Scalable and Sustainable Graph-Based Traffic Prediction With Adaptive Deep Learning

James Jianqiao Yu, *Senior Member, IEEE*

***Abstract*—Graph-based deep learning models are becoming prevalent for data-driven traffic prediction in the past years, due to their competence in exploiting the non-euclidean spatial-temporal traffic data. Nonetheless, these models are approaching a limit where drastically increasing model complexity in terms of trainable parameters cannot notably improve the prediction accuracy. Furthermore, the diversity of transportation networks requires traffic predictors to be scalable to various data sizes and quantities, and ever-changing traffic dynamics also call for capacity sustainability. To this end, we propose a novel adaptive deep learning scheme for boosting graph-based traffic predictor performance. The proposed scheme utilizes domain knowledge to decompose the traffic prediction task into sub-tasks, each of which is handled by deep models with low complexity and training difficulty. Further, a stream learning algorithm based on the empirical Fisher information loss is devised to enable predictors to incrementally learn from new data without re-training from scratch. Comprehensive case studies on five real-world traffic datasets indicate outstanding performance improvement of the proposed scheme when equipped to six state-of-the-art predictors. Additionally, the scheme also provides impressive autoregressive long-term predictions and incremental learning efficacy with traffic data streams.**

***Index Terms*—Traffic prediction, deep learning, capacity scalability, capacity sustainability, intelligent transportation systems, data mining.**

I. Introduction

RECENT years have witnessed the gradual adoption of intelligent transportation systems (ITS) as a vital component in modern smart cities along with the rapid urbanization process [1]. As an indispensable part of ITS, efficient traffic prediction systems provide transportation utilities with continuous road status information for subsequent traffic scheduling and operations [2], [3]. In support of their significant societal influence derived from the role of ITS in urban transportation, both the academia and the industry are contributing to traffic prediction algorithms aiming to forecast the traffic speed or flow volume of traffic participants in a future period based on current and historical observations [4].

A majority of research effort in the past decade has been devoted to data-driven approaches for abstracting latent patterns behind traffic data [1]. Among the existing data mining approaches, deep learning models have attracted remarkable attention due to their capability of modeling highly complex and non-linear functions, which just accords with the nature of traffic data [5]. The research community gradually exploits deeper data correlation with innovative deep learning models for traffic prediction over the past few years, embracing recurrent neural networks first for multivariate time-series learning, then temporal-convolutional neural networks for spatial-temporal data mining, and more recently graph-based learning models to learn from traffic data aligned in non-euclidean spaces [2]. Along this line of research, traffic prediction accuracy also advances by incorporating more latent information from data [3]. Increasingly complicated deep learning models are devised to squeeze every bit of prediction performance from the traffic data.

Nonetheless, the current development of graph-based deep learning models is reaching a state where the accuracy of traffic prediction is saturating regardless of the model complexity. The state of the arts in the past year have pushed the number of model parameters to be learned from data exponentially to the magnitude of millions, yet the performance gain in terms of relative predictive error is less than a few percent [6], [7], [8]. Considering the difficulty of training huge deep learning models and the corresponding requirement on the traffic data quantity, it is arguable how far can we proceed along this way. This challenge of model *complexity* is hindering the future development of traffic prediction approaches.

Granted that further increasing model complexity still contributed to significant prediction performance, there is one question that is infrequently asked: is it necessary to design such complex models to mine traffic data for prediction? As typical deep learning-based traffic predictors are presented in multi-layered structures, identifying the optimal number of layers is usually a trial-and-error process, which may render inferior performance with suboptimal settings [9]. Even though one particular configuration works best for an arbitrary set of data, the number may need to be re-adjusted for others. The learning process converges slowly with overly-deep models and is subject to overfitting, while the learning capacity is restricted if the model is too simple [10]. This open challenge is defined as capacity *scalability* for traffic prediction methods. Additionally, the ordinary prolonged model training time also hinders traffic predictors to adapt to the evolving traffic dynamics in the form of traffic data streams [11]. Consistently updating prediction models with the latest traffic data without the heavy re-training from scratch requires model capacity *sustainability*, whose research effort is minuscule in current traffic predictors.

Manuscript received 16 November 2022; revised 15 January 2024; accepted 22 June 2024. Date of publication 25 June 2024; date of current version 27 September 2024. Recommended for acceptance by Y. Shen.

The author is with the Department of Computer Science, University of York, YO10 5DD York, U.K. (e-mail: jqyu@ieee.org).

Digital Object Identifier 10.1109/TKDE.2024.3419036

1041-4347 © 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

To overcome the research gap and address these challenges, we propose a novel Traffic Slicing and Adaptive Learning (TSAIL) scheme for graph-based traffic predictors to jointly entertain the model complexity, capacity scalability, and sustainability requirements. Particularly, TSAIL comprises three major modules, namely, time-series slicer, adaptive traffic predictor, and prediction aggregator. By slicing the traffic time series along spatial and temporal horizons, the traffic data entanglement is segregated and can be effectively captured using smaller models with lower complexities. Additionally, auxiliary hidden data paths on each neural layer of the predictor are incorporated and aggregated by an attention network for the scheme to be adaptively scaled according to the latent data size. Finally, TSAIL enables the model training algorithm to discard dated and obsolete models and train new ones with minimal effort to incorporate the latest traffic data with a meta-attention network, so that the traffic predictors are furnished with capacity sustainability. To summarize, each of these modules primarily handles one of the aforementioned challenges, and TSAIL orchestrates the three to improve traffic prediction based on any graph-based predictors. Note that the primary objective of this study is not to devise a new traffic predictor. Instead, we rethink the nature of traffic data and patterns for traffic data mining and accordingly develop a universal scheme to be applied to existing and future graph-based traffic predictors as a performance booster add-on. The main contribution of this work is as follows:

- We propose a time-series slicing mechanism to utilize domain knowledge for traffic data disentanglement, resulting in less challenging data learning tasks.
- We devise an attention-based hidden feature aggregation mechanism to aid traffic predictors adapt to various data complexities and scales.
- We design an incremental training algorithm to learn the transient and steady traffic dynamics changes, making predictor models sustainable.
- Comprehensive case studies are conducted on five large-scale real-world traffic datasets with six state-of-the-art graph-based traffic predictors to show the efficacy of the proposed scheme.

The remainder of this paper is organized as follows. Section II presents a brief literature review on graph-based traffic prediction and existing efforts on scalability/sustainability. Section III elaborates on the proposed traffic slicing and adaptive learning scheme with a comprehensive analysis and discussion of the design principle. Section IV introduces a stream data learning mechanism based on the proposed scheme for capacity sustainability. Section V demonstrates the numerical results of case studies with in-depth interpretation. This paper is concluded in Section VI.

II. Related Work

Graph-based traffic prediction is gaining attention with the rapid advance of deep learning techniques since their renaissance in the early 2010 s. In this section, we briefly review the existing state of the arts on graph-based traffic prediction and take into account existing traffic prediction partitioning efforts. Interested readers are referred to recent surveys for thorough investigations of the context [2], [12].

In the past decade, the adoption of traffic monitoring systems and the produced traffic data in huge quantities enable advanced data mining techniques to learn the latent data dependency and thereupon make predictions [1]. These techniques gradually overwhelm canonical statistical and machine learning approaches, e.g., Support Vector Regressor [13], that heavily depend on feature engineering created by domain expertise. In this line of research, deep learning-based traffic predictors experienced a shift from recurrent neural networks to convolutional neural networks and, more recently, to graph learning-based neural networks [14], [15]. As traffic data is a type of time series continuously recorded at a semi-fixed frequency, all approaches above exploit the strong auto-correlation, temporal dependency, and optionally spatial dependency of traffic data for prediction [3]. Considering that traffic data are typically generated along transportation networks, which can be intuitively abstracted into graphs, graph-based deep learning approaches are natural fits to handle such non-euclidean space data correlation. Traffic prediction leaderboards are dominated by graph learning of late — [6], [7], [8], [16] to name a few.

Despite consistently improving the performance notably or not, contemporary traffic predictors are reaching a point where even exponentially increasing model complexity cannot lead to proper augmentations. Among the various rationales behind this observation, a prominent cause is that typical graph-based predictors take the complete chunk of data for training, which may comprise irrelevant spatial and temporal information degenerating the prediction [17], [18]. A straightforward solution is to partition the transportation network into sub-graphs. Consequently, multiple independent learning models can be parallelly trained and employed for these sub-graphs. Several approaches have been proposed utilizing this idea of domain decomposition if related transportation services are considered, e.g., traffic assignment [19], [20], network management [21], and travel time estimation [22]. Nevertheless, the research efforts were relatively scarce in the context of traffic prediction. Reference [17] provides a topology-partition-based approach to dividing huge transportation networks with tens of thousands of nodes into sub-graphs for traffic prediction. Reference [18] devises a speed-data-driven graph-partitioning approach to better address the traffic data correlation during sub-graph construction. This work takes a step further to consider data division on both the spatial domain as graph partitioning and the temporal domain as time-series segmentation.

In this work, inspired by the characteristics of traffic data and existing graph-based traffic predictors, we propose a novel traffic slicing and adaptive learning scheme based on the hypothesis that disentangling traffic data by domain decomposition can effectively reduce the required model complexity, which in turn contributes to better forecasts. The proposed scheme overcomes the aforementioned challenges and can be applied to existing and future graph-based traffic predictors as an add-on for possible performance improvement.

III. Traffic Slicing and Adaptive Learning for Traffic Prediction

To jointly address the three open challenges for graph-based traffic prediction as introduced in Section I, we propose TSAIL to adaptively learn from the historical traffic data for prediction. The proposed model differs from existing deep model-based traffic predictors by adopting relatively-shallow neural networks and utilizing an easy-to-train aggregation module to develop the final prediction. With a carefully designed spatial-temporal data slicing mechanism, the models in TSAIL aim to merely learn one traffic dynamics pattern and learn it well. Additionally, the hidden data paths in layered predictors enable TSAIL to adapt to various data slices of different scales, thus achieving model capacity scalability.

