# Pattern Expansion and Consolidation on Evolving Graphs for Continual Traffic Prediction

Binwu Wang
University of Science and Technology of China
Hefei, China
wbw1995@mail.ustc.edu.cn

Yudong Zhang
University of Science and Technology of China
Hefei, China
zyd2020@mail.ustc.edu.cn

Xu Wang
University of Science and Technology of China
Hefei, China
wx309@mail.ustc.edu.cn

Pengkun Wang
Suzhou Institute for Advanced Research, University of Science and Technology of China
Suzhou, China
pengkun@mail.ustc.edu.cn

Zhengyang Zhou
Suzhou Institute for Advanced Research, University of Science and Technology of China
Suzhou, China
zzy0929@mail.ustc.edu.cn

Lei Bai*
Shanghai AI Laboratory
Shanghai, China
baisanshi@gmail.com

Yang Wang*
University of Science and Technology of China
Hefei, China
angyan@ustc.edu.cn

## ABSTRACT

Recently, spatiotemporal graph convolutional networks are becoming popular in the field of traffic flow prediction and significantly improve prediction accuracy. However, the majority of existing traffic flow prediction models are tailored to static traffic networks and fail to model the continuous evolution and expansion of traffic networks. In this work, we move to investigate the challenge of traffic flow prediction on an evolving traffic network. And we propose an efficient and effective continual learning framework to achieve continuous traffic flow prediction without the access to historical graph data, namely **P**attern **E**xpansion and **C**onsolidation based on **P**attern **M**atching (PECPM). Specifically, we first design a bank module based on pattern matching to store representative patterns of the road network. With the expansion of the road network, the model configured with such a bank module can achieve continuous traffic prediction by effectively managing patterns stored in the bank. The core idea is to continuously update new patterns while consolidating learned ones. Specifically, we design a pattern expansion mechanism that can detect evolved and new patterns from the updated network, then these unknown patterns are expanded into the pattern bank to adapt to the updated road network. Additionally, we propose a pattern consolidation mechanism that includes both a bank preservation mechanism and a pattern traceability mechanism. This can effectively consolidate the learned patterns in the bank without requiring access to detailed historical graph data. Finally, we construct experiments on real-world traffic datasets to demonstrate the competitive performance, superior efficiency, and strong generalization ability of PECPM.

## CCS CONCEPTS

• **Information systems** → **Spatial-temporal systems**; *Data mining*.

## KEYWORDS

Spatiotemporal data mining, Traffic network, Continuous spatiotemporal learning.

**ACM Reference Format:**
Binwu Wang, Yudong Zhang, Xu Wang, Pengkun Wang, Zhengyang Zhou, Lei Bai, and Yang Wang. 2023. Pattern Expansion and Consolidation on Evolving Graphs for Continual Traffic Prediction. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3580305.3599463

## 1 INTRODUCTION

Traffic flow prediction is a crucial task in Intelligent Transportation Systems (ITS) that has the potential to have a significant impact on our daily routines [1, 6, 7, 10, 35, 37]. Accurate traffic prediction can help urban travelers choose appropriate travel routes and traffic managers proactively allocate resource in advance to avoid congestion.

Recently, researchers [15, 17, 21, 30, 31] focus on developing deep learning models to accurately represent the observed data in the high-dimension space because of their powerful capability

*Yang Wang and Lei Bai is the corresponding authors.


*KDD '23, August 6–10, 2023, Long Beach, CA, USA*

ACM ISBN 979-8-4007-0103-0/23/08…$15.00
https://doi.org/10.1145/3580305.3599463

to model nonlinear dependencies. These models generally consist of spatial modules (e.g., Convolutional Neural Network (CNN) or GCN) and temporal modules (e.g., RNN or Temporal Convolution Network (TCN)) to mine spatial and temporal dependencies, respectively. Then the generated node representation is input to a decoder (e.g., fully connected layers or more complex networks) to make predictions. While achieving encouraging success, most of these models are typically evaluated on a short-term dataset (e.g., one month), where the underlying graph structure (i.e., traffic network) is static and the flow distribution of the graph are relatively stable.

However, in practical traffic scenarios, most traffic networks are continuously expanding because new nodes are continuously added as the city develops. Thus, the underlying structure of the graph may change over time. Moreover, the flow distribution of graphs also gradually evolve over time (as shown in Figure.1). Although we can generate representations for new graphs through the inductive framework of GCN, the performance is unpromising because the parameters of the model fail to absorb new knowledge [28]. Hence, an arduous and infrequently explored issue is how to perform precise prediction on an evolving graph.

Due to the high time complexity of the retraining method, researchers move to the continual learning paradigm [3]. Continuous learning, also called lifelong learning, is a technology of continually learning a sequence of tasks as new data becomes available. And general goals are *knowledge expansion and knowledge consolidation* [14, 24]. Specifically, for continuous spatiotemporal learning, three major obstacles should be overcome simultaneously:

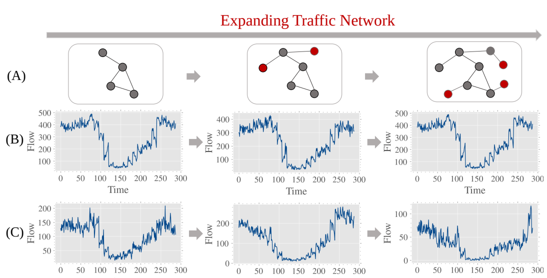



**Figure 1: Expansion and evolution of a traffic network. (A) shows that new nodes are emerging. (B) and (C) show the evolution of the node patterns, where node patterns in (B) are relatively stable, and the patterns of nodes in (C) change significantly. The data of two nodes from the PEMS dataset is respectively recorded in the same time on July 10th, 2011, 2012, and 2013.**

**First, new knowledge from the updated graph should be efficiently expanded into the model.** With the expansion of the traffic network, unknown traffic patterns from incremental nodes are emerging, and some patterns from original nodes are also evolving (as shown in Figure.1 (A) and (C)). It is imperative to learn new and evolved patterns to adapt to the updated graph.

**Second, previous knowledge of the saved model needs to be consolidated.** The naive incremental learning method (i.e., fine-tuning models) is challenged by catastrophic forgetting. This refers to the overriding of previously learned knowledge that is crucial for generating representations of stable nodes where traffic patterns are consistent (as shown in Figure.1(B)). To avoid catastrophic forgetting [8], TrafficStream [3] proposes a strategy based on experience-replay. While effective, this method requires detailed and complete historical graph data [1] of the entire city, which consumes lots of storage resources. Moreover, due to security concerns, storage volatility, and privacy terms, storing complete historical data is relatively challenging [29].

**Third, the generalization ability for new nodes of existing methods is limited.** Due to the possibility of insufficient updated graph data, models may fail to extract sufficient knowledge from observed traffic data. Thus, Accurate prediction of emerging nodes is challenging.

In this work, we investigate continuous traffic prediction without historical data and design a framework under the continuous learning paradigm. The core idea of this framework is to first manage a set of representative traffic patterns for the entire road network, and then iteratively update these representative patterns in streaming traffic data to achieve continual traffic prediction.

