32.97	29.71	23.63	22.90	20.97	20.62	19.83	N/A	**18.34**
	RMSE	155.14	50.15	45.32	36.51	35.44	34.10	34.02	32.87	N/A	**30.86**
	MAPE	39.69	15.43	14.14	12.28	11.98	9.05	8.86	8.53	N/A	**7.63**
PEMS08	MAE	22.32	23.25	22.19	18.19	17.84	15.31	13.29	13.58	12.51	**12.39**
	RMSE	33.83	36.15	33.59	28.18	27.12	24.92	23.33	23.51	22.10	**21.02**
	MAPE	14.47	14.71	18.74	11.24	11.21	10.13	9.03	9.05	8.55	**8.43**

Following this, two cross-attention distributions are generated based on encoder output sequences. The final output of the cross-attention layer will be calculated according to the Eq.4. In Eq.4, LayerNorm refers to layer normalization. $X_{self-attn}$ refers to self-attention output calculated over decoder input. x_f is the encoder output calculated for the sequence of feature type f.

$$h_{cross} = LayerNorm(X_{self-attn} + \sum_{f \in F} CrossAttn(x_f)) \tag{4}$$

5 Experiments

5.1 Experiment Setup

We evaluate our model on two groups of datasets that are widely used in the literature. The first group consists of three datasets; PEMS04, PEMS07, and PEMS08 [2] while the second group consists of two datasets; PEMS-BAY and METR-LA [8].

We evaluate STBGAT against a variety of baselines proposed in the literature on the aforementioned two dataset groups. Tested baseline models are listed below.

- **PEMS04-08**: *VAR [18], SVR, LSTM [6], DCRNN [15], STGCN [27], GMAN [29], ASTGNN [4], PDFormer [9]*
- **PEMS-BAY, METR-LA**: *VAR [18], SVR, FC-LSTM [6], DCRNN [15], STGCN [27], GMAN [29], STGM [13], STEP [20]*

The default PDFormer only supports input sequences with a maximum length of 12. However, to ensure a fair comparison, we adapted the default PDFormer to accept a 36-length input sequence that includes repetitive patterns. It is referred to as PDFormer(L) in Table 1. In contrast, other recent architectures STEP and ASTGNN support longer sequences by default.

Three evaluation matrices are used namely, Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE).

Table 2. Prediction accuracy results (PEMS-BAY, METR-LA)

		VAR	SVR	LSTM	DC-RNN	ST-GCN	GMAN	STGM	STEP	**ST-BGAT**
PEMS-BAY	MAE	2.93	3.28	2.37	2.07	2.49	1.86	1.86	1.79	**1.75**
	RMSE	5.44	7.08	4.96	4.74	5.69	4.32	4.37	4.20	**3.60**
	MAPE	6.50	8.00	5.70	4.90	5.79	4.37	4.34	4.18	**4.01**
METR-LA	MAE	6.52	6.72	4.37	3.60	4.59	3.44	**3.23**	3.37	3.94
	RMSE	10.11	13.76	8.69	7.60	9.40	7.35	7.10	**6.99**	7.25
	MAPE	15.80	16.70	14.00	10.50	12.70	10.07	**9.39**	9.61	9.79

Masked versions of MAE, RMSE, and MAPE matrices are used to alleviate the effect of missing values in the dataset. Experiments on each model are repeated 3 times and we report mean values for said matrices.

The test results of the STBGAT model presented in Table 1 and Table 2 are generated by passing input sequences consisting of repetitive patterns. The reported error values are the averaged error values computed across the entire prediction sequence length.

5.2 Comparison of Performance

The overall performance of baseline models and STBGAT is summarized in Table 1 and Table 2. It is important to highlight that the PDFormer(L) model fails to run on PEMS07 dataset due to memory overflow which suggests that it is not suitable for large road networks (tested on 128GB of RAM). The best results for each metric reported in each dataset are highlighted in bold. Our model outperforms all baselines in every performance metric across four datasets; PEMS07, PEMS08, PEMS04, and PEMS-BAY. However, the STBGAT model exhibits lower performance on the METR-LA dataset, particularly in MAE metric. We discover that this is attributed to the high discrepancy between train and test data distributions. Moreover, STBGAT model significantly outperforms all baselines in terms of RMSE metric. This metric is a useful indicator of performance when large errors between ground truth and predicted values are undesirable. Hence, it suggests that STBGAT can better approximate sudden fluctuations in traffic flow.

