SpoT-Mamba: Learning Long-Range Dependency on Spatio-Temporal Graphs with Selective State Spaces

Jinhyeok Choi[1], Heehyeon Kim[1], Minhyeong An[1] and Joyce Jiyoung Whang[1,*]

[1]*School of Computing, KAIST, Daejeon, Republic of Korea*

Abstract

Spatio-temporal graph (STG) forecasting is a critical task with extensive applications in the real world, including traffic and weather forecasting. Although several recent methods have been proposed to model complex dynamics in STGs, addressing long-range spatio-temporal dependencies remains a significant challenge, leading to limited performance gains. Inspired by a recently proposed state space model named Mamba, which has shown remarkable capability of capturing long-range dependency, we propose a new STG forecasting framework named SpoT-Mamba. SpoT-Mamba generates node embeddings by scanning various node-specific walk sequences. Based on the node embeddings, it conducts temporal scans to capture long-range spatio-temporal dependencies. Experimental results on the real-world traffic forecasting dataset demonstrate the effectiveness of SpoT-Mamba.

Keywords

Spatio-Temporal Graphs, Traffic Forecasting, Selective State Spaces, Random Walks

1. Introduction

The predictive learning methods on time series play a crucial role in diverse applications, such as traffic and weather forecasting. The intricate relationships and dynamic nature of time series are often represented as graphs, specifically spatio-temporal graphs (STGs), where node attributes evolve over time. Recently, spatio-temporal graph neural networks (STGNNs) have emerged as a powerful tool for capturing both spatial and temporal dependencies in STGs [1, 2, 3]. Many of those methods employ graph neural networks (GNNs) to exploit spatial dependencies inherent in the graph structures, integrating them with recurrent units or convolutions to capture temporal dependencies [2, 4, 5, 6, 7, 8, 9]. These approaches have facilitated the capturing of spatio-temporal dependencies within STGs. Despite their remarkable performance in predictive learning tasks, they often face challenges in handling long-range temporal dependencies among different time steps [10, 11].

STGs often exhibit repetitive patterns over both short and long periods, which is critical for precise predictions [12, 13]. Therefore, several methods have adopted self-attention mechanisms of transformer layers [14] rather than recurrent units to enhance their capability in exploiting global temporal information [10, 11, 15]. However, the significant computational overhead and complexity of attention mechanisms are being highlighted as major concerns [10, 12, 16, 17].

Meanwhile, structured state space sequence (S4) models have emerged as a promising approach for sequence modeling with linear scaling in sequence length [18]. Those models take the advantages of recurrent neural networks and convolutional neural networks, enabling them to handle long-range dependencies without relying on attention. However, due to their inability to select information depending on input data, they have shown limited performance.

A recent study has introduced a new S4 model overcoming the issue, named Mamba, which introduces a selection mechanism to filter information in an input-dependent manner [19]. Mamba has demonstrated notable performance over transformers across various types of sequence data, including language, audio, and genomics. In addition, there have been several studies towards replacing the transformer with Mamba in graph transformer frameworks [20, 21, 22].

In this paper, our focus lies on the predictive learning task on STGs, specifically STG forecasting. For STG forecasting, it is vital to capture the evolving behavior of individual nodes over time and how these changes propagate throughout the entire graph. Furthermore, leveraging these dynamics over long spatial and temporal ranges plays a crucial role in dealing with the intricate spatio-temporal correlations in STGs [1, 3, 23]. Building upon these insights and recent advances, we introduce SpoT-Mamba, a new **Sp**atio-**T**emporal graph forecasting framework with a **Mamba**-based sequence modeling architecture. With Mamba blocks, SpoT-Mamba extracts structural information of each node by scanning multi-way walk sequences and effectively captures long-range temporal dependencies with temporal scans. Experiments on the real-world dataset demonstrate that SpoT-Mamba achieves promising performance in STG forecasting. The official implementations of SpoT-Mamba are available at https://github.com/bdi-lab/SpoT-Mamba.

2. Preliminaries

State Space Model (SSM) The state space model (SSM) assumes that dynamic systems can be represented by their states at time step t [18]. SSM defines the evolution of a dynamic system's state with two equations: $\mathbf{h}'(t) = \mathbf{A}\mathbf{h}(t) + \mathbf{B}x(t)$ and $y(t) = \mathbf{C}\mathbf{h}(t) + \mathbf{D}x(t)$, where $\mathbf{h}(t) \in \mathbb{R}^D$ denotes the latent state, $x(t) \in \mathbb{R}$ represents the input signal, $y(t) \in \mathbb{R}$ denotes the output signal, and $\mathbf{A} \in \mathbb{R}^{D \times D}, \mathbf{B} \in \mathbb{R}^{D \times 1}, \mathbf{C} \in \mathbb{R}^{D \times D}$, and $\mathbf{D} \in \mathbb{R}$ are learnable parameters. SSM learns how to transform the input signal $x(t)$ into the latent state $\mathbf{h}(t)$, which is used to model the system dynamics and predict its output $y(t)$.

