STLGRU: Spatio-Temporal Lightweight Graph GRU for Traffic Flow Prediction

Kishor Kumar Bhaumik[1], Fahim Faisal Niloy[2],
Saif Mahmud[3], and Simon S. Woo[1]*

[1] Sungkyunkwan University, South Korea
[2] University of California, Riverside
[3] Cornell University

{kishor25,swoo}@g.skku.edu, fnilo001@ucr.edu, sm2446@cornell.edu

Abstract. Reliable forecasting of traffic flow requires efficient modeling of traffic data. Indeed, different correlations and influences arise in a dynamic traffic network, making modeling a complicated task. Existing literature has proposed many different methods to capture traffic networks' complex underlying spatial-temporal relations. However, given the heterogeneity of traffic data, consistently capturing both spatial and temporal dependencies presents a significant challenge. Also, as more and more sophisticated methods are being proposed, models are increasingly becoming memory-heavy and, thus, unsuitable for low-powered devices. To this end, we propose **S**patio-**T**emporal **L**ightweight Graph **GRU**, namely *STLGRU*, a novel traffic forecasting model for predicting traffic flow accurately. Specifically, our proposed *STLGRU* can effectively capture dynamic local and global spatial-temporal relations of traffic networks using memory-augmented attention and gating mechanisms in a continuously synchronized manner. Moreover, instead of employing separate temporal and spatial components, we show that our memory module and gated unit can successfully learn the spatial-temporal dependencies with reduced memory usage and fewer parameters. Extensive experimental results on three real-world public traffic datasets demonstrate that our method can not only achieve state-of-the-art performance but also exhibit competitive computational efficiency. Our code is available at https://github.com/Kishor-Bhaumik/STLGRU

Keywords: Traffic Forecasting, Time Series, Graph Convolution

1 Introduction

A traffic network can be represented as a graph, with the locations of the sensors and the connections among them acting as the nodes and edges, respectively. In the same way, flow at a particular junction or node is defined as the total number of people or vehicles passing through that junction at a given time. Specifically, the goal of traffic flow prediction algorithms is to predict the flow of future time steps by exploiting the complex spatialtemporal features of historical traffic data. Indeed, many cities are currently developing Intelligent Traffic

---

* Corresponding Author

Systems (ITS) [29] and predicting traffic flow is a key part of many of these systems' services. In particular, a large amount of collected traffic data have made urban data mining study much easier than ever before, such as traffic flow prediction [9], arrival time estimate [12], traffic speed analysis [2,4], and so on, thanks to the promising advancement of intelligent sensors. To be more specific, spatio-temporal traffic prediction aims to forecast future traffic trends by analyzing previous spatio-temporal features [30]. Furthermore, predicting traffic flow has become essential for several downstream applications, such as intelligent route planning [18], dynamic traffic management [27], and location-based services [16]. However, the efficiency and accuracy of traffic flow prediction algorithms are limited by the high variance in the spatial and temporal dimensions of traffic data. In addition, the observations made at different locations and time stamps are not independent, but they are rather dynamically correlated. Hence, traffic data has a nonlinear and complex spatial-temporal relationship, and its modeling is critical for designing effective prediction algorithms.

To address the aforementioned challenges, in this paper, we propose a novel traffic flow prediction model, called Spatio-Temporal Lightweight Graph GRU *(STLGRU)*. Our model takes advantage of graph convolution to model localized spatial relations. We then use an attention mechanism with a memory module to directly model the long-range local and non-local spatio-temporal dependencies. To update the memory, we use a gating mechanism, where our gating strategy records the key local and global spatio-temporal information and forgets the redundant ones when moving to the next time step. In addition, we carefully design our model to be lightweight, as the memory module uses fewer parameters than the existing baselines. Consequently, it can effectively learn long-range dependencies without the need to use multi-scale causal convolution or stacking past time step features. In summary, we make the following contributions:

- We propose *STLGRU*, a novel time series traffic flow prediction model. Our model captures the long-range global and local relationships of a traffic network more accurately by using memory-augmented attention module and gating mechanism.
- We carefully design our network to be lightweight by utilizing a memory module with minimal parameters, thus making it suitable for environments constrained by computational resources.
- We conduct extensive experiments on three popular traffic prediction benchmark datasets. Our results show that our model not only surpasses other baseline models in performance but also necessitates less memory usage in comparison.