In this section, we first provide the problem statement with notation definitions. Then, we present an overview of the proposed TSAIL, with subsequent detailed elaborations on the constituting components and the rationale behind the design.

A. Problem Definition

The traffic data of a transportation network depicts the dynamics of traffic flow over time, which can be naturally denoted by a time series. Let $\mathcal{V}$ be the geographic location of traffic sensors, which can be in the form of induction loops installed within the pavement, surveillance cameras along the road, or any other devices that constantly produce traffic measurement data. Following the common practice, the connectivity of these sensors — denoted by an edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ — can be developed according to their pair-wise euclidean distance or road connectivity. Consequently, traffic data is mapped onto a directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, and a further $\mathcal{A} \in \mathbb{B}^{|\mathcal{V}|\times|\mathcal{V}|}$ represents the adjacency matrix derived from $\mathcal{E}$.

Consider the current time $t = 0$ and a discrete time span $\mathcal{T}$, the historical traffic data (speed or flow) time-series is defined by $\mathbf{X}_{\leq 0} = \{\ldots, \mathbf{x}_{-2}, \mathbf{x}_{-1}, \mathbf{x}_0\}$, where $\mathbf{x}_t = \{x_{i,t}\} \in \mathbb{R}^{|\mathcal{V}|}$ denotes the traffic data measured at time t. The objective of traffic prediction is to find a generative function $\mathcal{F}(\cdot, \boldsymbol{\phi})$ such that

$$\mathbf{X}_{>0} = \mathcal{F}(\mathbf{X}_{\leq 0}, \boldsymbol{\phi}), \tag{1}$$

where $\mathbf{X}_{>0} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots\}$ and $\boldsymbol{\phi}$ is the set of parameters in $\mathcal{F}$. Typically, data-driven $\mathcal{F}$'s are formulated based on various parametric and non-parametric learning models extracting latent traffic knowledge from the history $\mathbf{X}_{\leq 0}$. Recent development of such models emphasizes both the spatial (euclidean first and then non-euclidean) and temporal data correlations. In the proposed TSAIL, we also consider these two factors with further model capacity scalability and sustainability designs.

B. TSAIL Framework

The design principle of TSAIL follows an intuitive hypothesis that the spatial-temporal traffic data dependency varies over the day and in different regions in a city. TSAIL adopts multiple deep learning models that are relatively shallower than existing solutions to independently learn the correlation. While the capacities are reduced, these models aim at extracting and extracting well the latent traffic dynamics of just one small portion of the city scale during the transient time. This strategy loosely resembles the concept of ensemble learning, where aggregated weak learners contribute to better model efficiency and performance.

Fig. 1 presents the overview of TSAIL. The proposed TSAIL framework comprises three major modules, namely, *time-series slicer*, *adaptive traffic predictor*, and *prediction aggregator*. In the first phase, the historical traffic data is heuristically divided into multiple traffic data "slices" following the principle of domain decomposition, each of which corresponds to a particular period and selected sensors. TSAIL first slices the data along the time axis to divide the ever-changing traffic dynamics into diverse periods, aiming at reducing the complexity of temporal correlation within each period. Subsequently, the data in the form of time-series graphs are further sliced into sub-graphs by the similarity of sensory data, further alleviating the learning difficulty of shallow models. To exploit the non-euclidean data learning capability of contemporary geometric deep learning approaches, these similar sensors after slicing goes through a node connector to establish their connectivity.

In the second phase, each traffic data slice is provided to a typical deep learning-based layered traffic predictor for parameter learning. To cope with the scalability challenge, each hidden layer of the predictor is appended with an independent traffic data regressor, whose output is combined and fed into an extra attention network to calculate the corresponding attention weight. These weights serve as an "importance" scoring of the layer output for the final sub-graph prediction. Finally, the third phase utilizes all combined sub-graph predictions and adopts another attention network for result aggregation.

C. Time-Series Slicer

Given the historical traffic data $\mathbf{X}_{\leq 0}$ over time span $\mathcal{T}$, the primary objective of the time-series slicer is to divide the data into multiple slices within the temporal and spatial domain so that the data in each slice have less complex spatial-temporal data correlation over the complete one. TSAIL follows the empirical conclusion in [23] that the spatial dependency of traffic data is typically easier to be extracted, $\mathbf{X}_{\leq 0}$ is first sliced along the time axis with a *Temporal Slicer* scheme. Particularly, the discrete time span $\mathcal{T}$ is split into multiple overlapping periods. All time periods are of the same length T and adjacent periods have a time difference defined by $\Delta \leq T$. Therefore, each day in the history corresponds to $\lceil 24\text{h}/\Delta \rceil$ periods. Considering that traffic data exhibits different dynamics over weekdays and weekends (incl. holidays) [24], [25], [26], we additionally aggregate the corresponding time periods concerning this feature. The resulting time periods are denoted by $\mathcal{T}_t^{\mathrm{D}} \subsetneq \mathcal{T}$ for those starting from time t of weekdays and $\mathcal{T}_t^{\mathrm{E}} \subsetneq \mathcal{T}$ for those starting from time t of weekends and holidays, respectively. Finally, the historical traffic data that fall into each $\mathcal{T}_t^{\mathrm{D}}$ and $\mathcal{T}_t^{\mathrm{E}}$ are jointly denoted by $\mathbf{X}_t^{\mathrm{D}}$ and $\mathbf{X}_t^{\mathrm{E}}$, and t can take $N = 2 \times \lceil 24\text{h}/\Delta \rceil$ different values (weekday/weekend multiplies $\lceil 24\text{h}/\Delta \rceil$ starting times of time periods).

After segregating the time periods, TSAIL further slices the resulting traffic data in the spatial domain. Intuitively, predicting

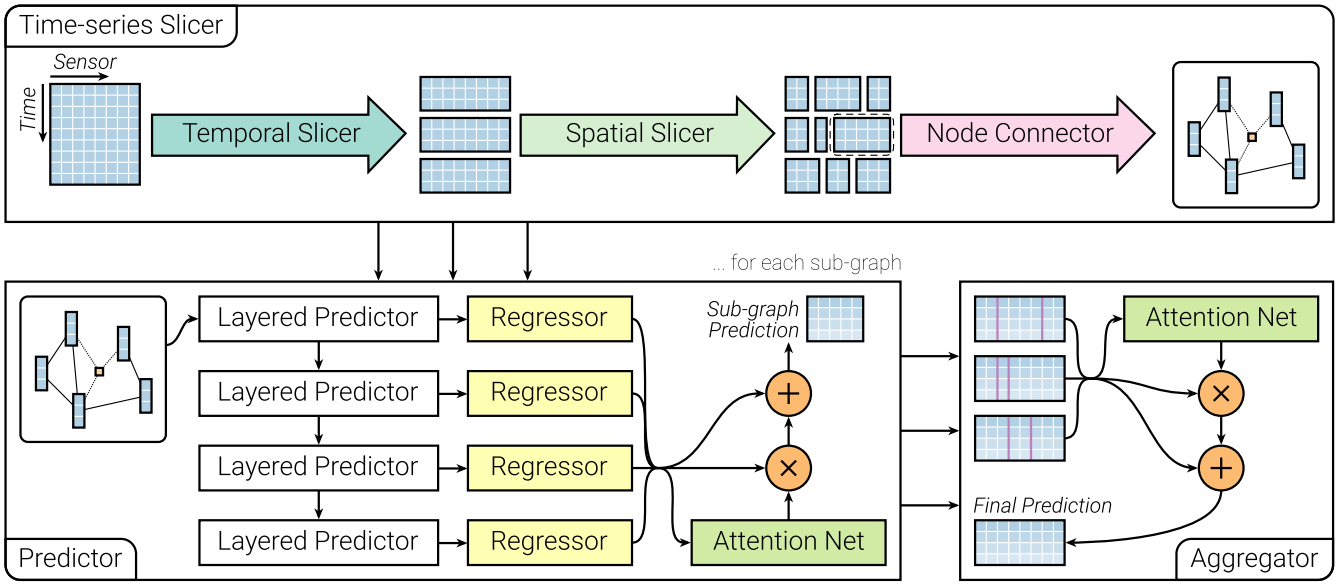

Fig. 1. Architecture of the proposed TSAIL framework for traffic prediction.

time series is simpler with data-driven approaches using past data that are more correlated. In the traffic prediction context, we interpret the correlation of traffic sensors from two aspects, namely, their statistical correlation and geographical connectivity via the road network. Given an intermediate data slice $\mathbf{X}_t^{\{\mathrm{D,E}\}}$ from the temporal slicer, the *Spatial Slicer* scheme first calculates the cross-correlation between the time-series of any two sensors $i, j \in \mathcal{V}$ using absolute Pearson's correlation coefficient:

$$\rho_{i,j,t}^{\{\mathrm{D,E}\}} = \left| \frac{\mathrm{cov}(\boldsymbol{x}_{i,t}, \boldsymbol{x}_{j,t})}{\sigma_{i,t}\sigma_{j,t}} \right|, \tag{2}$$

where cov is the covariance, $\boldsymbol{x}_{i,t} = \{x_{i,t}, x_{i,t+1}, \ldots, x_{i,t+T-1}\}$, and $\sigma_{i,t}$ is the standard deviation of $\boldsymbol{x}_{i,t}$, respectively. Based on the cross-correlation, additional edges that represent strong data correlation are added to the original traffic graph $\mathcal{G}$. For each time period $\mathcal{T}_t^{\{\mathrm{D,E}\}}$, the sensor pairs $(i, j) \notin \mathcal{E}$ that have the top ζ (edge count) cross-correlation measures among non-edge sensor pairs $\{\mathcal{V} \times \mathcal{V}\} \setminus \mathcal{E}$ are added to $\mathcal{E}$. When all pairs are calculated, sorted, and optionally included in the graph, each edge $e \in \mathcal{E}_t^{\{\mathrm{D,E}\}}$ after the addition is assigned with a weight equal to its corresponding cross-correlation according to (2). The new graph $\mathcal{G}_t^{\{\mathrm{D,E}\}}(\mathcal{V}, \mathcal{E}_t^{\{\mathrm{D,E}\}})$ is subsequently partitioned into K sub-graphs using the k-way graph partitioning algorithm METIS [27] by minimizing the total communications volume. Symbol $\mathcal{G}_{k,t}^{\{\mathrm{D,E}\}}$ is used to represent the k-th sub-graph during $\mathcal{T}_t^{\{\mathrm{D,E}\}}$.