The most notable observation across the experiments is the substantial performance enhancement achieved by spatial-temporal models in comparison to temporal prediction models alone. This accentuates the importance of discovering and exploiting spatial dependencies in road network graphs. LSTM exhibits the best performance among temporal prediction models leveraging its ability to identify long temporal dependencies compared to other temporal models. Based on our experiments, the spatial module can be considered as an enhancement to improve the accuracy of the temporal module of spatial-temporal models. Hence, having a comprehensive temporal module is also important to have better performance. DCRNN and STGCN models consist of RNN and convolution components in the temporal module that could hinder performance in modeling

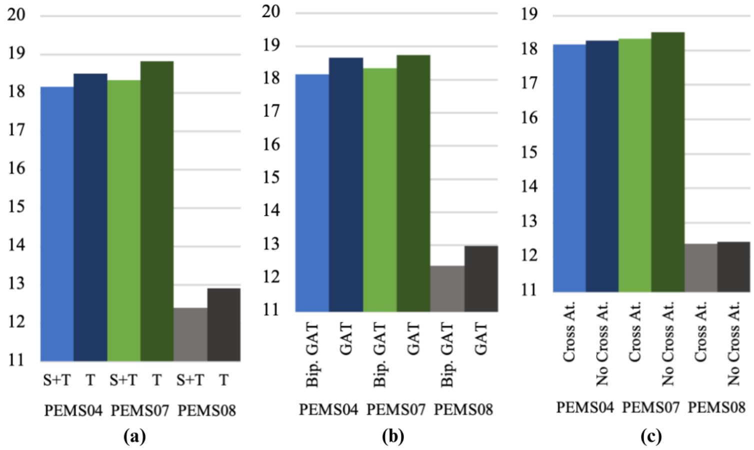

Fig. 3. Ablation study results (MAE Values)

temporal dependency effectively. In contrast, GMAN, ASTGNN, STEP, STGM, and PDFormer use various attention mechanisms that help achieve superior performance over traditional RNN-based models. In addition to the performance enhancements achieved through the two novel concepts presented here, two other existing concepts contributed to STBGAT's performance. First, STBGAT use CNN layers in temporal module for local context recognition which is also used in ASTGNN, PDFormer and STGM. Second, STBGAT effectively harnesses both short-term and long-term spatial dependencies as it uses two types of bipartite graphs in the encoder. A comparable approach is also utilized in PDFormer and STGM models. The contribution of the two novel concepts presented in this paper is discussed in the next section.

5.3 Ablation Study

Additional experiments are carried out to investigate the effectiveness of different components of STBGAT using PEMS04, PEMS07 and PEMS08 datasets. We first study the prediction capability of the temporal module by training the model without the spatial component of the model. We then test the impact and contribution of the spatial component to predictions. The evaluation results are depicted in Fig.3a. Results suggest that the temporal module plays a critical role in the prediction task. The model with only the temporal module outperforms the majority of the baselines except for ASTGNN and PDFormer in every performance metric. These results also indicate the importance of information propagation within the graph, particularly in PEMS07 and PEMS08 datasets.

Next, we assess the performance enhancement in the spatial module achieved by bipartite GAT in comparison to conventional GAT. The results of this experiment are presented in Fig.3b. A substantial improvement can be observed when utilizing the proposed bipartite GAT in contrast to conventional GAT.

Finally, an experiment is conducted to evaluate the impact of the heterogeneous cross-attention mechanism compared to the traditional cross-attention mechanism. The results of this experiment are presented in Fig.3c. According to the experiment, the effect of the heterogeneous cross-attention mechanism on the performance is not as pronounced as in bipartite GAT. Nevertheless, it contributes to enhancing the overall performance of the model compared to the conventional cross-attention mechanism.

6 Conclusion and Future Works

In this paper, we introduce a novel spatial-temporal graph neural network architecture for traffic forecasting that outperforms the latest state-of-the-art baselines across four real-world datasets. We proposed two novel concepts in this paper; bipartite graph attention network and heterogeneous cross-attention mechanism. The first concept enhances the spatial information propagation while the second concept improves the temporal dependency analysis of the model. The ablation study demonstrates the effectiveness of these two novel concepts in modeling spatial and temporal dynamics. As future work, the model can be extended and utilized in various downstream tasks in domains such as traffic analysis and social media.

Acknowledgements This research was supported by The University of Melbourne's Research Computing Services and the Petascale Campus Initiative.