2   Related Work

Spatio-temporal time series traffic forecasting. Deep learning has been successfully applied to many tasks, such as image analysis [21,22], natural language processing [8], activity recognition [20] etc. Recently, such learning techniques have been quite extensively applied to traffic flow prediction task. Amongst

these methods, STGCN [28] is the first pioneering work to model the traffic network with a fully convolutional structure. In this study, spatio-temporal relationships are effectively captured by including a graph convolution module inside temporal convolution modules. Moreover, DCRNN [17] introduces diffusion convolution to propagate information in the graph. PM-MemNet [15] learns to match input data to representative patterns with a key-value memory structure. Song et al. proposes STSGCN [23], which captures complex localized spatial-temporal correlation to find the heterogeneities in the spatial-temporal data. STSGCN [23] deals with spatial and temporal dimensions individually by utilizing various modules and calculates spatio-temporal attention within a restricted temporal frame.

In addition, Lin et al. [19] propose self-attention Conv-LSTM to capture long-range temporal dependencies for general spatio-temporal prediction task. A significant limitation of their approach is its reliance solely on convolution layers, confining their method to spatio-temporal prediction tasks representable by image grids. However, the traffic network has an inherent graph structure that needs to be exploited for reliable prediction. Yuzhou et al. [5] tackles this problem by enriching DL architectures with salient time-conditioned topological information of the traffic data. This study introduces the zig-zag persistence concept into time-aware graph convolutional networks.

However, most of the cutting-edge models fail to handle the challenge of being lightweight. RNN-based networks (including LSTM) are widely known to be difficult to train and computationally heavy [28]. For example, Mega-CRN [14] proposes Meta-Graph Convolutional Recurrent Network (MegaCRN) by plugging multiple Meta-Graph Learner powered by a MetaNode Bank into the encoder-decoder module. As a consequence, it becomes memory-heavy due to its large number of parameters. STSGCN [23] uses a certain length of time window to collect graph structure information and fuse the findings to forecast the following time steps. The computational cost is thus increased by employing repeated shots of graph aggregation. StemGNN [1] introduces a neural network that captures inter-series correlations and temporal dependencies in the spectral domain by aggregating numerous modules in separate blocks while disregarding the model's complexity. To solve the aforementioned issues, we present a simple but effective traffic forecasting model that is computationally cheap, lightweight, and capable of capturing both local and global long-range dependencies in a traffic network.

Attention Mechanism. Because of the high efficiency and versatility in modeling dependencies, attention mechanisms have been extensively used in a variety of domains [24,6]. The basic principle behind attention mechanisms is to concentrate on the most relevant features of the input data [6]. Recently, researchers used attention processes to graph-structured data to model spatial correlations for graph classification [25]. We expand the attention method to synchronize spatial and temporal dependencies while sequentially predicting traffic data.

Fig. 1: Overall architecture of STLGRU designed for multivariate traffic forecasting. Our model consists of a memory-augmented attention module and a gated unit, which capture the long-range local and global dependencies. It takes input from a single time step with an initial hidden state and outputs a hidden state for the next time step.

3   Proposed Model

3.1   Preliminaries and Problem Definition

A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ models the traffic topological network, where $\mathcal{V}$ represents nodes and $\mathcal{E}$ signifies edges. An edge $e_{ij} \in \mathcal{E}$ connects nodes v_i and v_j, where each node has junctional features (e.g. inflow, outflow). Then, we define the spatio-temporal traffic data forecasting problem using a mapping function f_θ, where it takes the historical series $\langle X_{(t-T+1)}, X_{(t-T+2)}, \ldots, X_t \rangle$. And, it predicts the future series $\langle X_{(t+1)}, X_{(t+2)}, \ldots, X_{(t+T')} \rangle$, where T is the length of the historical series and T' is the length of the target forecast series, and, $X_i \in \mathbb{R}^{N \times C}$, where N is the number of nodes and C is the number of information channels (speed, flows, etc.).Thus, the time series forecasting model can be defined as follows:

$$\langle X_{(t-T+1)}, X_{(t-T+2)}, \ldots, X_t \rangle \xrightarrow{f_\theta} \langle X_{(t+1)}, X_{(t+2)}, \ldots, X_{(t+T')} \rangle$$

3.2   Graph Convolution

Here, we first define the graph convolution, where the initial input matrix is denoted as $X \in \mathbb{R}^{N \times T \times C}$. As our focus is solely on traffic flow, C is consequently set to 1, resulting $X \in \mathbb{R}^{N \times T \times 1}$. And, we take $X_{t'} \in \mathbb{R}^{N \times 1}$ as input from a single time step t, where $t \in T$, and pass it through a convolutional layer ξ_θ to transform the input feature into high-dimensional space C' to increase the representation power of the network as follows:

$$X_t = \xi_\theta \left(X_{t'} \right); \theta \in \mathbb{R}^{1 \times C'} \tag{1}$$

Then, $X_t \in R^{N \times C'}$ is used as an input to the original network at time step t.