Discretized SSM To adapt SSM for discrete input sequences instead of continuous signals, discretization is applied with a step size Δ. The discretized SSM is defined in a recurrent form: $\mathbf{h}_t = \overline{\mathbf{A}}\mathbf{h}_{t-1} + \overline{\mathbf{B}}x_t$ and $y_t = \overline{\mathbf{C}}\mathbf{h}_t$, where $\overline{\mathbf{A}}$ and $\overline{\mathbf{B}}$ are approximated learnable parameters using a bilinear method with a step size Δ [18]. The term $\mathbf{D}$ is omitted from the equations as it can be considered as a skip connection. This formulation allows for capturing temporal

STRL'24: Third International Workshop on Spatio-Temporal Reasoning and Learning, 5 August 2024, Jeju, South Korea

[*] Corresponding author.

cjh0507@kaist.ac.kr (J. Choi); heehyeon@kaist.ac.kr (H. Kim); amh0360@kaist.ac.kr (M. An); jjwhang@kaist.ac.kr (J. J. Whang)

dependencies efficiently, resulting in similar computations to those in recurrent neural networks.

Meanwhile, due to its linear time-invariant (LTI) property, SSM can be reformulated as discrete convolution: $\mathbf{y} = \overline{\mathbf{K}} * \mathbf{x}$ and $\overline{\mathbf{K}} \in \mathbb{R}^L = (\overline{\mathbf{C}}\overline{\mathbf{B}}, \overline{\mathbf{C}}\overline{\mathbf{A}}\overline{\mathbf{B}}, \ldots, \overline{\mathbf{C}}\overline{\mathbf{A}}^{L-1}\overline{\mathbf{B}})$, where $\mathbf{x} \in \mathbb{R}^L$ denotes the input sequence, $\mathbf{y} \in \mathbb{R}^L$ denotes the output sequence, $*$ indicates the convolution operation, and L is the sequence length. This representation facilitates parallel training for SSM, thereby enhancing training efficiency.

The recurrent and convolutional representations of SSM for sequence modeling enable parallel training and linear scaling in sequence length. To further enhance the computational complexity of SSM, the structured state space sequence (S4) models have been proposed [18]. S4 models address the fundamental bottleneck of SSM, which involves repeated matrix multiplications, by employing a low-rank correction to stably diagonalize the transition matrix $\mathbf{A}$.

Mamba S4 models have demonstrated remarkable performance in handling long-range dependencies in continuous signal data, such as audio and time series. However, S4 models struggle with effectively handling discrete and information-dense data such as text [19]. This limitation arises from the LTI property inherent in the convolutional form of SSMs. While the LTI property enables linear time sequence modeling for S4 models, it requires that the learnable matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$, as well as the step size Δ, remain unchanged across all time steps. Consequently, S4 models cannot selectively recall previous tokens or combine the current token, treating each token in the input sequence uniformly. In contrast, Transformers dynamically adjust attention scores based on the input sequence, allowing them to effectively focus on different parts of the sequence [14].

To address both the lack of selectivity in S4 models and the efficiency bottleneck in sequence modeling, a recent study introduced a new S4 model called Mamba, which removes the LTI constraints [19]. Mamba incorporates a selection mechanism that allows its learnable parameters to dynamically interact with the input sequence. This mechanism is achieved by modifying the learnable parameters $\mathbf{B}$ and $\mathbf{C}$, as well as the step size Δ, to functions of the input sequence. Therefore, Mamba can selectively recall or ignore information in an input-dependent manner, while maintaining linear scalability in sequence length.

Inspired by the recent advancements in Mamba, we propose a Mamba-based sequence modeling architecture for predictive learning tasks on STGs. Our approach employs a selective mechanism to handle the dynamical changes in STGs, capturing long-range spatio-temporal dependencies. In addition, this allows for addressing computational inefficiencies in transformer-based STGNNs [10, 11, 15].

