International Journal of
Geo-Information

Article

Temporal-Spatial Traffic Flow Prediction Model Based on Prompt Learning

Siteng Cai, Gang Liu *, Jing He, Yulun Du, Zhichao Si and Yunhao Jiang

College of Geography and Planning, Chengdu University of Technology, Chengdu 610059, China; caisiteng@stu.cdut.edu.cn (S.C.); xiao00yao@163.com (J.H.); duyulun@stu.cdut.edu.cn (Y.D.); sizhichao@stu.cdut.edu.cn (Z.S.); jiangyunhao@stu.cdut.edu.cn (Y.J.)
* Correspondence: liugang2014@cdut.edu.cn; Tel.: +86-15928111254

Abstract: Traffic flow prediction is one of the most important and attractive topics in geographical information science (GIS), traffic management, and logistics. Traffic flows exhibit significant complexity and dynamics, requiring a thorough understanding of their spatiotemporal evolution patterns for accurate prediction and analysis. Existing studies utilizing deep learning for traffic flow prediction often suffer from distribution shift issues, leading to poor generalization capabilities when dealing with data that has different spatiotemporal distributions. Based on this, we propose a traffic flow prediction model based on prompt learning, leveraging graph convolutional networks to focus on the spatiotemporal dependencies of traffic flows. The model utilizes spatiotemporal context learning capabilities to capture the periodic states of traffic flows, enhancing the extraction of spatiotemporal features by integrating spatiotemporal information. Experimental results show that the spatiotemporal traffic flow prediction model equipped with a spatiotemporal prompt learning module outperforms several mainstream benchmark models in terms of predictive performance. The model presents efficient learning performance that reaches optimal state in a short period of time, reduces the impact of distribution shifts, and can be adapted to spatiotemporal traffic flow data under varying spatiotemporal contexts.

Keywords: spatiotemporal traffic flow prediction; distribution shift; prompt learning

Academic Editors: Hartwig H. Hochmair and Wolfgang Kainz

Received: 5 November 2024
Revised: 18 December 2024
Accepted: 29 December 2024
Published: 31 December 2024

Citation: Cai, S.; Liu, G.; He, J.; Du, Y.; Si, Z.; Jiang, Y. Temporal-Spatial Traffic Flow Prediction Model Based on Prompt Learning. *ISPRS Int. J. Geo-Inf.* **2025**, *14*, 11. https://doi.org/10.3390/ijgi14010011

1. Introduction

With the rapid pace of urbanization and a significant rise in vehicle ownership, the demand for transportation has surged, leading to increasingly severe traffic congestion and a higher incidence of traffic accidents [1]. Temporal-spatial prediction of traffic flow is always one of the most important subjects in GIS, spatial network, logistics, etc. In the future, the introduction of connected and autonomous vehicles (CAVs) into the market will add complexity to urban traffic systems. In this evolving landscape, connected automated vehicles (CAVs), connected vehicles (CVs), and regular human driven vehicles (RVs) will coexist, forming a mixed traffic flow [2]. This transition poses novel challenges for urban traffic management, necessitating innovative solutions to ensure efficient and safe mobility. In response to these challenges, many countries and regions worldwide are intensifying their research and application efforts in intelligent transportation systems (ITS) and transportation GIS. ITS is not only a critical component of modern smart city infrastructure but also a key technological means for solving traffic management problems [3]. Against this backdrop, accurate traffic flow prediction has become a core element of intelligent transportation systems and traffic information services. Precise traffic flow prediction can help relevant departments promptly adjust traffic scheduling plans and implement more reasonable traffic control measures, thereby effectively alleviating congestion. Additionally, it can significantly reduce traffic safety risks and ensure road safety. Accurate traffic predictions can provide real-time, reliable travel advice to the public, enabling citizens to flexibly adjust their travel plans according to forecasted information, improving travel efficiency and ultimately enhancing the quality of life and satisfaction of urban residents. The core of traffic flow prediction encompasses support for urban planning, real-time traffic monitoring and management, and the promotion of smart city applications. Among these, the main challenge of traffic flow prediction lies in effectively simulating and predicting traffic at different times and spatial locations, as well as exploring the correlations between time, space, and traffic flows.

Given the critical importance of accurate traffic flow prediction for effective traffic planning and management, numerous researchers have investigated a wide range of high-precision techniques. These methods include time series analysis, traditional machine learning models, and deep learning models.

