A Knowledge-Driven Memory System for Traffic Flow Prediction

Binwu Wang[1], Yudong Zhang[1], Pengkun Wang[1], Xu Wang[1], Lei Bai[2(✉)], and Yang Wang[1(✉)]

[1] University of Science and Technology of China, Hefei 230000, China
{wbw1995, zyd2020, pengkun, wx309 }@mail.ustc.edu.cn, angyan@ustc.edu.cn
[2] Shanghai AI Laboratory, Shanghai 200000, China
baisanshi@gmail.com

Abstract. Traffic flow prediction is critical for intelligent transportation systems. Recent studies indicate that performance improvement by designing new models is becoming marginal. Instead, we argue that the improvement can be achieved by using traffic-related facts or laws, which is termed exogenous knowledge. To this end, we propose a knowledge-driven memory system that can be seamlessly integrated into GCN-based traffic forecasting models. Specifically, the memory system includes three components: access interface, memory module, and feedback interface. The access interface based on the attention mechanism and the feedback interface based on the gate mechanism are used to guide the model to extract useful patterns and integrate these patterns into the model to enhance spatiotemporal representation respectively. The memory module is used to learn specific knowledge-based patterns, and this is achieved by constraining the learning process with unsupervised loss functions formulated inspired by exogenous knowledge. We construct three kinds of memory modules driven by different exogenous knowledge: the long-term trend memory to learn periodic patterns, the hierarchical effect memory to capture coarse-grained region patterns, and the representative pattern memory to extract representative patterns. Experiments combined with multiple existing models demonstrate the effectiveness of the memory system.

Keywords: Traffic forecasting · Spatiotemporal data mining · Graph convolutional network.

1 Introduction

Traffic forecasting plays a fundamental role in intelligent transportation systems (ITS) which is beneficial for practical traffic applications. For instance, road traffic speed and occupancy forecasting can provide insights for urban planning, dynamic management of urban traffic, the efficiency of the logistics industry, and route planning public.

Yang Wang is the corresponding author. Lei Bai is the joint corresponding author.

To achieve accurate traffic forecasting, most researchers are devoted to developing complex spatiotemporal learning models. Machine learning methods in this field mainly use time series analysis models, e.g, Auto-Regressive Integrated Moving Average (ARIMA) and Support Vector Regression (SVR), which fail to model complex spatiotemporal correlation among nodes of the traffic network and time points along the temporal dimension. In recent years, with the rise of deep learning, researchers [1–3] introduce various cutting-edge deep-learning models to learn spatiotemporal correlation, and then generated spatiotemporal representation is used as input to decoders (e.g fully connected layers) to predict traffic. For example, [1, 4] utilize convolutional neural networks (CNN) to learn spatial dependencies and combine CNN with time series models (e.g long short-term memory (LSTM) or temporal convolutional network(TCN)) to capture temporal dependencies. Recently, impressed by the promising performance of graph convolutional neural networks (GCN), researchers [5–8] move to integrate GNN into traffic forecasting for capturing dependencies among nodes. For example, STGCN [9] constructs a graph topology based on the road network and then uses GCN for graph representation learning. STSGCN [10] which is a well-designed synchronous model expands GCN into the spatiotemporal dimension to synchronously capture local spatiotemporal correlation with a local spatiotemporal graph.

However, recent studies indicate that the gain of the forecasting performance induced by modifying neural network structures has become marginal [11], and hence it is in great need to seek alternative approaches to further boost the performance of the traffic forecasting models. To this end, we note an overlooked aspect in the field of traffic: exogenous knowledge, which refers to the facts or laws related to traffic and is the external abstraction of the internal features of traffic data. Therefore, a natural idea is to introduce exogenous knowledge to help analyze the evolution laws of traffic networks, which can provide inspiration to learn more comprehensive spatiotemporal correlation. For example, based on the fact that traffic data is periodic, some models [1, 4] integrate different methods to explicitly capture periodic dependencies, which is proven to be effective for modeling more robust temporal dependencies.

In this paper, instead of designing advanced spatiotemporal learning models, we move to investigate another aspect: how to effectively leverage traffic exogenous knowledge to improve the prediction performance of the model, and finally propose a general module, knowledge-driven memory system, which uses the memory as the backbone due to its flexible capability of storing, abstracting and organizing the knowledge into a structural and addressable form. According to the exogenous knowledge, unsupervised loss functions are formulated to constrain the memory system to learn specific patterns, which are termed as knowledge-based patterns. The model can extract these patterns to enhance spatiotemporal representation.

The memory system includes three carefully-designed components: access interface, memory module, and feedback interface. The access interface provides a specific access address based on the query information from the model to guide