Specifically, for effectiveness and efficiency, we propose a continual spatiotemporal learning framework, **P**attern **E**xpansion and **C**onsolidation based on **P**attern **M**atching (PECPM). First, we generalize a set of representative patterns for the road network and a pattern-matching based pattern bank which is a memory module to learn and store higher-dimensional representations of these patterns. The representation generated by the spatiotemporal learning model is pattern-matched with these patterns in the pattern bank to predict future traffic flow. Then, during the continuous learning phase, knowledge expansion and knowledge consolidation can be achieved by managing this pattern bank. Specifically, the pattern extension mechanism is proposed to detect conflict nodes where patterns are significantly inconsistent with previous ones. And then we only use selected conflict nodes with new nodes to fine-tune the learned model for integrating evolved patterns and new patterns into the pattern bank, which is more efficient than retraining the model with data from all nodes. To consolidate the old knowledge, the pattern consolidation mechanism which consists of a bank preservation mechanism and a pattern traceability mechanism is designed to avoid forgetting learned patterns in the pattern bank from the model and data perspective, respectively. The experiment results on two large-scale datasets show that the continuous learning framework PECPM can significantly improve the training efficiency, prediction performance, and generalization ability of the model for continuous traffic prediction. We further deploy PECPM to multiple advanced spatiotemporal learning models to demonstrate the potential of the wide applicability of PECPM.

Our contributions can be summarized as follows:

- *PECPM:* We study the traffic prediction on an evolving graph without the use of complete historical data. And we propose a continuous learning framework with the potential to expand diverse GCN-based models into continuous spatiotemporal learning versions. And this framework integrates a

[1] For continuous traffic prediction, when the traffic network changes, we train (fine-tune) the model using the observed data over time. The data before traffic network changes are defined as historical data.

pattern-matching based pattern bank to manage representative patterns of the traffic network, and then we can iteratively manage these representative patterns in streaming traffic data to achieve continual traffic prediction.

- *Pattern Expansion:* The proposed pattern expansion mechanism only uses new nodes and some conflict nodes to learn new patterns from the updated road networks, which is more efficient than retraining the model with all nodes.
- *Pattern Consolidation:* We design a pattern consolidation mechanism, which includes a bank preservation mechanism and knowledge traceability mechanism, to preserve the learned patterns from the model and data perspective.
- *Generalizability:* Experiments conducted on the large-scale traffic datasets have demonstrated that PECPM exhibits competitive performance and superior efficiency. Notably, it also enhances the generalization ability for emerging nodes.

## 2 PROBLEM FORMULATION

An evolving traffic network is expanding and denoted as $\mathbb{G} = (G_1, G_2, \cdots, G_{\mathcal{T}})$, where $\tau \in \{1, ..., \mathcal{T}\}$ means a relatively long time interval (e.g., a year) [2], and $G_\tau = \{\mathcal{V}_\tau, \mathcal{E}_\tau, A_\tau\}$ represents the graph during the $\tau$-th year, where $\mathcal{V}_\tau$ is the node set with $|\mathcal{V}| = N_\tau$ and $\mathcal{E}_\tau$ is edge set. $A_\tau \in \mathbb{R}^{N_\tau \times N_\tau}$ is the adjacency matrix of $G_\tau$. We use $X_\tau = \left[x_\tau^t \in \mathbb{R}^{N_\tau \times F} | t = 0, \cdots, T_h\right]$ to denote $F$-dimentional signals generated by $N_\tau$ nodes in $\tau$-th year, where $x_\tau^t$ means the observed traffic data at time-step $t$.

**Problem 1 (Continual traffic prediction).** Given an evolving graph $\mathbb{G}$, we aims to sequentially train a function $\mathcal{F}$, which can make prediction for $\mathbb{G}$. Specifically, given the graph $G_\tau$ and observed data $\mathbf{X}_\tau$, the prediction funciton $\mathcal{F}_\tau$ in the $\tau$-th month can accurately predict future traffic flow of all nodes in $G_\tau$:

$$\mathcal{F}_\tau : (X_\tau, G_\tau) \rightarrow \left[x_\tau^{T_h+1}, \cdots, x_\tau^{T_h+T_P}\right] \tag{1}$$

where $T_h$ and $T_P$ are the look-back window size and forecasting window size.

## 3 METHODOLOGY

In this section, we elaborate on the proposed framework based on continual learning for continual traffic prediction (as shown in Figure.2). We first design a pattern bank based on pattern matching to manage knowledge from a pattern perspective. The pattern expansion and consolidation mechanism are used to balance learning new traffic patterns from new data and consolidating the saved patterns learned from previous data. New nodes and existing nodes whose traffic patterns change significantly with their 2-hop neighbors are constructed as a subgraph to mine the influence of network expansion. To consolidate the learned patterns, the bank preservation mechanism and pattern traceability mechanism are imposed on the current training model. The pseudo-code of PECPM is provided in Algorithm.1. The details of PECPM are described in the following.

### 3.1 Pattern Bank Based on Pattern Matching

Researchers [11] have found that traffic patterns of a road network are extremely redundant, we can generalize a set of representative patterns for the entire road network and predict future traffic by matching current traffic patterns with these representative patterns. For the generality of the framework, we combine the advantages of memory modules, namely pattern bank, to manage the spatiotemporal perspective of these patterns. Because the focus of this work is how to perform continuous traffic prediction instead of designing a cutting-edge spatiotemporal learning model, we first introduce a model as an encoder to extract spatiotemporal correlations, and the pattern bank can decode generated representation based on pattern matching for traffic prediction (as shown in Figure.2).

[2]During this period, the traffic network remains stable.

*3.1.1 Spatiotemporal Learning Model.* Current state-of-the-art traffic prediction models are based on GCNs, because of the powerful representation capability of GCNs for graph-structured data. To intuitively describe the proposed framework, we intrudoce a surrogate spatiotemporal graph learning model (**SurSTG**)to capture spatiotemporal correlations, which is similar to STGCN [32].

SurSTG is composed of two blocks, which is similar to STGCN [33]. Each block includes a GCN layer and a gated temporal layer. Given the input of $(l-1)$-th GCN as $h_\tau^{l-1} \in \mathbb{R}^{N_\tau \times D_l}$, where $D_l$ is the dimension of node features. The graph convolution operation can be well-approximated by 1st order Chebyshev polynomial expansion and generalized to high-dimensional GCN as:

$$h_\tau^l = \left(D^{-\frac{1}{2}} A_\tau D^{-\frac{1}{2}} + I_\tau\right) h_\tau^{l-1} W_\tau^l \tag{2}$$

where $W_\tau^l \in \mathbb{R}^{D_\tau^l \times D_\tau^{l+1}}$ is a learnable parameter. $A_\tau$ is the adjacency matrix in the $\tau$-th year, and $I_\tau$ is the corresponding identity matrix, and $D$ is the degree matrix. $h_\tau^l \in \mathbb{R}^{N_\tau \times D_\tau^{l+1}}$ is the output of this GCN layer and will be used as input of a gated temporal layer to capture temporal correlations. After two spatiotemporal blocks, we can get final spatiotemporal representation denoted as $H_\tau^S \in \mathbb{R}^{N_\tau \times D_s}$.

*3.1.2 Pattern Bank Based-on Pattern Matching.* To introduce the pattern matching mechanism into existing spatiotemporal models, we propose a hot-swapped pattern bank as a decoder to manage the spatiotemporal perspective of representative patterns.

