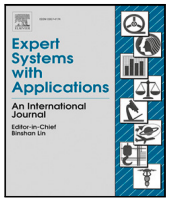
Deep knowledge distillation: A self-mutual learning framework for traffic prediction

Expert Systems With Applications 252 (2024) 124138

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa



Deep knowledge distillation: A self-mutual learning framework for traffic prediction

Ying Li [a], Ping Li [a], Doudou Yan [a], Yang Liu [b,*], Zhiyuan Liu [c]

[a] School of Information Engineering, Chang'an University, Xi'an, China
[b] Department of Architecture and Civil Engineering, Chalmers University of Technology, Gothenburg, Sweden
[c] Jiangsu Key Laboratory of Urban ITS, Jiangsu Province Collaborative Innovation Center of Modern Urban Traffic Technologies, School of Transportation, Southeast University, China

ARTICLE INFO

Keywords:
Traffic prediction
Spatio-temporal characteristics
Graph neural network
Knowledge distillation

ABSTRACT

Traffic flow prediction in spatio-temporal networks is a crucial aspect of Intelligent Transportation Systems (ITS). Existing traffic flow forecasting methods, particularly those utilizing graph neural networks, encounter limitations. When processing large-scale graph data, the depth of these models can restrict their ability to effectively capture complex relationships and patterns. Additionally, these methods often focus mainly on local neighborhood information, which can limit their capability to recognize and analyze global relationships and patterns within the graph data. Therefore, we proposed a deep knowledge distillation model, tailored to effectively capture spatio-temporal patterns in traffic flow prediction. This model incorporates a bidirectional random walk process on a directed graph, enabling it to effectively capture both spatial and temporal dependencies. Utilizing a blend of mutual learning and self-distillation, our approach enhances the detection of spatio-temporal relationships within traffic data and improves the feature perception ability at both local and global levels. We tested our model on two real-world datasets, achieving notable improvements in prediction accuracy, especially for predictions within a one-hour timeframe. In comparison to the baseline model, our proposed model achieved accuracy improvements of 0.19 and 0.18 on the respective datasets. These results highlight the success of using mutual learning and self-distillation to transfer knowledge effectively within and between models and to improve the model's capability in identifying and extracting features.

1. Introduction

The utilization of artificial intelligence (AI) has significant implications in the field of transportation (Fei et al., 2022; Liu et al., 2022; Ma, Wang, Yang, & Yang, 2020), with traffic prediction serving as a cornerstone of Intelligent Transportation Systems (ITS) (Jie, Xiaofei, Bo, & Zhigang, 2022; Liu, Liu, Lyu, & Ye, 2019; Liu, Lyu et al., 2021). Traffic prediction encompasses various aspects, including traffic flow data forecasting (Li, Yu, Shahabi and Liu, 2017), trajectory prediction (Bing et al., 2022; He, Liu, Yang, & Qu, 2024), vehicle dispatching (Li, Li, Jia, Zeng, & Wang, 2022; Xu et al., 2022; Yue, Abdel-Aty, & Wang, 2022) and traffic incident detection (Acharya & Mekker, 2022; Dabiri & Kulcsár, 2022). Traffic flow prediction is used to estimate upcoming traffic conditions, such as volume or speed, based on previously observed data from road networks (Fei, Shi, Li, Liu, & Qu, 2024; Liu et al., 2023; Zhong, Wu, Zhang, & Ma, 2023). Accurate and real-time traffic prediction is vital for our daily lives and can improve the efficiency of decision-making for transportation agencies. Since traffic data is spatially correlated and time-dependent (Gan et al., 2022), traffic prediction represents a typical spatio-temporal data forecasting task. Based on the prediction duration, traffic flow predictions can be primarily categorized into two types (Zhou et al., 2022): short-term prediction (within 30 min) and long-term prediction (more than 30 min).

Over the past few years, the mainstream traffic prediction methods have been divided into three categories: parametric, non-parametric, and hybrid methods. Parametric methods, also known as statistical-based models, primarily consider sequence correlations and have been widely utilized in traffic prediction. One popular parametric method is the auto-regressive integrated moving average (ARIMA) model (Ahmed & Cook, 1979) and its related variants. The second category is machine learning such as Support Vector Regression (SVR) (Li & Xu, 2021) etc., which shows promise in enhancing the accuracy and reliability of time series forecasting. However, they may need massive data to

* Corresponding author.
E-mail addresses: yingli@chd.edu.cn (Y. Li), 2021224096@chd.edu.cn (P. Li), 2020224050@chd.edu.cn (D. Yan), liuy@chalmers.se (Y. Liu), zhiyuanl@seu.edu.cn (Z. Liu).

https://doi.org/10.1016/j.eswa.2024.124138
Received 25 May 2023; Received in revised form 6 December 2023; Accepted 29 April 2024
Available online 10 May 2024

achieve high prediction accuracy. The third category comprises deep learning methods. Among them, recurrent neural network (RNN) (Ma, Yu, Wang, & Wang, 2015) and convolution neural network (CNN) (Ma et al., 2017) are represented, which can capture temporal and spatial correlation, respectively. However, they cannot fully capture the complex features of traffic data. To address this limitation, researchers have started modeling traffic data as a graph structure, with graph neural networks (GNNs) (Jiang & Luo, 2022) becoming cutting-edge methods for traffic prediction. This development has led to the emergence of graph convolutional neural networks (GCN) (Zhao et al., 2020). Current traffic flow prediction methods can still be improved in terms of feature extraction and accuracy. Our research focuses on reducing model complexity and improving prediction accuracy. We propose a hybrid network, called deep knowledge distillation, based on the graph neural network that combines self-distillation with mutual learning. The model is composed of an encoder, which handles traffic features, and a decoder, which produces sequence predictions. The contributions of this paper are as follows:

(1) We introduce a novel deep knowledge distillation model to capture spatio-temporal correlations, representing traffic flow as a bidirectional random walk process on a directed graph and capturing spatial dependencies. Furthermore, we employ an encoder–decoder structure to capture temporal dependencies. Unlike existing methods, our approach significantly enhances local and global feature perception and extraction capabilities within graph neural networks, resulting in improved accuracy in traffic flow forecasting.

(2) We add self-distillation and mutual learning based on the graph neural networks. Self-distillation creates a stronger link between shallow and deep structures, enabling the shallow structure to acquire more extensive knowledge from the deep structure, thereby enhancing the model's expressive and learning capabilities. Mutual learning encourages the two networks to learn global information together, optimizing the model to enhance connectivity within the graph neural network and effectively utilize global information for prediction and reasoning.

(3) We have performed experiments on two well-known traffic datasets to validate the model's effectiveness. Our results suggest that deep knowledge distillation accurately captures overall traffic flow patterns. Additionally, our model surpasses other commonly utilized models in terms of predictive accuracy. Compared to the baseline model, the MAE (mean absolute error) decreases by 0.19 on the METR-LA dataset. Meanwhile, the PEMS-BAY dataset, which contains more sensors, exhibits a 0.18 decrease in MAE for the one-hour prediction performance index.

The subsequent sections of this paper are organized as follows: Section 2 reviews the existing methods of knowledge distillation and traffic prediction. Section 3 elaborates on the architecture and specific details of Deep Knowledge Distillation. Section 4 presents a thorough experimental analysis, including a comparison of our results with those of other models. Finally, Section 5 concludes the study, summarizing the main findings and contributions.

2. Literature review

This section will examine the pertinent literature on knowledge distillation and traffic flow forecasting, focusing on technical principles and associated applications.

2.1. Knowledge distillation

Knowledge distillation (Hinton, Vinyals, Dean, et al., 2015) is a powerful method for compressing models, where the "knowledge" from a complex teacher model with superior learning capabilities is transferred to a simpler student model, creating a teacher–student network. By utilizing knowledge distillation, there is potential to enhance the performance of the student model by learning from the soft targets provided by the teacher model. Due to its superior performance, knowledge distillation has emerged as a popular research area in deep learning. Some researchers have also applied it to transportation (Ji, Yu, & Lei, 2022). The current categorization of knowledge distillation methods is based on the nature of the knowledge being transferred, which includes logits-based knowledge, feature-based knowledge, relational-based knowledge, and related variations like self-distillation and mutual learning (Gou, Yu, Maybank, & Tao, 2021). Next, we will introduce the research status of knowledge distillation.

2.1.1. Classic knowledge distillation

Logit-based knowledge distillation: It focuses on the logit output of the teacher's last layer, which was proposed and published by (Hinton et al., 2015), they introduced this novel approach and the notion of distillation temperature T. This technology works by training a teacher network first and then distilling its knowledge to a student network at a high temperature T, where the distillation temperature represents the softening degree of the label. As T increases, the output label is smoother. The choice of T is linked to the dimensions of the student network. When the parameter amount in the student network is relatively small, a relatively low T is sufficient. Lately, (Li, Yang et al., 2017) demonstrated that in some cases, the student network can exceed the performance of the teacher network, such as using it to solve the noise label problem in supervised learning and achieving better results. And (Kobayashi, 2022) introduced extractive distillation. This method is based on analyzing the temperature and uniformity of the teacher probability, extracting the knowledge contained in the teacher model. While the above model is easy to implement, it may be challenging for small-scale student networks to absorb the knowledge imparted by the teacher network.

Feature-based knowledge distillation: Its goal is to extract features from a teacher model and transmit these features to a student model, and to help the student model learn and generalize better. (Adriana et al., 2015) first introduced this method, named FitNets, which used the intermediate representation of the teacher model as a "hint" to assist the student in training. During this time, the student model is designed to approximate the intermediate representation of the teacher model layer by layer to learn the "knowledge" of the teacher model. (Liu, Huang et al., 2021) proposed a novel inter-channel correlation method for knowledge distillation that extracts retained feature correlations between the channels. The above model can provide more effective information when students are learning online. However, the operation is difficult due to the dimension disparity between the teacher and the intermediate layer of the student.

Relational-based knowledge distillation: This method combines the output of multiple teacher models into structural units and focuses on the correlation between the feature maps emphatically. (Park, Kim, Lu, & Cho, 2019) first proposed this concept. By closely mirroring the structural characteristics of the teacher model, we can furnish the student with more efficient guidance. This model has stronger generalization and a better effect, but it has randomness when selecting the intermediate layer's output, so the interpretability is not strong.

2.1.2. Variants based on classic knowledge distillation

The student model may not learn all the teacher's knowledge due to limitations of original knowledge distillation (Tzelepi, Passalis, & Tefas, 2021). Training a large teacher model can also be computationally expensive, which may not be feasible in certain scenarios. To address these issues, researchers have introduced enhancements such as mutual learning and self-distillation, building upon traditional knowledge distillation.

Mutual Learning: (Zhang, Xiang, Hospedales, & Lu, 2018) introduced the Deep Mutual learning (DML) approach. It disrupts the conventional hierarchical relationship between teacher and student models in knowledge distillation, allowing multiple student models to interact and collaborate during the training process. This communication enables them to perform better and achieve higher accuracy in

Table 1
Summary of literature review on knowledge distillation.