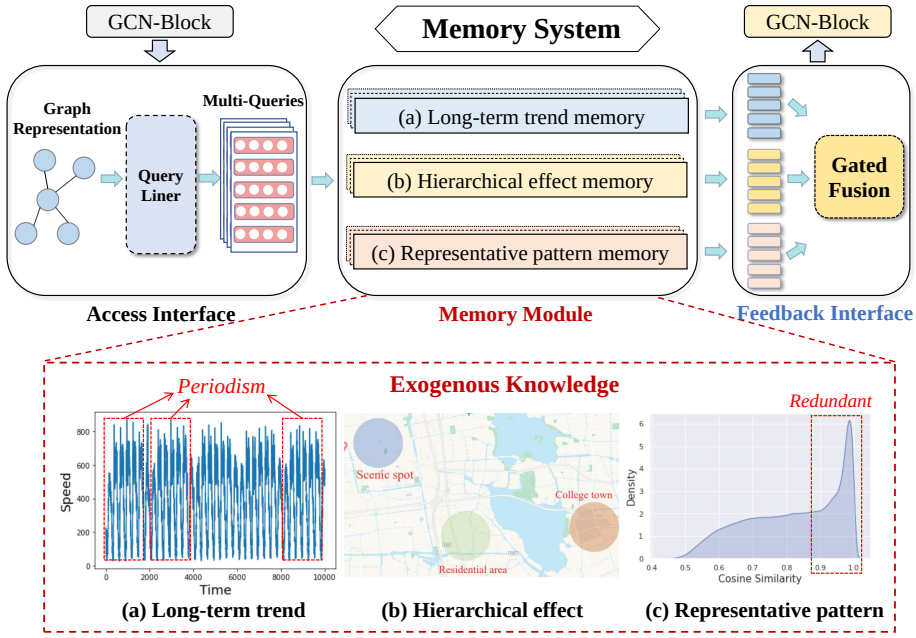


Fig. 1: The details of the memory system. Subfigure (a) illustrates that the traffic is periodic. Subfigure (b) shows the macro level of the transportation system (three hotspots). Subfigure (c) shows the cosine similarity distribution of spatiotemporal patterns, most of which have extremely high similarity.

the model to extract useful patterns, and the memory module is parameterized and end-to-end updated with models to learn knowledge-based patterns according to exogenous knowledge. The feedback interface refers to how to integrate the information from the memory system into the models. Specifically,

Access interface. The access interface is based on the attention mechanism. The advantage is that the model can adaptively extract useful patterns by matching the query vector of the model with the memory module.

Memory module. Three types of traffic exogenous knowledge (as shown in Fig.1) are introduced to enrich the memory system ecology. That is, three kinds of memory modules are configured in the memory system to store the corresponding patterns.

– **Long-term trend memory**. Based on the fact that traffic data is periodic (as shown in Fig.1 (a)), long-term trend memory is used to explicitly store periodic patterns of the traffic network, which can be used to model stronger temporal dependence.

- **Hierarchical effect memory**. Urban traffic network is a hierarchical structure (as shown in Fig.1 (b)), including not only micro-structure where fine-grained roads or nodes are regarded as entities, but also macro-structure with coarse-grained regions gathered by micro-entities as entities. In a coarse-grained region, the associations among nodes may be more intimate. To this end, we propose the hierarchical effect memory to model this effect and learn coarse-grained patterns of the traffic network. Considering the availability of external data (e.g POI), we introduce a graph pooling loss function to constrain the model to adaptively learn a friendly hierarchical structure of the traffic network.
- **Representative pattern memory**. A recent study [12] reveals that traffic patterns of road networks are redundant, and the traffic status of the entire road network can be effectively represented by generalizing a set of representative patterns. Based on this discovery, we propose the representative pattern memory to extract representative patterns of the traffic network.

Feedback interface. To efficiently integrate extracted information from the memory module into the model, we provide a feedback interface based on the gating mechanism to filter out redundant information and achieve efficient information fusion.

In conclusion, we propose a novel memory system driven by exogenous knowledge for traffic forecasting. Our contributions are summarized as follows:

- We investigate leveraging exogenous knowledge to improve the prediction performance of the model and propose a knowledge-driven memory system that can broadly boost the representational power of GCN-based traffic forecasting models.
- We carefully customize three components of the memory system. The access interface based on the attention mechanism and the feedback interface are used to extract knowledge-based patterns from memory modules and integrate these patterns into the models to enhance the spatiotemporal representation. And three kinds of memory modules driven by exogenous knowledge are introduced to learn and store periodic patterns, coarse-grained patterns, and representative patterns respectively.
- The memory system is deployed to diverse traffic forecasting models to evaluate the validity, and experiments on two real-world datasets demonstrate that traffic forecasting models can widely benefit from the memory system.

2 Preliminaries

In this section, we first define some terms that will be used in the problem statement and then formulate the traffic forecasting problem.

Def.1 (Traffic Network) We use a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ to denote a traffic network, where $\mathcal{V}$ is the node (e.g, traffic sensors) set with $|\mathcal{V}| = N$ nodes. $\mathcal{E}$ is

a set of edges representing the connectivity among vertices, and $\mathcal{A} \in R^{N \times N}$ is the adjacency matrix of the graph.

Def.2 (Traffic data) Traffic data is collected from devices (e.g, traffic sensors) deployed on roads. We denote the traffic condition at time step t as a graph signal $X_t \in R^{N \times C}$, where C is the number of traffic conditions of interest (e.g traffic speed, traffic flow, etc.).

Problem formulation.1 (Traffic forecasting) In this paper, traffic forecasting with the memory system can be formulated as: **Input**: a GCN-based traffic forecasting model Ψ, the memory system $\mathcal{M}$, and the observed data of L time steps of graph $\mathcal{G}$, $X = (X_{t-L+1}, X_2, ..., X_t) \in R^{L \times N \times C}$. **Output**: a forecasting function Ψ with the memory system which can effectively infer the traffic data next P time-steps $\Psi(X) = (X_{t+1}, X_2, ..., X_{t+P-1}) \in R^{P \times N \times C}$:

$$\Psi^*, \mathcal{M}^* = \arg\min_{\Psi, \mathcal{M}} \|\Psi(X) - Y\|^2, \tag{1}$$

where Ψ^* and $\mathcal{M}^*$ mean the optimized model function and memory system.

3 Method

In this section, we first introduce the learning process of the GCN-based traffic forecasting model, then show the interaction between the model and the memory system (as shown in Fig.1). Finally, three core components of the memory system are elaborated.