While it is already possible to aggregate the historical traffic data of nodes in any $\mathcal{G}_{k,t}^{\{\mathrm{D,E}\}}$ during the corresponding $\mathcal{T}_t^{\{\mathrm{D,E}\}}$, there is one more step before feeding the data into a learning model, namely, *Node Connector*. The idea behind this scheme is intuitive. Multi-hop spatial correlation among data can be extracted in the original graph $\mathcal{G}$. However, the correlation may be discarded during the graph partitioning process due to possible edge cuts along the correlation hops. To overcome this issue, we introduce the concept of *virtual node* into the partitioned sub-graph $\mathcal{G}_{k,t}^{\{\mathrm{D,E}\}}$. In particular, the node set of each $\mathcal{G}_{k,t}^{\{\mathrm{D,E}\}}$ is added with a virtual node, which is connected to any other nodes i which has one or more connecting edges $(i, j) \in \mathcal{E}_t^{\{\mathrm{D,E}\}}$ cut by METIS. This virtual node also receives input data in the following traffic prediction step just like other nodes, only that the data are all zeros to prevent adverse impact on the data learning process.

D. Adaptive Traffic Predictor

With the aforementioned time-series slicer, the traffic graph is partitioned into multiple sub-graphs $\mathcal{G}_{k,t}^{\{\mathrm{D,E}\}}$, each of which corresponds to a particular time period $\mathcal{T}_t^{\{\mathrm{D,E}\}}$. Contemporary data-driven traffic predictors can utilize the information to train a learning model and forecast traffic data. However, modern cities are of different sizes. It is hard to figure out a unified model architecture to fit all, i.e., traffic predictors have scalability issues.

In TSAIL, an adaptive traffic predictor is devised to provide scalability to traffic predictors. Without loss of any generality, an arbitrary data-driven traffic predictor is with L hidden layers. Here we do not enforce the use of any particular models, only that the maximum model capacity is with L layers considering the computing ability. Typically, the predictor employs the final feature representation at the L-th layer as the prediction $\mathbf{h}_L$. TSAIL takes a step forward and utilizes hidden feature representations $\{\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_L\}$ of all layers, which are derived by the base traffic predictor. In particular, independent traffic data regressors $f_l(\cdot)$ are appended to the corresponding representations to transform the embeddings back to predictions:

$$f_l(\mathbf{h}_l) = \mathrm{ReLU}(\mathbf{w}_l\mathbf{h}_l + b_l), \forall l = 1, 2, \ldots, L, \tag{3}$$

where $\mathbf{w}_l$ and b_l are the regressor weight and bias, respectively. The multiple predictions yielded by L layers are aggregated with attention-based pooling to develop the final prediction:

$$f(\mathbf{x}) = \sum_{l=1}^{L} \alpha_l f_l(\mathbf{h}_l), \tag{4}$$

where the attention coefficient for each layered prediction α_l is calculated by an attention network, which is a softmax-activated shallow neural network whose output denotes the relative weights of each $f_l(\mathbf{h}_l)$. By training, this attention network aims to establish the relationship among the hierarchical traffic regressors. The model training can be correspondingly achieved with the following loss:

$$\mathcal{L}(\mathbf{x}_\tau, \mathbf{y}_\tau) = \ell\left(\sum_{l=1}^{L} \alpha_l f_l(\mathbf{h}_l), \mathbf{y}_\tau\right). \tag{5}$$

The predictive loss $\ell(\cdot,\cdot)$ is set following the original traffic predictor.

When training with sliced traffic time-series, each $\mathcal{G}_{k,t}^{\{\mathrm{D,E}\}}$ and $\mathcal{T}_t^{\{\mathrm{D,E}\}}$ pair corresponds to one adaptive traffic predictor, denoted by $\mathrm{ATP}_{k,t}^{\{\mathrm{D,E}\}}(\cdot)$. The training data are prepared according to the particular requirement of the underlying traffic prediction model. In general, for all $\tau \in \mathcal{T}_\tau^{\{\mathrm{D,E}\}}$, historical time-series $\mathbf{X}_{<\tau}$ including nodal data from $\mathcal{G}_{k,\tau}^{\{\mathrm{D,E}\}}$ is employed as the input $\mathbf{x}_\tau$ of the predictor, and $\mathbf{x}_{k,\tau} = \{x_{i,\tau}\}_{i\in\mathcal{V}_{k,\tau}^{\{\mathrm{D,E}\}}}$ is the prediction objective, i.e., $\mathbf{y}_\tau$ in (5).

To accelerate the training of adaptive traffic predictor and improve the capacity scalability, a layer-wise back-propagation scheme is devised to substitute the typical back-propagation which propagates the loss derivatives from the last layer L. In (5), the derivatives are propagated from each regressor f_l to update layer-wise parameters, namely,

$$\Theta_l \leftarrow \Theta_l - \eta \nabla_{\Theta_l} \ell\left(\sum_{l=1}^{L} \alpha_l f_l(\mathbf{h}_l), \mathbf{y}\right), \tag{6}$$

where Θ_l is the collection of the traffic predictor parameters in the l-th layer, which also includes the added regressor parameters $\mathbf{w}_l$ and b_l, and η is the learning rate, respectively. Following the idea that shallow models converge faster than deep ones [28], the attention network focus on the shallow layers at the beginning of training. With the increase of training data volume, larger attention coefficients are learned for deeper layers, contributing to capacity scalability. Consequently, the optimal network depth is learned automatically and adaptively.

E. Prediction Aggregator

TSAIL leverages the adaptive traffic predictor to forecast sub-graph traffic at a particular time. The k predictors combined can intuitively provide the whole picture of the prediction. Nonetheless, we argue that predictors trained for other time periods can provide auxiliary information to better assist in developing the final prediction. To put it another way, predictors on $\mathcal{G}_{k,t}^{\{\mathrm{D,E}\}}, \forall t \neq \tau$ may help forecast $t = \tau$ traffic data. Following this principle, we devise a prediction aggregator to utilize the additional information. For any arbitrary time τ, the training data $\langle \mathbf{x}_\tau, \mathbf{y}_\tau \rangle$ are input into $\mathrm{ATP}_{k,t}^{\{\mathrm{D,E}\}}(\cdot), \forall t$ whose output can be combined to yield N complete traffic predictions. These predictions are input into another two-layer fully-connected attention network to adaptively learn the relative importance β_t of each traffic prediction from data. The output of this attention network $\sum_{t=1}^{N} \beta_t \cdot \|_{k=1}^{K} \mathrm{ATP}_{k,t}^{\{\mathrm{D,E}\}}(\cdot)$ is considered as the final prediction of TSAIL, and the network can be trained by reconstruction loss. Note that this prediction aggregator is optional: without this module, the other two can still generate complete traffic predictions following the standard practice of inference. In the case studies, we will empirically demonstrate the efficacy of this prediction aggregator.

IV. Model Capacity Sustainability for Traffic Prediction

Besides the model capacity scalability challenge addressed in the previous section, another outstanding issue in traffic prediction is capacity sustainability. It is notable that traffic data are presented in the form of data streams and are evolving, resulting in an ever-changing ground truth distribution. Models trained on previous distributions may experience a significant performance drop if the changes are ignored. Re-training the model from scratch is a waste of existing distilled knowledge on the historical traffic dynamics, yet re-training with well-trained parameters may experience the catastrophic forgetting problem.

In TSAIL, we primarily utilize the prediction aggregator to provide sustainable model capacity for learning traffic data streams instead of unregulated increases in model capacity by enlarging its size. Recall that the prediction aggregator leverages an attention network to "score" the importance of different traffic predictions, each of which is developed by a set of adaptive traffic predictors. Indeed, there is no limitation on the number of predictions the attention can handle. This makes it possible to adopt new predictors for new data from the traffic data stream, and then integrate them into the existing attention mechanism. Particularly, new adaptive traffic predictors $\mathrm{ATP}_{k,t}^{\{\mathrm{D,E}\}}(\cdot)$ can be constructed and trained with newly available data when re-training the model is deemed necessary, e.g., every day. This process is independent of the other existing predictors, and since each one handles a sub-graph, is relatively lightweight. The trained new predictors, together with the existing ones, calculate traffic predictions upon being provided with historical data. Subsequently, all predictions go through the previously trained prediction aggregator to develop the final prediction, whose reconstruction loss is back-propagated through the comprising attention network to update the parameters. The training continues until the aggregator performance converges again, which is typically much faster than training from scratch as the starting one serves as a pre-trained model. In such a manner, new information from the traffic data stream can be integrated timely for prediction.