Category	Approach	Related research
Classic knowledge distillation	Logit-based knowledge distillation	(Hinton et al., 2015) and (Li, Yang et al., 2017)
	Feature-based knowledge distillation	(Adriana et al., 2015) and (Liu, Huang et al., 2021)
	Relational-based knowledge distillation	(Park et al., 2019)
Variants based on classic knowledge distillation	Mutual learning	(Wu, Feng et al., 2019; Zhai et al., 2021) and (Zhang et al., 2018)
	Self-distillation	(Ji et al., 2022; Kim et al., 2021) and (Zhang et al., 2019)

completing their tasks. At each training iteration, DML computes the predictions of both models and updates the parameters of both networks based on others' predictions. (Wu, Feng et al., 2019) suggested using multi-task interleaving supervision to guide deep networks for salient detection. In (Zhai et al., 2021), the concept of mutual learning has been expanded from regular grids to graphs by implementing mutual graph learning.

Self-Distillation: This method was first proposed by (Zhang et al., 2019), where a single neural network acts as both the teacher and student network. The deeper portion is used to supervise the shallower ones during self-learning, with the deeper portion used to supervise the shallower ones during self-learning. This approach can drastically reduce model size and computational requirements without sacrificing accuracy, making the model easier to deploy in a resource-constrained environment. (Kim, Ji, Yoon, & Hwang, 2021) introduced a method called progressive self-knowledge distillation, it can improve the performance and generalization ability of deep neural networks through gradual learning while reducing the risk of overfitting. (Ji et al., 2022) proposed to use the self-distillation mechanism in the pre-training phase. This can assist the model in acquiring more abstract and high-level feature representations by using the output of the pre-trained model as the target distribution. Specifically, self-distillation does not increase computational costs during inference. Table 1 is a summary of literature related to knowledge distillation.

2.2. Traffic prediction

Traffic prediction is a typical task in analyzing time series data (Qu, Lin, & Liu, 2023; Wu & Qu, 2022). It involves forecasting future traffic conditions based on historical traffic data that has been provided. The dynamic and constantly changing nature of traffic data across both time and space (Chaniotakis, Abouelela, Antoniou, & Goulias, 2022; Zheng, Chai, & Katos, 2022), as well as its susceptibility to various uncertain factors like weather, pedestrians (Yuanzhi, Tao, Xi, & Youning, 2022), and traffic accidents (Dabiri & Kulcsár, 2022), makes the task particularly difficult (Xu et al., 2023). It can be classified into short-term prediction and long-term prediction based on the length of prediction time. At present, some classic traffic prediction algorithms, such as ARIMA (Chen, Hu, Meng, & Zhang, 2011) and SVR (Li & Xu, 2021), only consider sequence information. However, the traffic flow on nearby roads can significantly affect each other. Therefore, effectively integrating all temporal and spatial dependencies into the prediction model has become a major challenge. The objective is to create a model that can accurately capture the complex interaction between various traffic variables and offer dependable predictions based on this data.

With the advances in deep learning, researchers have found that neural networks are effective in identifying and analyzing the complex non-linear systems present in traffic networks (Mohammadian, Zheng, Haque, & Bhaskar, 2023). CNNs are particularly powerful in extracting spatial features, while RNNs are well-suited for modeling temporal correlations. As a result, they have become a popular choice for traffic forecasting. (Qu, Lyu, Li, Ma, & Fan, 2021) incorporated features into RNNs, and used stacked RNNs to extract sequential features from traffic data. However, RNNs can encounter issues such as gradient explosion when processing lengthy sequences. These problems are effectively addressed through the implementation of gating structures, which are present in variants like Long Short-Term Memory (LSTM) (Cui, Ke, Pu, & Wang, 2018) and Gated Recurrent Neural Networks (GRU). (Dai, Ma, & Xu, 2019) primarily discussed the utilization of GRU to forecast short-term traffic flow in urban road segments. The method also incorporates spatiotemporal information. Besides, CNNs are commonly used to address spatial correlation in traffic flow analysis. For instance, (Ma et al., 2017) proposed a new method to transform traffic flow data into an image-like form and used a deep convolutional neural network to process and predict it. (Liu, Zheng, Feng, & Chen, 2017) introduced a mixed approach called Conv-LSTM, which integrates convolution and LSTM techniques to accurately capture the spatial–temporal characteristics of traffic data.

However, CNN can only model Euclidean data and cannot directly extract the topological relationship of road space. It needs to convert the topological structure of the road into the form of a traffic grid to model. Therefore, CNN has limitations in extracting the spatial characteristics of traffic data. There has been increasing interest in using GNN and GCN for traffic prediction tasks (Huang, Ye, Yang, & Xiong, 2023; Yuanzhi et al., 2022). These methods are more appropriate for modeling the complex relationships between various traffic variables and can effectively capture the non-linear dependencies found within traffic data (Jiang & Luo, 2022). Li, Yu, Shahabi, and Liu (2018) proposed a diffusion convolutional RNN for predicting traffic flow. They utilized diffusion convolutional networks and GRU to extract and analyze spatial and temporal features. This combination allows the model to capture complex dependencies within traffic data, leading to more accurate predictions. (Yu, Yin, & Zhu, 2018) proposed Spatial–Temporal Graph Convolutional Networks (STGCN), combining one-dimensional convolution and graph convolution to extract spatiotemporal features, while (Song, Lin, Guo, & Wan, 2020) introduced a spatial–temporal synchronous GCN, this is capable of capturing local spatiotemporal correlations without necessitating the merging of multiple modules. Moreover, (Zheng, Fan, Wang, & Qi, 2020) introduced a graph multi-attention network (Gman) that employs an encoder–decoder architecture to forecast traffic situations. This method takes into account spatial and temporal factors at different positions within the road network. Furthermore, the application of other neural networks like the attention and the transformer to graphs has shown promising outcomes. (Zhang & Guo, 2020) proposed a new graph with attention to LSTM, aiming to model the interaction between spatial and temporal factors in traffic flow dynamics. The research uses graph attention to effectively model non-Euclidean data structures and LSTM cells to extract time-dependent features from time series data. In (Yan, Ma, & Pu, 2021), a traffic transformer was proposed to address spatial–temporal features and mitigate challenges in long-term traffic prediction. (Wu, Pan, Long, Jiang and Zhang, 2019) introduced the Graph WaveNet (GWN), a graph convolutional network using the adaptive adjacency matrix for spatial features and dilated convolutions for temporal features. Table 2 is a summary of the current mainstream traffic flow prediction models.

3. Methods

This section introduces our proposed deep knowledge distillation model, depicted in Fig. 1. It consists of two networks, Net-Θ_1 and Net-Θ_2, both with encoder–decoder structures. The main idea of our model is: (1) Two networks engage in bidirectional distillation through

Table 2
Summary of literature review on traffic prediction.

Category		Application task	Related studies
Statistical methods		Flow	ARIMA (Chen et al., 2011)
		Flow	SVR (Li & Xu, 2021)
Deep Learning	Based on temporal	Speed	FI-RNN (Qu et al., 2021)
		Flow	LSTM (Cui et al., 2018)
		Flow	GRU (Dai et al., 2019)
		Flow	Transformer (Yan et al., 2021)
	Based on spatial	Flow	CNN (Ma et al., 2017)
		Flow	GCN (Jiang & Luo, 2022)
		Speed	STGCN (Yu et al., 2018)
		Speed	GWN (Wu, Pan et al., 2019)
	Mixed model	Speed	DCRNN (Li et al., 2018)
		Flow & speed	GMAN (Zheng et al., 2020)
		Flow	Conv-LSTM (Liu et al., 2017)
		Speed	ADSTGCN (Zhao et al., 2022)

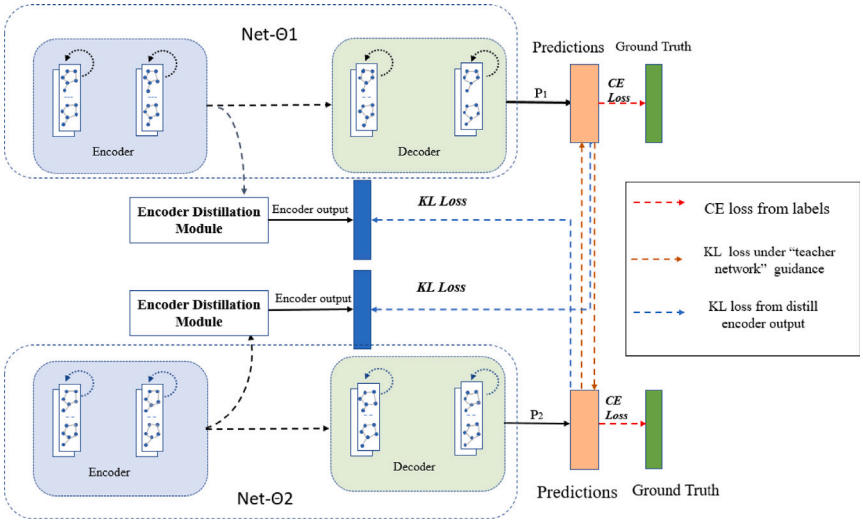


Fig. 1. The overall architecture of deep knowledge distillation.

mutual learning. (2) Two networks engage in cross-distillation through self-distillation.

To model traffic flow, we utilize the concept of graph neural networks as described in prior research (Li, Yu et al., 2017). Specifically, we model traffic flow as a directed graph and apply a bidirectional random walk process on a graph to model spatial correlation. Additionally, we incorporate temporal dependence by employing the GRU. Based on this, we introduce self-distillation and mutual learning to further extract spatio-temporal features from traffic flow data and enhance prediction accuracy.

In Fig. 1, our model integrates the GNN architecture with mutual learning and self-distillation techniques. To evaluate the similarity of predictions made by the two student networks, our model implements two distinct loss functions. The Kullback–Leibler loss measures the disparity between probability distributions, while the Cross-Entropy loss evaluates how closely the predictions align with the actual ground truth. Additionally, both networks utilize each other's shallow structures for self-distillation, where 'P_1' and 'P_2' in Fig. 1 represent the outputs from the softmax layer of the respective network decoders. For ease of reference, throughout the remainder of the paper, 'Cross Entropy loss' will be abbreviated as 'CE loss', and 'Kullback–Leibler divergence' as 'KL loss'.

3.1. Preliminary

The basic idea of knowledge distillation is to allow a student model to learn the behavior of a teacher model, including the output probability distribution. This enables the student model to mimic the prediction results of the teacher model. In our research, we refer to the teacher and student networks as Net-T and Net-S, respectively. The traditional knowledge distillation process begins with training Net-T and then transferring its knowledge to Net-S, using the 'temperature' parameter T specific to knowledge distillation. This process can be represented as follows:

$$L = \alpha L_{soft} + \beta L_{hard} \tag{1}$$

where α is the weight hyperparameter used to weight the soft loss and β is the weight hyperparameter used to weight the hard loss. The soft target is the output of Net-T, so L_{soft} is calculated as the KL divergence between the softmax output of Net-S and the soft target. In addition, L_{hard} is computed as the CE loss between the softmax output of Net-T and the ground truth.