3.1 GCN-based Models for Traffic Forecasting

Recently, researchers move to study GCN-based traffic forecasting models due to the powerful capability of modeling graph structure data. They modify or extend GNNs to extract spatial features and combine GCNs with time series models (e.g RNN or Transformer) to learn spatiotemporal correlation. Finally, the generated spatiotemporal representation is used as input to a decoder (e.g fully connected layers or more complicated designs) to predict future traffic states. Given the spatiotemporal representation $\mathbf{X}$ as input, GCN performs convolution on the graph topology $\mathcal{G}$ and aggregates features from the neighborhood. The calculation process of general GCN can be represented as:

$$\bar{\mathbf{X}} = \sigma\left(\mathbf{W} \cdot \mathcal{F}_g\left(\{\mathbf{X}_v\} \cup \{\mathbf{X}_u, \forall u \in \mathcal{N}(v)\}\right)\right) \tag{2}$$

where $\mathbf{W}$ are learnable parameters. $\mathbf{X}_v$ specifies the representation of node v. $\mathcal{N}(v)$ means neighborhood of node v. $\mathcal{F}_g(\cdot)$ is a aggregate function (such as the mean function) and σ is the activation function. $\bar{\mathbf{X}} \in R^{N \times D_h}$ represents the generated spatiotemporal representation, where D_h is the number of channels.

3.2 The Model with Memory System

The memory system, which includes three components: access interface, memory module, and feedback interface, can be integrated into GCN-based traffic forecasting models to boost the representational power.

Specifically, the spatiotemporal learning model use generated spatiotemporal representation $\mathbf{X}$ as a query vector to retrieve knowledge-based patterns stored in the memory module, which are used to obtain enhanced spatiotemporal representation. The process of interactions between the model and the memory system is as follows:

$$\bar{\mathbf{X}} = \mathcal{F}_m(\mathbf{C}_\star \mathcal{M}_\star; \mathbf{X}) \tag{3}$$

where $\mathcal{M}_\star$ represents one kind of memory module, where $\star \in \{L, H, R\}$ means the long-term trend memory, the hierarchical effect memory, and the representative pattern memory. $\mathbf{C}_\star$ is an access matrix provided by the access interface and records the slot location information that the model should access according to the query vector X. Based on the access matrix, the model extracts stored features from the memory module $\mathcal{M}_\star$. The feedback interface function $\mathcal{F}_m(\cdot)$ is used to integrate the extracted information into the model to obtain enhanced spatiotemporal representation $\bar{\mathrm{X}}$.

Access interface. The access interface is based on the attention mechanism and returns an access matrix $\mathbf{C}$ according to the query vector to guide the model to access the memory module, ensuring that the model can accurately extract useful information and prevent the disturbance of other irrelevant features.

Specifically, we first linearly map the query vector $X^{(l)}$ to a high-dimensional space and the result is denoted as $\mathbf{Q}_\star \in R^{N \times d_q}$. The patterns stored in the memory are treated as key vectors, then, the similarity between the query vector and key vectors can be computed by dot product operation:

$$C_\star(k) = \frac{\exp\left(\langle \mathbf{Q}_\star, \mathcal{M}_\star(k) \rangle\right)}{\sum_{k'=1}^{K} \exp\left(\langle \mathbf{Q}_\star, \mathcal{M}_\star(k') \rangle\right)} \tag{4}$$

where $\mathcal{M}_\star(k)$ means k-th slot of the memory module. To increase the capacity, we model the query vector as a multi-head array and can obtain a similarity matrix sequence $[C_1, ..., C_h] \in R^{N_h \times N \times K}$, where N_h is the number of heads. The access matrix $\mathbf{C}$ can be obtained by aggregating the similarity sequence with convolution operation across heads:

$$\mathbf{C}_\star = \mathrm{softmax}\left(\Gamma_\phi\left(\|_{m=0}^{N_h} C_m\right)\right) \tag{5}$$

where Γ_ϕ means the convolution operator with 1×1 size kernel. With the access matrix, the model can extract patterns stored in the slots:

$$\widetilde{\mathbf{X}} = \mathbf{C}_\star \mathcal{M}_\star \tag{6}$$

Feedback interface. The feedback interface function $\mathcal{F}_m(;)$ is used to integrate the extracted patterns $\widetilde{\mathbf{X}}$ from the memory module into the model to enhance spatiotemporal representation $\mathbf{X}$. In order to filter out redundant information and achieve effective integration, we use the gate mechanism as the feedback interface function $\mathcal{F}_m(;)$. Specifically, we first compute a filter gate $\mathbf{O}$:

$$\mathbf{O} = \sigma\left(\mathbf{W}_g\left[\widetilde{\mathbf{X}}, \mathbf{X}\right] + \mathbf{b}_g\right) \tag{7}$$

where $\mathbf{W}_g$ and $\mathbf{b_g}$ are learnable parameters. Based on the filter gate $\mathbf{O}$, we integrate two representation vectors to obtain an enhanced spatiotemporal representation:

$$\mathcal{F}_m(\mathbf{X}_1; \mathbf{X}_2) = \mathbf{O} \odot \widetilde{\mathbf{X}} + \mathbf{X} \tag{8}$$

where $\odot$ is Hadamard product.