Time series analysis is a statistical technique used to analyze and forecast data points collected over time. This approach is particularly effective in capturing temporal patterns in traffic volume, such as daily, weekly, and seasonal trends, as well as short-term fluctuations. For instance, S. Vasantha Kumar et al. [4] selected a three-lane arterial road in Chennai, India, as their study site. They applied necessary differencing and then plotted the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) to determine the appropriate order of the Seasonal Autoregressive Integrated Moving Average (SARIMA) model. This method allowed them to accurately capture both the seasonal and nonseasonal components of the traffic flow. Emami et al. [5] developed a Kalman Filter to predict traffic flow on urban arterial roads using data obtained from connected vehicles. The proposed algorithm is computationally efficient and provides real-time predictions. While their method effectively captures the temporal information in traffic flow, it does not account for spatial information, which can also significantly influence traffic flow prediction.

Traditional machine learning models are particularly adept at handling structured data and are distinguished by their high interpretability and relatively modest computational requirements. Ata et al. [6] use the traditional machine learning method Support Vector Machine (SVM) to predict traffic flow. They propose a TCC-SVM system model to analyze traffic congestion in the environment of a smart city. While traditional machine learning models, such as decision trees and support vector machines (SVM), often perform well with smaller datasets and offer interpretability, they have certain limitations, particularly in handling spatial information. Although these models can be enhanced by incorporating spatial features like distance and direction, their spatial processing capabilities are generally weaker compared to deep learning models.

Deep learning models excel in feature representation, making them a prominent approach for spatiotemporal prediction tasks, which are widely used in the field of traffic flow prediction. Recurrent Neural Networks (RNNs) [7–9] and their variants, such as Long Short-Term Memory (LSTM) [10,11] and Gated Recurrent Units (GRUs) [12,13], have achieved excellent performance in extracting temporal features. Convolutional Neural Networks (CNNs) can effectively capture local spatial features in the traffic network through convolutional layers, extracting local spatial patterns [14–16]. When predicting traffic flow at nodes within a road network, these methods often fail to effectively leverage the network's topology to capture essential spatial information. The emergence of graph methods solves this problem and has become a mainstream traffic flow prediction method. Graph Neural Networks (GNNs) emerged as effective models for capturing complex spatial dependencies, using message-passing mechanisms to learn relationships between spatial units such as regions and road segments [17–21]. To construct adjacency matrices for graphs, researchers have explored various factors, taking into consideration static geographical distances as well as time-aware regional correlations [22]. Additionally, some researchers have used attention mechanisms. Transformer is a classic implementation of attention mechanisms, using an encoder-decoder architecture constructed through self-attention modules and feedforward neural networks [23–25]. Meanwhile, cutting-edge technologies such as meta-learning [26,27] and transfer learning [28,29] have also been widely adopted in recent years. Prompt learning [30–33] is a technique that involves fine-tuning pretrained models, particularly prevalent in the field of Natural Language Processing (NLP). The core idea of prompt learning is to design specific prompts that enable pretrained models to better adapt to downstream tasks without requiring extensive modifications to the model's parameters or the training data.

Whether these models can be applied to different datasets and tasks, which is an interesting issue [34]. For existing models, if there is a discrepancy between the distribution of training and testing data, it may lead to inaccurate predictions in real world urban traffic scenarios. Additionally, if the spatiotemporal features of different data distributions vary significantly, directly applying the parameters learned from dataset A to dataset B may result in performance discrepancies. Therefore, it is necessary to effectively adjust the traffic flow prediction model to handle such distribution changes, thereby improving its generalization capability.

The main contributions of this work are as follows: (1) We propose a novel traffic flow prediction model that incorporates prompt learning into Graph Convolutional Networks (GCNs) to enhance the model's generalization capability; (2) Based on prompt learning, we design a traffic flow prediction prompt network (TPPN) to extract spatiotemporal features from traffic flow data and generate soft prompts, which are used to adapt the model to specific tasks, thereby enhancing its adaptive capability.

2. Materials and Methods

2.1. Problem Description and Definition

Traffic flow prediction is a critical component of modern transportation management systems, playing a pivotal role in optimizing traffic operations, enhancing road safety, and reducing congestion. Accurate predictions enable traffic authorities to make informed decisions, such as dynamically adjusting traffic signals, rerouting vehicles, and planning infrastructure improvements. Moreover, efficient traffic management can lead to reduced travel times, lower fuel consumption, and decreased environmental impact. Therefore, developing robust and reliable traffic flow prediction models is of paramount importance for improving urban mobility and quality of life.

A traffic road network is a typical non-Euclidean topological structure that can be represented in the form of an undirected graph G(V,E,A). Here, V represents the nodes on the graph, with the number of nodes $|V| = N$; E represents the edges connecting two nodes; and $A \in R^{N\times N}$ is the adjacency matrix of the graph, storing the weights of the edges. Traffic flow prediction involves predicting traffic flow sequences for future time periods given a traffic road network graph G and historical traffic flow sequences over T time periods. This prediction is based on a mapping relationship, as described by Equation (1).

$$\mathrm{X_{t+1}, \cdots X_{t+T} = f(X_{t-n} \cdots, X_{t-1}, X_t)} \tag{1}$$

where X is a three-dimensional vector which is used encode spatiotemporal information, $X\epsilon R^{R\times T\times F}$. In this representation, R denotes the number of regions, T represents the time intervals, and F indicates the number of features. Each tensor $X_{R,T,F}$ corresponds to the value of the F-th feature at the R-th node and the T-th time interval. For example, in the context of traffic flow prediction, the vector X represents traffic flow data quantified as the number of vehicles passing through a specific region within fixed time intervals (e.g., every 5 min).