Problem definition: In simpler terms, traffic prediction involves using historical data about traffic to create a model that can predict future traffic patterns. The road network is represented as a graph network G, which is composed of nodes that represent sensors located on the road, and edges that symbolize the links or connections between these sensors. The spatially weighted adjacency matrix captures the relationships between different sensors. At each time step, features of traffic flow are observed and represented as $X^{(m)}$. Using this graph representation, we can frame the prediction problem as the search for a function $f(\cdot)$ that can make accurate predictions about traffic patterns in the future using data from the past. Given a graph G, we can formulate the prediction problem as follows:

$$[X^{(m-M_0+1)}, \ldots, X^{(m)}; G] \xrightarrow{f(\cdot)} [X^{(m+1)}, \ldots, X^{(m+M)}] \tag{2}$$

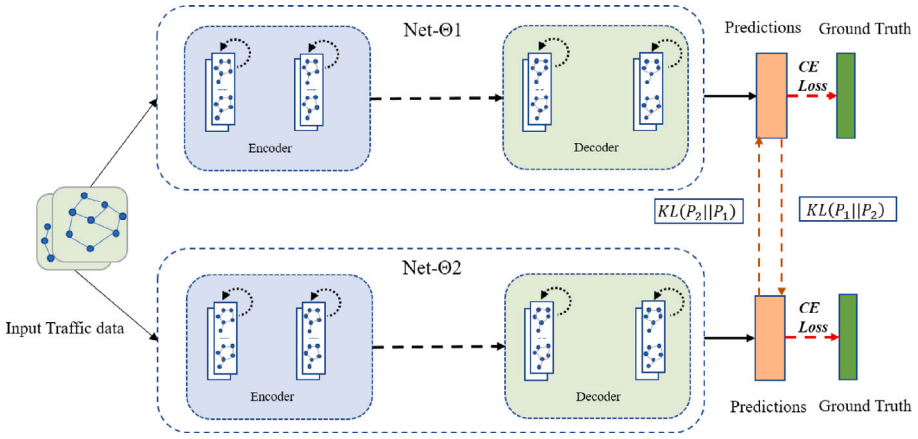


Fig. 2. The architecture of the mutual learning.

where $X^{(t)} \in R^{N\times C}$ is the traffic flow features at time step t, and C is the number of traffic features. $f(\cdot)$ is a mapping function that maps traffic data at historical moments to the future.

3.2. Overall architecture

Mutual learning describes two student networks learning from each other, without the need for supervision from a teacher network. It is an extension of traditional knowledge distillation. and has the advantage of breaking the pre-defined strong and weak relationships. Another benefit is its applicability to various network architectures, including heterogeneous networks with a mix of large and small networks, enabling them to learn from each other. Moreover, This approach can also be used to create a queue of networks trained in this way, which can serve as an ensemble to further improve performance. Based on the above advantages, we apply mutual learning in our method, letting two small neural networks learn simultaneously, and supervise each other.

In Fig. 2, we illustrate the application of mutual learning in the model's architecture. Two neural-network graphs, Net-Θ_1 and Net-Θ_2, are employed to create a mutual learning model, enabling reciprocal learning. Each network has two loss functions: a Cross-Entropy (CE) loss, represented by a red dotted line, and a Kullback–Leibler (KL) divergence loss, shown by an orange dotted line. Traffic flow features $X^{(m)}$ pass through Net-Θ_1 and Net-Θ_2, resulting in two different predicted probabilities P_1 and P_2. The KL divergence between P_1 and P_2 is then calculated to determine if the predictions of the two networks match, thus enhancing the model's generalization performance. The KL divergence formulas for P_1 and P_2 are:

$$D_{KL}(P_2\|P_1) = P_2 \log \frac{P_2}{P_1} \tag{3}$$

Besides, traditional supervision loss is essential, represented by CE loss. To simplify the notation, we represent the self-supervision loss and the matching loss between the two networks as L_1 and L_2, respectively.

$$L_1 = CrossEntropy \tag{4}$$

$$L_2 = D_{KL}(P_2\|P_1) = P_2 \log \frac{P_2}{P_1} \tag{5}$$

where P_1 and P_2 represent input data into two networks for training, and two different predictions output after training. $D_{KL}(P_2\|P_1)$ to calculate the difference between the two network prediction values. It is smaller, which means that the difference between the two prediction values is smaller.

Self-distillation can improve accuracy without adding computational cost. The main idea is to use a single network as both the teacher and student, removing the need to train a separate teacher model. Distillation is usually performed between different layers within the network. Expanding upon mutual learning, we incorporate self-distillation to enhance prediction accuracy even further. Fig. 3 depicts the architecture of self-distillation applied in the graph neural network. Self-distillation divides the model into shallow and deep structures, with dimension transformation performed on the encoder layer, and uses this part as a shallow network. During training, the deeper structure distills the shallow, with the deep structure acting as a teacher network to transfer knowledge to the shallow student network. In Fig. 3, two types of losses are depicted: the Cross-Entropy (CE) loss, represented by the red dotted line, and the Kullback–Leibler (KL) divergence loss from the distillation of encoder output, shown by the blue dotted line.

Table 3
Deep knowledge distillation prediction algorithm process.

1. Algorithm: Deep knowledge distillation to predict traffic flow
2. Input: training data, learning rate r
3. Initialization: Network θ_1 and Network θ_2
4. t = 0
5. do:
6. t = t + 1
7. Randomly extract data
8. inputs, labels = get_random_data()
9. Compute predictions P_1 and P_2 from the two networks respectively
10. **P_1 = network_1(inputs)**
11. **P_2 = network_2(inputs)**
12. Calculate mutual learning loss: **CE loss and KL loss**
13. Calculate self-distillation loss: **KL loss**
14. Compute the total loss: **Mutual learning loss + Self-distillation loss**
15. Update network θ_1: $\theta_1 \leftarrow \theta_1 + r\frac{\partial L_{\theta_1}}{\partial \theta_1}$
16. Update prediction P_1: **P1 = network_1(inputs)**
17. Update network θ_2: $\theta_2 \leftarrow \theta_2 + r\frac{\partial L_{\theta_2}}{\partial \theta_2}$
18. Update prediction P_2: **P2 = network_2(inputs)**
19. While: The objective function has not converged

Deep knowledge distillation: To enhance the spatio-temporal feature extraction ability of two graph neural networks for traffic data, our method employs deep knowledge distillation through a deep mixture modeling approach. The process of extracting features is depicted in Fig. 1, where the traffic data is passed through two networks Θ_1 and Θ_2 as input to extract features. Firstly, we employ the self-distillation algorithm to enhance the sensitivity of the shallow and deep structures of the model toward the features. Then the mutual learning algorithm is utilized to guide the two graph neural networks to improve their feature learning ability, thereby enhancing the model's performance in prediction. Each network has a classification loss function for the hard label and a loss function that mimics another student network. Table 3 is the overall algorithm flow of our model.

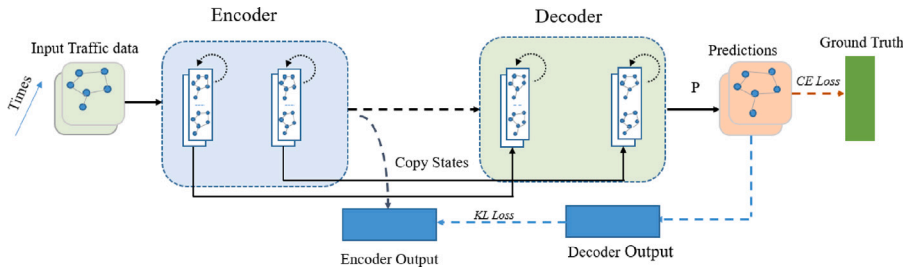


Fig. 3. The architecture of self-distillation.

Typically, we utilize the decoder output from Net-Θ_2 to extract and enhance the encoder output of Net-Θ_1 via a distillation process, since they are of equal size. The distillation supervision of the two models can be represented as L_3:

$$L_3 = KL(q^{E\theta_1}, q^{D\theta_2}) \tag{6}$$

where $q^{E\theta_1}$ is the encoder output of Net-Θ_1, $q^{D\theta_2}$ is the decoder output of Net-Θ_2, which is KL divergence between $q^{E\theta_1}$ and $q^{D\theta_2}$.

Each network in our approach incurs multiple losses. For Net-Θ_1, the overall loss function comprises three components: (1) the self-supervised loss function; (2) the matching loss function from Net-Θ_2; (3) the loss from encoder output.

$$L_{\theta_1} = (1-\alpha)CELoss1 + \alpha D_{KL}(P_2 \| P_1) + KL(q^{E\theta_1}, q^{D\theta_2}) \tag{7}$$

Similarly

$$L_{\theta_2} = (1-\alpha)CELoss2 + \alpha D_{KL}(P_1 \| P_2) + KL(q^{E\theta_2}, q^{D\theta_1}) \tag{8}$$

where α is the weight coefficient α that measures the proportion of mutual learning loss to the total loss. It comes from the process of calculating the total loss. Different weight coefficients indicate that the two parts of the loss have different proportions to the network supervision, which will affect the performance of the network. Normally, the value of α can be adjusted between 0 and 1, we experiment with it in subsequent chapters.

In particular, we attempt to use a hyperparameter to balance these different losses, but the effect is not significant.

4. Experiments

In this section, we will start by introducing two commonly used datasets 4.1, along with the relevant experiment settings 4.2, including evaluation metrics and baseline models. Then, we will systematically analyze the experimental results 4.3. After that, we will validate several parameters and conduct ablation experiments of the model in supplementary experiments 4.4. Additionally, to ensure the efficiency of every module within our model, we will carry out extended experiments and hyperparameter experiments 4.5. The details are as follows:

4.1. Datasets

Our method's effectiveness is validated through experiments on two commonly used datasets, collected at five-minute intervals. We divided them into training, validation, and testing subsets with respective proportions of 70%, 20%, and 10%. Detailed information about these datasets is presented in Table 4.

METR-LA includes data from 207 sensors placed on highways in Los Angeles County. The study mainly used four months of data, from March 1, 2012, to June 30, 2012. Subsequently will be recorded as dataset 1.

PEMS-BAY contains 325 sensors deployed in the San Francisco Bay Area, mainly experimenting with data collected over 6 months from January 1, 2017, to May 31, 2017. Subsequently will be recorded as dataset 2.

Table 4 provides a concise overview of two extensive traffic flow datasets, each equipped with a significant number of vehicle sensors. These sensors offer comprehensive traffic information and wide spatial coverage, making them highly valuable for traffic forecasting. Fig. 4 displays the locations of the 207 sensors in dataset 1.