3.3 Knowledge-driven Memory Module

The memory module $\mathcal{M}$ is used to store knowledge-based patterns and initialized as the parameterized matrix, which can be updated end-to-end with the model. To effectively learn these patterns, we formulate the loss function based on exogenous knowledge to constrain the learning of the memory. As mentioned before, we consider three kinds of exogenous knowledge and construct different memory modules. For example, for exogenous knowledge that traffic data is periodic, the long-term trend memory $\mathcal{M}_L \in R^{K_L \times D_m}$ is used to learn periodic patterns, where K_L is the number of slots in $\mathcal{M}_L$ and D_L means the number of pattern channels. Similarly, we construct a hierarchical effect memory $\mathcal{M}_H$ to model macro-regional patterns and a representative pattern memory $\mathcal{M}_R$ to capture representative patterns.

Inspired by three types of traffic exogenous knowledge, we reconstruct the input sequence from different perspectives and design special loss functions to align the mapping matrix and the access matrix provided $\mathbf{C}$ by the access interface, ensuring the memory to learn structural and addressable patterns.

1. Long-term trend memory. Traffic data is considered to be periodic [13, 14], thus, long-term trends play an important role in the traffic forecasting task. Driven by this knowledge, we propose long-term trend memory $\mathcal{M}_L$ to capture periodic patterns of the traffic network.

The periodic patterns of the nodes with close spatial functionality are consistent [10], thus, to improve the learning effect, we first daily traffic patterns of each node into K_L clusters, which can reflect the functional properties of the nodes. And the clustering matrix is denoted as $\mathbf{S}_L \in R^{N \times K_L}$, where $\mathbf{S}_L[i, g] = 1$ means that the daily traffic patterns of node v_i belongs to the g-th cluster. In order to constrain each slot of the memory $\mathcal{M}_L$ to store corresponding long-term trends of a cluster, we propose a clustering loss function which is calculated by the clustering matrix $\mathbf{S}_L$ and the access matrix $\mathbf{C}_L$ of memory $\mathcal{M}_L$:

$$\mathcal{L}_L = \sum_i \sum_j \mathbf{S}_L[i, j] \log\left(\mathbf{C}_L[i, j]\right) \tag{9}$$

2. Hierarchical effect memory. The transportation system is a hierarchical structure that includes not only basic micro-levels (e.g nodes or road networks) but also macro-levels (e.g hot spots) [15]. In a macro region, the correlation between nodes may be closer, thus, modeling the hierarchical effect and learning coarse-grained patterns of the macro region can provide a broader perspective for capturing the spatial correlation among nodes. Some researchers use external information (e.g POIs, land attributes, or population density) to analyze the macro-structure of road networks. However, the information may be not readily available due to privacy policies. We propose a hierarchical effect memory $\mathcal{M}_H$ which can adaptively learn the coarse-grained patterns of the road network without external information.

Specifically, rethinking the access matrix $\mathbf{C}_H \in R^{N \times K_H}$ of hierarchical effect memory $\mathcal{M}_H$, if we set K_H much smaller than the number of nodes N, $\mathbf{C}_H$ can be viewed as the mapping matrix of microscopic nodes to the macroscopic regions, and features stored in the memory $\mathcal{M}_H$ can reflect hierarchical information of the road network. To promote the model to learn friendly coarse-grained patterns in the latent space, we introduce an unsupervised graph clustering loss widely used in deep clustering methods [16–18]. Specifically, we first model auxiliary target distribution $\mathbf{V}$ as an auxiliary which can be computed as:

$$\mathbf{V}[i,j] = \frac{\left(\mathbf{C}_H[i,j]\right)^2 / \sum_i \mathbf{C}_H[i,j]}{\sum_{j'} \left(\mathbf{C}_H[i,j']\right)^2 / \sum_i \mathbf{C}_H[i,j']} \tag{10}$$

The auxiliary target distribution $\mathbf{V}$ can improve the cluster purity by normalizing the contributions. The Kullback-Leibler (KL) divergence between $\mathbf{V}$ and the access matrix $\mathbf{C}_H$ is used as unsupervised loss:

$$\mathcal{L}_H = \mathrm{KL}\left(\mathbf{V} \| \mathbf{C}_H\right) = \sum_i \sum_j \mathbf{V}[i,j] \log \frac{\mathbf{V}[i,j]}{\mathbf{C}_H[i,j]} \tag{11}$$

3. Representative pattern memory. Recently, researchers discovered that the traffic patterns of the entire road network are extremely redundant [12], so a few representative patterns shared by all nodes can effectively prompt the spatiotemporal information of the entire road network. And these representative patterns can help the model better understand the spatiotemporal state of the road network. Thus, we propose a representative pattern memory $\mathcal{M}_R$ to store the high-dimensional representation of representative patterns of the road network.

Specifically, for the road network $\mathcal{G}$, we first calculate the daily average flow vector F of each node. If the sensor records data every five minutes (e.g PeMS system), the shape of F is equal to $(N \times 288)$. Then we split it to obtain the pattern set using a time window, where the length of the time window is equal to the time step of the input sequence L. The pattern set is denoted as $\mathbf{B} \in R^{N \times L_k}$, where $L_k = \lfloor \frac{288}{L} \rfloor$, and this set is proved redundant [12]. We show this fact with PeMS dataset (as shown in Fig.1 (c)), which shows the cosine similarity

distribution between the various patterns in the pattern set $\mathbf{B}$. We can see that pattern set is biased distribution. So we perform cluster-based downsampling and the center vector of each cluster is regarded as a representative pattern. Thus, we use $\mathbf{P} \in R^{N_P \times L}$ to denote the representative pattern set, where N_P means the number of representative patterns.

To retrieve the representative pattern that best matches the input traffic features, We take a pattern $x \in R^{1 \times L}$ of the input sequence $X \in R^{N \times L}$ as an example. First, we compute the cosine similarity between it and representative patterns. Then we select the **top-k** representative patterns with the highest cosine similarity into the candidate set, and the corresponding similarity matrix is denoted as $s_r \in R^{1 \times N_P}$. If a representative pattern is in the candidate set, the corresponding position of s_r is the cosine similarity of the two patterns. Otherwise, it is equal to 0. Similarly, for the entire input sequence X, we get the matching degree matrix $\mathbf{S}_r \in R^{N \times N_P}$.