Spatio-Temporal Graph Forecasting A spatio-temporal graph (STG) is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{X})$, where $\mathcal{V}$ is a set of N nodes, $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is a set of edges, $\mathcal{X} = [\mathbf{X}_1, \ldots, \mathbf{X}_\tau]$ is a sequence of observed data for all nodes at each historical time step, and τ is a length of the sequence. Here, $\mathbf{X}_t \in \mathbb{R}^{N \times D_{\text{in}}}$ denotes the observed data at time step t, where D_{in} denotes the dimension of the input node attributes. STG forecasting aims to predict future observations for T' time steps, given historical observations for the previous T time steps. This is formulated as $[\mathbf{X}_{t-T+1}, \ldots, \mathbf{X}_t] \xrightarrow{f(\cdot)} [\mathbf{X}_{t+1}, \ldots, \mathbf{X}_{t+T'}]$, where $f(\cdot)$ represents the STG forecasting model.

3. Spatio-Temporal Graph Forecasting with SpoT-Mamba

We propose SpoT-Mamba (Figure 1), which captures spatial dependencies from node-specific walk sequences and learns temporal dependencies across time steps leveraging Mamba-based sequence modeling. By utilizing the selection mechanisms of Mamba blocks, SpoT-Mamba can selectively propagate or forget information in an input-dependent manner on both temporal and spatial domain.

Multi-way Walk Sequence In STGs, the temporal sequences for nodes are naturally defined by the time-series data. On the other hand, since the topological structure does not have a specific order, a tailored method is required to define the spatial sequences of nodes in graphs. Hence, we employ three well-known walk algorithms: depth-first search (DFS), breadth-first search (BFS), and random walks (RW), to extract diverse local and global structural information from each node's neighborhood. The walk sequences of length K for node v_i using these walk algorithms are defined as $\mathcal{W}_{BFS}(i)$, $\mathcal{W}_{DFS}(i)$, and $\mathcal{W}_{RW}(i)$, respectively. These node-specific walk sequences are extracted M times to exploit more comprehensive structural information.

Walk Sequence Embedding SpoT-Mamba generates embeddings for node-specific walk sequences by scanning each sequence. Here, SpoT-Mamba performs bi-directional scans through Mamba blocks, which makes the model robust to permutations and captures the long-range spatial dependency of the sequence more effectively [21]. Then, SpoT-Mamba aggregates the representations of node-specific walk sequences with pointwise convolution. This allows for incorporating representations of neighboring nodes to generate walk sequence embedding for each type of walk sequence.

Subsequently, SpoT-Mamba integrates the walk sequence embeddings into a node embedding $\mathbf{w}_i \in \mathbb{R}^D$ using Multi-Layer Perceptron (MLP), where i represents the node index and D denotes the embedding dimension. Rather than simply stacking GNN layers, we employ Mamba-based sequence modeling to generate node embeddings from diverse types of walk sequences. Therefore, our approach effectively captures local and long-range dependencies within the graph by scanning the neighborhood structure of each node.

Temporal Scan with Mamba Blocks We adopt the learnable day-of-week and timestamps-of-day embeddings to capture the repetitive patterns over both short and long periods in STGs [24]. SpoT-Mamba concatenates these embeddings with the node embedding for each time step to effectively model temporal dynamics from historical observations. Then, it performs selective recurrent scans with Mamba blocks across the sequence of embeddings along the time axis, observing changes in temporal dynamics over time. This process helps identify critical portions of the sequence for forecasting and captures periodic patterns in both short- and long-term intervals. Consequently, the results effectively encompass spatio-temporal dependencies, thereby enriching the predictive capabilities.

STG forecasting of SpoT-Mamba Finally, SpoT-Mamba enhances the representations of nodes scanned along the temporal axis by incorporating global information from the entire graph at each time step through transformer layers.