2.2. Experimental Setup and Datasets

The experiments were conducted using a network architecture built on the PyTorch framework and executed on an NVIDIA (CA, USA) GeForce RTX 4090. The dataset was split into training, validation, and testing sets at a ratio of 6:2:2. During training, the mean squared error (MSE) function was used as the loss function, and the Adam optimizer was used to update the parameters. To evaluate the model's performance in traffic flow prediction, we used three widely adopted evaluation metrics: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE). These metrics quantify the differences between the predicted data and the ground truth data. Lower values for these metrics indicate better performance.

To assess the model's generalization capability across different urban spatiotemporal contexts, the experimental setup was as follows: (1) In the pre-training phase, we use four datasets—PEMS03, PEMS04, PEMS07M [35], and the Chengdu Didi dataset (shown in Figure 1)—as our training set. The PEM datasets consist of records detailing traffic flow conditions across different streets and cities in California, USA. The Chengdu Didi dataset records the traffic flow index of the road network in Chengdu. (2) In the subsequent fine-tuning phase, we focus on the Chengdu Didi dataset to fine-tune and evaluate our framework. These datasets represent traffic speeds in Los Angeles, traffic flows in California, and traffic flows in Chengdu, respectively. Each target dataset was split into training, validation, and testing sets at a ratio of 6:2:2. The data collection frequency was 5 min.

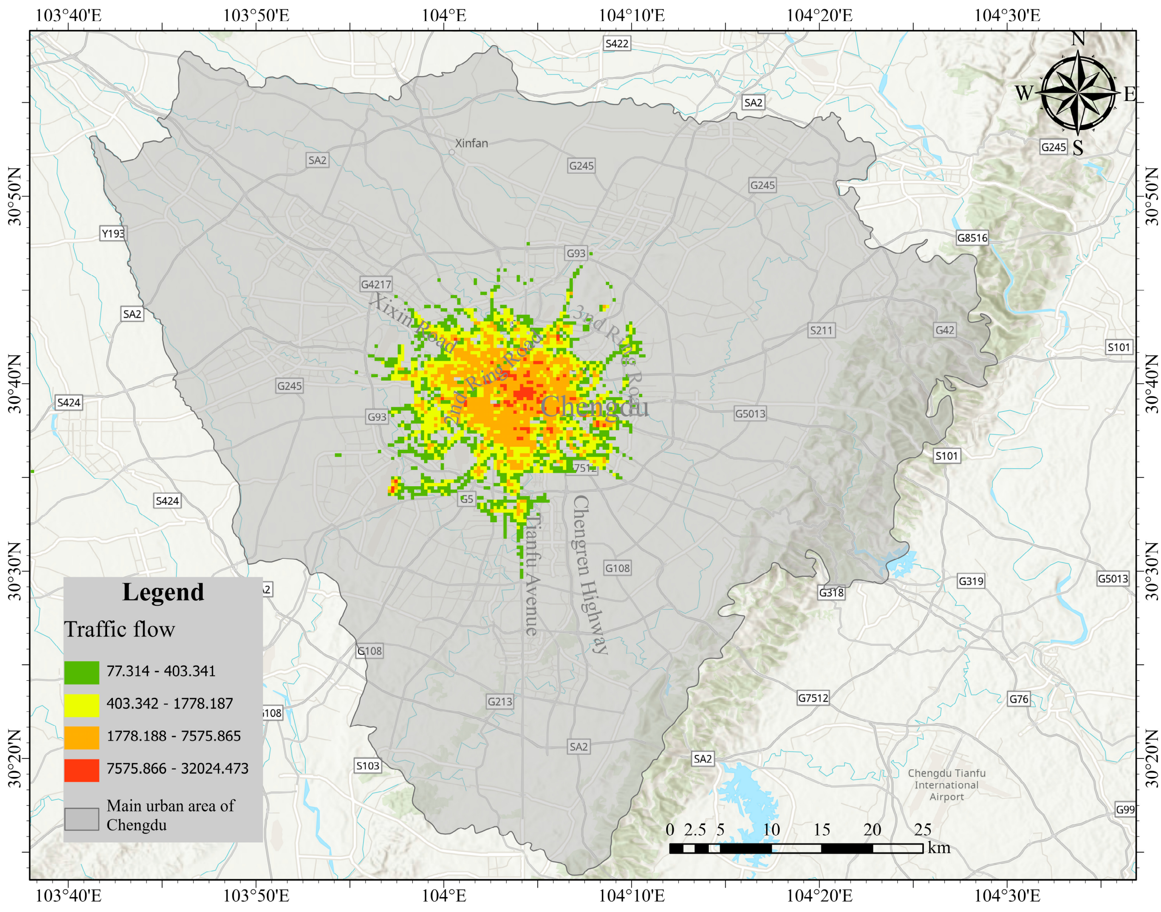

Figure 1. Chengdu Didi Datasets.