**Representative patterns.** We use the data collected from PEMS as a case study. Inspired by [11], we first split the data using a time window of length $T$ time steps (as shown in Figure.3(A)). Then, we calculated the cosine similarity of traffic patterns, and the similarity distribution is shown in Figure.3(B), and the results show that the traffic patterns are highly similar. Thus, to extract representative patterns of the road network, we first get averaged daily traffic data vectors of each node denoted as $\mathbb{C} = \{C^1, ..., C^N\} \in \mathbb{R}^{N \times L}$, where $L$ indicates the number of recorded data points in a day (e.g., $L$ is equal to 288 in total 24 hours with 5-minute intervals in the PEMS dataset), this is because the traffic flow follows a periodic distribution. Then we use a time window of length $T_h$ time steps to slice $\mathbb{C}$, so that we can get an original pattern set denoted as $\widetilde{\mathbb{P}}$, and $|\widetilde{\mathbb{P}}| = N \times \left\lfloor \frac{L}{T_h} \right\rfloor$, which has a biased distribution [11]. So we perform cluster-based downsampling (e.g., K-means) on $\widetilde{\mathbb{P}}$, and the center of each cluster is regarded as a representative pattern. Finally, a representative pattern set $\mathbb{P} \in \mathbb{R}^{K \times T_h}$ is extracted, where $K$ is the number of clusters and also the number of representative patterns.

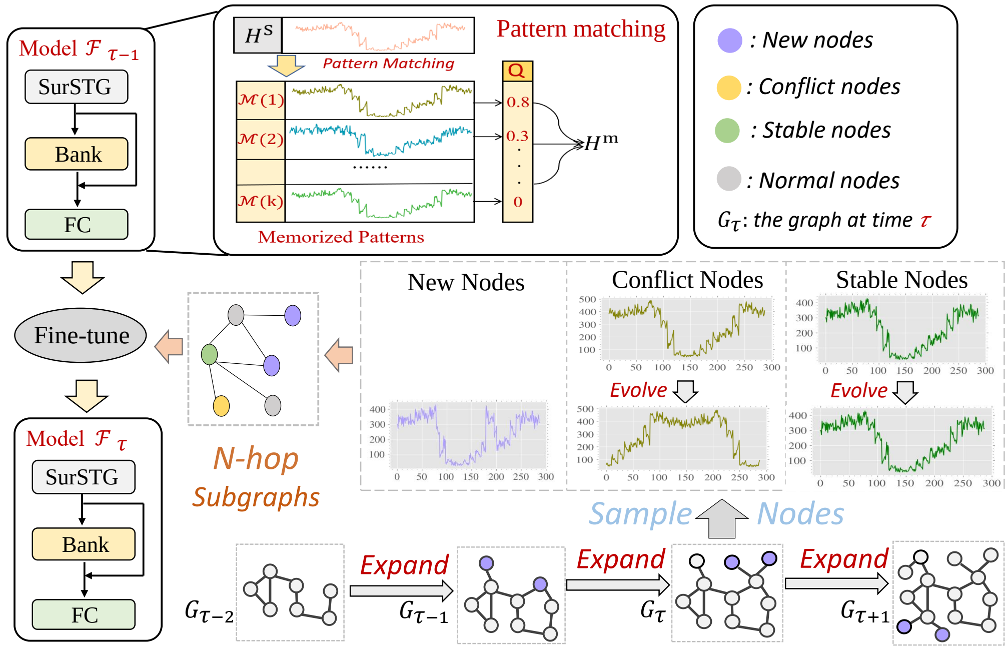
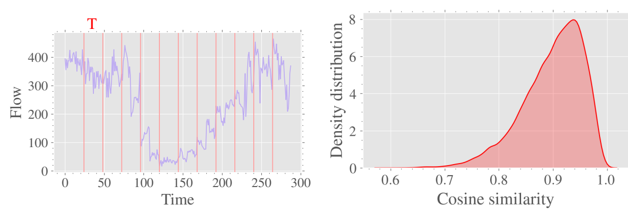



**Figure 2: PECPM for continuous traffic prediction. 'Conflict Node' and 'Stable Node' represent the nodes where the patterns are significantly evolved and stable, relatively.**




**(A): An example of daily patterns. Each part between the red dash lines is denoted as a pattern sliced by a time window. (B): Cosine similarity distribution of patterns.**

**Pattern bank.** Based on the extracted representative pattern set, we first retrieve the best matching patterns with the input patterns in this set to predict future traffic. Specifically, given the input sequence of node $i$ at $t$ time step in the $\tau$-th year $X^t_{\tau,i} \in \mathbb{R}^{1\times T_h}$ and the representative pattern set in the $\tau$-th year $\mathbb{P}_\tau$. First, we compute the cosine similarity between $X^t_{\tau,i}$ and $\mathbb{P}_\tau$. In this way, we obtain a similarity matrix $Q_i \in \mathbb{R}^{1\times K}$. To filter redundant information, we retain $k_c$ elements that have the biggest value (and set the remaining elements to 0) in $Q_i$, and corresponding representative patterns are denoted as matching candidates. Hence we can get a matching degree matrix of all nodes $\mathbf{Q}_\tau \in \mathbb{R}^{N_\tau\times K}$.

In the spatiotemporal perspective, we first construct a memory module $\mathcal{M} \in \mathbb{R}^{K\times D}$ named pattern bank, which is a parameterized matrix. Each row of the memory is the high-dimensional representation of a representative pattern. The representation of input patterns from the spatiotemporal learning model $H^S_\tau$ is used to match with patterns stored in the pattern bank to infer future traffic. Specifically, $H^S_\tau$ is used as a query vector to compute the cosine similarity with the pattern bank:

$$\mathbf{P}_\tau(k) = \text{Softmax}\left(\frac{H^S_\tau\,(\mathcal{M}_\tau(k))^\top}{\sqrt{D}}\right), \tag{3}$$

where $\mathcal{M}_\tau(k)$ means the $k$-th row of the pattern bank in the $\tau$-th year. $\mathbf{P}_\tau \in \mathbb{R}^{N_\tau\times K}$ is an attention score matrix. Then we can extract a feature vector $H^M_\tau$ as:

$$H^M_\tau = \sum_{k=1}^{K} \mathbf{P}_\tau(k) * \mathcal{M}(k) \tag{4}$$

Then we apply skip connection to $H^M_\tau$ with the spatiotemporal representation $H^S_\tau$, finally, a last fully connected layer is used as decoder to predict traffic flow in the next $T'$ time-steps. The prediction loss function in this paper is as follows:

$$\mathcal{L}_r = \left\|\hat{Y}_\tau - Y_\tau\right\|^2 + \mu\left\|\mathbf{P}_\tau - \mathbf{Q}_\tau\right\|^2 \tag{5}$$

where $\mu$ is a hyperparameter to balance two parts. The former part of the loss function measures the loss between the predicted value and the ground-truth value. The latter enforces alignment of the attention matrix $\mathbf{P}_\tau$ with the matching degree matrix $\mathbf{Q}_\tau$, ensuring that the model only accesses matching representative patterns in the bank.

## 3.2 Pattern Expansion and Consolidation Mechanism

To achieve continuous traffic prediction, we propose a pattern expansion and consolidation mechanism. The pattern expansion mechanism allows the model to efficiently mining the influence of network expansion by fine-tuning the saved model with only new nodes and a few conflict nodes. The pattern consolidation mechanism is used to consolidate learned patterns of the model from data and parameter perspectives.