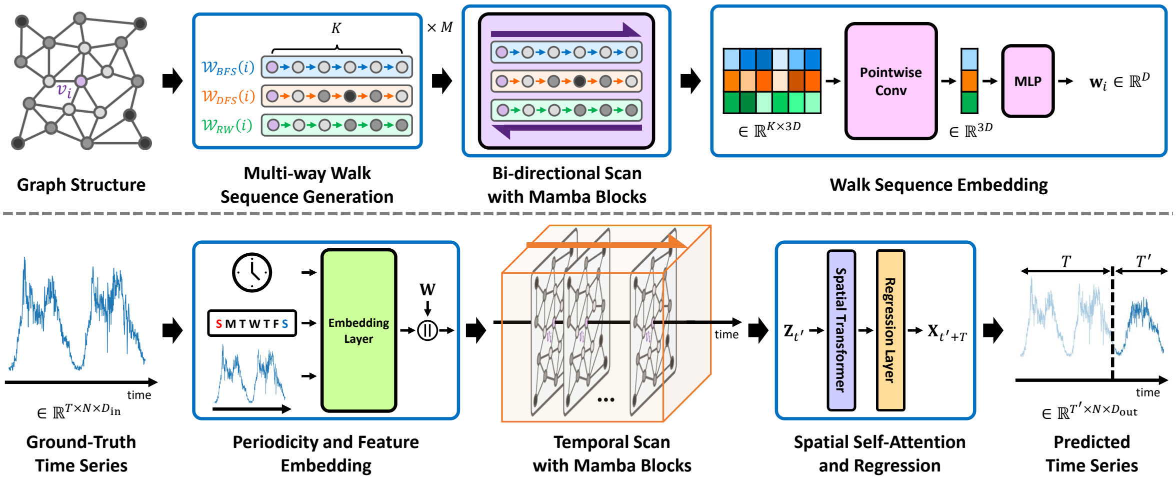
Graph Structure
v_i
K
$\times M$
$w_{BFS}(i)$
$w_{DFS}(i)$
$w_{RW}(i)$
Multi-way Walk Sequence Generation
Bi-directional Scan with Mamba Blocks
$\in \mathbb{R}^{K\times 3D}$
Pointwise Conv
$\in \mathbb{R}^{3D}$
MLP
$\mathbf{w}_i \in \mathbb{R}^D$
Walk Sequence Embedding
time
$\in \mathbb{R}^{T\times N\times D_{\text{in}}}$
Ground-Truth Time Series
S M T W T F S
Embedding Layer
W
Periodicity and Feature Embedding
...
time
Temporal Scan with Mamba Blocks
$\mathbf{Z}_{t'}$
Spatial Transformer
Regression Layer
$\mathbf{X}_{t'+T}$
Spatial Self-Attention and Regression
T
T'
time
$\in \mathbb{R}^{T'\times N\times D_{\text{out}}}$
Predicted Time Series

Figure 1: Overview of SpoT-Mamba. The first row represents the node-specific walk sequence embedding. The second row represents the overall procedure of STG forecasting. $\mathbf{W} \in \mathbb{R}^{N\times D}$ denotes the node embeddings for all nodes in the graph and $\mathbf{Z}_{t'} \in \mathbb{R}^{N\times 4D}$ denotes one of the outcomes from the temporal scan, corresponding to the time step t'.

Then, MLP is applied to forecast the attributes of each node for future time steps. To accurately predict the temporal trajectory while ensuring robustness to outliers that deviate significantly from the expected trajectory, we train SpoT-Mamba utilizing the Huber loss, which is less sensitive to outliers while maintaining the smoothness of the squared error for small errors [24].

4. Experiments

We compare the performance of SpoT-Mamba with state-of-the-art baselines. Additionally, we conduct ablation studies to further demonstrate the effectiveness of SpoT-Mamba.

4.1. Dataset and Experimental Setup

Dataset We evaluate SpoT-Mamba on *PEMS04* [25], a real-world traffic flow forecasting benchmark, following the experimental setup in [24]. *PEMS04* contains highway traffic flow data collected from the California Department of Transportation's Performance Measurement System (PEMS). The nodes represent the sensors, and edges are created when two sensors are on the same road. Traffic data in *PEMS04* is collected every 5 minutes. We set the input and prediction intervals to 1 hour, corresponding to $T = T' = 12$. The statistic of *PEMS04* is shown in Table 1. In the experiments, *PEMS04* is divided into training, validation, and test sets in a 6:2:2 ratio in temporal order.

Baselines We compare SpoT-Mamba with several baselines using various methods, including GNNs and Transformers. For STGNNs, we consider GWNet [9], DCRNN [26], AGCRN [27], GTS [28], and MTGNN [29]. For attention-based methods, we include STAEformer [24], GMAN [30], and PDformer [31]. Additionally, other methods such as HI [16], STNorm [32], and STID [33] are also considered.

Evaluation Metrics We use three standard metrics for traffic flow prediction: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE). MAE is the unweighted mean of the absolute differences between predictions and ground truth values. RMSE is calculated as the square root of the average of the squared differences. MAPE is similar to MAE but normalizes each error by the corresponding ground truth value and expresses it as a percentage. For all three metrics, lower values indicate better performance. Additionally, we rank all the methods used in our experiments across these evaluation metrics and compute their average ranks.