It can be figured that the above scheme avoids re-training any adaptive traffic predictors. However, the aggregator is re-trained with predictions from the new predictors, which may still lead

to the catastrophic forgetting problem, although on a smaller neural network. To resolve this issue, we follow recent efforts on the Fisher information matrix [29], [30] and modify the loss function of the aggregator. Considering the nature of time-series prediction, the output of the aggregator can be defined as a sample drawn from the likelihood distribution $\Pr_{\boldsymbol{\phi}}(\mathbf{y}|\mathbf{x})$. Correspondingly, the empirical Fisher information matrix $F_{\boldsymbol{\phi}} = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}}\left[\frac{\partial \log \Pr_{\boldsymbol{\phi}}(\mathbf{y}|\mathbf{x})}{\partial \boldsymbol{\phi}} \frac{\partial \log \Pr_{\boldsymbol{\phi}}(\mathbf{y}|\mathbf{x})}{\partial \boldsymbol{\phi}}^{\mathsf{T}}\right] \in \mathbb{R}^{|\boldsymbol{\phi}|\times|\boldsymbol{\phi}|}$, where $\mathcal{D}$ is the training data domain. The matrix has three key properties [31], namely, equivalent to the second-order derivative of the loss near local minima, easily computable from first-order derivatives, and guaranteed positive semi-definite. Therefore, using it as regularization guides the training algorithm to keep critical network parameters close to previously learned values, thus alleviating catastrophic forgetting [10]. We further follow [32] and assume that neural network parameters $\boldsymbol{\phi}$ are independent to circumvent the huge size of $F_{\boldsymbol{\phi}}$. As such, only the diagonal entries of $F_{\boldsymbol{\phi}}$ are calculated and stored, which leads to the following loss function for the prediction aggregator:

$$\mathcal{L}^*(\mathbf{x}_\tau, \mathbf{y}_\tau) = \ell\left(\sum_{t=1}^{N^*} \beta_t \big\|_{k=1}^{K} \mathrm{ATP}_{k,t}^{\{\mathrm{D},\mathrm{E}\}}(\mathbf{x}_\tau), \mathbf{y}_\tau\right) + \frac{\lambda}{2}\sum \mathrm{diag}(F_{\boldsymbol{\phi}}) \odot (\boldsymbol{\phi} - \boldsymbol{\phi}^*), \tag{7}$$

where N^* is the total number of adaptive traffic predictors including the new ones and previous N, λ is a weight parameter describing how close the newly trained model shall be with the previous one[1], and $\odot$ is the element-wise multiplication operation. This gives an effective approach for the aggregator to adapt to new predictors with the learned knowledge from the previous ones. The utility of the prediction aggregator in improving model capacity sustainability will be illustrated in the case studies.

V. Case Studies

In this work, we propose a novel TSAIL scheme to bring both scalability and sustainability to graph-based traffic predictors. We conduct a series of comprehensive case studies to illustrate the efficacy of TSAIL on six state-of-the-art predictors with five distinct real-world datasets from different sources. In particular, we first investigate the performance and efficiency improvements brought by TSAIL to the predictors. TSAIL-driven performance enhancements in long-term autoregressive prediction are then assessed. Subsequently, we carry out a thorough ablation test on the constituting modules of TSAIL to verify their necessity, and stream-data-driven traffic prediction is further investigated. Finally, a hyperparameter test is performed to demonstrate the sensitivity of the proposed scheme.

A. Experimental Configurations

In subsequent case studies, we employ five real-world traffic speed datasets for performance evaluation, namely, NavInfo road speed data of Beijing, China and Shanghai, China (NI-BJ and NI-SH), Hong Kong Real-time Road Traffic data (HK-RT), and Caltrans Performance Measurement System traffic data in the Bay Area and District 7 (PeMS-BAY and PeMS-D7) from the respective three data sources. Particularly,

- NI-BJ and NI-SH include proprietary floating car data of the two metropolises in China from Jan. 2019 to Jun. 2019 with a constant 5min sampling interval. 1569 and 1830 roads (or segments) are covered in respective datasets, whose adjacency matrices are derived from the road connectivity information with a midway-located sensors assumption following the practice [26], [33], [34].
- HK-RT dataset comprises publicized traffic speed data of 608 major routes and urban roads in Hong Kong from Jun. 2021 to Mar. 2022 with a constant 5min sampling interval. The adjacency matrix of the dataset is generated from the route linking data published by the same data source.
- PeMS-BAY and PeMS-D7 datasets are developed from the 325 and 228 traffic sensors in the sampling area from Jan. 2017 to May 2017 and from May to Jun. 2012, respectively. Traffic readings are aggregated into consecutive 5min windows. Different from the other two data sources but in accordance with the literature [6], [35], pairwise sensor distances with a thresholded Gaussian kernel are adopted to construct the default adjacency matrices for the two datasets.

All datasets are subjected to z-score normalization and are split in chronological order with a 7:1:2 ratio for model training, validation, and testing. We adopt the widely recognized Mean Average Percentage Error (MAPE) and Root Mean Square Error (RMSE) as the performance metrics in the sequel. We run the following tests with the implementation of an environment on nVidia RTX 2080Ti GPUs for neural network computing acceleration.

Unless otherwise stated, the prediction granularity is set to 5min using historical data in the immediate past 1h, i.e., 12 samples. The length of time periods $T = 2\mathrm{h}$ with a time difference $\Delta = 1\mathrm{h}$. The node-correlation threshold $\zeta = |\mathcal{E}|$, i.e., the number of additional cross-correlation-based edges is identical to that of connectivity-based ones. The number of partitions $K = \lceil |\mathcal{V}|/100 \rceil$ so that each sub-graph has approximately 100 nodes. The attention networks in the adaptive traffic predictor and the prediction aggregator have two fully connected layers, each with 64 nodes. The adaptive traffic predictor is trained by the respective training method of the underlying data-driven traffic predictor. The prediction aggregator is trained by Adam optimizer with a base learning rate 5×10^{-3}.

B. Quantitative Results

In the case study, we adopt the following six state-of-the-art graph-based traffic predictors as the base model of TSAIL, and investigate its respective performance improvement w.r.t. prediction accuracy and computation cost:

[1]We empirically set λ to 10^3. Other values do not significantly change the model performance in our offline tests.

- *Graph Wavenet (GWN) [6]:* GWN is an end-to-end graph neural network model for traffic prediction, which incorporates an adaptive dependency matrix learned by node embedding for capturing the hidden spatial data correlation. An additional stacked delated 1D convolution component is also utilized to exponentially grow the receptive field for long sequence data mining.
- *Graph Multi-attention Network (GMAN) [7]:* GMAN adopts an encoder-decoder structure consisting of multiple spatio-temporal attention blocks to exploit the spatio-temporal characteristics of traffic dynamics. A transform attention layer is devised to directly model the historical and future time step correlation, rendering an attenuated error propagation issue for multistep predictions.
- *Spatial-Temporal Fusion Graph Neural Networks (STFGNN) [8]:* STFGNN employs a data-driven method to generate temporal graphs for compensating missing but genuine data correlations that spatial graphs cannot reflect. A fusion operator on multiple spatial and temporal graphs is proposed to learn the hidden spatial-temporal dependencies at different time periods in parallel.
- *Temporal Graph Convolutional Network (T-GCN) [16]:* T-GCN captures the spatial and temporal dependencies in traffic data by an integrated GCN and gated recurrent unit module, in which the complex topological structures and dynamic changes of traffic can be exploited for traffic forecasting.
- *Graph and Attentive Multi-Path Convolutional Network (GAMCN) [36]:* GAMCN devises a variant of graph convolutional network and embed road network indices into a latent space, and develops a multi-path convolutional neural network to exploit the joint impact of past traffic conditions to future.
- *Spatio-Temporal Sequence-to-Sequence Network (STSSN) [37]:* STSSN employs an encoder-decoder framework with convolution and diffusion modules to capture time-varying node representations for daily and weekly traffic patterns, utilizing dilated causal convolution for short-term temporal correlations and an encoder-decoder attention module to address long-term temporal correlations and mitigate error propagation.

For the first four base models, we utilize the published source code provided in the respective literature with minuscule non-algorithmic changes to adapt to TSAIL, and the latter two models are implemented according to the respective publication. The hyperparameters are configured according to the suggestions listed in the literature. In Table I, we summarize the prediction accuracy of base models and the performance variation led by incorporating TSAIL, labeled by "+TSAIL". Note that the relative performance among the base models is not a primary focus. We are more interested in investigating whether TSAIL can bring performance improvements and to what extent if any.

From the comparative results presented in the table, it is clear that TSAIL can positively affect the performance of all investigated graph-based traffic predictors. In particular, an average 5.66% prediction error reduction to MAPE and RMSE can be obtained by introducing TSAIL to all predictors on all datasets. This statistical summary of raw traffic prediction results indicates that the proposed TSAIL effectively disentangles the traffic prediction task considering the domain-specific spatial-temporal dependency. Additionally, an interesting conclusion can be developed from the result that not all datasets are equal for TSAIL: more significant improvements are introduced to those "harder" ones. Notably, NI-BJ and NI-SH witness 6.75% and 6.21% MAPE boost across all predictors compared to PeMS-BAY's 4.62%, respectively. We hypothesize that as PeMS-BAY contains sensors installed along the major highway in Bay Area, the traffic is primarily free-flow and is less convoluted than urban ones, which are the cases for NI-BJ and NI-SH. As a result, the utilized time-series slicer effectively breaks down the relatively weak spatial-temporal data correlations with domain knowledge. The slices require much less model capacity for the predictors to exploit the strong but diverse data dependencies, rendering notably improved predictions.

To further reveal the efficacy of TSAIL with small models on traffic prediction, we change the hyperparameters of base predictors to reduce the volume of trainable parameters. Specifically, GWN halves the total number of layers from the default eight to four, and the number of filters from 32 to 16; GMAN employs two attention blocks, each with four attention heads; STFGNN reduces to two graph convolution layers with 32 filters; and T-GCN possesses 128 hidden neurons in each layer; GAMCN uses 32-dimensional latent space and 15min slots in time-of-day embedding; STSSN follows GWN to halve the total number of layers and reduces the number of attention heads to four. This scheme is labeled by "TSAIL/S" for TSAIL with small models, and the respective performance is also summarized in Table I. The results imply that, although not as superior as full-sized TSAIL, TSAIL/S can still introduce more precise traffic predictions over the base predictors with comparable total model size. An average 4.71% MAPE improvement is developed over all test cases. Additionally, the training and inference speed boost can be observed from the training and inference times as shown in Fig. 2, and the models' FLOPs, where GWN's are reduced from 1.460×10^9 to 4.079×10^8, GMAN's are reduced from 2.335×10^9 to 4.452×10^8 STFCNN's are reduced from 2.334×10^9 to 7.411×10^8, TGCN's are reduced from 9.943×10^7 to 2.494×10^7, GAMCN's are reduced from 9.426×10^9 to 3.430×10^9, and STSSN's are reduced from 2.074×10^9 to 8.529×10^8. It is true that the alleviation of model computation footprints is caused by the shrinkage of model sizes, yet TSAIL enables the modification and provides even better aggregated performance. This case study reveals an alternative usage of TSAIL for traffic prediction performance boost. Under the cases where multiple deep models can be trained parallelly, full-sized TSAIL can achieve the best overall prediction accuracy with less parallel training time. For serial-only training scenarios, notably reducing the base model size can still obtain better performance with TSAIL. To go one step further, it is possible to adjust base model hyperparameters to discover an optimal strategy for the trade-off between model training speed and prediction accuracy. This is a promising future research topic and automated machine learning techniques can be applied.