*3.2.1 Pattern Expansion Mechanism.* In modern transportation systems, traffic road networks keep expanding gradually, and unknown patterns of new nodes are constantly emerging, while some traffic patterns of original nodes are also constantly evolving. These new patterns are inconsistent with the patterns learned by the model, which leads to the unsatisfactory prediction performance of the saved model for the updated graph. Thus, it is necessary to integrate these patterns into the saved model for adapting to the new road network. Although fine-tuning the saved model or retraining a new model with the data of all nodes in the updated graph could achieve this goal, this way is not efficient when the graph and model are large. We propose a pattern extension mechanism, which only uses new nodes and conflict nodes to fine-tune the saved model.

First, we extract a new representative pattern set $\mathbb{P}_\tau$ for the updated road network. Then we need to synchronize them into the pattern bank, a efficient method is to select the new nodes and conflict nodes where current patterns are sharply conflict with previous ones. To detect these conflict nodes, we design a detection algorithm based on Wasserstein distance [10].

The core idea is to calculate the similarity of the daily average flow of each node in two consecutive years $\mathbb{C}_{\tau-1}$ and $\mathbb{C}_\tau$, if a node with lower similarity, this means that the traffic patterns of this node change significantly, and Wasserstein distance is adopted as the distance function. For example, the average daily flow vector of node $v_i$ [3] in the $\tau$-th year is denoted as $C_\tau^i \in \mathbb{R}^L = \left(x_\tau^1, ..., x_\tau^l, ..., x_\tau^L\right)$, where $x_\tau^l$ means the $l$-th data point and $L$ is equal to the number of recorded data points in a day (e.g., $L$=288 in the PeMS dataset). We first compute a probability distribution $P_\tau$ for $C_\tau$:

$$p_\tau^t = \frac{\boldsymbol{x}_\tau^t}{\sqrt{\sum_{t=1}^L \left(x_{\tau-1}^t\right)^2}} \tag{6}$$

The probability distribution $P_\tau = \{p_\tau^1, ...p_\tau^t..., p_\tau^L\}$ denotes the proportion of each data point in the total flow. Similarly, we extract the probability distribution $P_{\tau-1}$ of $C_{\tau-1}^i$, and then calculate the distance of two sequences [10]:

$$\begin{gathered} DIS\left(C_\tau^i, C_{\tau-1}^i\right) = \\ \inf_{\gamma\in\Pi[P_\tau,P_{\tau-1}]} \int_v \int_u \gamma(u,v) \left(1 - \frac{x_\tau^u \times x_{\tau-1}^v}{\sqrt{\sum_{t=1}^L \left(x_\tau^t\right)^2} \times \sqrt{\sum_{t=1}^L \left(x_{\tau-1}^t\right)^2}}\right) dudv \\ \text{s.t.} \int \gamma(u,v)du = \frac{x_\tau^u}{\sqrt{\sum_{t=1}^L \left(x_\tau^t\right)^2}}, \int \gamma(u,v)dv = \frac{x_{\tau-1}^v}{\sqrt{\sum_{t=1}^L \left(x_{\tau-1}^t\right)^2}} \end{gathered} \tag{7}$$

Where inf (infimum) means that the solution with the smallest cumulative moving distance from all the schemes to convert one probability distribution $P_{\tau-1}$ to another $P_\tau$. Thus, the similarity is equal to 1- $DIS\left(C_\tau^i, C_{\tau-1}^i\right)$. We select the top 5% of nodes with low similarity and new nodes to construct as a subgraph to integrate new patterns into the saved model.

*3.2.2 Pattern Consolidation Mechanism.* When the model continuously learns new knowledge from the updated graph, one problem is that it may forget learned knowledge if no strategies are taken. To this end, the bank preservation mechanism is used to consolidate the learned patterns in the pattern bank. The pattern traceability mechanism detects the nodes whose patterns are relatively stable to replay learned patterns.

**Bank Preservation Mechanism.** As mentioned in the pattern extension mechanism, we need to continually perform clustering on the data from the updated graph to obtain an adaptive representative pattern set. However, K-means uses a random strategy to initialize the centers of clusters, which may lead to learning a new set significantly different from the previous one. This means that the parameterized patterns stored in the pattern bank need to be significantly updated, which may lead to catastrophic forgetting. Therefore, we propose to restrict random initialization to make K-means clustering yield familiar results. Specifically, the centers of clusters in $\mathbb{P}_{\tau-1}$ are used as the initial position, and then we perform clustering to derive the new one, ensuring that $\mathbb{P}_\tau$ is familiar with $\mathbb{P}_{\tau-1}$.

In addition, during the training process, some abnormal data also causes a large deviation of the parameters. Thus we inherit elastic weight consolidation (EWC) [8] on the model to further preserve parametric patterns stored in the pattern bank (also protect other important parameters). EWC can approximate true posterior distribution for continuous learning parameters by the diagonal precision given by a Fisher information matrix. And it can avoid catastrophic change on learned parameters. Specifically,the previous model $\mathcal{F}_{\tau-1}$ is used to impose constraints on the parameters of the current one $\mathcal{F}_\tau$:

$$\mathcal{L}_\mathrm{s} = \lambda\,\mathrm{vec}\left(\theta^{\tau-1} - \theta^\tau\right)^\top \Omega\,\mathrm{vec}\left(\theta^{\tau-1} - \theta^\tau\right) \tag{8}$$

where the balance factor $\lambda$ is a hyperparameter. $\theta^{\tau-1}$ means the parameters of the model $\mathcal{F}_{\tau-1}$. The operator $vec(\cdot)$ stacks a tensor into the corresponding vector. $\Omega$ is Fisher Information, and we use $\Omega_i$ to denote the importance of $i$-th parameter in the model $\mathcal{F}_{\tau-1}$, It can be measured by first order derivatives of the loss:

$$\Omega_i = \frac{1}{|D_t|}\sum_{x\in D_t} \frac{\partial L_{pr}(\mathcal{F}_{\tau-1}; x)^2}{\partial\theta_i^2} \tag{9}$$

where $|D_t|$ means the size of training data. During the incremental learning phase (e.g.$\tau$ >1), we add this constrain loss to the prediction loss.

**Pattern Traceability Mechanism.** Tracing the sources where old knowledge of the learned model is from and replaying information is an effective way to consolidate knowledge [23]. Conversely to the pattern expansion mechanism, the nodes with high similarity are more stable, whose patterns are relatively consistent with the ones learned by the model. Similar to conflict nodes, we select the top 5% of nodes with high similarity to construct a sub-graph for the saved model. Note that these nodes are selected according to

[3] Node $v_i$ must appear in both the $(\tau - 1)$-th and $\tau$-th year.

the evolutionary degree of their daily flow features, which does not require detailed historical graph data.

### 3.3 Framework Efficiency Study

Traffic prediction models based on GCNs suffers from huge time complexity challenges due to the matrix multiplication in the GNN layer. For a graph $G_\tau$, the time complexity is $O(N_{\tau-1}^2)$. With the expansion of the road network, the time complexity of retraining a new model becomes $O\left((N_{\tau-1}+\Delta N_\tau)^2\right)$, where $\Delta N_\tau$ means the number of new nodes in the $\tau$-th year. Instead, our continual framework PECPM, which use only the new nodes and $d$ sampled nodes (including conflict nodes and stable nodes) to fine-tune the model, could reduce the time complexity of GCN layers to $O\left((\Delta N_\tau + d)^2\right)$. In a real-world traffic road, the nodes of previous large road network are much larger than the new nodes, i.e, $\Delta N_\tau \ll N_\tau$. Thus, our framework is theoretically more efficient.