Table 1
Statistic of *PEMS04* dataset.

$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	#Time Steps	Time Interval	Time Range
307	338	16,992	5 min.	01/2018 - 02/2018

Implementation Details SpoT-Mamba is implemented using the Deep Graph Library [34] and PyTorch [35]. For the transformer, we utilize the off-the-shelf transformer encoder available in PyTorch, and for Mamba, we employ the official implementation [19] and apply pre-normalization. We train SpoT-Mamba for 300 epochs using the Adam optimizer [36], with early stopping if there is no improvement over 20 epochs. Additionally, we apply the learning rate decay, reducing the learning rate at the 20th, 40th, and 60th epochs. To determine the optimal hyperparameters for SpoT-Mamba, we conduct a grid search. The grid search covers $M \in \{2, 4\}$, learning rates of $\{0.001, 0.0005\}$), weight decays of $\{0.001, 0.0001\}$, and learning rate decay rates of $\{0.1, 0.5\}$. We fix the feed-forward dimension to 256, $D = 32$, $K = 20$, dropout probability to 0.1, batch size to 32, and the number of layers for Mamba and the transformer to 3. All experiments are conducted using GeForce RTX 3090 24GB.

4.2. Traffic Forecasting Performance

Table 2 shows the traffic forecasting performance of the baselines and SpoT-Mamba on *PEMS04* in terms of MAE, RMSE, and MAPE. Note that we report the baseline results from [24] since we strictly followed the experimental settings described in [24]. For each evaluation metric, the best results are boldfaced, and the second-best results are under-

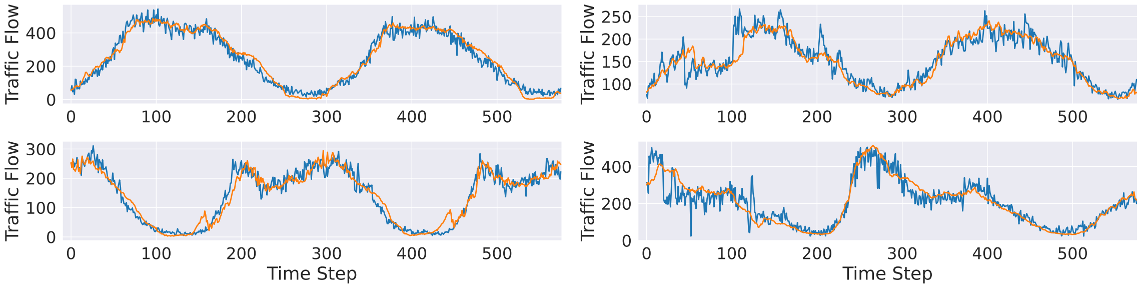

Figure 2: Traffic flow forecasting results for four randomly selected nodes in *PEMS04*. The blue line represents the ground truth, and the orange line denotes the predictions made by SpoT-Mamba.

Table 2
Traffic forecasting performance on *PEMS04*. 'Avg.' denotes the average rank across the three evaluation metrics.

	MAE	Rank	RMSE	Rank	MAPE	Rank	Avg.
HI	42.35	13	61.66	13	29.92	13	13
GWNet	18.53	5	**29.92**	**1**	12.89	6	4
DCRNN	19.63	11	31.26	8	13.59	11	10
AGCRN	19.38	9	31.25	7	13.40	9	8.33
STGCN	19.57	10	31.38	9	13.44	10	9.67
GTS	20.96	12	32.95	12	14.66	12	12
MTGNN	19.17	8	31.70	11	13.37	8	9
STNorm	18.96	6	30.98	6	12.69	5	5.67
GMAN	19.14	7	31.60	10	13.19	7	8
PDformer	18.36	3	30.03	3	12.00	3	3
STID	18.38	4	29.95	2	12.04	4	3.33
STAEformer	**18.22**	**1**	30.18	5	11.98	2	2.67
SpoT-Mamba	18.31	2	30.11	4	**11.86**	**1**	**2.33**

lined. It is observed that SpoT-Mamba consistently achieves high rankings across all metrics: MAE, RMSE, and MAPE, recording the highest average rank among all methods. This suggests the effectiveness of Mamba's selective recurrent scan in modeling spatio-temporal dependency. Compared to other metrics, SpoT-Mamba demonstrates its best performance in MAPE, achieving the highest ranking. Among the baselines, STAEformer [24] shows the most comparable performance to SpoT-Mamba.