TABLE I
PERFORMANCE IMPROVEMENT OF TSAIL ON GRAPH-BASED TRAFFIC PREDICTORS

	NI-BJ		NI-SH		HK-RT		PeMS-BAY		PeMS-D7	
	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE
GWN	10.16%	5.08	11.06%	5.20	8.62%	6.66	2.56%	2.15	4.83%	3.72
+TSAIL	9.62%	4.81	10.36%	4.88	8.01%	6.19	2.48%	2.09	4.52%	3.50
	+5.28%	+5.23%	+6.28%	+6.23%	+7.02%	+7.11%	+3.03%	+2.85%	+6.42%	+5.96%
+TSAIL/S	9.70%	4.86	10.44%	4.91	8.10%	6.26	2.51%	2.10	4.55%	3.52
	+4.48%	+4.44%	+5.58%	+5.65%	+5.97%	+5.99%	+2.11%	+2.35%	+5.77%	+5.37%
GMAN	10.10%	5.06	10.75%	5.05	9.57%	7.39	4.12%	3.46	7.00%	5.42
+TSAIL	9.22%	4.61	9.80%	4.60	8.82%	6.82	3.85%	3.23	6.50%	5.03
	+8.70%	+8.85%	+8.80%	+8.82%	+7.85%	+7.73%	+6.54%	+6.53%	+7.09%	+7.25%
+TSAIL/S	9.30%	4.66	9.89%	4.64	8.90%	6.88	3.90%	3.27	6.58%	5.10
	+7.97%	+7.90%	+7.99%	+7.96%	+7.01%	+6.88%	+5.35%	+5.39%	+5.97%	+5.95%
STFGNN	9.42%	4.73	9.92%	4.65	6.92%	5.34	2.36%	1.98	5.23%	4.03
+TSAIL	8.94%	4.47	9.43%	4.43	6.76%	5.22	2.26%	1.90	4.93%	3.81
	+5.19%	+5.39%	+4.94%	+4.82%	+2.32%	+2.36%	+4.20%	+4.22%	+5.71%	+5.52%
+TSAIL/S	9.03%	4.53	9.51%	4.47	6.83%	5.28	2.30%	1.93	4.96%	3.83
	+4.19%	+4.23%	+4.07%	+4.02%	+1.25%	+1.08%	+2.67%	+2.73%	+5.11%	+4.96%
T-GCN	10.30%	5.16	10.78%	5.07	9.02%	6.98	3.65%	3.07	6.94%	5.37
+TSAIL	9.51%	4.77	9.79%	4.60	8.59%	6.65	3.45%	2.90	6.45%	4.98
	+7.68%	+7.68%	+9.16%	+9.27%	+4.74%	+4.70%	+5.51%	+5.65%	+7.07%	+7.28%
+TSAIL/S	9.59%	4.80	9.92%	4.66	8.67%	6.70	3.49%	2.93	6.52%	5.03
	+6.94%	+6.93%	+7.97%	+7.96%	+3.93%	+4.00%	+4.51%	+4.70%	+6.03%	+6.28%
GAMCN	9.78%	4.89	10.10%	4.75	7.58%	5.86	2.43%	2.04	4.77%	3.69
+TSAIL	9.19%	4.61	9.85%	4.63	7.26%	5.61	2.32%	1.95	4.68%	3.62
	+6.05%	+5.81%	+2.49%	+2.52%	+4.19%	+4.24%	+4.35%	+4.33%	+1.94%	+1.90%
+TSAIL/S	9.29%	4.65	9.94%	4.67	7.32%	5.67	2.34%	1.97	4.76%	3.67
	+4.94%	+4.94%	+1.57%	+1.64%	+3.36%	+3.32%	+3.63%	+3.55%	+0.27%	+0.59%
STSSN	10.35%	5.18	10.97%	5.16	8.38%	6.47	2.61%	2.19	5.08%	3.94
+TSAIL	9.56%	4.80	10.36%	4.87	8.13%	6.29	2.50%	2.10	4.73%	3.65
	+7.60%	+7.43%	+5.60%	+5.63%	+2.94%	+2.87%	+4.10%	+4.12%	+6.99%	+7.33%
+TSAIL/S	9.69%	4.85	10.46%	4.91	8.19%	6.33	2.53%	2.13	4.77%	3.68
	+6.38%	+6.51%	+4.71%	+4.75%	+2.29%	+2.29%	+2.99%	+3.04%	+6.21%	+6.61%

While TSAIL can effectively be considered as an ensemble of adaptive traffic predictors, its design inherently bestows a lightweight advantage over typical ensemble predictors. Although it may seem plausible to form an ensemble by combining multiple base models (e.g., GWN) with the TSAIL prediction aggregator to merge individual outputs, our offline experiments indicate that the large number of full-fledged base models employed in this mixture-of-expert model makes training nearly intractable. Indeed, attempting to fit the model parameters and gradient information for the numerous base models on the available GPUs is unfeasible. In an attempt to address this challenge, we conducted experiments with a variant of such an ensemble, involving sequentially and asynchronously training the base GWN models and the aggregator using the NI-BJ dataset. Subsequently, predictions from this ensemble were tested against those from a single GWN model using Wilcoxon rank sum tests on the null hypothesis that the GWN ensemble performs similarly to the standalone GWN model. The statistical test results at a 95% significance level, indicate that these two approaches show no noticeable difference. This finding underscores the advantage of the proposed TSAIL framework, demonstrating its efficacy when compared to general ensemble approaches.

C. Autoregressive Prediction

All statistical findings from earlier tests are based on predictions for the following five minutes. In practice, transport operators may adopt an autoregressive approach that uses predictions from previous time instances as input to predict the next ones. This approach inevitably accumulates forecast errors when the prediction horizon expands, rendering degrading model performance over time. Notwithstanding, we argue that the proposed TSAIL decomposes the complex spatial-temporal data correlation for models to exploit so that the degradation is attenuated. We conduct an autoregressive prediction test in this subsection and present empirical support for the aforementioned hypothesis.

Fig. 3 depicts the MAPE and RMSE trends of TSAIL(/S) on the six base predictors with all investigated datasets. In the figure, the horizontal axis stands for the time horizon of

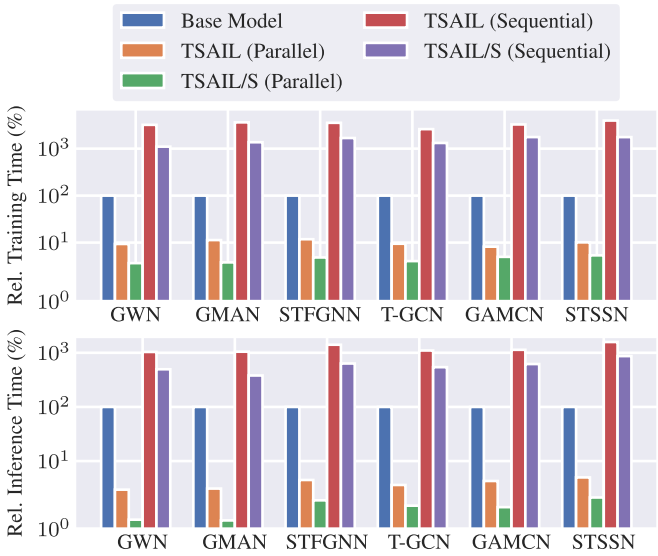

Fig. 2. Relative training and inference time of base models, +TSAIL, and +TSAIL/S on NI-BJ dataset. The latter two can be executed parallelly where all base models are trained in parallel, or sequentially where models are trained in a sequence. The training and inference times of the base models are set to 100%.

prediction, ranging from 5min to 60min with a constant 5min interval. Hence, the slopes of metric curves indicate the rate of performance degradation over time, i.e., the less tilt the better. From the statistical results, several conclusions can be drawn. In particular, all tested approaches — with or without TSAIL — experience error accumulation due to the autoregressive prediction nature yet with different strengths. While the best-performing base predictors are not identical for all datasets, incorporating TSAIL always introduces performance improvements. Additionally, the degree of improvements generally are expanding over time, which can be observed from the increasing performance gap along the prediction horizon. The average MAPE improvement obtained by TSAIL on each of the five datasets is 6.75%/6.21%/4.85%/4.62%/5.87% for 5min-ahead predictions, respectively. The improvements are almost doubled to 10.58%/8.22%/8.62%/8.76%/7.45% when 30min-ahead predictions are generated and are constantly growing to 11.91%/9.46%/9.67%/9.47%/8.59% for 60min-ahead ones. The result implies that TSAIL effectively weakens the effect of predicting error accumulation and in turn helps graph-based traffic predictors better accommodate long-term traffic prediction tasks.