**Algorithm 1:** PECPM for Continuous Spatiotemporal Learning at time $\tau$>0

**Input:** Observed flow data $X_\tau$, graph $G_\tau$, and model $\mathcal{F}_{\tau-1}$
**Output:** Prediction model functions $\mathcal{F}_\tau$
1 Initialize the parameters of $\mathcal{F}_\tau$ with $\mathcal{F}_\tau - 1$
2 Initialize $\mathbb{P}_{\tau-1}$ as centers of the clusters, and perform clustering to get new representative pattern set $\mathbb{P}_\tau$
3 Sample new nodes, conflict nodes, and stable nodes with their 2-hop neighbors to construct a subgraph
4 Use this subgraph to fine-tune $\mathcal{F}_\tau$

## 4 EXPERIMENTS

In this section, we evaluate the overall performance and efficiency (Subsection 4.2) of our framework. Then, we separately evaluate the effects of learning new knowledge (Subsection 4.3) and consolidating old knowledge (Subsection 4.4). We then construct ablation experiments (Subsection 4.5) and hyperparameter sensitivity experiments (Subsection 4.6). Finally, to verify the general applicability of PECPM, we provide additional experimental analysis: (1). Experiments on another dataset (Subsection 4.7), (2) PECPM with other spatiotemporal learning models (Subsection 4.8).

**Table 1: The details of the evolving traffic network.**

| Year | 2011 | 2012 | 2013 | 2014 | 2015 | 2016 | 2017 |
|---|---|---|---|---|---|---|---|
| Nodes | 655 | 715 | 786 | 822 | 834 | 850 | 871 |
| Edges | 1577 | 1929 | 2316 | 2536 | 2594 | 2691 | 2788 |

### 4.1 Experimental Setup

**Dataset.** We evaluate the proposed framework PECPM on a real-world dataset PEMS3-Stream [3], where the traffic network is extending, and PEMS3-Stream is collected by California Transportation Agencies (CalTrans) Performance Measurement System (PeMS) from July 10th to August 9th from 2011 to 2017. Tbale.1 shows the details of the evolving traffic network. We follow standard protocol and split the dataset into training set, validation set, and test set in chronological order with the ratio of 6:2:2 [12].

**Experiment Setting.** We use data from the past 12 time steps to predict the next 12 time steps (ie. $T_h = T_p = 12$), and then calculate the average metrics for three granularities (i.e. 15 minutes, 30 minutes, and 60 minutes). All the models are optimized by the Adamw [19] optimizer with a learning rate of $10^{-3}$ and batch size of 128. The maximum epoch is set to 100 with early stop strategy. We only use the training dataset to extract representative pattern set in every year and the representative pattern set size is 50 (ie. $|\mathbb{P}| = K$=50). The most similar 5 representative patterns are selected as the candidates (ie. $k_c$=5). Two loss balance factors $\mu$ and $\lambda$ are 50 and 0.001 respectively. Three metrics are used in the experiments: (1) Mean Absolute Error (MAE), (2) Root Mean Square Error (RMSE), and (3) Mean Absolute Percentage Error (MAPE).

**Baselines.** The baselines include continual training strategy and retrained training strategy.

- GRU [4]: Gated Recurrent Unit (GRU) is a variant of RNN using a gating mechanism. We train a new GRU model with all training data every year.
- SurSTG: Surrogate spatiotemporal learning model (SurSTG) is similar to STGCN [33]. We retrain a new SurSTG by using the data of all nodes every year.
- SurSTG-Retrain: We use the training data of all nodes each year are used to retrain a SurSTG (STGCN), which is initialized with the models trained at the last year.
- SurSTG-Static: We use data in the first year (i.e., 2011) to train a SurSTG model. And the trained model without further training is used to directly predict traffic flow after 2011.
- SurSTG-Expand: From 2012 to 2016, we use only **N**ew **No**des to fine-tune the saved SurSTG for expanding new knowledge every year.
- SurSTG-TrafficStream(SurSTG-TS) [3]: TrafficStream is a continual learning strategy based on Jensen-Shannon divergence. In 2012-2016, We continually train SurSTG with TrafficStream.
- PECPM: Our framework is a continual learning method, which only use new nodes, conflict nodes, and stable nodes to fine-tune the saved SurSTG from 2012 to 2016.

### 4.2 Performance Analysis on All Nodes

We first analyze the prediction performance of the models for all nodes, and the average MAE, RMSE, and MAPE in seven years are shown in Table.2. And Figure.4 shows the performance of the models every year.

As can be observed, GRU achieves the largest errors because it fails to capture spatial correlations. SurSTG-Retrain which uses all nodes to fine-tune the saved model achieves better performance than SurSTG, this indicates that old knowledge is beneficial to improve the performance of the model. However, SurSTG-Retrain and SurSTG are extremely inefficient compared with continual methods, since they both access the data of all nodes for training (or fine-tuning) every year. PECPM achieves the best prediction performance compared with the other continual training methods (i.e., SurSTG-Static, SurSTG-Expand, and SurSTG-TS), which is also

Table 2: Prediction performance of the models for all nodes.

| Model | | 15 min | | | 30 min | | | 60 min | | | Time |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | MAE | RMSE | MAPE | MAE | RMSE | MAPE | MAE | RMSE | MAPE | Total(s) |
| Retrained | GRU | 14.71 | 25.01 | 20.36 | 17.15 | 29.02 | 22.76 | 19.98 | 29.78 | 26.29 | 5630.17 |
| | SurSTG | 12.34 | 20.31 | 16.93 | 13.74 | 22.87 | 18.51 | 15.28 | 25.76 | 21.36 | 8891.32 |
| | SurSTG-Retrain | 12.13 | 19.93 | 16.78 | 13.64 | 22.36 | 18.43 | 15.19 | 25.53 | 21.10 | 8504.67 |
| Continual | SurSTG-Static | 13.37 | 21.10 | 28.35 | 14.78 | 23.54 | 30.88 | 16.32 | 27.20 | 34.28 | 1208.98 |
| | SurSTG-Expand | 12.98 | 21.01 | 18.12 | 14.97 | 24.63 | 18.78 | 16.91 | 29.32 | 21.12 | 2296.43 |
| | SurSTG-TS | 12.29 | 20.22 | 16.35 | 13.93 | 22.74 | 17.67 | 15.45 | 25.64 | 19.87 | 4321.85 |
| | PECPM | **11.81** | **19.39** | **15.91** | **13.41** | **22.17** | **17.07** | **14.76** | **25.03** | **18.30** | 3035.21 |