Fig. 2: Graph generation from learnable node embeddings E

As shown in Fig. 2, let $E \in \mathbb{R}^{N \times d}$ be the learned node embedding matrix, where d represents the embedding dimension. In addition, Ω represents the probability matrix, and each $\Omega_{ij} \in \Omega$ corresponds to the probability of preserving the edge between time series i and j, respectively. This relationship is formally expressed as follows:

$$\Omega = EE^T \tag{2}$$

In particular, we use the Gumbel softmax method [13] to obtain the final sparse adjacency matrix $A \in R^{N \times N}$ to effectively assure a sufficient amount of sparsity in the graph structure. And, let σ and τ be the activation function and the temperature variable, respectively. Then, we can define sparse adjacent matrix A as follows:

$$A = \sigma((log(\Omega_{ij}/(1 - \Omega_{ij}) + (n_{ij}^1 - n_{ij}^2)/\tau)$$
$$s.t.\ n_{ij}^1, n_{ij}^2 \sim Gumbel(0, 1) \tag{3}$$

Eq. (3) implements Gumbel Softmax for our task, where $A_{i,j} = 1$ with probability $\Omega_{i,j}$ and $A_{i,j} = 0$ with the remaining probability. In particular, Gumbel Softmax maintains the same probability distribution as the normal Softmax, ensuring statistical consistency in generating the trainable probability matrix for the graph forecasting network. Next, let I be an identity matrix and D be a diagonal degree matrix satisfying $D_{ii} = \Sigma_j A_{ij}$. Then, the specific operation of graph convolution network (GCN) with the learnable weight $W \in R^{C' \times C'}$ can be expressed as follows:

$$GCN\left(X_t\right) = W(I + D^{-\frac{1}{2}}AD^{-\frac{1}{2}})X_t \in \mathbb{R}^{N \times C'} \tag{4}$$

3.3   Memory-Augmented Attention (MAA) Module

As discussed before, many state-of-the-art models struggle with maintaining a lightweight design. For instance, models proposed by Jiang et al. [1] and Yu et al. [2] have learned spatio-temporal relations by combining GCN and GRU modules and they further stack these fused modules multiple times. However, when stacking multiple layers to capture long-term dependencies in traffic data, they encounter a significant increase in memory usage during inference. To mitigate this issue, we introduce a memory-augmented attention (MAA) mechanism by

continuously synchronizing relevant features in both spatial and temporal data in each timestep. In particular, Figure 1 illustrates the MAA module's structure, where it combines the graph convolution output $J_r \in \mathbb{R}^{N \times C'}$ with the randomly initialized hidden input $H_{t-1} \in \mathbb{R}^{N \times C'}$ through concatenation, and pass it through a convolutional function as follows:

$$J_r = GCN\left(X_t\right), \tag{5}$$

$$M = J_r \oplus H_t, \tag{6}$$

$$P = \psi_w\left(M\right), \tag{7}$$

where ψ is a 1D convolutional function with parameter $w \in \mathbb{R}^{C' \times C'}$, and M and P both have the same dimension of $\mathbb{R}^{2N \times C'}$. We use the softmax specified by $P_{u,v}$ to calculate the attention score for both spatial characteristics from the GCN output and temporal features from the hidden input in the following way:

$$P_{u,v} = \frac{\exp P_{u,v}}{\sum_{v=1}^{N} \exp P_{u,v}}, u, v \in \{1, 2, \ldots, N\}. \tag{8}$$