4.2. Experiment settings

We trained using the Pytorch framework on a server with four GeForce RTX 3090 graphics cards. We applied Z-Score to standardize the two data sets separately to improve convergence speed during gradient descent. Additionally, we used end-to-end training with the "Adam" optimizer and a learning rate of 0.001 to simplify the project and optimize the model. To prevent overfitting, we implemented dropout and early stopping during training. Specific settings are shown in Table 5.

Metrics: Our study uses various metrics to assess performance, such as mean absolute error (MAE), mean absolute percentage error (MAPE), and root mean squared error (RMSE). Here are their detailed explanations (Moreno, Mariani, & dos Santos Coelho, 2021):

Mean Absolute Error (MAE): represents the average absolute error between the predicted value and the ground truth, the formula is:

$$MAE = \frac{1}{n}\Sigma_{i=1}^{n}|y_i - f(x_i)| \tag{9}$$

Mean Absolute Percentage Error (MAPE): represents the average absolute percentage error between the predicted value and the true value, the formula is:

$$MAPE = \frac{100\%}{n}\frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - f(x_i)}{y_i}\right| \tag{10}$$

Root Mean Squared Error (RMSE): represents the root mean square error between the predicted value and the true value, the formula is:

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}\left(y_i - f(x_i)\right)^2} \tag{11}$$

where n is the time step, y_i and $f(x_i)$ represent the true value and predicted value of the ith time step respectively. The smaller the calculated results of the three metrics, the higher the accuracy of the prediction results.

Baseline Models: In this paper, when comparing with other models, we use the same optimizer (Adam) and learning rate (0.001). These models are as follows:

(1) HA (Historical Average Liu & Guan, 2004): Calculates the average of historical data points.

(2) ARIMA (Auto-Regressive Integrated Moving Average Ahmed & Cook, 1979): A method for time series analysis, implemented using the 'statsmodels' Python package, which is a software library for statistical modeling in Python.

(3) VAR (Vector Auto-Regression Lütkepohl, 2013): An advanced time series model capturing pairwise relationships between traffic flow sequences, also implemented using the 'statsmodels' Python package.

Table 4
Basic information of the datasets.

Dataset	#sensors	Region	#training_set	#validation_set	#testing_set
METR-LA	207	Los Angeles County	23 974	3425	6850
PEMS-BAY	325	Bay Area	36 465	5209	10 419

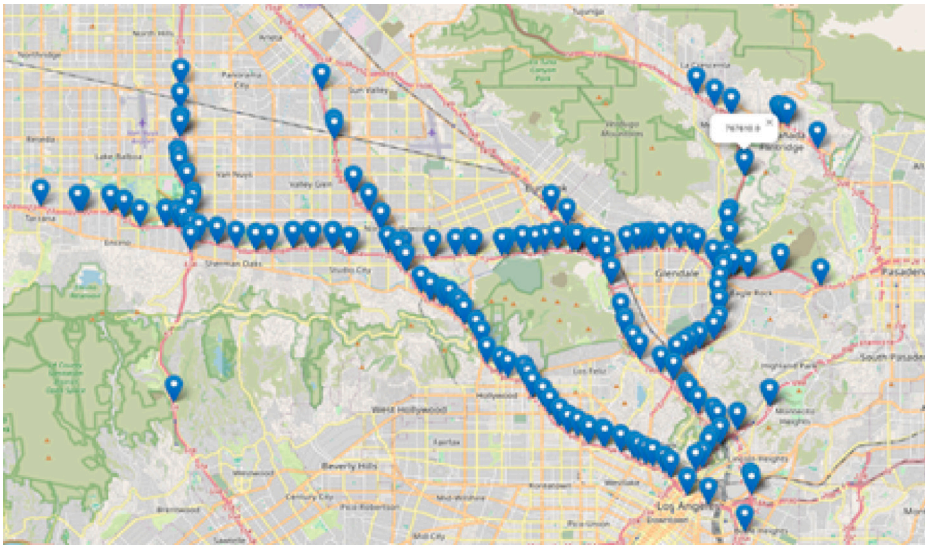


Fig. 4. Visualization of sensors distribution in dataset 1.

Table 5
Basic parameter setting.

RNN layers	2	Optimizer	Adam
Batch size	64	Initial learning rate	0.001
Training epoach	100	Maximum migration step	2
Number of RNN neurons	64	Distillation temperature T	8

(4) SVR (Support vector regression model Li & Xu, 2021): Utilizes linear support vector machine models for regression problems. Key parameters include a penalty term C = 0.1 and a historical observation count of 5.

(5) FNN (Feedforward Neural Network More, Mugal, Rajgure, Adhao, & Pachghare, 2016): Configured with two hidden layers, each containing 256 units. The model is trained with a batch size of 64 and uses MAE as the loss function.

(6) FC-LSTM (Fully Connected Long Short-Term Memory Sutskever, Vinyals, & Le, 2014): A variant of RNN with 4 layers of LSTM units. The input sequence is encoded with one LSTM layer, and the output sequence is decoded with another. Each layer has 256 LSTM units, the batch size is 64, and the model is trained using the MAE loss function.

(7) DCRNN (Diffusion Convolutional Recurrent Neural Network Li et al., 2018): Combines diffusion convolutional networks and GRU to extract spatial and temporal features. Both the encoder and decoder have two recurrent layers, with each layer comprising 64 units.

(8) STGCN (Spatial–Temporal Graph Convolutional Networks Yu et al., 2018): Consists of multiple spatial–temporal convolution blocks (ST-Conv Block), with three-layer channels in each ST-Conv block having 64, 16, and 64 units, respectively. The graph convolution kernel size is 3.

(9) GWN (Graph Wave Net Wu, Pan et al., 2019): Features adaptive graphs for capturing hidden spatial features and dilated convolutions for temporal features. The network uses 8 layers, each with a different dilation factor (1,2,1,2,1,2,1,2), and the diffusion step size in the graph convolution layer is 2.

4.3. Experiment results

Tables 6 and 7 present a comparison of our deep knowledge distillation model with other baselines in terms of performance on two datasets. The evaluation metrics used are MAE, MAPE, and RMSE, with a prediction horizon of 15, 30, and 60 min. The results indicate that our model performs well on both datasets. In comparison to the latest prediction graph model GWN, our model exhibits a slightly higher prediction error at 15 min, but as the prediction horizon increases (i.e., beyond 30 min), our model's prediction error is lower than that of GWN. This suggests that the deep knowledge distillation model is more suitable for long-term prediction.

Compared to the benchmark model, our model achieves lower MAE (3.41 vs. 3.60), MAPE (9.80% vs. 10.50%), and RMSE (7.10 vs. 7.59) values on dataset 1. Similarly, on dataset 2, our model also outperformed DCRNN, with a lower MAE by 0.18, a lower MAPE by 0.40%, and a lower RMSE by 0.41. In summary, our model improves prediction accuracy on both datasets, with dataset 2 showing a more significant improvement.

Through Tables 6 and 7, we can also draw the following conclusions:

(1) Among the four graph convolution methods, DCRNN and STGCN utilize prior knowledge to capture spatial dependencies in constructing the graph topology. They use a static graph. In contrast, GWN uses an adaptive graph matrix, allowing it to capture hidden spatial features, resulting in better prediction effects than DCRNN and STGCN.

(2) In traffic flow prediction, the longer the forecast time, the more uncertain factors are involved, and our approach is similar to DCRNN in short-term prediction. However, as the forecast time increases, the performance gap between the two models gradually widens, demonstrating that our model has greater robustness.

(3) Based on the benchmark model, we incorporated mutual learning and self-distillation to enhance feature extraction and perception, resulting in improved prediction performance.

For prediction, both DCRNN and our method randomly select a sensor's 1-day data on two datasets, and Figs. 5 and 6 present the speed trends. Blue, green, and orange colors correspond to the ground truth,

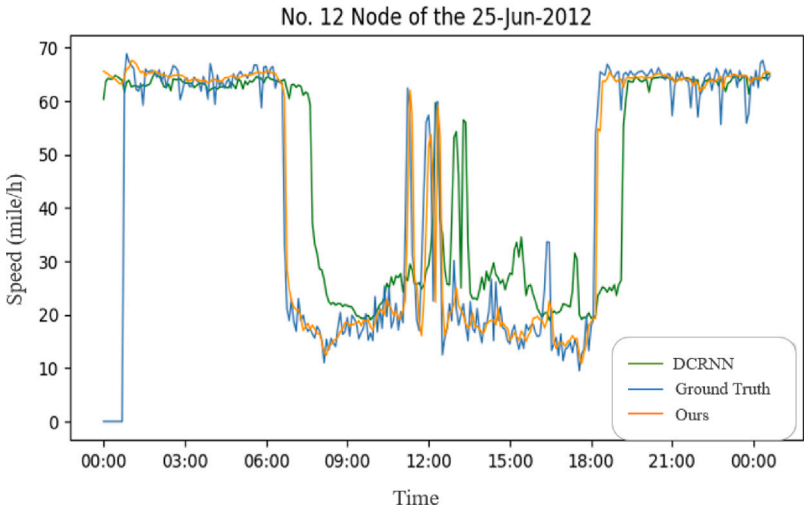


Fig. 5. We conduct an evaluation of our model and DCRNN's performance on dataset 1 by assessing the level of alignment between their predictions and the ground truth values.

Table 6
Performance comparison of deep knowledge distillation and other benchmark models on dataset 1.

Model	15 min			30 min			60 min		
Metric	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE
HA	4.16	13.00%	7.80	4.16	13.00%	7.80	4.16	13.00%	7.80
ARIMA	3.99	9.60%	8.21	5.15	12.70%	10.45	6.90	17.40%	13.23
FNN	3.99	9.90%	7.94	4.23	12.90%	8.17	4.49	14.00%	8.69
FC-LSTM	3.44	9.60%	6.30	3.77	10.90%	7.23	4.37	13.20%	8.69
DCRNN	2.77	7.30%	5.38	3.15	8.80%	6.45	3.60	10.50%	7.59
STGCN	2.88	7.60%	5.74	3.47	9.60%	7.24	4.59	12.70%	9.40
GWN	**2.69**	**6.90%**	**5.15**	3.07	8.40%	6.22	3.53	10.00%	7.37
Ours	2.70	7.00%	5.17	**3.05**	**8.30%**	**6.10**	**3.41**	**9.80%**	**7.01**

Table 7
Performance comparison of deep knowledge distillation and other benchmark models on dataset 2.