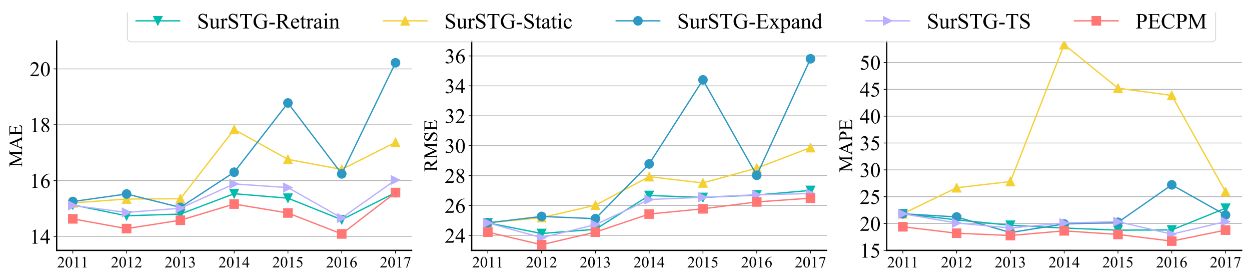



Figure 4: Prediction performance for all nodes in terms of MAE, RMSE, and MAPE in consecutive years.

better than the retraining methods (SurSTG-Retrain), however, it takes only a third of the training time to the retraining methods. SurSTG-Static which directly predicts flow using SurSTG trained in 2011 achieves the worst performance because of failing to learn the new patterns of the updated road network. Thus, SurSTG-Expand, which only uses new nodes to fine-tune the model every year for learning new patterns, achieves unpromising prediction because it fails to consolidate the learned knowledge. Especially when the patterns of previous nodes are inconsistent with new nodes (e.g., 2016), the prediction errors of SurSTG-Static and SurSTG-Expand increase significantly. Although, SurSTG-TS as an unfair baseline uses more information (i.e. detailed and complete historical data in the past year), PECPM still outperforms it in prediction performance and efficiency.

In short, the results show that PECPM achieves excellent performance and high efficiency for continual traffic flow prediction.

### 4.3 Framework Generalization Analysis

It is a priority for traffic managers to predict future traffic conditions at new nodes for developing new traffic strategies, but achieving accurate predictions is a challenge due to the limited data available from these nodes, which depends on the generalization ability of models to learn new knowledge from the updated network. The average of the three metrics over the six-year period (i.e., 2012-2017) on the new nodes is shown in Table.3.

The SurSTG-Static model, which directly predicts traffic states using a trained model from 2011, exhibits the highest errors due to its inability to absorb new knowledge in the updated road network, and learned knowledge is no longer applicable to new nodes. SurSTG-Expand achieve better performance in MAE and RMSE for new nodes than SurSTG-TS and SurSTG-Retrain because it focuses only on the patterns of new nodes using the data of new nodes every year to fine-tune the model. It also reveals us that achieving a balance between consolidating learned patterns and learn new patterns remains a challenge.

However, PECPM significantly outperforms the other models and achieves a balance of integrating new knowledge and consolidating old knowledge. The performance of PECPM is significantly higher than that of SurSTG+TS in terms of MAE and RMSE. However, we observe that the gap between PECPM and other baselines in MAPE is not as large as that in the other two metrics. Considering that MAPE is naturally sensitive to small values, we further ablate the performance for small data in new nodes. We compute the average MAPE of the value between 0 and 5 in six years as an example and the time granularity is 15 minutes, the results of PECPM are 112.43 larger than the 110.13 of SurSTG-TS. This means PECPM has deficient prediction performance for these small values which are not very crucial for the traffic system.

Table 3: Average prediction performance of the models for new nodes.

| Model | 60-min | | |
|---|---|---|---|
| | MAE | RMSE | MAPE |
| SurSTG-Retrain | 12.43 | 21.03 | 40.24 |
| SurSTG-Static | 14.64 | 23.48 | 79.59 |
| SurSTG-Expand | 12.26 | 20.47 | 39.57 |
| SurSTG-TS | 14.89 | 26.74 | 26.96 |
| PECPM | **10.47** | **16.68** | **24.32** |

## 4.4 Performance Analysis on Original Nodes

We further analyze the prediction performance on the original nodes to evaluate the effectiveness of each method in consolidating previous knowledge (as shown in Table. 4).

SurSTG-Expand, which only uses the data of new nodes to fine-tune the saved SurSTG, exhibits inferior performance compared to TrafficStream and PECPM which both emphasizing preserving prior knowledge. Especially when the patterns of new nodes are significantly different from the patterns of original nodes (e.g., 2017), as shown in Figure 4, because SurSTG-Expand only focuses on learning new patterns and suffers from catastrophic forgetting, thus it achieves the worst performance in 2017. This highlights the necessity of consolidating knowledge. SurSTG-Retrain achieves lower errors than SurSTG-TS due to using all the data to retrain a new model. Compared to TrafficStream (SurSTG-TS), PECPM uses a pattern bank explicitly to store important patterns and adopts two strategies (i.e., bank preservation mechanism and pattern traceability mechanism) to further consolidate learned patterns, thus it achieves better prediction performance for original nodes and surpasses the retraining method SurSTG-Retrain. Even in the most challenging years, PECPM still achieves competitive performance.

**Table 4: Average prediction performance of the models for original nodes.**

| Model | 60-min | | |
|---|---|---|---|
| | MAE | RMSE | MAPE |
| SurSTG-Retrain | 15.96 | 26.32 | 21.74 |
| SurSTG-Static | 17.02 | 28.75 | 36.54 |
| SurSTG-Expand | 17.98 | 30.88 | 24.69 |
| SurSTG-TS | 16.23 | 26.57 | 22.08 |
| PECPM | **15.51** | **25.89** | **19.67** |

## 4.5 Ablation Experiment Study

We evaluate the effectiveness of each component of PECPM by removing each main component separately. The prediction performance of different variants in 60 minutes is shown in Figure.5. w/o Bank means PECPM without the pattern bank based on pattern matching and the bank preservation mechanism, which is less effective, and it indicates that explicitly storing representative patterns can help models integrate new patterns and consolidate old knowledge. w/o PT and w/o MP respectively refer to training the model without a bank preservation mechanism and pattern traceability strategy. Their performance is worse than PECPM, which demonstrates that two strategies can avoid the forgetting problem. w/o PE means that we only use new nodes (without conflict nodes) to expand new patterns into the model, which achieves higher errors than PECPM, this suggests that only integrating the patterns of new nodes into the model is not enough. In contrast, PECPM has lower errors due to learning both new and evolved patterns.

## 4.6 Hyperparameter Sensitivity Study

In this section, we investigate the sensitivity of the hyperparameter $K$, which denotes the number of representative patterns set

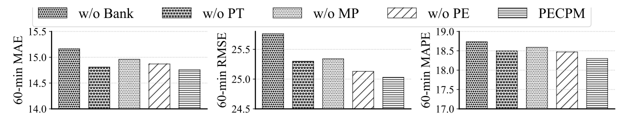



**Figure 5: Ablation experiment of component effectiveness.**

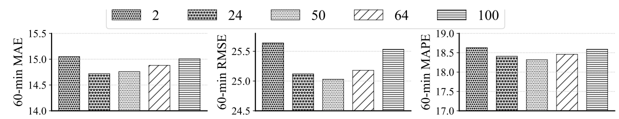



**Figure 6: Sensitivity analysis of hyperparameters $K$.**

$K$ (equal to $|\mathbb{P}|$). The prediction performance with different $K$ is shown in Figure 6. We can observe that our model achieves the best performance when $K$=50. If $K$ is too small, the pattern bank would fail to provide enough information. Conversely, too large $K$ means that there will be an excessive number of patterns in the bank that hinder the ability to capture representative patterns of the road network.