4.3. Qualitative Analysis & Ablation Studies

In Figure 2, we visualize the predictions of SpoT-Mamba and the ground-truth time series on *PEMS04*. For this visualization, predictions are made for four randomly selected nodes, starting from an arbitrary time step within a test split, covering 576 consecutive time steps (equivalent to two days). Since we set $T = T' = 12$, multiple predictions are concatenated to represent the duration. When multiple predictions exist at a single time step, we average them. It is observed that the predicted time series closely aligns with the ground-truth data.

Additionally, we conduct the ablation studies of SpoT-Mamba on *PEMS04*. We replace the Mamba blocks used for the walk sequence embedding (indicated as Walk Scan) and for scanning along the time axis (indicated as Temporal Scan) with transformer encoders. Note that when replacing the Mamba blocks for the Temporal Scan with a transformer encoder, we reduce the batch size from 32 to 8 due to Out-of-Memory issues. Results are shown in Table 3. We observed differences in performance depending on which type of scan module is replaced. Specifically, when the Walk Scan is conducted by a transformer encoder, the overall performance of SpoT-Mamba decreases (first and second rows). On the other hand, replacing only the Mamba blocks for the Temporal Scan with a transformer encoder shows negligible performance differences (third row).

Table 3
Ablation studies of SpoT-Mamba on *PEMS04*, varying the types of the walk scan and temporal scan modules.

Walk Scan	Temporal Scan	MAE	RMSE	MAPE
Transformer	Transformer	18.41	30.32	12.12
Transformer	Mamba	18.69	30.17	12.28
Mamba	Transformer	18.29	30.06	11.93
Mamba	Mamba	18.31	30.11	11.86

This disparity can be attributed to the inherent differences between Mamba and Transformer, along with the application of learnable embeddings that impose biases on the sequence. Mamba blocks scan inputs recurrently, inherently considering the sequence order. In contrast, the transformer encoder does not recognize input sequence order by itself. Furthermore, while SpoT-Mamba utilizes learnable embeddings for temporal sequences, i.e., day-of-week and timestamps-of-day, it does not apply such embeddings for walk sequences. As a result, the Transformer encoder struggles to perceive the order in walk sequences, despite performing well with temporal sequences.

5. Conclusion and Future Work

In this paper, we explore STG forecasting and introduce a new Mamba-based STG forecasting model, SpoT-Mamba. SpoT-Mamba utilizes Mamba blocks to scan multi-way walk sequences and temporal sequences. This approach allows the model to effectively capture the long-range spatio-temporal dependencies in STG, enhancing forecasting accuracy on complex graph structures. SpoT-Mamba shows promising results on the real-world traffic forecasting benchmark *PEMS04*. For future work, we will extend SpoT-Mamba to handle graphs with complex relations [37, 38] and evolving graphs [39, 40].

Acknowledgments

This research was supported by an NRF grant funded by MSIT 2022R1A2C4001594 (Extendable Graph Representation Learning) and an IITP grant funded by MSIT 2022-0-

00369 (Development of AI Technology to support Expert Decision-making that can Explain the Reasons/Grounds for Judgment Results based on Expert Knowledge).

References

[1] Z. Diao, X. Wang, D. Zhang, Y. Liu, K. Xie, S. He, Dynamic spatial-temporal graph convolutional neural networks for traffic forecasting, in: Proceedings of the 33rd AAAI Conference on Artificial Intelligence, 2019, pp. 890–897. doi:10.1609/aaai.v33i01.3301890.

[2] D. Cao, Y. Wang, J. Duan, C. Zhang, X. Zhu, C. Huang, Y. Tong, B. Xu, J. Bai, J. Tong, Q. Zhang, Spectral temporal graph neural network for multivariate time-series forecasting, in: Proceedings of the 34th International Conference on Neural Information Processing Systems, volume 33, 2020, pp. 17766–17778. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/cdf6581cb7aca4b7e19ef136c6e601a5-Paper.pdf.

[3] M. Li, Z. Zhu, Spatial-temporal fusion graph neural networks for traffic flow forecasting, in: Proceedings of the 35th AAAI Conference on Artificial Intelligence, 2021, pp. 4189–4196. doi:10.1609/aaai.v35i5.16542.

[4] Y. Chen, I. Segovia, Y. R. Gel, Z-gcnets: Time zigzags at graph convolutional networks for time series forecasting, in: Proceedings of the 38th International Conference on Machine Learning, 2021, pp. 1684–1694. URL: https://proceedings.mlr.press/v139/chen21o.html.