Model	15 min			30 min			60 min		
Metric	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE
HA	2.88	6.80%	5.59	2.88	6.80%	5.59	2.88	6.80%	5.59
ARIMA	1.62	3.50%	8.21	2.33	5.40%	4.76	3.38	8.30%	6.50
VAR	1.74	3.60%	3.16	2.32	5.00%	4.25	2.93	6.50%	5.44
SVR	1.85	3.80%	3.59	2.48	5.50%	5.18	3.28	8.00%	7.08
FNN	2.20	5.20%	4.42	2.30	5.40%	4.63	2.46	5.90%	4.98
FC-LSTM	2.05	4.80%	4.19	2.20	5.20%	4.55	2.37	5.70%	4.96
DCRNN	1.38	2.90%	2.95	1.74	3.90%	3.97	2.07	4.90%	4.74
STGCN	1.36	2.90%	2.96	1.81	4.20%	4.27	2.49	5.80%	5.69
GWN	**1.30**	2.90%	**2.74**	1.63	3.70%	3.70	1.95	4.60%	4.52
Ours	1.31	**2.80%**	2.75	**1.63**	**3.70%**	**3.65**	**1.89**	**4.50%**	**4.33**

DCRNN, and our model's speed predictions, respectively. The figures indicate that our model's speed predictions are more accurate, as they closely align with the ground truth. In contrast, the DCRNN model's predictions exhibit significant hysteresis. It can be seen that there is an obvious hysteresis in the prediction of the DCRNN model when there is a peak change, and our model's predictions are closer to the true value. Besides, there is a considerable slowdown in traffic during peak hours, such as morning rush hour, afternoon rush hour, and evening rush hour. This indicates a higher volume of traffic during these times. The traffic management department may consider implementing measures such as traffic restrictions or route planning reminders to address this issue (Siri, Siri, & Sacone, 2022).

To be more precise, when using the validation subset of dataset 1 as an example, we compare the performance of our model and DCRNN across three metrics in our experimental results, as shown in Figs. 7, 8, and 9. It is clear that on the validation set, the DCRNN model starts to converge after 20 training epochs, while the error of the deep knowledge distillation model continues to decrease, indicating that the performance of our model continues to improve. We attribute this to the increased mixed distillation method. On one hand, the two models perform self-distillation through each other's shallow network, and on the other hand, they complete the transfer of knowledge through mutual learning. Therefore, the model converges faster during the training process, the final convergence value is lower, and the predicted error value is also smaller.

Figs. 10 and 11 display boxplots that compare the errors of our model and DCRNN on the test datasets. The red box represents our model, while the blue box represents DCRNN. The upper and lower boundaries of each box indicate the highest and lowest error values, respectively. The lines in the middle of each box indicate the median error. To generate these visual representations, we individually compute the prediction errors of both our model and DCRNN by comparing them against the ground truth values. Next, we calculate the mean errors over intervals of two hours, which is represented by the center line of the box in the figure. As shown in the figures, the median line for our model is slightly below that of DCRNN, which indicates that our model has a lower error and is more robust.

4.4. Supplementary experiment

To further corroborate our experimental results, we conduct relevant supplementary experiments in this part, including ablation experiments 4.4.1 and extension experiments 4.4.2.

4.4.1. Ablation experiment

In this section, we conduct experiments with various parameters in the model to determine the optimal ones. For instance, we experiment with the learning rate and optimizer using the METR-LA dataset. We test the learning rate at 0.01, 0.001, and 0.0001, and compare the optimization results of the Adam optimizer and the SGD optimizer. Ultimately, we select 0.001 as the initial learning rate for our experiment and opt for the Adam optimizer. Table 8 is the experimental results of different learning rates, and Table 9 is the results of different optimizers.

Then we verify the effect of each part of the model, the deep knowledge distillation model is mainly composed of three important parts, namely the DCRNN module, the self-distillation module, and the mutual learning module. we compare the effects of self-distillation and mutual learning modules on two datasets to optimize the benchmark model alone and optimize it after the fusion of the two modules on the experimental results. The results are shown in Table 10. We can see:

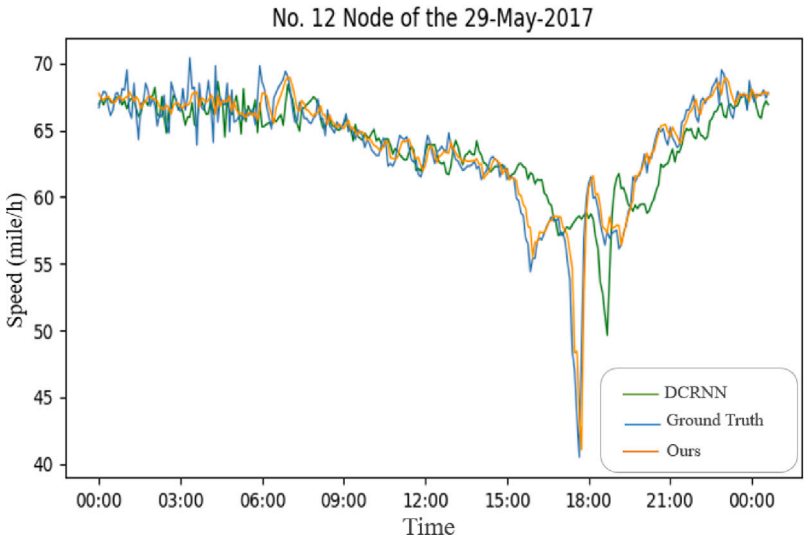


Fig. 6. We conduct an evaluation of our model and DCRNN's performance on dataset 2 by assessing the level of alignment between their predictions and the ground truth values.

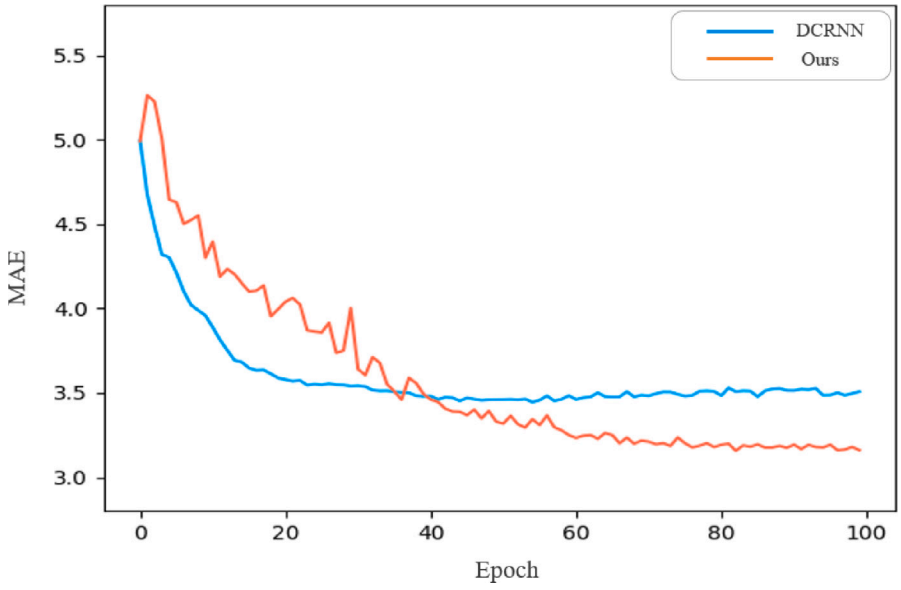


Fig. 7. MAE curve visualization between two models.

Table 8
Training results at different learning rates.

Learning	15			30			60		
rate	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE
0.01	2.81	7.73%	5.45	3.17	9.39%	6.44	3.70	11.70%	7.69
0.001	**2.70**	**7.00%**	**5.17**	**3.05**	**8.30%**	**6.10**	**3.41**	**9.80%**	**7.01**
0.0001	2.77	7.46%	5.45	3.19	9.24%	6.51	3.76	11.79%	7.78

(1) Combining the benchmark model with self-distillation and mutual learning modules, the experimental results are significantly better.

(2) Mutual learning alone improves model performance more significantly than self-distillation.

(3) The performance of adding mutual learning and self-distillation models at the same time is better than adding mutual learning or self-distillation alone.

We analyze the reason because after adding the hybrid distillation, the two models not only perform self-distillation through each other's shallow network but also complete the transfer of knowledge through mutual learning. Therefore, the model converges faster during the training process, the final convergence value is lower, and the prediction error is also smaller. Compared with the self-distillation model, the mutual learning optimization method has smaller errors, which proves that the mutual learning-based optimization method performs better than the self-distillation. We speculate that the reason is that mutual learning makes the knowledge learned between the networks more sufficient.

Furthermore, we also verify the impact of the three different self-distillation methods on the prediction performance through experiments. In addition to distillation based on an encoder and decoder, we also use self-distillation from the Last Mini-Batch (DLB) method (Shen, Xu, Yang, Li, & Guo, 2022), which uses its historical information for self-distillation. The results are shown in Table 11. Among the three methods based on self-distillation and mutual learning, the prediction results of the Encoder and Decoder methods that divide the model according to the structure and distill the shallow structure on the two datasets are more significant than the DLB. We guess the reason is that after combining the mutual learning module, the self-distillation based

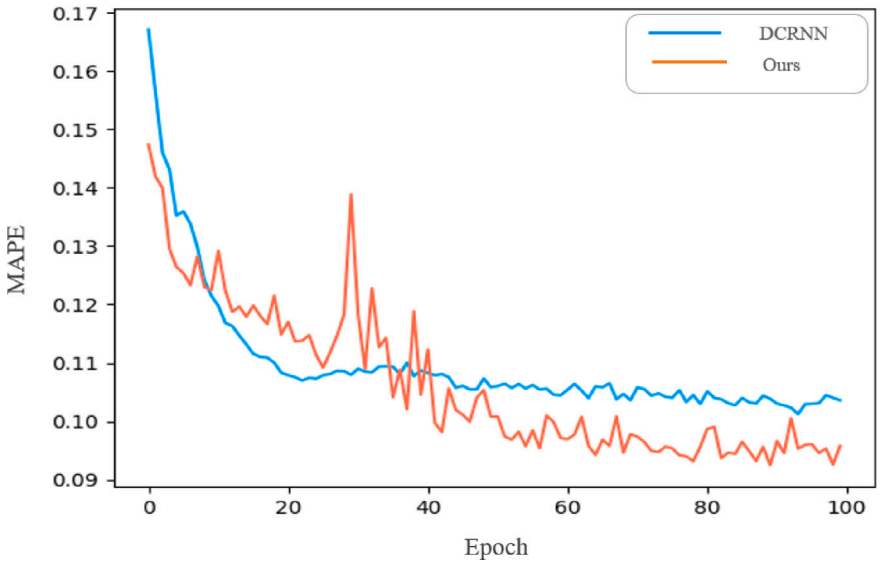


Fig. 8. MAPE curve visualization between two models.