D. Ablation Study

Section III elaborates on the modular design of TSAIL, whose components are depicted in Fig. 1. While previous tests demonstrate notable performance improvement brought by TSAIL on graph-based predictors, it is worth investigating which constituting components of the model contribute most to the improvement. In this subsection, we present four ablation variants of TSAIL and TSAIL/S to empirically evaluate each component's necessity. GWN is used as the base traffic predictor

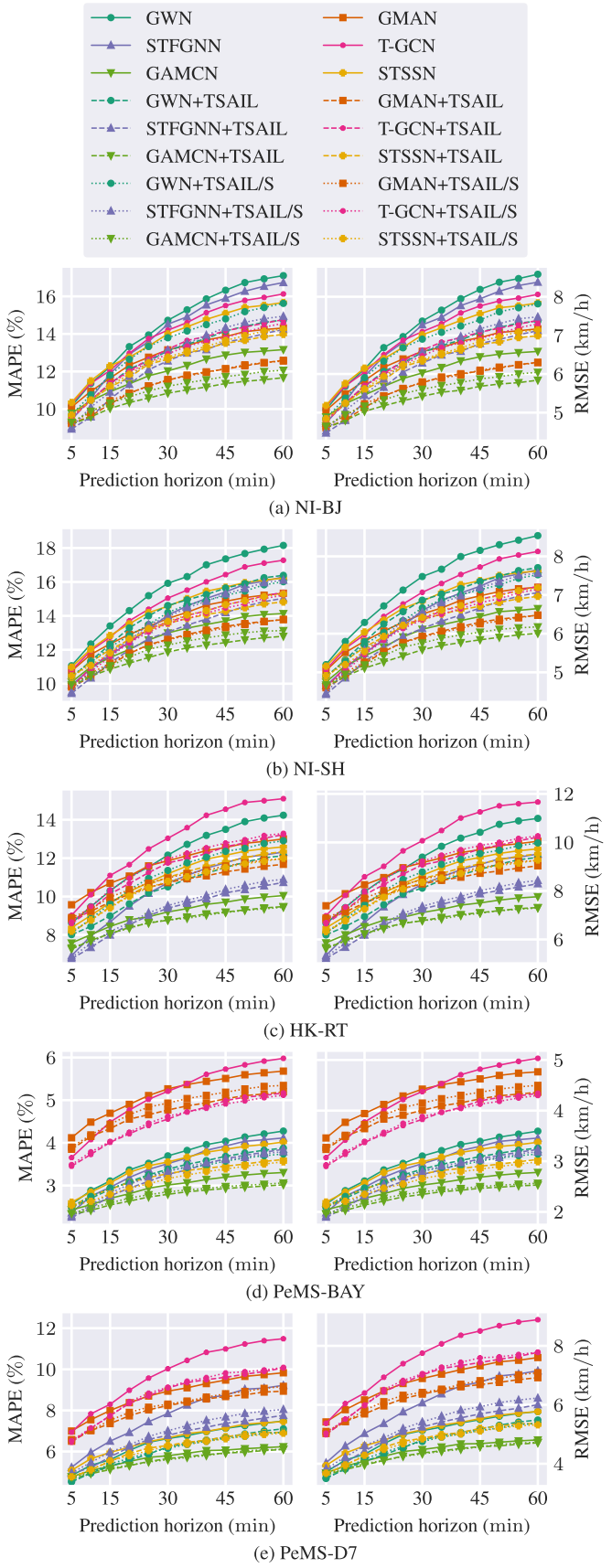

Fig. 3. Performance of base models, +TSAIL, and +TSAIL/S with lengthening prediction horizons on five datasets.

TABLE II
PERFORMANCE OF TSAIL VARIANTS ON GWN

	NI-BJ		NI-SH		HK-RT		PeMS-BAY		PeMS-D7	
	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE
GWN	10.16%	5.08	11.06%	5.20	8.62%	6.66	2.56%	2.15	4.83%	3.72
+TSAIL	9.59%	4.80	10.36%	4.88	8.01%	6.19	2.48%	2.09	4.52%	3.50
-AGG	9.66%	4.83	10.40%	4.89	8.06%	6.23	2.50%	2.10	4.55%	3.52
	−0.73%	−0.73%	−0.31%	−0.19%	−0.55%	−0.67%	−0.83%	−0.62%	−0.58%	−0.51%
-ADP	9.66%	4.85	10.48%	4.93	8.08%	6.24	2.52%	2.12	4.56%	3.52
	−0.82%	−1.01%	−1.09%	−1.04%	−0.84%	−0.86%	−1.43%	−1.25%	−0.96%	−0.68%
-TSS	9.76%	4.89	10.57%	4.97	8.14%	6.29	2.53%	2.13	4.60%	3.55
	−1.80%	−1.84%	−2.02%	−1.97%	−1.59%	−1.65%	−2.03%	−1.91%	−1.89%	−1.56%
-VTN	9.64%	4.82	10.42%	4.89	8.05%	6.21	2.49%	2.09	4.54%	3.50
	−0.56%	−0.52%	−0.51%	−0.34%	−0.42%	−0.44%	−0.24%	−0.23%	−0.45%	−0.05%
GWN	10.16%	5.08	11.06%	5.20	8.62%	6.66	2.56%	2.15	4.83%	3.72
+TSAIL/S	9.68%	4.85	10.44%	4.90	8.10%	6.26	2.51%	2.10	4.55%	3.52
-AGG	9.77%	4.88	10.51%	4.94	8.16%	6.30	2.53%	2.12	4.59%	3.54
	−0.92%	−0.62%	−0.72%	−0.74%	−0.69%	−0.61%	−0.77%	−1.06%	−0.82%	−0.53%
-ADP	9.78%	4.89	10.54%	4.95	8.19%	6.34	2.53%	2.13	4.59%	3.55
	−1.04%	−0.79%	−0.96%	−1.01%	−1.07%	−1.20%	−1.09%	−1.40%	−0.88%	−0.89%
-TSS	9.88%	4.95	10.62%	4.99	8.28%	6.40	2.55%	2.14	4.68%	3.61
	−2.08%	−2.07%	−1.76%	−1.75%	−2.12%	−2.16%	−1.78%	−2.11%	−2.76%	−2.55%
-VTN	9.72%	4.87	10.48%	4.92	8.15%	6.29	2.52%	2.12	4.58%	3.54
	−0.44%	−0.29%	−0.38%	−0.27%	−0.59%	−0.49%	−0.52%	−0.79%	−0.71%	−0.60%

and offline preliminary tests show that all other base approaches demonstrate highly similar performance variations.

Table II summarizes the model performance of the four variants on GWN with TSAIL and TSAIL/s. Particularly,

- "-AGG" removes the final prediction aggregator module from TSAIL. In this variant, each sub-graph is predicted once via the adaptive traffic predictor. The results are directly concatenated without post-processing adjustments.
- "-ADP" discards the layer-wise regressor and the subsequent attention-based pooling in the adaptive traffic predictor module. The last-layer output of GWN is considered final in this module.
- "-TSS" drops the time-series slicer module of TSAIL so that the GWN in the adaptive traffic predictor accepts complete time series from all nodes in the transportation network. This variant is conceptually identical to an attention-based ensemble without spatial-temporal decomposition.
- "-VTN" scraps the virtual nodes added to METIS subgraphs for preserving multi-hop spatial correlation. By abandoning these nodes, this variant may have isolated sub-graphs for one GWN.

Besides the absolute MAPE and RMSE obtained by each variant, the relative performance deviation in percentage from TSAIL and TSAIL/S are also presented for better comprehension.

The comparison indicates that all three major modules in TSAIL contribute to performance improvement, although their importance is not identical. On average, 0.60%/0.81%/1.16% MAPE degradation is observed by removing the prediction aggregator, adaptive traffic predictor, and time-series slicer modules, respectively. The virtual nodes in the time-series slicer also provide an average 0.98% MAPE improvement. Furthermore, the results also imply that these modules are conjugated in terms of spatial-temporal dependency decomposition and exploitation: the inter-correlation among the modules also gives a share of the average 6.19% MAPE improvement considering that they alone add up to 3.55% only. Finally, reducing base predictor capacity (TSAIL/S) does not diminish their merits. Indeed, MAPE gaps are slightly enlarged to 0.78%/0.90%/1.29%/1.10%, respectively. This is due to that the smaller model capacities of traffic predictors rely more on the fine spatial-temporal decomposition provided by TSAIL modules, and layer-wise regression and attention-pooling help them adaptive select the optimal network depth by learning the data.

E. Training With Data Stream

In Section IV, we present a TSAIL-based traffic predictor training scheme with streaming data, so that the underlying deep models are consistently updated with the latest data reflecting the ever-changing traffic dynamics. In this subsection, we investigate the efficacy of this scheme in improving model sustainability. Same as Section V-D, GWN is selected to be the base traffic predictor. Vanilla GWN, +TSAIL, and +TSAIL/S flavors all employ data of the first month from datasets NI-BJ and HK-RT for model warm-up. Notwithstanding, the latter two both updates the model parameters every day with the immediate latest ground truth in the past day with the stream learning scheme in Section IV. It is acknowledged that the comparison is not a fair one as TSAIL variants employ more training data in the model, which should lead to better performance by intuition. The case study is instead more an illustration of whether stream learning eliminates potential model aging as intended.