After the above step, $P \in \mathbb{R}^{2N \times C'}$ is divided into P_s and P_t, which have the same size, $(P_s, P_t) \in \mathbb{R}^{N \times C'}$. Next, we element-wise multiply P_s and P_t with J_r and H_{t-1} respectively, as follows:

$$a_s = P_s \odot J_r, \tag{9}$$

$$a_t = P_t \odot H_{t-1}, \tag{10}$$

where $\odot$ represents the Hadamard product. Rather than exclusively representing spatial context, a_s also includes temporal information for a specific timestamp, while a_t serves a similar dual role, encompassing both spatial and temporal context. We then add these two context vectors, finally producing J_z as follows:

$$J_z = a_s + a_t; J_z \in \mathbb{R}^{N \times C'} \tag{11}$$

3.4   Memory Updating

Prior traffic forecasting models [28,10,11] often use graph and temporal convolution independently, overlooking the heterogeneities within spatial-temporal data. To tackle this problem, our approach involves a continuously synchronized gating mechanism to update the hidden state H_t, allowing MAA to capture long-range dependencies across both spatial and temporal domains effectively. The update process is defined as follows:

$$g = \sigma\left(W_z \cdot J_z + U_z \cdot H_{t-1}\right), \tag{12}$$

$$r = \sigma\left(W_r \cdot J_r + U_r \cdot H_{t-1}\right), \tag{13}$$

$$\tilde{h} = \tanh\left(W_h \cdot X_t + r * U_h \cdot H_{t-1}\right), \tag{14}$$

$$H_t = g * H_{(t-1)} + (1 - g) * \tilde{h}, \tag{15}$$

where $(W, U) \in \mathbb{R}^{C' \times C'}$ are the learnable parameters and σ is the sigmoid function. Compared with the original memory cell in the GRU [7] that is updated only by current input X_t and previous hidden state H_{t-1}, Our proposed memory cell updates based on the original input X_t, graph convolution J_r, aggregated context vector J_z, and the previous hidden state H_{t-1}, which effectively captures both local and global spatio-temporal dependencies in real-time.

On the other hand, similar to the standard GRU mechanism, we use the final output at the last time step, denoted as $H_T \in R^{N \times C'}$, and process it through two fully connected layers for prediction as follows:

$$\hat{\mathcal{Y}} = \text{Re} \, LU \, (H_T W_1 + b_1) \cdot W_2 + b_2, \tag{16}$$

where $\hat{\mathcal{Y}} \in \mathbb{R}^{N \times T'}$ denotes the prediction of the overall network, and $W_1 \in \mathbb{R}^{C' \times C'}, b_1 \in \mathbb{R}^{C'}, W_2 \in \mathbb{R}^{C' \times T'}$, and $b_2 \in \mathbb{R}^{T'}$ are learnable parameters. Finally, to train the model, we use the loss function as follows:

$$\mathcal{L}(\theta) = \left\| \widetilde{\mathcal{Y}} - \hat{\mathcal{Y}} \right\|_2^2 \tag{17}$$

where $\widetilde{\mathcal{Y}}$ denotes the ground truth and $\hat{\mathcal{Y}}$ denotes the prediction of the model, respectively. **The detail of our training algorithm is provided in the supplementary material.**

4   Experimental Results and Analysis

Datasets. We perform experiments on three publicly available popular benchmark traffic datasets, which are PeMSD4, PeMSD7, and PeMSD8 from California Transportation Agencies [3]. In these datasets, each vertex on the graph represents a sensor node to collect the traffic flow data and the flow data is aggregated to 5 minutes. Thus, each hour has 12 data points in the flow data. We apply zero-mean normalization for preprocessing these datasets.

Baselines. We compare our proposed $STLGRU$ against the following popular as well as SoTA baseline models on spatio-temporal prediction task: 1) Spatial-temporal synchronous modeling mechanism (STSGCN [23]), 2) Spectral Temporal Graph Neural Network for time series forecasting (StemGNN [1]), 3) Time Zigzags at Graph Convolutional Networks (Z-GCNETs [5]), 4) Graph-Wavenet (GW-Net [26]), 5) Pattern Matching Memory Networks (PM-MemNet [15]), and 6) Meta-Graph Convolutional Recurrent Network (Mega-CRN[14]). We use default settings for each baseline when performing comparisons.

Evaluation Metrics. We apply three widely used metrics to evaluate the performance of our model, (1) Mean Absolute Error (MAE), (2) Mean Absolute Percentage Error (MAPE), and (3) Root Mean Squared Error (RMSE).

Implementation Details. We divide all the datasets with a ratio 6:2:2 into training, testing, and validation sets, respectively. We use Adam optimizer with a learning rate of 0.001 and set 16 as the batch size. We conduct experiments

Table 1: The overall performance of STLGRU and baseline methods.