[5] K. Guo, Y. Hu, Y. Sun, S. Qian, J. Gao, B. Yin, Hierarchical graph convolution network for traffic forecasting, in: Proceedings of the 35th AAAI Conference on Artificial Intelligence, 2021, pp. 151–159. doi:10.1609/aaai.v35i1.16088.

[6] B. Lu, X. Gan, H. Jin, L. Fu, H. Zhang, Spatiotemporal adaptive gated graph convolution network for urban traffic flow forecasting, in: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, 2020, pp. 1025–1034. doi:10.1145/3340531.3411894.

[7] J. Ye, L. Sun, B. Du, Y. Fu, H. Xiong, Coupled layerwise graph convolution for transportation demand prediction, 2021, pp. 4617–4625. doi:10.1609/aaai.v35i5.16591.

[8] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, H. Li, T-gcn: A temporal graph convolutional network for traffic prediction, IEEE Transactions on Intelligent Transportation Systems 21 (2020) 3848–3858. doi:10.1109/TITS.2019.2935152.

[9] Z. Wu, S. Pan, G. Long, J. Jiang, C. Zhang, Graph wavenet for deep spatial-temporal graph modeling, in: Proceedings of the 28th International Joint Conference on Artificial Intelligence, 2019, p. 1907–1913. URL: https://dl.acm.org/doi/abs/10.5555/3367243.3367303.

[10] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, W. Zhang, Informer: Beyond efficient transformer for long sequence time-series forecasting, in: Proceedings of the 35th AAAI Conference on Artificial Intelligence, 2021, pp. 11106–11115. doi:10.1609/aaai.v35i12.17325.

[11] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, R. Jin, FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting, in: Proceedings of the 39th International Conference on Machine Learning, 2022, pp. 27268–27286. URL: https://proceedings.mlr.press/v162/zhou22g.html.

[12] S. Liu, H. Yu, C. Liao, J. Li, W. Lin, A. X. Liu, S. Dustdar, Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting, in: Proceedings of the 10th International Conference on Learning Representations, 2022. URL: https://openreview.net/forum?id=0EXmFzUn5I.

[13] G. Lai, W.-C. Chang, Y. Yang, H. Liu, Modeling long- and short-term temporal patterns with deep neural networks, in: Proceedings of the 42th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2018, p. 95–104. doi:10.1145/3209978.3210006.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017, pp. 6000–6010.

[15] S. Guo, Y. Lin, N. Feng, C. Song, H. Wan, Attention based spatial-temporal graph convolutional networks for traffic flow forecasting, in: Proceedings of the 33rd AAAI Conference on Artificial Intelligence, 2019, pp. 922–929. doi:10.1609/aaai.v33i01.3301922.

[16] Y. Cui, J. Xie, K. Zheng, Historical inertia: A neglected but powerful baseline for long sequence time-series forecasting, in: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, 2021, p. 2965–2969. doi:10.1145/3459637.3482120.

[17] R.-G. Cirstea, B. Yang, C. Guo, T. Kieu, S. Pan, Towards spatio- temporal aware traffic time series forecasting, in: Proceedings of the IEEE 38th International Conference on Data Engineering, 2022, pp. 2900–2913. doi:10.1109/ICDE53745.2022.00262.

[18] A. Gu, K. Goel, C. Re, Efficiently modeling long sequences with structured state spaces, arXiv preprint arXiv:2111.00396 (2022). doi:10.48550/arXiv.2111.00396.

[19] A. Gu, T. Dao, Mamba: Linear-time sequence modeling with selective state spaces, arXiv preprint arXiv:2312.00752 (2023). doi:10.48550/arXiv.2312.00752.

[20] C. Wang, O. Tsepa, J. Ma, B. Wang, Graph-mamba: Towards long-range graph sequence modeling with selective state spaces, arXiv preprint arXiv:2402.00789 (2024). doi:10.48550/arXiv.2402.00789.

[21] A. Behrouz, F. Hashemi, Graph mamba: Towards learning on graphs with state space models, arXiv preprint arXiv:2402.08678 (2024). doi:10.48550/arXiv.2402.08678.

[22] L. Li, H. Wang, W. Zhang, A. Coster, Stg-mamba: Spatial-temporal graph learning via selective state space model, arXiv preprint arXiv:2403.12418 (2024). doi:10.48550/arXiv.2403.12418.

[23] Z. Fang, Q. Long, G. Song, K. Xie, Spatial-temporal graph ode networks for traffic flow forecasting, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, pp. 364—-373. doi:10.1145/3447548.3467430.