## 4.7 Experiment Study on Another Dataset

In order to investigate the performance of PECPM more broadly, we construct another dataset as a supplement: NYC-BikeStream which is collected from the New York sharing bike service system and records station-level bike flow from 2014 to 2018. We regard one station as a node and construct a graph based on the geographic coordinate of the stations. Because the sharing-bike system continues to expand and new stations are built, the graph in the NYC-BikeStream dataset is constantly expanding, and the number of nodes each year is shown in Table.5. We use the past 6-hour observed data to predict bike flow in the next hour.

The experimental results are reported in Table.6. As can be observed, SurSTG-Retrain achieves lower errors than SurSTG, and this proves that leveraging old knowledge is beneficial for improving prediction performance. Our proposed framework PECPM still maintains excellent performance on the NYC-BikeStream dataset. It achieves lower errors than the continuous learning framework TrafficStream (SurSTG-TS) and the retrained method SurSTG-Retrain while maintaining high training efficiency.

**Table 5: The number of stations in different years of the NYC-BikeStream dataset.**

| Year | 2014 | 2015 | 2016 | 2017 | 2018 |
|---|---|---|---|---|---|
| Stations | 526 | 564 | 612 | 657 | 689 |

## 4.8 PECPM with other SurSTGs

In this section, we combine PECPM with two additional spatiotemporal learning models, SGCN [3] and STSGCN [25]. The experiment results on the PEMS3-Stream dataset are reported in Table.7. Compared with other learning methods, PECPM still achieves the best prediction performance while maintaining high efficiency. Because

**Table 6: Average prediction performance of STGCN with PECPM on NYC-BikeStream Dataset.**

| | Model | MAE | RMSE | MAPE | Time |
|---|---|---|---|---|---|
| Retrained | GRU | 6.41 | 13.53 | 27.05 | 3854.94 |
| | SurSTG | 4.57 | 8.12 | 21.14 | 7734.25 |
| | SurSTG-Retrain | 4.46 | 8.03 | 20.98 | 7680.58 |
| Continual | SurSTG-Static | 4.94 | 8.77 | 21.89 | 1132.76 |
| | SurSTG-Expand | 4.88 | 8.56 | 21.67 | 1476.31 |
| | SurSTG-TS | 4.51 | 8.11 | 20.76 | 3010.68 |
| | PECPM | **4.32** | **7.79** | **20.20** | 2011.29 |

the structure of SGCN is relatively simple, including only two layers of GCN and one layer of TCN, the performance of SGCN with PECPM is less competitive than that of SGCN as SurSTG. However, due to a large number of parameters, STSGCN consumes more training time. With STSGCN as SurSTG, the performance advantage of PECPM is more prominent compared with TrafficStream. One potential reason is that as the number of model parameters increases, TrafficStream is insufficient to preserve prior knowledge. For example, the 60-minute MAE of SurSTG for original nodes is 16.52, which is larger than the one of PECPM 15.39. Because PECPM uses a pattern bank to explicitly preserve important patterns, and the bank consolidation mechanism is used to constrain the update of important parameters, the pattern traceability mechanism can further preserve prior knowledge by replaying historical data.

## 5 RELATED WORKS

**Traffic flow prediction.** Recently, efforts [9, 16, 20, 26, 36, 40] have been made to develop various traffic forecasting techniques based on deep learning. Currently, the most advanced approach is Spatio-Temporal Graph Neural Convolutional Networks (STG), which integrates Graph Neural Networks (GNN) [41] with sequence models to jointly model spatiotemporal correlations. For example, STGCN [32] and T-GCN [38] combine graph convolutional networks with causal convolutional modules. DCRNN [13] and MFDGCN [5] use diffusion GNN with RNN to capture long-term temporal patterns. [11] proposes a traffic prediction model based on pattern matching. However, existing models are designed for static graphs. In practical applications, the distribution and underlying structure of spatiotemporal graphs evolves over time.

**Continuous learning.** Continuous learning [2, 18, 22, 34] is an emerging technology that can complete new tasks by effectively adapting the acquired knowledge without forgetting the learned knowledge. In the field of graph learning [28], ER-GNN [39] and Feature-Graph [27] develop a continuous GNN model based on experience-replay strategies, which grant the model with the access to the historical graph from previous tasks for the rehearsal of previous experience. SGNN-GR [29] employs a generator to learn the distribution of historical graph data, which can generate synthetic historical nodes with the same distribution features. Then these nodes are replayed those distribution features in the next task. However, these models fail to handle the effective update of the traffic flow prediction model when the traffic patterns of nodes and the road network structure evolve simultaneously [3]. To realize the continuous traffic flow prediction, TrafficStream [3] develops a framework based on the historical-data replay strategy. However, compared with both TrafficStream, PECPM can achieve this goal more effectively by alternatively maintaining representational patterns.

**Table 7: 60-minute average prediction performance of SGCN (top part) and STSGCN (bottom part) as SurSTG on PEMS3-Stream dataset.**

| | Model | MAE | RMSE | MAPE | Time |
|---|---|---|---|---|---|
| Retrained | GRU | 19.98 | 29.78 | 26.29 | 630.17 |
| | SGCN | 16.91 | 27.81 | 25.56 | 1550.05 |
| | SurSTG-Retrain | 16.30 | 26.92 | 24.97 | 1286.32 |
| Continual | SurSTG-Static | 17.42 | 27.81 | 37.24 | 236.54 |
| | SurSTG-Expand | 18.26 | 30.47 | 23.86 | 385.51 |
| | SurSTG-TS | 16.37 | 27.04 | 22.94 | 476.42 |
| | PECPM | **16.02** | **26.51** | **22.30** | 421.29 |

| | Model | MAE | RMSE | MAPE | Time |
|---|---|---|---|---|---|
| Retrained | GRU | 19.98 | 29.78 | 26.29 | 630.17 |
| | STSGCN | 14.97 | 25.13 | 220.15 | 15831.76 |
| | SurSTG-Retrain | 14.82 | 25.03 | 19.93 | 14983.32 |
| Continual | SurSTG-Static | 16.23 | 27.12 | 33.98 | 2601.23 |
| | SurSTG-Expand | 17.12 | 29.42 | 21.34 | 4350.91 |
| | SurSTG-TS | 15.73 | 25.56 | 21.03 | 8812.30 |
| | PECPM | **14.85** | **24.62** | **17.90** | 4361.39 |

## 6 CONCLUSION

This work is an attempt to propose a continuous spatiotemporal learning framework without complete historical data. We introduce the pattern-matching mechanism into the traffic flow predicting task and propose a series of strategies to achieve continual spatiotemporal graph learning by integrating new knowledge into the learned model and consolidating old knowledge of the saved model. The experiment results on the large-scale datasets show that the continuous learning framework PECPM can significantly improve the training efficiency and prediction performance of the model for continuous traffic prediction. In the future, we will investigate the potential of PECPM applications in other spatiotemporal domains (e.g., the atmosphere). Moreover, some nodes in the graph may disappear (e.g. sensor failure), thus, investigating the impact of vanishing nodes is also a future research direction.

## 7 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] Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. 2020. Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in Neural Information Processing Systems* 33 (2020), 17804–17815.

[2] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and M Ranzato. 2019. Continual learning with tiny episodic memories. (2019).

[3] Xu Chen, Junshan Wang, and Kunqing Xie. 2021. TrafficStream: A Streaming Traffic Flow Forecasting Framework Based on Graph Neural Networks and Continual Learning. *arXiv preprint arXiv:2106.06273* (2021).