The simulation results are depicted in Fig. 4, in which the day-average MAPE of the three tested models and schemes are

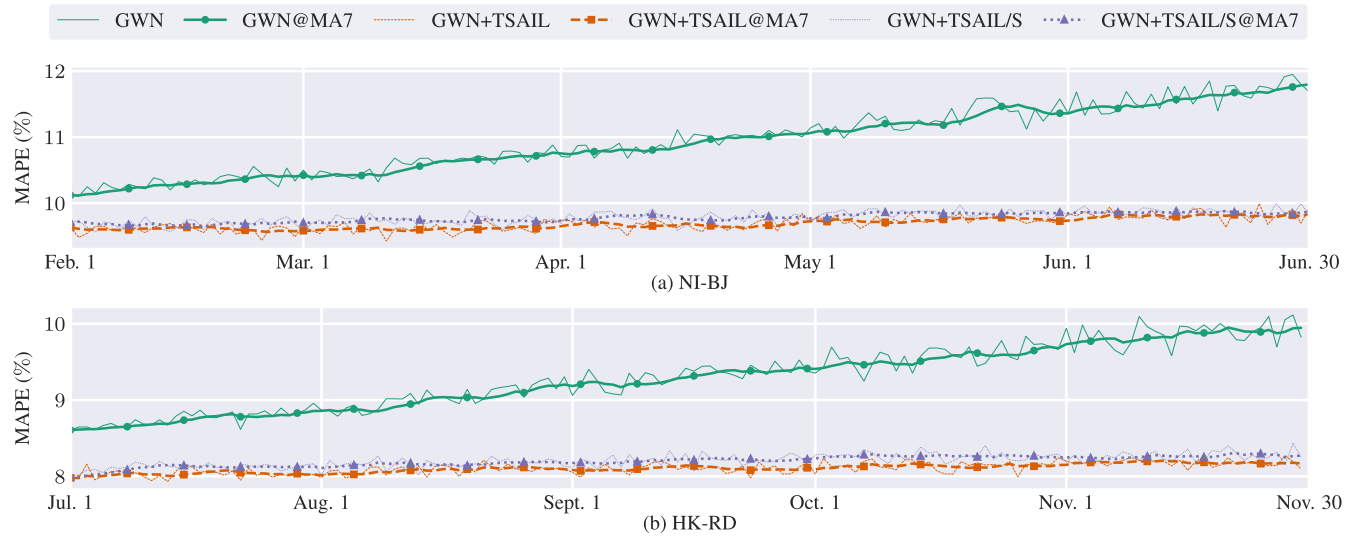

Fig. 4. Performance of GWN, +TSAIL, and +TSAIL/S learning from data stream.

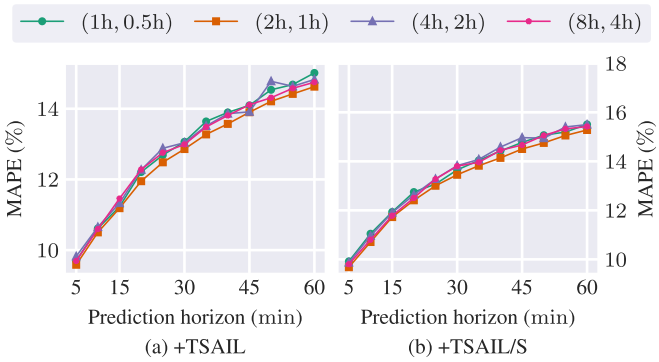

Fig. 5. TSAIL sensitivity to hyperparameter tuple (T, Δ).

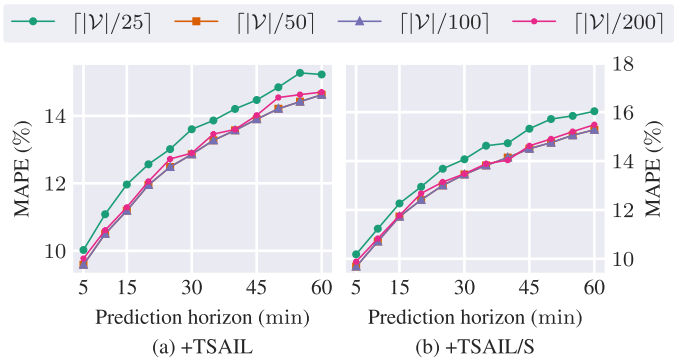

Fig. 7. TSAIL sensitivity to hyperparameter K.

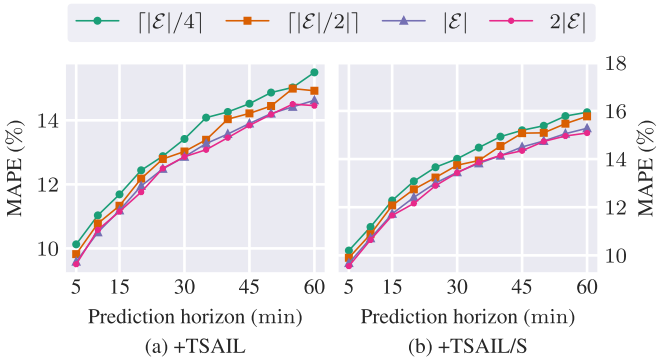

Fig. 6. TSAIL sensitivity to hyperparameter ζ.

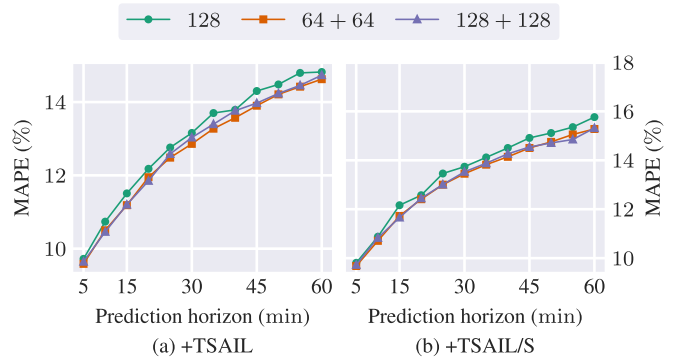

Fig. 8. TSAIL sensitivity to attention network structure.

presented with their respective 7-day moving average plotted for better comprehension, labeled by "@MA7". From the comparison, it can be found that model aging is indeed a notable issue in GWN, as the prediction performance gradually degenerates with time. This reflects the necessity of constantly re-training traffic predictors to cope with changing traffic dynamics. Meanwhile, both TSAIL variants demonstrate quite stable performance variations during the tested horizon. Minuscule MAPE degradation is developed by comparing the prediction performance at the beginning and end of the horizons for both datasets. This observation reveals that TSAIL with stream learning is a highly competitive method for handling both the model accuracy and

sustainability challenges. The scheme effectively reduces the frequency of model re-training from scratch — or even obviates the need — so that the overall computation can be alleviated thanks to the unique prediction aggregator design in TSAIL.

F. Hyperparameter Sensitivity

In the proposed TSAIL scheme, a few hyperparameters are adopted to control the behavior of the time-series slicer and prediction aggregator, namely, time slicing parameters T and Δ (default 2h and 1h), cross-correlation edge count ζ (default $|\mathcal{E}|$), number of sub-graphs K (default $\lceil|\mathcal{V}/100|\rceil$), and neuron count in attention networks (default two layers with 64 each). In this subsection, we carry out a series of hyperparameter tests to examine the sensitivity of TSAIL on them. Multiple values of each hyperparameter are tested on GWN+TSAIL(/S) and NI-BJ dataset. All other simulation configurations are set identical to previous case studies.

We first discuss the influence of changing the time horizon slicing parameter, i.e., T and Δ first introduced in Section III-C. Besides the default value tuple, we further test the GWN performance with $(T, \Delta) \in \{(1\,\mathrm{h}, 0.5\,\mathrm{h}), (4\,\mathrm{h}, 2\,\mathrm{h}), (8\,\mathrm{h}, 4\,\mathrm{h})\}$. The simulation results are presented in Fig. 5. A general conclusion can be developed from the plot that TSAIL is not notably sensitive to the length of time slices. While the default $(2\mathrm{h}, 1\mathrm{h})$ is preferred by the test, changing the values leads to less than 2% performance deviations in one-step predictions. This observation can be credited to the outstanding time-series learning capability of the base model, i.e., GWN. Nonetheless, we hypothesize that further shrinking or expanding the time slices would eventually undermine the TSAIL efficacy. An adaptive time slicing scheme that considers the traffic dynamics can be the answer to time slicing, which is among the future directions of this work.

We further present the sensitivity of the other two spatial slicing parameters, i.e., ζ and K introduced in Section III-C. The summarized performance of $\zeta \in \{\lceil|\mathcal{E}/4|\rceil, \lceil|\mathcal{E}/2|\rceil, |\mathcal{E}|, 2|\mathcal{E}|\}$ and $K \in \{\lceil|\mathcal{V}/25|\rceil, \lceil|\mathcal{V}/50|\rceil, \lceil|\mathcal{V}/100|\rceil, \lceil|\mathcal{V}/200|\rceil\}$ are depicted in Figs. 6 and 7, respectively. Increasing ζ renders more node-correlation edges added during the spatial slicing step, and decreasing K (left to right in the legend of Fig. 7) reduces the number of sub-graphs. The results indicate that adding node-correlation edges to sub-graphs typically improves the overall model performance, albeit the gain saturates at a particular count ($|\mathcal{E}|$ in our case). In the meantime, TSAIL is quite tolerant of the total number of sub-graphs as long as the local nodal connectivity is guaranteed, i.e., sufficient nodes exist in each sub-graph. Apart from extreme values for either hyperparameter, TSAIL develops satisfactory performance on prediction accuracy, and we recommend the default $|\mathcal{E}|$ and $\lceil|\mathcal{V}/100|\rceil$ as a general setting for typical prediction models and datasets.

Finally, TSAIL adopts a two-layered fully connected neural network as the attention network in the adaptive traffic predictor and the prediction aggregator. The objective of this network is to combine predictions developed from multiple base predictors and generate the final one. We also test two other structures as alternatives to the attention network with one layer of 128 neurons and two layers of 128 neurons each, respectively, where the results are demonstrated in Fig. 8. From the comparison, it can be concluded that TSAIL is not sensitive to the attention network structure as long as sufficient non-linearity can be provided, i.e., two or more layers are preferred. Considering that the network training time is proportional to the model capacity defined by the number of layers and neurons, the default $64 + 64$ setting is favored over other tested variants.