Each slot in representative pattern memory $\mathcal{M}_R$ is used to store the high-dimensional representation of each representative pattern in $\mathbf{P}$. This is achieved through a loss function which can force the model to align access matrix $\mathbf{C}_r$ and matching degree matrix $\mathbf{S}_r$, ensuring that the model only accesses the slots which store representative patterns matching with the input sequence. Specifically, the following loss function is computed:

$$\mathcal{L}_P = \|S_r - \mathbf{C}_r\|^2 \tag{12}$$

3.4 Loss Function with Memory System

For three kinds of exogenous knowledge, we design different loss functions respectively to constrain the model to store specific features. Thus, the total loss function for deploying the memory system to a traffic forecasting model can be defined as:

$$\mathcal{L} = \left\| \hat{Y} - Y \right\|^2 + \alpha \mathcal{L}_L + \beta \mathcal{L}_H + \mu \mathcal{L}_P \tag{13}$$

where the first part represents the loss between the predicted values and the ground-true values. α, β, and μ are hyperparameters to balance each loss.

4 Experiment

4.1 Experiment Settings and Traffic forecasting Models

Dataset We evaluate the effectiveness of the memory system on two widely used public traffic network datasets, PeMSD3 and PeMSD7. All datasets are collected from the Caltrans Performance Measurement System (PeMS) and aggregated into 5-minutes windows, thus, there are 288 data points per day. The peMSD3 dataset records traffic data from 358 sensors from September 1, 2018, to November 30, 2018. And the PeMSD7 dataset collects monitoring data from 883 sensors from July 1, 2016, to August 31, 2016.

Data Preprocess Linear interpolation is utilized to fill in the missing values in each dataset. Min-max normalization is adopted to normalize the data into the range of $[-1; 1]$ to make the training process more stable. And two datasets are divided into training sets, validation sets, and testing sets according to the ratio of 6:2:2 in chronological order, i.e., the earliest 60% of samples are split into the training set, the subsequent 20% of samples are used for validation, and the last 20% of samples are used for testing. And we use one-hour historical data to predict the traffic data after one hour (ie.L=P=12).

Experiment Settings. We optimize all the models with the Adamw optimizer. The initial learning rates in the PeMSD3 dataset and PeMSD7 dataset are set to 0.008 and 0.005 respectively. And the learning rate decays to 1% of the initial value if the loss on the validation set does not decrease for 15 epochs. The hyperparameters of the models are chosen through a carefully parameter-tuning process on the validation set. The number of heads in the access interface N_h is set to 4. And the number of slots in each memory module (i.e K_L, K_H, and K_R) on the PeMSD3 dataset are equal to 8, 12, and 50 respectively, and they on the PeMSD7 dataset are set to 12, 16, and 64 respectively.

Metrics. Three metrics - Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE) are used to evaluate the prediction performance of the models.

Traffic forecasting models. The memory system is deployed into existing advanced GCN-based traffic forecasting models.

- **STGCN [9]** uses graph convolution and temporal convolution for learning spatial and temporal dependencies, respectively.
- **DCRNN [9]** combines diffusion graph convolution and recurrent units to capturing spatiotemporal correlation.
- **ASTGCN [19]** is a traffic predicting model based on self-attention, which learns dynamic spatiotemporal correlation in a flexible manner.
- **GraphWaveNet [6]** proposes node embedding vectors to construct graph structures, and uses GCN and dilated casual convolution to predict traffic.
- **STGNN [20]** uses GCN to learn spatial correlation with GRU and Transformer to learn global and local temporal dependencies.
- **AGCRN [21]** proposes an adaptive graph learning method for GCN to capture the dynamic features of the traffic road network and combines it with RNN for traffic forecasting.
- **HGCN [15]** designs a hierarchical GCN to learn the hierarchical features of traffic networks.
- **STFGNN [10]** constructs temporal graphs based on DTW algorithm and distance-based spatial graphs to learn spatiotemporal correlation.

Model	PeMSD3		
	MAE	RMSE	MAPE
STGCN	17.55 16.59 + **4.46%**	30.82 29.23 + **5.16%**	17.34 16.67 + **3.83%**
DCRNN	17.98 17.18 + **4.45%**	30.31 29.40 + **3.00%**	18.34 17.73 + **3.32%**
ASTGCN	17.34 16.87 + **2.71%**	29.56 28.78 + **2.64%**	17.21 16.90 + **1.80%**
GraphWaveNet	19.12 18.45 + **3.50%**	32.77 31.21 + **4.76%**	19.37 18.59 + **4.03%**
STGNN	17.24 17.31 - 0.41%	29.62 29.23 + **1.30%**	17.38 17.02 + **2.11%**
AGCRN	15.98 15.54 + **2.75%**	28.25 27.65 + **2.11%**	15.34 15.48 - 0.91%
HGCN	17.21 16.41 + **4.61%**	29.34 27.84 + **5.01%**	17.15 16.56 + **3.45%**
STFGNN	16.77 16.33 + **2.62%**	28.34 27.81 + **1.88%**	16.30 16.18 + **0.74%**