[4] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

[5] Zhengyan Cui, Junjun Zhang, Giseop Noh, and Hyun Jun Park. 2022. MFDGCN: Multi-Stage Spatio-Temporal Fusion Diffusion Graph Convolutional Network for Traffic Prediction. *Applied Sciences* 12, 5 (2022), 2688.

[6] Liangzhe Han, Bowen Du, Leilei Sun, Yanjie Fu, Yisheng Lv, and Hui Xiong. 2021. Dynamic and multi-faceted spatio-temporal deep learning for traffic speed forecasting. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*. 547–555.

[7] Liangzhe Han, Xiaojian Ma, Leilei Sun, Bowen Du, Yanjie Fu, Weifeng Lv, and Hui Xiong. 2022. Continuous-Time and Multi-Level Graph Representation Learning for Origin-Destination Demand Prediction. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 516–524.

[8] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences* 114, 13 (2017), 3521–3526.

[9] Jianlei Kong, Xiaomeng Fan, Xuebo Jin, Sen Lin, and Min Zuo. 2023. A Variational Bayesian Inference-Based En-Decoder Framework for Traffic Flow Prediction. *IEEE Transactions on Intelligent Transportation Systems* (2023).

[10] Shiyong Lan, Yitong Ma, Weikang Huang, Wenwu Wang, Hongyu Yang, and Pyang Li. 2022. Dstagnn: Dynamic spatial-temporal aware graph neural network for traffic flow forecasting. In *International Conference on Machine Learning*. PMLR, 11906–11917.

[11] Hyunwook Lee, Seungmin Jin, Hyeshin Chu, Hongkyu Lim, and Sungahn Ko. 2021. Learning to Remember Patterns: Pattern Matching Memory Networks for Traffic Forecasting. *arXiv preprint arXiv:2110.10380* (2021).

[12] Mengzhang Li and Zhanxing Zhu. 2021. Spatial-temporal fusion graph neural networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35. 4189–4196.

[13] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2017. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926* (2017).

[14] Zhizhong Li and Derek Hoiem. 2017. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence* 40, 12 (2017), 2935–2947.

[15] Chenxi Liu, Zhu Xiao, Dong Wang, Minhao Cheng, Hongyang Chen, and Jiawei Cai. 2022. Foreseeing private car transfer between urban regions with multiple graph-based generative adversarial networks. *World Wide Web* 25, 6 (2022), 2515–2534.

[16] Chenxi Liu, Zhu Xiao, Dong Wang, Lei Wang, Hongbo Jiang, Hongyang Chen, and Jiangxia Yu. 2021. Exploiting Spatiotemporal Correlations of Arrive-Stay-Leave Behaviors for Private Car Flow Prediction. *IEEE Transactions on Network Science and Engineering* 9, 2 (2021), 834–847.

[17] Chi Harold Liu, Chengzhe Piao, Xiaoxin Ma, Ye Yuan, Jian Tang, Guoren Wang, and Kin K Leung. 2021. Modeling Citywide Crowd Flows using Attentive Convolutional LSTM. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 217–228.

[18] David Lopez-Paz and Marc'Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. *Advances in neural information processing systems* 30 (2017).

[19] Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* (2017).

[20] Chuishi Meng, Yanhua Li, Yu Zheng, Jieping Ye, Qiang Yang, Philip S Yu, and Ouri Wolfson. 2022. The 11th International Workshop on Urban Computing. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4886–4887.

[21] Huiling Qin, Xianyuan Zhan, Yuanxun Li, Xiaodu Yang, and Yu Zheng. 2021. Network-wide traffic states imputation using self-interested coalitional learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1370–1378.

[22] Amal Rannen, Rahaf Aljundi, Matthew B Blaschko, and Tinne Tuytelaars. 2017. Encoder based lifelong learning. In *Proceedings of the IEEE International Conference on Computer Vision*. 1320–1328.

[23] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. 2017. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2001–2010.

[24] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. 2017. Incremental learning of object detectors without catastrophic forgetting. In *Proceedings of the IEEE international conference on computer vision*.

[25] Chao Song, Youfang Lin, Shengnan Guo, and Huaiyu Wan. 2020. 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. 914–921.

[26] Yongxin Tong, Yuqiang Chen, Zimu Zhou, Lei Chen, Jie Wang, Qiang Yang, Jieping Ye, and Weifeng Lv. 2017. The simpler the better: a unified approach to predicting original taxi demands based on large-scale online platforms. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 1653–1662.

[27] Chen Wang, Yuheng Qiu, Dasong Gao, and Sebastian Scherer. 2022. Lifelong graph learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 13719–13728.

[28] Junshan Wang, Guojie Song, Yi Wu, and Liang Wang. 2020. Streaming graph neural networks via continual learning. In *Proceedings of the 29th ACM international conference on information & knowledge management*. 1515–1524.

[29] Junshan Wang, Wenhao Zhu, Guojie Song, and Liang Wang. 2022. Streaming Graph Neural Networks with Generative Replay. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1878–1888.

[30] Yuandong Wang, Hongzhi Yin, Tong Chen, Chunyang Liu, Ben Wang, Tianyu Wo, and Jie Xu. 2021. Gallat: A spatiotemporal graph attention network for passenger demand prediction. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 2129–2134.

[31] Junchen Ye, Leilei Sun, Bowen Du, Yanjie Fu, Xinran Tong, and Hui Xiong. 2019. Co-prediction of multiple transportation demands based on deep spatio-temporal neural network. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 305–313.

[32] Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2017. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875* (2017).

[33] Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2018. Spatio-temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*.

[34] Peiyan Zhang, Yuchen Yan, Chaozhuo Li, Senzhang Wang, Xing Xie, Guojie Song, and Sunghun Kim. 2023. Continual Learning on Dynamic Graphs via Parameter Isolation. *arXiv preprint arXiv:2305.13825* (2023).

[35] Yunchao Zhang, Yanjie Fu, Pengyang Wang, Xiaolin Li, and Yu Zheng. 2019. Unifying inter-region autocorrelation and intra-region structures for spatial embedding via collective adversarial learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1700–1708.

[36] Yudong Zhang, Binwu Wang, Ziyang Shan, Zhengyang Zhou, and Yang Wang. 2022. CMT-Net: A mutual transition aware framework for taxicab pick-ups and drop-offs co-prediction. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 1406–1414.

[37] Liang Zhao, Min Gao, and Zongwei Wang. 2022. ST-GSP: Spatial-Temporal Global Semantic Representation Learning for Urban Flow Prediction. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 1443–1451.

[38] Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li. 2019. T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE transactions on intelligent transportation systems* 21, 9 (2019), 3848–3858.

[39] Fan Zhou and Chengtai Cao. 2021. Overcoming catastrophic forgetting in graph neural networks with experience replay. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 4714–4722.

[40] Zhengyang Zhou, Yang Wang, Xike Xie, Lianliang Chen, and Hengchang Liu. 2020. RiskOracle: a minute-level citywide traffic accident forecasting framework. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 1258–1265.

[41] Zhengyang Zhou, Yang Wang, Xike Xie, Lianliang Chen, and Chaochao Zhu. 2020. Foresee urban sparse traffic accidents: A spatiotemporal multi-granularity perspective. *IEEE Transactions on Knowledge and Data Engineering* 34, 8 (2020), 3786–3799.