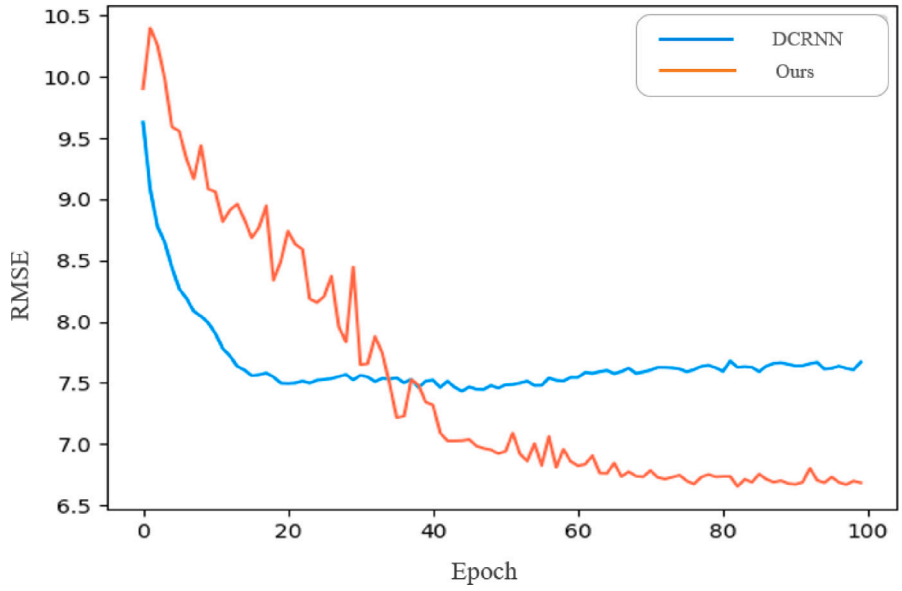


Fig. 9. RMSE curve visualization between two models.

Table 9
Training results under different optimizers.

Optimizer	15			30			60		
	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE
Adam	**2.70**	**7.00%**	**5.17**	**3.05**	**8.30%**	**6.10**	**3.41**	**9.80%**	**7.01**
SGD (momentum = 0.9)	3.03	8.13%	5.81	3.57	10.23%	7.03	4.33	13.45%	8.55

Table 10
Ablation experiment: verify the effectiveness of each part of deep knowledge distillation.

DCRNN	Mutual learning	Self -distillation	METR-LA			PEMS-BAY		
			MAE	MAPE	RMSE	MAE	MAPE	RMSE
✓			3.54	10.20%	7.45	1.97	4.70%	4.63
✓	✓		3.45	10.00%	7.07	1.95	4.60%	4.43
✓		✓	3.59	9.80%	7.48	1.96	4.60%	4.53
✓	✓	✓	**3.41**	**9.80%**	**9.80%**	**1.89**	**4.50%**	**4.33**

on the network structure enables the learning of diverse knowledge between the two networks.

When selecting different shallow structures, the prediction results based on the Encoder are better than those of the Decoder. We speculate that this is due to the shallower network structure of the Encoder, which allows for better knowledge retention from the deep network. This, in turn, has a stronger impact on the overall network. By enhancing the feature extraction and perception of the shallow part between the two networks, the features of the shallow part are fully extracted, thereby improving the learning ability of the entire network

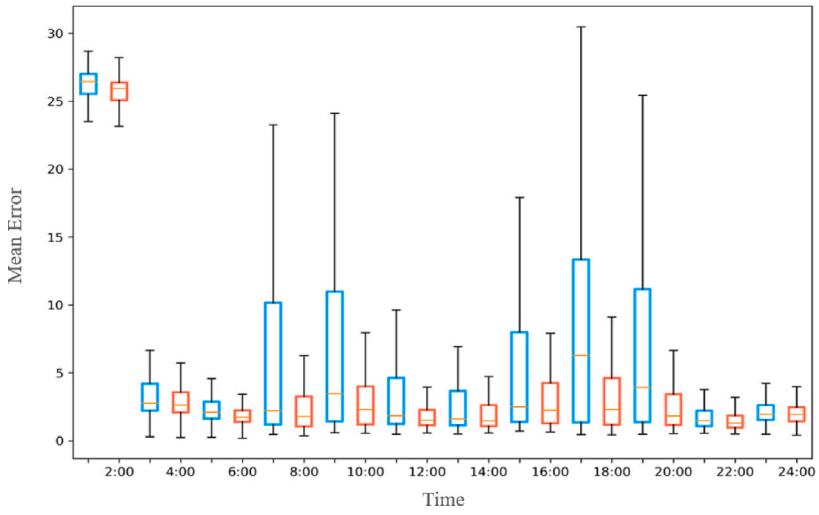


Fig. 10. Error analysis boxplots about our model and DCRNN on dataset 1: For the METR-LA dataset, we select all sensor predictions on June 25, 2012, for processing.

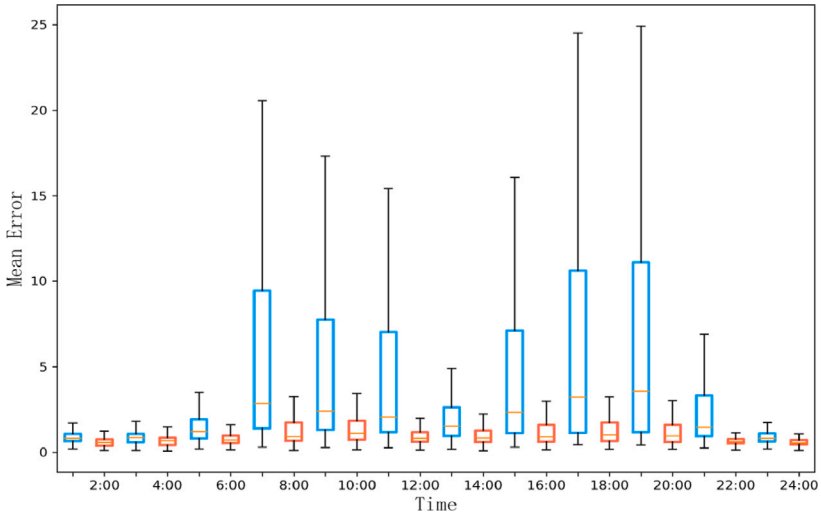


Fig. 11. Error analysis boxplots about our model and DCRNN on dataset 2: For the PEMS-BAY data set, we select the predicted values on all sensors on May 29, 2017, for processing.

Table 11
Controlled experiment: verify the effect of different self-distillation methods combined with mutual learning.

Self-Mutual	Encoder	Decoder	DLB	METR-LA			PEMS-BAY		
				MAE	MAPE	RMSE	MAE	MAPE	RMSE
✓	✓			**3.41**	**9.80%**	7.01	**1.89**	**4.50%**	**4.33**
✓		✓		3.43	9.90%	**6.99**	1.91	4.60%	4.37
✓			✓	3.50	10.20%	7.12	1.92	4.60%	4.41

and enhancing prediction performance. This is also the reason why the deep knowledge distillation model is based on Encoder self-distillation.

To better compare the experimental results of three self-distillation-mutual learning methods, we visualize their prediction results on the PEMS-BAY dataset. Fig. 12 shows that the self-distillation-prediction curve based on the Encoder method closely aligns with the real value curve.

Overall, our proposed method utilizes both mutual learning and self-distillation modules. Experimental findings prove that incorporating mutual learning or self-distillation modules individually contributes to enhanced performance. However, when they are combined, the performance is further enhanced.

4.4.2. Extended experiment

To further validate the effectiveness of deep knowledge distillation in enhancing traffic flow prediction based on graph neural network, we conducted experimental verification on the DCRNN model and compared it with the GTS (Graph For Time Series, GTS) model proposed by (Shang & Chen, 2021), as a benchmark method. A novel method is proposed in GTS for optimizing graph structures, which aims to enhance the forecasting of multiple multivariate time series when the graph is unknown. This method utilizes the DCRNN model as the benchmark and improves the prediction performance by optimizing the graph structure. GTS is used as the benchmark model, and a mutual learning module is added. Two GTS networks are also employed as the student model, and they learn from each other to facilitate knowledge

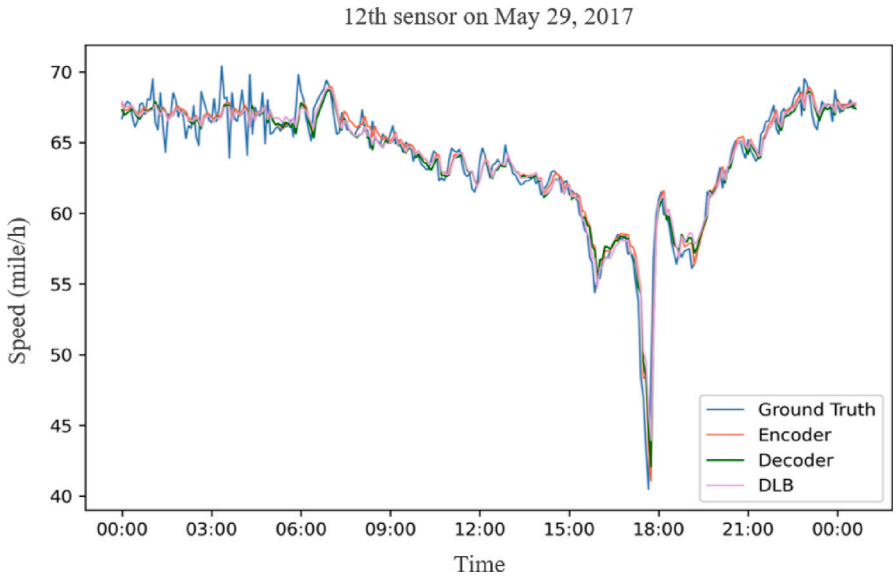


Fig. 12. Prediction results of three different self-distillation methods on the PEMS-BAY dataset: We chose the 1-day true value on 1 sensor on May 29, 2019, and the predicted results of these three different self-distillation methods.

Table 12
Extended experiment: using GTS as a benchmark model to verify the universality of mutual learning.

GTS	Mutual learning	METR-LA			PEMS-BAY		
		MAE	MAPE	RMSE	MAE	MAPE	RMSE
✓		3.44	10.00%	7.28	1.95	4.60%	4.46
✓	✓	**3.37**	**10.00%**	**6.98**	**1.93**	**4.50%**	**4.43**

transfer. The experiment was conducted on the METR-LA and PEMS-BAY datasets, and the prediction results for the next 15 min, 30 min, and 1 h were evaluated. The 1-h forecast results were selected for comparison.

Table 12 demonstrates that the inclusion of mutual learning leads to reductions in MAE, MAPE, and RMSE when compared to GTS. This effect is particularly notable in the METR-LA dataset, where the RMSE value decreases by 0.3 in comparison to the GTS model. This provides further evidence that mutual learning has the potential to enhance the performance of traffic flow prediction models based on graph neural networks.

4.5. Hyperparametric experiments

We conduct a parameter sensitivity analysis on two important hyperparameters when optimizing the deep knowledge distillation model, both of which come from the loss function: one is the weight coefficient α that measures the proportion of different losses to the total loss, which has almost no effect on self-distillation. Therefore, we only analyze the weight coefficient α of the proportion of mutual learning loss to the total loss. The other is the effect of distillation temperature T on model prediction performance.