Model	PEMSD7		
	MAE	RMSE	MAPE
STGCN	25.33 24.32 +**4.02%**	39.34 37.65 +**4.30%**	11.21 10.66 +**4.91%**
DCRNN	25.21 24.23 +**3.89%**	38.61 37.09 +**3.94%**	11.82 11.48 +**2.88%**
ASTGCN	24.21 23.44 +**3.18%**	37.87 36.21 +**4.27%**	10.73 10.33 +**3.73%**
GraphWaveNet	26.39 24.96 +**5.41%**	41.50 39.41 +**5.04%**	11.97 11.54 +**3.59%**
STGNN	24.23 24.19 +**0.17%**	38.22 37.61 +**1.60%**	12.01 11.98 +**0.25%**
AGCRN	22.37 21.87 +**2.24%**	36.55 35.98 +**1.56%**	9.12 9.14 -0.22%
HGCN	26.61 25.11 +**5.61%**	40.03 38.59 +**3.69%**	11.57 10.87 +**6.05%**
STFGNN	23.46 22.91 +**2.34%**	36.62 35.80 +**2.79%**	9.21 9.01 +**2.17%**

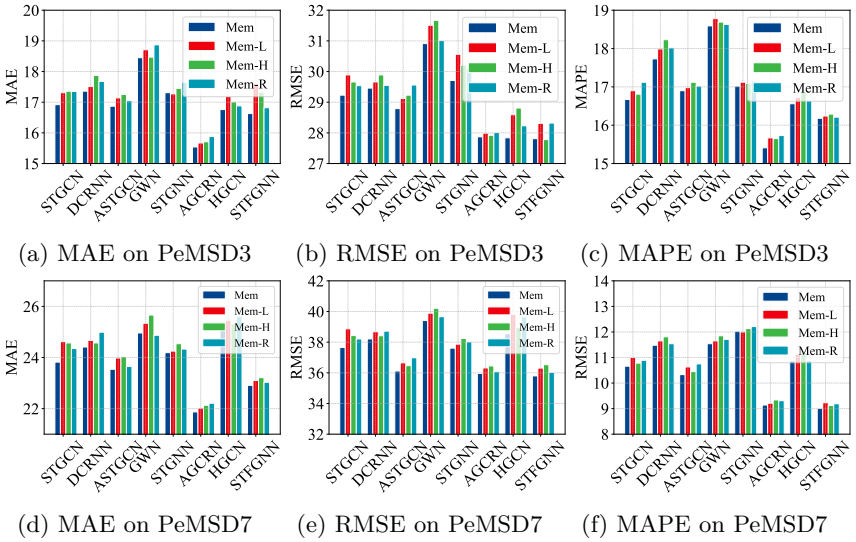
Table 1: The results of the models with the memory system on two datasets. The front and second parts of each metric are the original performance of the models and the performance of the models with the memory system respectively.

4.2 Experiment Result Analysis

The experiment results on the two datasets are shown in Table 1. We observe that the memory module has a positive effect on the predictive performance of the models, because various exogenous knowledge provides insights from multiple perspectives on analyzing the evolutionary patterns of traffic data, and thus knowledge feature stored in the memory system can help models learn comprehensive spatiotemporal correlation and enhance prediction performance of the models. We find that STGCN only constructs the graph structure based on geographic coordinates to model spatial dependencies, and it fails to learn complex spatiotemporal correlation. And the memory system can guide STGCN to learn comprehensive spatiotemporal correlation with exogenous knowledge.

Although DCRNN and AGCRN integrate RNN to capture long-term temporal trends, hierarchical effect memory and representative pattern memory in the memory system can provide them with hierarchical features and representative pattern perspectives of the traffic network, and these features as supplements can boost spatiotemporal representation learning of model. HGCN uses multi-level graphs to extract the hierarchical features of the traffic network, so it achieves better prediction performance than the simple GCN-based model STGCN, and

it still can benefit from the other two memory modules (i.e long-term trend memory and representative pattern memory) in the memory system.



(a) MAE on PeMSD3 (b) RMSE on PeMSD3 (c) MAPE on PeMSD3

(d) MAE on PeMSD7 (e) RMSE on PeMSD7 (f) MAPE on PeMSD7

Fig. 2: Three kinds of memory modules validity analysis.

4.3 Ablation Experiment Analysis

In this section [3], we evaluate the effectiveness of three types of memory modules on two datasets (i.e. long-trend memory, hierarchical effect memory, and representative pattern memory). We remove each kind of memory module respectively, and the variants are denoted as Mem-L, Mem-H, and Mem-R. The experiment results are shown in Fig.2, which show that each kind of memory module is beneficial to the improvement of prediction performance.

For capturing long-term trends, ASTGCN forms periodic sequences as input by sampling data points one week apart. DCRNN and AGCRN rely on the special components RGU to capture long-term trends of traffic data. However, we find that each variant Mem-L, which combines the memory system without long-term trend memory, achieves higher errors than each model with three kinds of memory modules Mem. Because the long-term trend memory module provides more accurate periodic insights of the entire road network by aggregating daily patterns of all nodes.

To model complex spatial correlation among nodes, GraphWaveNet, STGNN, AGCRN, and STFGNN design various methods to generate graph structures, but they only consider microscopic nodes as entities and fail to learn the macroscopic features of the traffic network, and the hierarchical effect memory can

[3] We abbreviate **G**raph**W**ave**N**et as GWN.

complement these features to these models. So the models without the hierarchical effect memory achieve higher errors. For HGCN which is a multi-level GCN-based model to capture the hierarchical structure, the hierarchical effect memory can still improve the prediction performance. It may be that HGCN performs spectral clustering on the adjacency matrix to get the multi-layer graphs, which are static and may not accurately describe the hierarchical structure of the road network. On the contrary, the memory module can adaptively perceive the road network structure.