References

1. Bai, L., Yao, L., Kanhere, S.S., Wang, X., Liu, W., Yang, Z.: Spatio-temporal graph convolutional and recurrent networks for citywide passenger demand prediction. In: Proceedings of the 28th ACM CIKM. pp. 2293–2296 (2019)
2. Chen, C., Petty, K., Skabardonis, A., Varaiya, P., Jia, Z.: Freeway performance measurement system: mining loop detector data. TRR **1748**(1), 96–102 (2001)
3. Giorgino, T.: Computing and visualizing dynamic time warping alignments in r: the dtw package. Journal of statistical Software **31**, 1–24 (2009)
4. Guo, S., Lin, Y., Wan, H., Li, X., Cong, G.: Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting. IEEE TKDE **34**(11), 5415–5428 (2021)
5. He, H., Ye, K., Xu, C.Z.: Multi-feature urban traffic prediction based on unconstrained graph attention network. In: 2021 IEEE BigData. pp. 1409–1417 (2021)
6. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
7. Huang, R., Huang, C., Liu, Y., Dai, G., Kong, W.: Lsgcn: Long short-term traffic prediction with graph convolutional networks. In: IJCAI. vol. 7, pp. 2355–2361 (2020)
8. Jagadish, H.V., Gehrke, J., Labrinidis, A., Papakonstantinou, Y., Patel, J.M., Ramakrishnan, R., Shahabi, C.: Big data and its technical challenges. Communications of the ACM **57**(7), 86–94 (2014)
9. Jiang, J., Han, C., Zhao, W.X., Wang, J.: Pdformer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction. arXiv preprint arXiv:2301.07945 (2023)

10. Johnson, D.B.: A note on dijkstra's shortest path algorithm. JACM **20**(3), 385–388 (1973)
11. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
12. Kong, X., Xing, W., Wei, X., Bao, P., Zhang, J., Lu, W.: Stgat: Spatial-temporal graph attention networks for traffic flow forecasting. IEEE Access **8**, 134363–134372 (2020)
13. Lablack, M., Shen, Y.: Spatio-temporal graph mixformer for traffic forecasting. Expert Systems with Applications **228**, 120281 (2023)
14. Li, W., Wang, X., Zhang, Y., Wu, Q.: Traffic flow prediction over muti-sensor data correlation with graph convolution network. Neurocomputing **427**, 50–63 (2021)
15. Li, Y., Yu, R., Shahabi, C., Liu, Y.: Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. arXiv preprint arXiv:1707.01926 (2017)
16. Li, Y., Moura, J.M.: Forecaster: A graph transformer for forecasting spatial and time-dependent data. In: ECAI 2020, pp. 1293–1300. IOS Press (2020)
17. Lu, Z., Lv, W., Cao, Y., Xie, Z., Peng, H., Du, B.: Lstm variants meet graph neural networks for road speed prediction. Neurocomputing **400**, 34–45 (2020)
18. Lütkepohl, H.: Vector autoregressive models. Handbook of research methods and applications in empirical macroeconomics **30** (2013)
19. Roy, A., Roy, K.K., Ali, A.A., Amin, M.A., Rahman, A.M.: Unified spatio-temporal modeling for traffic forecasting using graph neural network. In: 2021 IJCNN. pp. 1–8. IEEE (2021)
20. Shao, Z., Zhang, Z., Wang, F., Xu, Y.: Pre-training enhanced spatial-temporal graph neural network for multivariate time series forecasting. In: Proc. 28th ACM SIGKDD Conf. Know. Disc. Data Min. pp. 1567–1577 (2022)
21. Tian, Y., Zhang, K., Li, J., Lin, X., Yang, B.: Lstm-based traffic flow prediction with missing data. Neurocomputing **318**, 297–305 (2018)
22. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Adv. NIPS **30** (2017)
23. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y., et al.: Graph attention networks. stat **1050**(20), 10–48550 (2017)
24. Wang, X., Ma, Y., Wang, Y., Jin, W., Wang, X., Tang, J., Jia, C., Yu, J.: Traffic flow prediction via spatial temporal graph neural network. In: Proc. web conf. 2020. pp. 1082–1092 (2020)
25. Williams, B.M., Hoel, L.A.: Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results. Journ. trans. eng. **129**(6), 664–672 (2003)
26. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Philip, S.Y.: A comprehensive survey on graph neural networks. IEEE TNNLS **32**(1), 4–24 (2020)
27. Yu, B., Yin, H., Zhu, Z.: Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting
28. Yu, H., Wu, Z., Wang, S., Wang, Y., Ma, X.: Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks. Sensors **17**(7), 1501 (2017)
29. Zheng, C., Fan, X., Wang, C., Qi, J.: Gman: A graph multi-attention network for traffic prediction. In: Proc. of the AAAI conf. on art. intell. vol. 34, pp. 1234–1241 (2020)