Firstly, analyzing the influence of different weights α, set the α coefficient to [0.1,0.3,0.5,0.7,0.9] to conduct experiments on the METR-LA dataset, the results are shown in Fig. 13, when $\alpha = 0.9$ or $\alpha = 0.1$, with the worst effect. $\alpha = 0.9$ means that the mutual learning loss weight is too large, and the loss weight of the network itself is too low, which greatly affects the prediction performance of the model. $\alpha = 0.1$ means that the weight of mutual learning loss is too low, and the mutual learning ability between networks becomes weaker, which also affects the prediction performance. The influence of the other three different weight coefficients on the model is not much different. Among them, the value of $\alpha = 0.5$ on the validation set and the final convergence of the MAE and loss curves is a little lower than the curve convergence results of the other two weight coefficients.

Table 13
Hyperparameter experiment: verify the influence of different distillation temperatures T on the experimental results.

T	15 min MAE/RMSE/MAPE	30 min MAE/RMSE/MAPE	60 min MAE/RMSE/MAPE
1	2.71/5.19/7.13%	3.07/6.13/8.48%	3.43/7.07/9.95%
2	2.71/5.19/7.10%	3.06/6.11/8.42%	3.42/7.03/9.91%
3	2.71/5.18/7.06%	3.06/6.10/8.36%	3.43/7.01/9.81%
4	2.70/5.17/7.06%	3.05/6.10/8.39%	3.41/7.01/9.82%
5	2.72/5.18/7.12%	3.07/6.09/8.46%	3.43/6.99/9.96%
6	2.71/5.19/7.10%	3.06/6.09/8.39%	3.44/7.00/9.82%

It is worth noting that we don't add the self-distillation to the model for the sensitivity analysis of α, so the optimal distillation temperature should be the T when the benchmark only adds mutual learning, which is different from the selection of the next distillation temperature T different.

Next, we explore the effect of the distillation temperature T. When $T = 1$, an ordinary softmax is used. When $T > 1$, the value after softmax will be more evenly and gently distributed. The experimental results obtained through different distillation temperatures are shown in Table 13. During our experiments, we adjust the value of different distillation temperature parameters T and observe its effect on the output of intermediate layer self-distillation and mutual learning loss. Fig. 14 illustrates the MAE obtained for different values of T and durations used in our model. The comparison on the METR-LA dataset reveals that the optimal loss value is achieved when the distillation temperature is set to $T = 4$, irrespective of the time intervals (15 min, 30 min, or 60 min), which outperforms other distillation temperatures.

5. Conclusions

Our primary research goal is to develop a hybrid model for traffic prediction. In our case study, we introduce a deep knowledge distillation model that combines self-distillation and mutual learning techniques to enhance its feature extraction capabilities and improve prediction accuracy. To evaluate the performance of our model, we conducted experiments on two real-world datasets. We used a GCN model

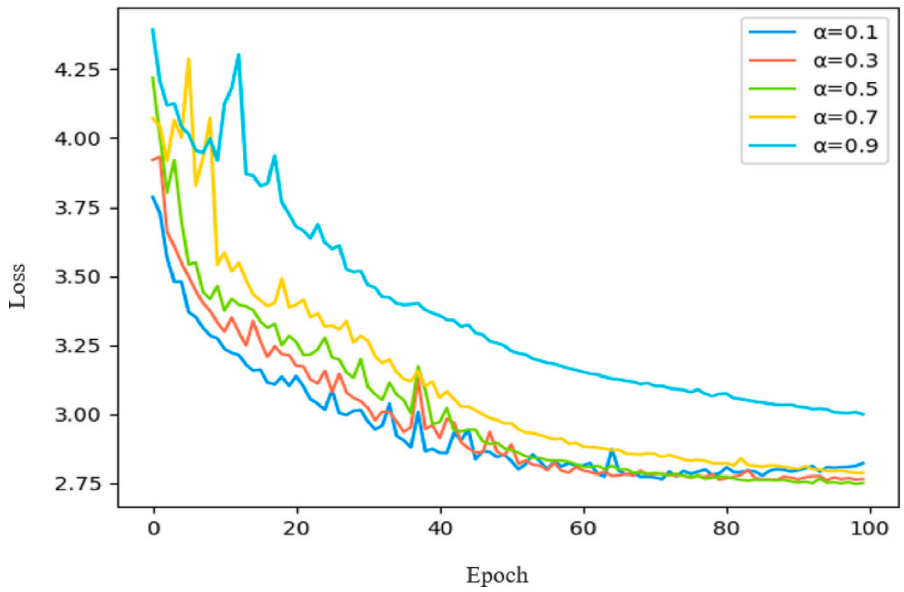


Fig. 13. The influence of different weight coefficients α: α adopts [0.1,0.3,0.5,0.7,0.9] and plot their Loss variation.

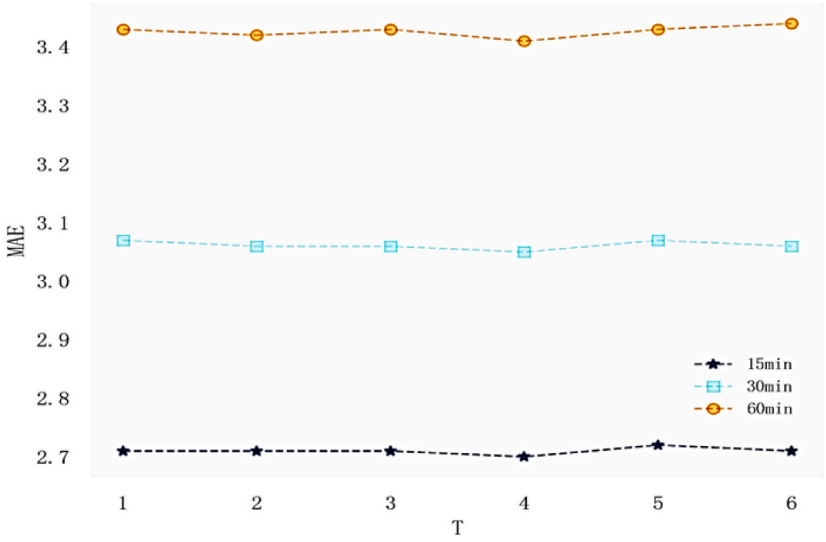


Fig. 14. The results obtained at different distillation temperatures T. In the experiment, the coefficient of temperature T adopts [1,2,3,4,5,6].

with an encoder–decoder structure as the baseline and incorporated self-distillation and mutual learning to create a self-learning knowledge distillation module. Upon verification, integrating this module into the baseline model results in a noticeable improvement in prediction accuracy. This approach is applicable for forecasting traffic-related variables such as volume and density and can be extended to prediction tasks in various domains.

Our model faces challenges due to using the same neural network for mutual learning during training. This creates two sets of parameters, which increases computing resources and yields minimal additional knowledge. Additionally, we selected two graph neural networks with the same structure as the student network, resulting in little difference between their structures and limited additional knowledge gained during mutual learning. Furthermore, when extracting traffic flow data characteristics, we only take into account temporal correlation and spatial dependence, neglecting other factors such as complex weather conditions that also affect traffic flow changes.

To tackle the challenges of the model, firstly, we plan to reduce the parameter count of both networks. One potential approach is refining the parallel connection between the two student networks. Alternatively, employing two distinct networks as student models could enable mutual learning, thereby extracting more comprehensive knowledge. Furthermore, we intend to consider other factors impacting traffic flow, such as weather, and integrate them into the model. For instance, we propose adopting a multi-task learning framework that simultaneously predicts traffic flow and weather patterns. Utilizing the interdependence between these factors can result in more holistic and accurate traffic predictions, overcoming the current model's limitation of relying solely on time and spatial dependencies. Finally, we aim to conduct further data analysis to uncover potential patterns and trends, thereby enhancing the model's performance.

CRediT authorship contribution statement

Ying Li: Conceptualization, Determination of research questions, Selection of research methods. **Ping Li:** Literature review, Data processing, Experimental operation, Writing – original draft. **Doudou Yan:** Experimental design, Data curation, Experimental operation. **Yang Liu:**

Writing – review & editing, Ensuring that the logic, grammar, format, etc. are correct. **Zhiyuan Liu:** Provide important ideological guidance and substantive guidance on research direction, Methodology, Conclusions.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (52002031), the National Natural Science Foundation of China (52172325), the Key Research and Development Project of China (2021YFB1600104), and the State Key Laboratory of Intelligent Green Vehicle and Mobility, Tsinghua University (KFY2421).

References

Acharya, S., & Mekker, M. (2022). The verbiage in variable message signs and traffic diversion during crash incidents. *Journal of Intelligent and Connected Vehicles*, *5*(3), 333–344.

Adriana, R., Nicolas, B., Ebrahimi, K. S., Antoine, C., Carlo, G., & Yoshua, B. (2015). Fitnets: Hints for thin deep nets. *Vol. 2*, In *Proc. ICLR*.

Ahmed, M. S., & Cook, A. R. (1979). *Analysis of freeway traffic time-series data by using Box-Jenkins techniques*. 722.

Bing, L., Jinrui, W., Jiming, X., Jinhong, C., Guozhong, D., Baoquan, Y., et al. (2022). Microscopic trajectory data-driven probability distribution model for weaving area of channel change. *Journal of Automotive Safety and Energy*, *13*(2), 333.

Chaniotakis, E., Abouelela, M., Antoniou, C., & Goulias, K. (2022). Investigating social media spatiotemporal transferability for transport. *Communications in Transportation Research*, *2*, Article 100081.

Chen, C., Hu, J., Meng, Q., & Zhang, Y. (2011). Short-time traffic flow prediction with ARIMA-garch model. In *2011 IEEE intelligent vehicles symposium* (pp. 607–612). IEEE.

Cui, Z., Ke, R., Pu, Z., & Wang, Y. (2018). Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction. arXiv preprint arXiv:1801.02143.

Dabiri, A., & Kulcsár, B. (2022). Incident indicators for freeway traffic flow models. *Communications in Transportation Research*, *2*, Article 100060.

Dai, G., Ma, C., & Xu, X. (2019). Short-term traffic flow prediction method for urban road sections based on space-time analysis and GRU. *IEEE Access*, *7*, 143025–143035.

Fei, D., Guanyu, M., En, T., Nan, Z., Jianmin, B., & Dengyin, Z. (2022). Multi-channel high-resolution network and attention mechanism fusion for vehicle detection model. *Journal of Automotive Safety and Energy*, *13*(1), 122.

Fei, Y., Shi, P., Li, Y., Liu, Y., & Qu, X. (2024). Formation control of multi-agent systems with actuator saturation via neural-based sliding mode estimators. *Knowledge-Based Systems*, *284*, Article 111292.