The experiment shows representative pattern memory module is widely applicable because learned representative patterns can represent the traffic state of the road network and help models more accurately infer the future traffic.

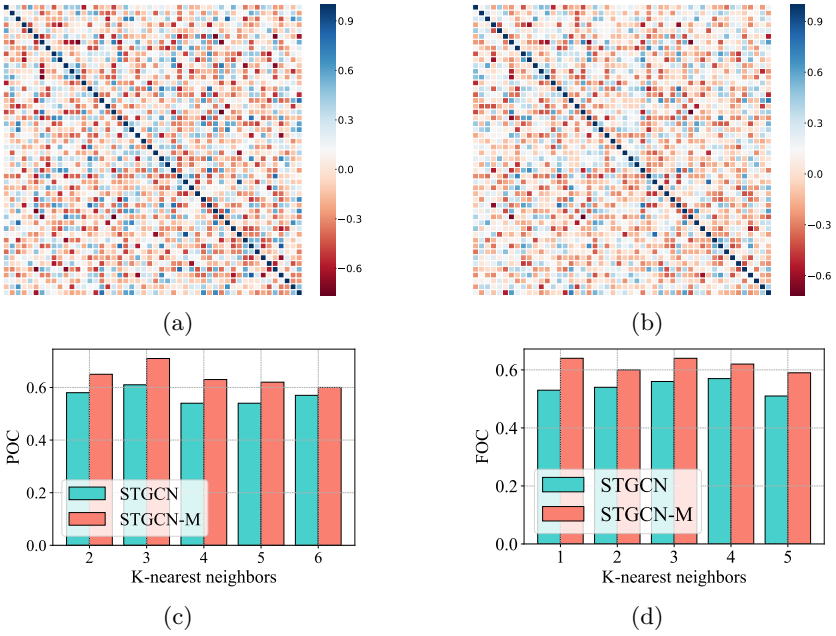


Fig. 3: Subgraph (a) shows the similarity heatmap of the spatiotemporal representation of 100 nodes by STGCN with the memory system. Subgraph (b) shows the similarity heatmap of spatiotemporal representation by only STGCN. Subgraph (c) and Subgraph (d) show the traffic similarity between each node and its k-nearest neighbors in the embedding space.

4.4 Case Study

It is crucial to accurately describe a high-dimensional spatiotemporal representation of nodes for traffic forecasting. A good spatiotemporal learning model should learn node representation which can reflect traffic pattern similarity [22]. In this section, we use STGCN as an example to investigate the effect of the memory system on the spatiotemporal representation learning of the models.

First, we show two heatmaps of node representation learned by STGCN with the memory system (as shown in Fig.3 (a)) and node representation learned by STGCN (as shown in Fig.3 (b)) on the test dataset. We find that the memory system can help the model learn a well-discriminated representation space. That is, the representation of nodes with similar traffic patterns is as close as possible, which benefits the decoder to analyze the representation for predicting traffic.

We further calculate the predicted traffic similarity between each node and its neighbors. Pearson correlation (POC) and the First-order temporal correlation (FOC) [22] are used as similarity functions. We observe that the node embedding of STGCN with the memory system (STGCN-M) shows significant improvement over embeddings of only STGCN. And it indicates that accessing the memory system can effectively learn better traffic-related representation.

4.5 Related Work

Traffic forecasting. In recent years, with the development of deep learning, researchers are devoted to designing advanced deep learning models for traffic forecasting. For example, ST-ResNet [1] exploits convolutional neural networks to mine spatiotemporal correlation for predicting the inflow and outflow of each region. DMVST-Net[4] proposes a local CNN module to learn local spatial dependencies, while LSTM is integrated to learn temporal dependencies. ST-GSP [23] is a semantic encoder composed of ResNet to capture urban-scale spatial correlation and the influence of external factors.

However, CNNs cannot effectively process graph-structured data. Driven by recent advances in graph convolutional neural networks, GCNs are introduced to model spatial dependencies among nodes. For example, T-GCN [24] integrates GCN and GRU to learn spatiotemporal correlation for traffic forecasting. DCRNN [25] proposes a GCN-based layered coupling method for adaptively capturing multi-level spatial correlation of traffic networks. ST-GDN [26] uses diffusion graph convolution to learn local regional geographic dependencies and global spatial semantics. ST-ChebNet [27] uses Chebyshev graph neural network to learn complex topology in traffic networks.

Neural networks with memory. Researchers combine memory modules with neural networks for more powerful learning and reasoning capabilities to solve several challenging tasks such as one-shot learning [28, 29] and question answering [30, 31]. [31] designs a memory based network that designs an inference components with a readable and writable memory module to remember historical supporting information for question answering. [29] designs a memory module that can record network activations of rare events for one-shot learning.

4.6 Conclusion

In this paper, we investigate leveraging exogenous traffic knowledge to improve model prediction performance and propose a knowledge-driven memory system,

which can be easily deployed to GCN-based traffic forecasting models to boost representational power. Three components of the system are carefully designed, and the access interface is based on the attention mechanism to provide precise access information for models. Three memory modules including long-term trend memory, hierarchical effect memory, and representative pattern memory are used to learn and store knowledge-based patterns according to different exogenous knowledge, and the models can enhance spatiotemporal representation by accessing these patterns. And the feedback interface based on the gate mechanism is used to integrate extracted information from the memory system into the model. To evaluate the effectiveness of the memory system, we apply the memory system to existing traffic forecasting models and conduct experiments on two datasets, which demonstrate the effectiveness of the memory system.