[24] H. Liu, Z. Dong, R. Jiang, J. Deng, J. Deng, Q. Chen, X. Song, Spatio-temporal adaptive embedding makes vanilla transformer sota for traffic forecasting, in: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, 2023, p.

4125–4129. doi:10.1145/3583780.3615160.

[25] C. Song, Y. Lin, S. Guo, H. Wan, Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting, in: Proceedings of the 34th AAAI Conference on Artificial Intelligence, 2020, pp. 914–921. doi:10.1609/aaai.v34i01.5438.

[26] Y. Li, R. Yu, C. Shahabi, Y. Liu, Diffusion convolutional recurrent neural network: Data-driven traffic forecasting, in: Proceedings of the 6th International Conference on Learning Representations, 2018. URL: https://openreview.net/forum?id=SJiHXGWAZ.

[27] L. Bai, L. Yao, C. Li, X. Wang, C. Wang, Adaptive graph convolutional recurrent network for traffic forecasting, in: Proceedings of the 34th International Conference on Neural Information Processing Systems, 2020, p. 17804–17815. URL: https://dl.acm.org/doi/abs/10.5555/3495724.3497218.

[28] C. Shang, J. Chen, J. Bi, Discrete graph structure learning for forecasting multiple time series, in: Proceedings of the 9th International Conference on Learning Representations, 2021. URL: https://openreview.net/forum?id=WEHSlH5mOk.

[29] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, C. Zhang, 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, 2020, p. 753–763. doi:10.1145/3394486.3403118.

[30] C. Zheng, X. Fan, C. Wang, J. Qi, Gman: A graph multi-attention network for traffic prediction, in: Proceedings of the 34th AAAI Conference on Artificial Intelligence, 2020, pp. 1234–1241. doi:10.1609/aaai.v34i01.5477.

[31] J. Jiang, C. Han, W. X. Zhao, J. Wang, Pdformer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction, in: Proceedings of the 37th AAAI Conference on Artificial Intelligence, 2023, pp. 4365–4373. doi:10.1609/aaai.v37i4.25556.

[32] J. Deng, X. Chen, R. Jiang, X. Song, I. W. Tsang, Stnorm: Spatial and temporal normalization for multivariate time series forecasting, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, p. 269–278. doi:10.1145/3447548.3467330.

[33] Z. Shao, Z. Zhang, F. Wang, W. Wei, Y. Xu, Spatial-temporal identity: A simple yet effective baseline for multivariate time series forecasting, in: Proceedings of the 31st ACM International Conference on Information & Knowledge Management, 2022, p. 4454–4458. doi:10.1145/3511808.3557702.

[34] M. Wang, D. Zheng, Z. Ye, Q. Gan, M. Li, X. Song, J. Zhou, C. Ma, L. Yu, Y. Gai, T. Xiao, T. He, G. Karypis, J. Li, Z. Zhang, Deep graph library: A graph-centric, highly-performant package for graph neural networks, arXiv preprint arXiv:1909.01315 (2020). doi:10.48550/arXiv.1909.01315.

[35] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, Pytorch: An imperative style, high-performance deep learning library, in: Proceedings of the 33th International Conference on Neural Information Processing Systems, 2019, pp. 8024–8035. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf.

[36] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: Proceedings of the 3rd International Conference on Learning Representations, 2015. URL: https://doi.org/10.48550/arXiv.1412.6980.

[37] H. Kim, J. Choi, J. J. Whang, Dynamic relation-attentive graph neural networks for fraud detection, in: Proceedings of 2023 IEEE International Conference on Data Mining Workshops (ICDMW), 2023, pp. 1092–1096. doi:10.1109/ICDMW60847.2023.00143.

[38] C. Chung, J. Lee, J. J. Whang, Representation learning on hyper-relational and numeric knowledge graphs with transformers, in: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2023, pp. 310–322.

[39] X. Huang, Y. Yang, Y. Wang, C. Wang, Z. Zhang, J. Xu, L. Chen, M. Vazirgiannis, Dgraph: A large-scale financial dataset for graph anomaly detection, in: Proceedings of the 36th International Conference on Neural Information Processing Systems, 2022, pp. 22765–22777. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/8f1918f71972789db39ec0d85bb31110-Paper-Datasets_and_Benchmarks.pdf.

[40] J. Lee, C. Chung, J. J. Whang, InGram: Inductive knowledge graph embedding via relation graphs, in: Proceedings of the 40th International Conference on Machine Learning, 2023, pp. 18796–18809.