Gan, N., Zhang, M., Zhou, B., Chai, T., Wu, X., & Bian, Y. (2022). Spatio-temporal heuristic method: a trajectory planning for automatic parking considering obstacle behavior. *Journal of Intelligent and Connected Vehicles*, *5*(3), 177–187.

Gou, J., Yu, B., Maybank, S. J., & Tao, D. (2021). Knowledge distillation: A survey. *International Journal of Computer Vision*, *129*, 1789–1819.

He, Y., Liu, Y., Yang, L., & Qu, X. (2024). Deep adaptive control: Deep reinforcement learning-based adaptive vehicle trajectory control algorithms for different risk levels. *IEEE Transactions on Intelligent Vehicles*, *9*(1), 1654–1666. http://dx.doi.org/10.1109/TIV.2023.3303408.

Hinton, G., Vinyals, O., Dean, J., et al. (2015). Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531, 2..

Huang, X., Ye, Y., Yang, X., & Xiong, L. (2023). Multi-view dynamic graph convolution neural network for traffic flow prediction. *Expert Systems with Applications*, *222*, Article 119779.

Ji, J., Yu, F., & Lei, M. (2022). Self-supervised spatiotemporal graph neural networks with self-distillation for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*.

Jiang, W., & Luo, J. (2022). Graph neural network for traffic forecasting: A survey. *Expert Systems with Applications*, Article 117921.

Jie, X., Xiaofei, P., Bo, Y., & Zhigang, F. (2022). Learning-based automatic driving decision-making integrated with vehicle trajectory prediction. *Journal of Automotive Safety and Energy*, *13*(2), 317.

Kim, K., Ji, B., Yoon, D., & Hwang, S. (2021). Self-knowledge distillation with progressive refinement of targets. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 6567–6576).

Kobayashi, T. (2022). Extractive knowledge distillation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision* (pp. 3511–3520).

Li, Y., Li, S. E., Jia, X., Zeng, S., & Wang, Y. (2022). FPGA accelerated model predictive control for autonomous driving. *Journal of Intelligent and Connected Vehicles*, *5*(2), 63–71.

Li, C., & Xu, P. (2021). Application on traffic flow prediction of machine learning in intelligent transportation. *Neural Computing and Applications*, *33*, 613–624.

Li, Y., Yang, J., Song, Y., Cao, L., Luo, J., & Li, L.-J. (2017). Learning from noisy labels with distillation. In *Proceedings of the IEEE international conference on computer vision* (pp. 1910–1918).

Li, Y., Yu, R., Shahabi, C., & Liu, Y. (2017). Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. arXiv preprint arXiv:1707.01926.

Li, Y., Yu, R., Shahabi, C., & Liu, Y. (2018). Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International conference on learning representations*.

Liu, J., & Guan, W. (2004). A summary of traffic flow forecasting methods. *Journal of Highway and Transportation Research and Development*, *21*(3), 82–85.

Liu, L., Huang, Q., Lin, S., Xie, H., Wang, B., Chang, X., et al. (2021). Exploring inter-channel correlation for diversity-preserved knowledge distillation. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 8271–8280).

Liu, Y., Liu, Z., Lyu, C., & Ye, J. (2019). Attention-based deep ensemble net for large-scale online taxi-hailing demand prediction. *IEEE Transactions on Intelligent Transportation Systems*, *21*(11), 4798–4807.

Liu, Y., Lyu, C., Zhang, Y., Liu, Z., Yu, W., & Qu, X. (2021). Deeptsp: Deep traffic state prediction model based on large-scale empirical data. *Communications in Transportation Research*, *1*, Article 100012.

Liu, Y., Wu, F., Liu, Z., Wang, K., Wang, F., & Qu, X. (2023). Can language models be used for real-world urban-delivery route optimization? *The Innovation*, *4*(6).

Liu, Y., Wu, F., Lyu, C., Li, S., Ye, J., & Qu, X. (2022). Deep dispatching: A deep reinforcement learning approach for vehicle dispatching on online ride-hailing platform. *Transportation Research Part E: Logistics and Transportation Review*, *161*, Article 102694.

Liu, Y., Zheng, H., Feng, X., & Chen, Z. (2017). Short-term traffic flow prediction with conv-LSTM. In *2017 9th international conference on wireless communications and signal processing* (pp. 1–6). IEEE.

Lütkepohl, H. (2013). Vector autoregressive models. In *Handbook of research methods and applications in empirical macroeconomics* (pp. 139–164). Edward Elgar Publishing.

Ma, X., Dai, Z., He, Z., Ma, J., Wang, Y., & Wang, Y. (2017). Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction. *Sensors*, *17*(4), 818.

Ma, Y., Wang, Z., Yang, H., & Yang, L. (2020). Artificial intelligence applications in the development of autonomous vehicles: A survey. *IEEE/CAA Journal of Automatica Sinica*, *7*(2), 315–329.

Ma, X., Yu, H., Wang, Y., & Wang, Y. (2015). Large-scale transportation network congestion evolution prediction using deep learning theory. *PLoS One*, *10*(3), Article e0119044.

Mohammadian, S., Zheng, Z., Haque, M. M., & Bhaskar, A. (2023). Continuum modeling of freeway traffic flows: State-of-the-art, challenges and future directions in the era of connected and automated vehicles. *Communications in Transportation Research*, *3*, Article 100107.

More, R., Mugal, A., Rajgure, S., Adhao, R. B., & Pachghare, V. K. (2016). Road traffic prediction and congestion control using artificial neural networks. In *2016 international conference on computing, analytics and security trends* (pp. 52–57). IEEE.

Moreno, S. R., Mariani, V. C., & dos Santos Coelho, L. (2021). Hybrid multi-stage decomposition with parametric model applied to wind speed forecasting in Brazilian Northeast. *Renewable Energy*, *164*, 1508–1526.

Park, W., Kim, D., Lu, Y., & Cho, M. (2019). Relational knowledge distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 3967–3976).

Qu, X., Lin, H., & Liu, Y. (2023). Envisioning the future of transportation: Inspiration of ChatGPT and large models. *Communications in Transportation Research*, *3*.

Qu, L., Lyu, J., Li, W., Ma, D., & Fan, H. (2021). Features injected recurrent neural networks for short-term traffic speed prediction. *Neurocomputing*, *451*, 290–304.

Shang, C., & Chen, J. (2021). Discrete graph structure learning for forecasting multiple time series. In *Proceedings of international conference on learning representations*.

Shen, Y., Xu, L., Yang, Y., Li, Y., & Guo, Y. (2022). Self-distillation from the last mini-batch for consistency regularization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 11943–11952).

Siri, E., Siri, S., & Sacone, S. (2022). A topology-based bounded rationality day-to-day traffic assignment model. *Communications in Transportation Research*, *2*, Article 100076.

Song, C., Lin, Y., Guo, S., & Wan, H. (2020). Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. *Vol. 34*, In *Proceedings of the AAAI conference on artificial intelligence* (pp. 914–921).

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems*, *27*.

Tzelepi, M., Passalis, N., & Tefas, A. (2021). Online subclass knowledge distillation. *Expert Systems with Applications*, *181*, Article 115132.

Wu, R., Feng, M., Guan, W., Wang, D., Lu, H., & Ding, E. (2019). A mutual learning method for salient object detection with intertwined multi-supervision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8150–8159).

Wu, Z., Pan, S., Long, G., Jiang, J., & Zhang, C. (2019). Graph wavenet for deep spatial-temporal graph modeling. arXiv preprint arXiv:1906.00121.

Wu, J., & Qu, X. (2022). Intersection control with connected and automated vehicles: a review. *Journal of Intelligent and Connected Vehicles*, *5*(3), 260–269.

Xu, M., Di, Y., Ding, H., Zhu, Z., Chen, X., & Yang, H. (2023). AGNP: Network-wide short-term probabilistic traffic speed prediction and imputation. *Communications in Transportation Research*, *3*, Article 100099.

Xu, Q., Li, K., Wang, J., Yuan, Q., Yang, Y., & Chu, W. (2022). The status, challenges, and trends: an interpretation of technology roadmap of intelligent and connected vehicles in China (2020). *Journal of Intelligent and Connected Vehicles*, *5*(3), 1–7.

Yan, H., Ma, X., & Pu, Z. (2021). Learning dynamic and hierarchical traffic spatiotemporal features with transformer. *IEEE Transactions on Intelligent Transportation Systems*, *23*(11), 22386–22399.

Yu, B., Yin, H., & Zhu, Z. (2018). Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In *Proceedings of the 27th international joint conference on artificial intelligence* (pp. 3634–3640).

Yuanzhi, H., Tao, J., Xi, L., & Youning, S. (2022). Pedestrian-crossing intention-recognition based on dual-stream adaptive graph-convolutional neural-network. *Journal of Automotive Safety and Energy*, *13*(2), 325.

Yue, L., Abdel-Aty, M., & Wang, Z. (2022). Effects of connected and autonomous vehicle merging behavior on mainline human-driven vehicle. *Journal of Intelligent and Connected Vehicles*, *5*(3), 36–45.

Zhai, Q., Li, X., Yang, F., Chen, C., Cheng, H., & Fan, D.-P. (2021). Mutual graph learning for camouflaged object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 12997–13007).

Zhang, T., & Guo, G. (2020). Graph attention LSTM: A spatiotemporal approach for traffic flow forecasting. *IEEE Intelligent Transportation Systems Magazine*, *14*(2), 190–196.

Zhang, L., Song, J., Gao, A., Chen, J., Bao, C., & Ma, K. (2019). Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 3713–3722).

Zhang, Y., Xiang, T., Hospedales, T. M., & Lu, H. (2018). Deep mutual learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4320–4328).

Zhao, J., Liu, Z., Sun, Q., Li, Q., Jia, X., & Zhang, R. (2022). Attention-based dynamic spatial-temporal graph convolutional networks for traffic speed forecasting. *Expert Systems with Applications*, *204*, Article 117511.

Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, P., Lin, T., et al. (2020). T-GCN: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, *21*(9), 3848–3858. http://dx.doi.org/10.1109/TITS.2019.2935152.

Zheng, G., Chai, W. K., & Katos, V. (2022). A dynamic spatial-temporal deep learning framework for traffic speed prediction on large-scale road networks. *Expert Systems with Applications*, *195*, Article 116585.

Zheng, C., Fan, X., Wang, C., & Qi, J. (2020). Gman: A graph multi-attention network for traffic prediction. *Vol. 34*, In *Proceedings of the AAAI conference on artificial intelligence* (pp. 1234–1241).

Zhong, C., Wu, P., Zhang, Q., & Ma, Z. (2023). Online prediction of network-level public transport demand based on principle component analysis. *Communications in Transportation Research*, *3*, Article 100093.

Zhou, Z., Yang, Z., Zhang, Y., Huang, Y., Chen, H., & Yu, Z. (2022). A comprehensive study of speed prediction in transportation system: From vehicle to traffic. *Iscience*, Article 103909.