VI. Conclusion

In this paper, we propose a traffic slicing and adaptive learning scheme to provide scalable and sustainable data learning for graph-based traffic predictors. The proposed scheme aims at addressing the current low model complexity efficiency challenge while resolving the difficulty in determining the optimal learning model size and integrating the live traffic data stream. By adopting a unique time-series slicing module, the proposed TSAIL avoids the common practice of enhancing prediction accuracy via increasing the model complexity but instead breaks down the big prediction task into exponentially smaller ones handled by simple predictors. The subsequent adaptive traffic predictor further employs layer-wise hidden data paths to enable predictors to adaptively select model depths, leading to model capacity scalability. Finally, an attention-driven prediction aggregator resembles the principle of ensemble learning for prediction calibration and allows for a novel stream data learning algorithm tailor-made for the scheme.

To evaluate the efficacy of the proposed scheme, we conduct a series of comprehensive case studies on five real-world datasets from different data sources. The simulation results indicate that an average 6.19% prediction error reduction can be achieved over the existing state of the arts. Considering that the scheme serves as an add-on to current and future models, TSAIL is a universal performance booster for graph-based traffic predictors. Further, the case studies also reveal that the scheme can provide better autoregressive long-term predictions and incorporate traffic dynamics changes lively with traffic data streams. A set of hyperparameter tests are also carried out to illustrate the impact of model hyperparameters, and to develop guidelines for selecting TSAIL parameters.

In the future, we plan to investigate advanced multitask learning techniques to further alleviate the computation effort of training multiple models. Futher, recent development of traffic predictors that incorporates abnormal/incident data has the potential of being integrated with TSAIL for better all-condition practicality. We look forward to follow-up research efforts on general schemes for boosting traffic predictor performance.

References

[1] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, "Big data analytics in intelligent transportation systems: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 1, pp. 383–398, Jan. 2019.

[2] X. Yin, G. Wu, J. Wei, Y. Shen, H. Qi, and B. Yin, "Deep learning on traffic prediction: Methods, analysis, and future directions," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 4927–4943, Jun. 2022.

[3] R. Jiang et al., "DL-Traff: Survey and benchmark of deep learning models for urban traffic prediction," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2021, pp. 4515–4525.

[4] A. M. Nagy and V. Simon, "Survey on traffic prediction in smart cities," *Pervasive Mobile Comput.*, vol. 50, pp. 148–163, 2018.
[5] I. Lana, J. Del Ser, M. Velez, and E. I. Vlahogianni, "Road traffic forecasting: Recent advances and new challenges," *IEEE Intell. Transp. Syst. Mag.*, vol. 10, no. 2, pp. 93–109, Summer 2018.
[6] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," in *Proc. Int. Joint Conf. Artif. Intell.*, Macao, China, 2019, pp. 1907–1913.
[7] C. Zheng, X. Fan, C. Wang, and J. Qi, "GMAN: A graph multi-attention network for traffic prediction," in *Proc. AAAI Conf. Artif. Intell.*, New York, USA, 2020, pp. 1234–1241.
[8] M. Li and Z. Zhu, "Spatial-temporal fusion graph neural networks for traffic flow forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 4189–4196.
[9] S. Pouyanfar et al., "A survey on deep learning: Algorithms, techniques, and applications," *ACM Comput. Surv.*, vol. 51, no. 5, pp. 1–36, Sep. 2018.
[10] Y. Yang, D.-W. Zhou, D.-C. Zhan, H. Xiong, and Y. Jiang, "Adaptive deep models for incremental learning: Considering capacity scalability and sustainability," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, Anchorage, AK, USA, 2019, pp. 74–82.
[11] H. M. Gomes, J. Read, A. Bifet, J. P. Barddal, and J. Gama, "Machine learning for streaming data: State of the art, challenges, and opportunities," *ACM SIGKDD Explorations Newslett.*, vol. 21, no. 2, pp. 6–22, Nov. 2019.
[12] D. A. Tedjopurnomo, Z. Bao, B. Zheng, F. M. Choudhury, and A. K. Qin, "A survey on modern deep neural network for traffic prediction: Trends, methods and challenges," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 4, pp. 1544–1561, Apr. 2022.
[13] L. Vanajakshi and L. R. Rilett, "A comparison of the performance of artificial neural networks and support vector machines for the prediction of traffic speed," in *Proc. IEEE Intell. Veh. Symp.*, 2004, pp. 194–199.
[14] K.-H. N. Bui, J. Cho, and H. Yi, "Spatial-temporal graph neural network for traffic forecasting: An overview and open research issues," *Appl. Intell.*, vol. 52, no. 3, pp. 2763–2774, Jun. 2021.
[15] W. Jiang and J. Luo, "Graph neural network for traffic forecasting: A survey," *Expert Syst. Appl.*, vol. 207, 2022, Art. no. 117921.
[16] L. Zhao et al., "T- GCN: A temporal graph convolutional network for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3848–3858, Sep. 2020.
[17] T. Mallick, P. Balaprakash, E. Rask, and J. Macfarlane, "Graph-partitioning-based diffusion convolutional recurrent neural network for large-scale traffic forecasting," *Transp. Res. Rec.*, vol. 2674, no. 9, pp. 473–488, 2020.
[18] C. Zhang, S. Zhang, X. Zou, S. Yu, and J. J. Q. Yu, "Towards large-scale graph-based traffic forecasting: A data-driven network partitioning approach," *IEEE Internet Things J.*, vol. 10, no. 5, pp. 4506–4519, Mar. 2023.
[19] C. N. Yahia, V. Pandey, and S. D. Boyles, "Network partitioning algorithms for solving the traffic assignment problem using a decomposition approach," *Transp. Res. Record: J. Transp. Res. Board*, vol. 2672, no. 48, pp. 116–126, Oct. 2018.
[20] P. Johnson, D. Nguyen, and M. Ng, "Large-scale network partitioning for decentralized traffic management and other transportation applications," *J. Intell. Transp. Syst.*, vol. 20, no. 5, pp. 461–473, Apr. 2016.
[21] K. An, Y.-C. Chiu, X. Hu, and X. Chen, "A network partitioning algorithmic approach for macroscopic fundamental diagram-based hierarchical traffic network management," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 4, pp. 1130–1139, Apr. 2018.
[22] R. Saedi, M. Saeedmanesh, A. Zockaie, M. Saberi, N. Geroliminis, and H. S. Mahmassani, "Estimating network travel time reliability with network partitioning," *Transp. Res. Part C: Emerg. Technol.*, vol. 112, pp. 46–61, Mar. 2020.
[23] Y. H. Lau and R. C.-W. Wong, "Spatio-temporal graph convolutional networks for traffic forecasting: Spatial layers first or temporal layers first?," in *Proc. Int. Conf. Adv. Geographic Inf. Syst.*, Beijing, China, 2021, pp. 427–430.
[24] Y. Wu, H. Tan, L. Qin, B. Ran, and Z. Jiang, "A hybrid deep learning based traffic flow prediction method and its understanding," *Transp. Res. Part C: Emerg. Technol.*, vol. 90, pp. 166–180, 2018.
[25] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proc. AAAI Conf. Artif. Intell.*, San Francisco, CA, USA, 2017, pp. 1655–1661.
[26] J. J. Q. Yu, C. Markos, and S. Zhang, "Long-term urban traffic speed prediction with deep learning on graphs," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 7359–7370, Jul. 2022.
[27] G. Karypis and V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 359–392, 1998.
[28] T. Chen, I. Goodfellow, and J. Shlens, "Net2Net: Accelerating learning via knowledge transfer," in *Proc. Int. Conf. Learn. Representations*, San Juan, Puerto Rico, 2016, pp. 1–12.
[29] A. Chaudhry, P. K. Dokania, T. Ajanthan, and P. H. S. Torr, "Riemannian walk for incremental learning: Understanding forgetting and intransigence," in *Proc. Eur. Conf. Comput. Vis.*, Munich, Germany, 2018, pp. 556–572.
[30] S.-W. Lee, J.-H. Kim, J. Jun, J.-W. Ha, and B.-T. Zhang, "Overcoming catastrophic forgetting by incremental moment matching," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, Long Beach, CA, USA, 2017, pp. 4655–4665.
[31] R. Pascanu and Y. Bengio, "Revisiting natural gradient for deep networks," in *Proc. Int. Conf. Learn. Representations*, Banff, Canada, 2014, pp. 1–12.
[32] J. Kirkpatrick et al., "Overcoming catastrophic forgetting in neural networks," *Proc. Nat. Acad. Sci.*, vol. 114, no. 13, pp. 3521–3526, 2017.
[33] J. J. Q. Yu, "Graph construction for traffic prediction: A data-driven approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 15015–15027, Sep. 2022.
[34] J. J. Q. Yu, "Citywide traffic speed prediction: A geometric deep learning approach," *Knowl.-Based Syst.*, vol. 212, Nov. 2020, Art. no. 106592.
[35] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *Proc. Int. Conf. Learn. Representations*, Vancouver, Canada, 2018, pp. 1–16.
[36] J. Qi, Z. Zhao, E. Tanin, T. Cui, N. Nassir, and M. Sarvi, "A graph and attentive multi-path convolutional network for traffic prediction," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 7, pp. 6548–6560, Jul. 2023.
[37] S. Cao, L. Wu, J. Wu, D. Wu, and Q. Li, "A spatio-temporal sequence-to-sequence network for traffic flow prediction," *Inform. Sci.*, vol. 610, pp. 185–203, 2022.

James Jianqiao Yu (Senior Member, IEEE) received the BEng and PhD degrees in electrical and electronic engineering from the University of Hong Kong, Pokfulam, Hong Kong, in 2011 and 2015, respectively. He was a post-doctoral fellow with the University of Hong Kong from 2015 to 2018, and was an assistant professor with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China from 2018 to 2023. He is currently a lecturer with the Department of Computer Science, the University of York, United Kingdom. His general research interests are in intelligent transportation systems, privacy computing, deep learning, and smart cities. His work is now mainly on spatial-temporal data mining, forecasting and logistics of future transportation systems, and artificial intelligence techniques for industrial applications. He has published more than 100 academic papers in top international journals and conferences, and representative papers have been selected as ESI highly cited papers. He was the World's Top 2% Scientists by Stanford University. He is an editor of the *IET Smart Cities Journal*.