Datasets	Model	15 min			30 min			60 min		
		MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
PeMSD4	STSGCN	19.41	30.69	14.82	21.83	31.33	15.54	23.19	33.65	16.90
	StemGNN	20.24	28.15	13.03	20.68	30.88	14.21	22.92	33.74	15.65
	Z-GCNETs	19.50	28.61	12.78	23.21	30.09	13.12	29.24	32.95	16.14
	GW-Net	18.15	25.24	13.27	22.12	30.62	16.28	21.85	33.70	17.29
	PM-MemNet	18.95	30.16	13.79	20.01	31.47	14.17	26.85	32.14	17.21
	Mega-CRN	19.25	24.88	12.72	19.60	25.96	13.84	22.82	26.33	14.87
	STLGRU(Ours)	**17.59**	**23.24**	**11.02**	**18.73**	**24.61**	**12.85**	**21.05**	**25.41**	**13.87**
PeMSD7	STSGCN	16.17	23.15	16.51	22.19	34.87	19.88	24.26	39.03	20.21
	StemGNN	15.77	22.68	13.97	22.38	33.69	18.99	24.54	34.41	19.45
	Z-GCNETs	15.64	25.19	15.47	23.78	33.64	19.05	26.12	34.78	23.47
	GW-Net	18.74	26.14	16.58	23.64	34.82	24.65	24.15	34.12	29.02
	PM-MemNet	15.25	24.14	15.17	21.12	34.41	19.97	25.39	33.50	21.29
	Mega-CRN	14.23	21.05	13.11	22.86	33.19	18.40	23.55	33.54	19.29
	STLGRU(Ours)	**13.79**	**19.12**	**12.31**	**20.89**	**31.45**	**15.56**	**23.06**	**32.19**	**19.12**
PeMSD8	STSGCN	15.97	23.14	14.79	16.45	24.78	18.47	19.13	29.80	18.96
	StemGNN	15.83	24.93	10.26	15.95	23.88	19.98	24.10	28.13	23.79
	Z-GCNETs	15.76	25.11	10.01	15.64	23.29	16.67	17.55	29.67	19.19
	GW-Net	14.95	24.92	12.79	15.92	24.99	18.97	17.69	28.92	22.67
	PM-MemNet	14.10	22.15	10.41	16.65	24.17	13.77	19.13	28.16	16.68
	Mega-CRN	14.07	22.53	9.54	16.10	22.42	17.97	18.12	27.29	21.05
	STLGRU(Ours)	**13.93**	**20.94**	**8.84**	**15.03**	**22.18**	**12.64**	**16.83**	**26.35**	**14.74**

with our model using non-overlapping time windows in the time series data. The entire experiments are run on a single GPU (Nvidia TITAN RTX). If the test scores of a baseline are unknown for a dataset, we run their publicly available code based on their suggested settings to obtain the results.

Results. Table 1 compares the performance of our model to the baseline models in 15, 30 and 60 minutes traffic forecasting, respectively. As shown in Table 1, our model outperforms all of the baseline models in both long and short-term forecasting. StemGNN, Z-GCNET, STSGCN, GW-Net, and PM-MemNet stack multiple layers of spatio-temporal modules by optimizing a probabilistic graph model. Our proposed method demonstrates improvements over the comparative models, achieving an average increase of 2.7%, 3.1%, and 2.3% in MAE, RMSE, and MAPE, respectively. Mega-CRN, which utilizes trainable adjacency matrices to understand node relationships and employs an encoder-decoder structure to manage traffic data heterogeneity, is also surpassed by our STLGRU model. Overall, STLGRU demonstrates superior performance, exhibiting average improvements of 2.9%, 3.1%, and 2.6% in MAE, RMSE, and MAPE, respectively.

Furthermore, Table 2 presents the maximum memory footprint, computational complexity, and the number of parameters of the baseline models on PeMSD4 dataset. Because we use the same model for each dataset, we present the experimental results with one dataset. Results with additional datasets are provided in Suppl. To compute a model's GPU memory usage during inference, we use the Linux command line "gpustat" with a minibatch size of 1 and with no gradients. We can observe that $STLGRU$ requires the least memory during

Table 2: In-depth comparison of different model efficiency on PeMSD4. We show that STLGRU achieves high memory efficiency with less computational power and parameters in all three datasets. The second best is shown with underline (See Supp. for more results with additional datasets).