Acknowledgements. This paper is partially supported by the National Natural Science Foundation of China (No.62072427, No.12227901), the Project of Stable Support for Youth Team in Basic Research Field, CAS (No.YSBR-005), Academic Leaders Cultivation Program, USTC.

References

1. J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proc. of AAAI*, 2017.
2. L. Liu, Z. Qiu, G. Li, Q. Wang, W. Ouyang, and L. Lin, "Contextualized spatial–temporal network for taxi origin-destination demand prediction," *IEEE Transactions on Intelligent Transportation Systems*, 2019.
3. K.-F. Chu, A. Y. Lam, and V. O. Li, "Deep multi-scale convolutional lstm network for travel demand and origin-destination predictions," *IEEE Transactions on Intelligent Transportation Systems*, 2019.
4. H. Yao, F. Wu, J. Ke, X. Tang, Y. Jia, S. Lu, P. Gong, J. Ye, and Z. Li, "Deep multi-view spatial-temporal network for taxi demand prediction," in *Proc. of AAAI*, 2018.
5. Z. Li, G. Xiong, Y. Chen, Y. Lv, B. Hu, F. Zhu, and F.-Y. Wang, "A hybrid deep learning approach with gcn and lstm for traffic flow prediction," in *2019 IEEE intelligent transportation systems conference (ITSC)*, 2019.
6. Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," *arXiv preprint arXiv:1906.00121*, 2019.
7. X. Zhang, C. Huang, Y. Xu, L. Xia, P. Dai, L. Bo, J. Zhang, and Y. Zheng, "Traffic flow forecasting with spatial-temporal graph diffusion network," 2020.
8. A. Ali, Y. Zhu, and M. Zakarya, "Exploiting dynamic spatio-temporal graph convolutional neural networks for citywide traffic flows prediction," *Neural networks*, 2022.
9. B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," *arXiv preprint arXiv:1709.04875*, 2017.
10. M. Li and Z. Zhu, "Spatial-temporal fusion graph neural networks for traffic flow forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 5, 2021, pp. 4189–4196.
11. Y. Tang, A. Qu, A. H. Chow, W. H. Lam, S. Wong, and W. Ma, "Domain adversarial spatial-temporal network: A transferable framework for short-term traffic forecasting across cities," *arXiv preprint arXiv:2202.03630*, 2022.

12. H. Lee, S. Jin, H. Chu, H. Lim, and S. Ko, "Learning to remember patterns: Pattern matching memory networks for traffic forecasting," in *Proc. of ICLR*, 2021.
13. M. Zheng, Z. Ruan, M. Tang, Y. Do, and Z. Liu, "Influence of periodic traffic congestion on epidemic spreading," *International Journal of Modern Physics C*, 2016.
14. A. Zonoozi, J.-j. Kim, X.-L. Li, and G. Cong, "Periodic-crn: A convolutional recurrent model for crowd density prediction with recurring periodic patterns." in *Proc. of IJCAI*, 2018.
15. K. Guo, Y. Hu, Y. Sun, S. Qian, J. Gao, and B. Yin, "Hierarchical graph convolution networks for traffic forecasting," in *Proc. of AAAI*, 2021.
16. J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *Proc. of ICML*, 2016.
17. A. Razavi, A. Van den Oord, and O. Vinyals, "Generating diverse high-fidelity images with vq-vae-2," *Proc. of NeurIPS*, 2019.
18. E. Aljalbout, V. Golkov, Y. Siddiqui, M. Strobel, and D. Cremers, "Clustering with deep learning: Taxonomy and new methods," *arXiv preprint arXiv:1801.07648*, 2018.
19. S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *Proc. of AAAI*, 2019.
20. X. Wang, Y. Ma, Y. Wang, W. Jin, X. Wang, J. Tang, C. Jia, and J. Yu, "Traffic flow prediction via spatial temporal graph neural network," in *Proc. of WWW*, 2020.
21. L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," *Proc. of NeurIPS*, 2020.
22. Z. Pan, Y. Liang, W. Wang, Y. Yu, Y. Zheng, and J. Zhang, "Urban traffic prediction from spatio-temporal data using deep meta learning," in *Proc. of KDD*, 2019.
23. L. Zhao, M. Gao, and Z. Wang, "St-gsp: Spatial-temporal global semantic representation learning for urban flow prediction," in *Proc. of WSDM*, 2022.
24. L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, "T-gcn: A temporal graph convolutional network for traffic prediction," *IEEE Transactions on Intelligent Transportation Systems*, 2019.
25. J. Ye, L. Sun, B. Du, Y. Fu, and H. Xiong, "Coupled layer-wise graph convolution for transportation demand prediction," in *Proc. of AAAI*, 2021.
26. X. Zhang, C. Huang, Y. Xu, L. Xia, P. Dai, L. Bo, J. Zhang, and Y. Zheng, "Traffic flow forecasting with spatial-temporal graph diffusion network," in *Proc. of AAAI*, 2021.
27. B. Yan, G. Wang, J. Yu, X. Jin, and H. Zhang, "Spatial-temporal chebyshev graph neural network for traffic flow prediction in iot-based its," *IEEE Internet of Things Journal*, 2021.
28. A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "Meta-learning with memory-augmented neural networks," in *Proc. of ICML*, 2016.
29. Ł. Kaiser, O. Nachum, A. Roy, and S. Bengio, "Learning to remember rare events," *arXiv preprint arXiv:1703.03129*, 2017.
30. S. Sukhbaatar, J. Weston, R. Fergus *et al.*, "End-to-end memory networks," *Proc. of NeurIPS*, 2015.
31. J. Weston, S. Chopra, and A. Bordes, "Memory networks," *arXiv preprint arXiv:1410.3916*, 2014.