Model	Memory (MB)	FLOPs	Parameters
STSGCN	<u>1028</u>	282.24G	<u>550.48K</u>
GW-Net	1031	<u>189.16G</u>	610.25K
StemGNN	1220	378.98G	1.64M
Z-GCNETs	1473	389.49G	1.08M
Mega-CRN	1409	311.97G	669.14K
PM-MemNet	1052	421.49G	1.34M
STLGRU (Ours)	**990**	**77.93G**	**348.54K**

inference than baseline models. It also has the least computation complexity and number of parameters. In Table 2, we present the memory footprint, computational complexity, and the number of parameters used to train *STLGRU*. Our model stands out, as it demands the least memory, with fewer parameters, and ours exhibits reduced computational complexity. This efficiency positions our model as an ideal choice for integration into real-world, low-powered devices.

5   Ablation Study

We verify the effectiveness of *STLGRU* with additional ablation experiments. We dissect our model and focus on two main components: the Gumbel softmax and the memory augmented attention (MAA). As illustrated in Table 3, the absence of MAA leads to a remarkable decline in performance. The role of Gumbel softmax is pivotal in ensuring optimal sparsity within the graph. When we substitute Gumbel softmax with only the learnable embedding matrix, there is a noticeable decline in our model's performance. However, this is not surprising, given that irrelevant connections can reduce the model's ability to capture the dynamic interrelations between nodes accurately.

Table 3: Ablation study for the effectiveness of the memory augmented attention (MAA) and gumble softmax module used in our method.

Gumble Softmax	MAA	Error Score (MAE)
×	×	23.12
×	✓	21.74
✓	×	19.83
✓	✓	**16.83**

Fig. 3: Performance comparison of spatio-temporal models and *STLGRU* with different settings. MAE, RMSE and MAPE of 1-hour forecasting on three datasets are plotted.

Afterwards, as demonstrated in Figure 3, we compare our model against traditional spatio-temporal configurations. Specifically, we evaluate (1) Graph convolution for capturing spatial knowledge and 1D CNN to capture temporal dependencies, (2) Graph convolution for capturing spatial knowledge and LSTM to capture temporal dependencies, (3) Graph convolution for capturing spatial knowledge and vanilla GRU to capture temporal dependencies. From our observations in Figure 3, it is evident that *STLGRU* consistently outperforms other methods significantly. We thus argue that memory-augmented attention can capture more fine-grained spatio-temporal patterns and trace the crucial interdependencies among the road network.

6 Conclusion

In this work, we introduce *STLGRU*, a uniquely lightweight and efficient model for traffic flow prediction task. Our model incorporates a memory module enhanced with attention mechanism, capable of synchronizing spatial correlations within node networks and long-term temporal patterns in a continuous manner. Our experimental results showcase its superior performance across three benchmark traffic prediction datasets while maintaining a significantly reduced computational overhead compared to baseline models. For future work, we plan to adapt *STLGRU* for other spatial-temporal forecasting challenges, and explore how to model spatio-temporal dependencies when long-term data is scarce.

References

1. Cao, D., Wang, Y., Duan, J., Zhang, C., Zhu, X., Huang, C., Tong, Y., Xu, B., Bai, J., Tong, J., et al.: Spectral temporal graph neural network for multivariate time-series forecasting. Advances in neural information processing systems **33**, 17766–17778 (2020)
2. Chen, C., Li, K., Teo, S.G., Zou, X., Wang, K., Wang, J., Zeng, Z.: Gated residual recurrent graph neural networks for traffic prediction. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 485–492 (2019)
3. Chen, C., Petty, K., Skabardonis, A., Varaiya, P., Jia, Z.: Freeway performance measurement system: mining loop detector data. Transportation Research Record **1748**(1), 96–102 (2001)
4. Chen, W., Chen, L., Xie, Y., Cao, W., Gao, Y., Feng, X.: Multi-range attentive bicomponent graph convolutional network for traffic forecasting. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 3529–3536 (2020)
5. Chen, Y., Segovia, I., Gel, Y.R.: Z-gcnets: Time zigzags at graph convolutional networks for time series forecasting. In: International Conference on Machine Learning. pp. 1684–1694. PMLR (2021)
6. Cheng, W., Shen, Y., Zhu, Y., Huang, L.: A neural attention model for urban air quality inference: Learning the weights of monitoring stations. In: Proceedings of the AAAI conference on artificial intelligence. vol. 32 (2018)
7. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014)
8. Deb, T., Sadmanee, A., Bhaumik, K.K., Ali, A.A., Amin, M.A., Rahman, A.: Variational stacked local attention networks for diverse video captioning. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 4070–4079 (2022)
9. Diao, Z., Zhang, D., Wang, X., Xie, K., He, S., Lu, X., Li, Y.: A hybrid model for short-term traffic volume prediction in massive transportation systems. IEEE Transactions on Intelligent Transportation Systems **20**(3), 935–946 (2018)
10. Fang, S., Zhang, Q., Meng, G., Xiang, S., Pan, C.: Gstnet: Global spatial-temporal network for traffic flow prediction. In: IJCAI. pp. 2286–2293 (2019)
11. Guo, S., Lin, Y., Feng, N., Song, C., Wan, H.: Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 922–929 (2019)
12. He, P., Jiang, G., Lam, S.K., Tang, D.: Travel-time prediction of bus journey with multiple bus trips. IEEE Transactions on Intelligent Transportation Systems **20**(11), 4192–4205 (2018)
13. Jang, E., Gu, S., Poole, B.: Categorical reparameterization with gumbel-softmax. arXiv preprint arXiv:1611.01144 (2016)

14. Jiang, R., Wang, Z., Yong, J., Jeph, P., Chen, Q., Kobayashi, Y., Song, X., Fukushima, S., Suzumura, T.: Spatio-temporal meta-graph learning for traffic forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 8078–8086 (2023)
15. Lee, H., Jin, S., Chu, H., Lim, H., Ko, S.: Learning to remember patterns: Pattern matching memory networks for traffic forecasting. arXiv preprint arXiv:2110.10380 (2021)
16. Lee, W.H., Tseng, S.S., Shieh, J.L., Chen, H.H.: Discovering traffic bottlenecks in an urban network by spatiotemporal data mining on location-based services. IEEE Transactions on Intelligent Transportation Systems **12**(4), 1047–1056 (2011)
17. Li, Y., Yu, R., Shahabi, C., Liu, Y.: Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. arXiv preprint arXiv:1707.01926 (2017)
18. Liebig, T., Piatkowski, N., Bockermann, C., Morik, K.: Dynamic route planning with real-time traffic predictions. Information Systems **64**, 258–265 (2017)
19. Lin, Z., Li, M., Zheng, Z., Cheng, Y., Yuan, C.: Self-attention convlstm for spatiotemporal prediction. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 11531–11538 (2020)
20. Mahmud, S., Tonmoy, M., Bhaumik, K.K., Rahman, A.M., Amin, M.A., Shoyaib, M., Khan, M.A.H., Ali, A.A.: Human activity recognition from wearable sensor data using self-attention. arXiv preprint arXiv:2003.09018 (2020)
21. Niloy, F.F., Amin, M.A., Ali, A.A., Rahman, A.M.: Attention toward neighbors: A context aware framework for high resolution image segmentation. In: 2021 IEEE International Conference on Image Processing (ICIP). pp. 2279–2283. IEEE (2021)
22. Niloy, F.F., Bhaumik, K.K., Woo, S.S.: Cfl-net: Image forgery localization using contrastive learning. arXiv preprint arXiv:2210.02182 (2022)
23. Song, C., Lin, Y., Guo, S., Wan, H.: Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 914–921 (2020)
24. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
25. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
26. Wu, Z., Pan, S., Long, G., Jiang, J., Zhang, C.: Graph wavenet for deep spatial-temporal graph modeling. arXiv preprint arXiv:1906.00121 (2019)
27. Yang, Q., Koutsopoulos, H.N., Ben-Akiva, M.E.: Simulation laboratory for evaluating dynamic traffic management systems. Transportation Research Record **1710**(1), 122–130 (2000)
28. Yu, B., Yin, H., Zhu, Z.: Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. arXiv preprint arXiv:1709.04875 (2017)
29. Zhang, J., Wang, F.Y., Wang, K., Lin, W.H., Xu, X., Chen, C.: Data-driven intelligent transportation systems: A survey. IEEE Transactions on Intelligent Transportation Systems **12**(4), 1624–1639 (2011)
30. Zhao, X., Fan, W., Liu, H., Tang, J.: Multi-type urban crime prediction. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 4388–4396 (2022)

Supplementary Material for
STLGRU: Spatio-temporal Lightweight Graph GRU for Traffic Flow Prediction

1 Proposed Training Algorithm for STLGRU

We describe the training algorithm of our proposed *STLGRU* in Algorithm 1.

Algorithm 1 *STLGRU*

1: **Input:** $X = \langle X_{(t-T+1)}, X_{(t-T+2)}, \ldots, X_t \rangle; \quad X \in \mathbb{R}^{N \times T \times 1}$
2: **Output:** $\widetilde{Y} = \langle X_{(t+1)}, X_{(t+2)}, \ldots, X_{(t+T')} \rangle; \quad \widetilde{Y} \in \mathbb{R}^{N \times T \times 1}$
3: **Parameters:** Randomly initialize Θ and hidden state H_{t-1}
4: **for all** T **do**
5:     $X_{t'} \leftarrow X[:, t, :]; \quad X_{t'} \in \mathbb{R}^{N \times 1}$
6:     $X_t = \xi_\theta(X_{t'}); \quad X_t \in R^{N \times C'}$ ▷ followed by eq. 1
7:     $H_{t-1} = STLGRU_\Theta(X_t, H_{t-1})$
8: **end for**
9: $H_t = H_{t-1}$
10: $\hat{Y} = OutputLayer(H_t)$ ▷ followed by eq. 16
11: calculate loss L using eq. 17
12: **return** $\hat{Y}$

---

2 Memory consumption

In this section, we compare our proposed model with existing baselines using PeMSD7 and PeMSD4 datasets.

Table 4: In-depth comparison of different model efficiency. We show that *STLGRU* achieves high memory efficiency with less computational power and parameters in all three datasets. The second best is shown with underline.

Model	PeMSD7			PeMSD8		
	Memory (MB)	FlOPs	Parameter	Memory (MB)	FlOPs	Parameter
STSGCN	1420	384.27G	895.73K	920	88.55G	98.7K
StemGNN	1816	589.94G	1.87M	1111	271.51G	1.22M
Z-GCNETs	1753	442.629G	1.45M	1314	298.26G	987.25K
Mega-CRN	1638	421.85G	1.12M	1312	245.68G	889.79K
GW-Net	1920	497.64G	827K	1037	144.91G	247.63K
PM-MemNet	1267	512.73G	1.64M	934	437.61G	1.07M
STLGRU(Ours)	**1328**	**295.54G**	**634.89K**	**893**	**52.15G**	**79.82K**

3 Prediction visualization

We visually plot the time series alongside its ground truth in Figure 4. This comparative visualization underscores STLGRU's superior predictive capabilities compared to the baseline Mega-CRN.

Fig. 4: Visualization of the predicted traffic flow.

4 Dataset discription

The summary statistics of the key elements of the datasets are shown in Table 5.

Table 5: Dataset Statistics.

Datasets	Nodes	Edges	Timesteps	Periods
PeMSD4	307	340	16,992	2018/01/01 - 2018/02/28
PeMSD7	883	866	28,224	2016/07/01 - 2016/08/31
PeMSD8	170	277	17,856	2016/07/01 - 2016/08/31

Fig. 5: Sensor Distribution of three traffic datasets, where the dots are the traffic-sensor locations.

5 Baseline discription

We compare our proposed *STLGRU* against the following baseline models on spatial-temporal prediction task.

✧ STSGCN: STSGCN captures the complex localized spatial-temporal correlations through a spatial-temporal synchronous modeling mechanism.
✧ StemGNN: StemGNN combines Graph Fourier Transform (GFT) which models inter-series correlations and Discrete Fourier Transform (DFT) which models temporal dependencies in an end-to-end framework.

✧ Z-GCNETs: Z-GCNETs proposes to enhance DL architectures with the most salient time-conditioned topological information of the data and introduce the concept of zigzag persistence into time-aware graph convolutional networks.
✧ GW-Net: Graph-Wavenet developed a novel adaptive dependency matrix and learned it through node embedding which can precisely capture the hidden spatial dependency in the data.
✧ PM-MemNet: PM-MemNet learns to match input data to representative patterns with a key-value memory structure.
✧ Mega-CRN: Meta-Graph Convolutional Recurrent Network (MegaCRN) uses the Meta-Graph Learner incorporating a MetaNode Bank into GCRN encoder-decoder.

6   Evaluation metrics

Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) and Root Mean Squared Error (RMSE) are derived as follows:

$$MAE = \frac{1}{n} \sum_{t=1}^{t=n} |y' - y| \tag{18}$$

$$MAPE = \frac{1}{n} \sum_{t=1}^{t=n} \frac{|y' - y|}{y} * 100\% \tag{19}$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^{t=n} (y' - y)^2} \tag{20}$$