- PEMS03 contains 26,208 data points collected by 358 loop detectors.
- PEMS04 contains 16,992 data points collected by 307 loop detectors.
- PEMS07 contains 26,208 data points collected by 883 loop detectors.
- PEMS08 contains 17,856 data points collected by 170 loop detectors.
- The Chengdu Didi dataset contains 17,280 data points collected by 524 loop detectors.

2.3. Structure of the Prediction Model

The overall structure of the prediction model is shown in Figure 2. The model is divided into two main parts: the pre-training stage and the prediction stage. During the pre-training stage, data is processed and fed into the model. Through the prompt network module, historical data and various embeddings are fused to obtain the prompts required for pre-training. Based on these prompts and using prompt learning methods, the model parameters are continuously adjusted until the optimal pre-trained model is achieved.

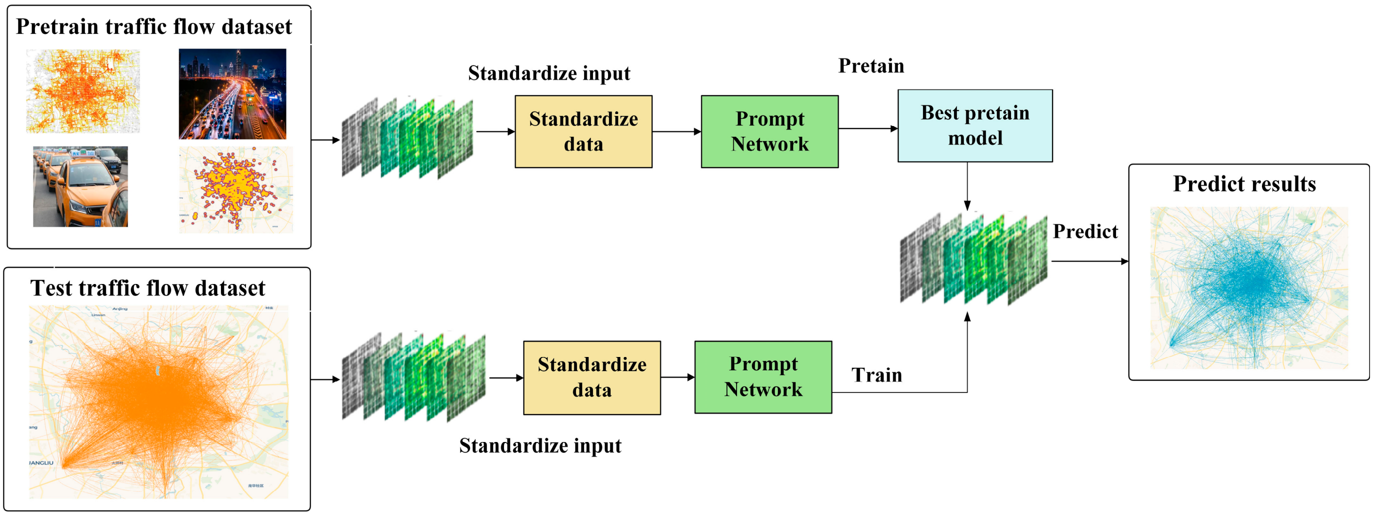

Figure 2. Model Architecture Diagram.

The data processing in the prediction stage is identical to that of the pre-training stage. After passing through the prompt network, the generated prompts are used to train the model, and the results of this training are propagated forward to obtain the predicted outcomes.

2.4. Prompt Learning

Prompt Learning is an emerging machine learning paradigm, particularly suited for the field of Natural Language Processing (NLP). It guides pre-trained models to better adapt to downstream tasks by adding specific prompts to the input data. Prompt learning avoids the gap between pre-training and fine-tuning by adding templates to introduce additional. In this way, language models can achieve ideal results in few-shot or zero-shot scenarios. Traditional fine-tuning methods typically require updating a large number of model parameters, which can lead to the model forgetting the knowledge learned during the pre-training phase (a phenomenon known as “catastrophic forgetting”). In contrast, prompt learning guides the model to better adapt to new tasks by adding specific prompts to the input data, without modifying the model’s parameters. This approach not only reduces the reliance on large-scale labeled data but also retains the general knowledge acquired during pre-training. To enhance the model’s adaptability to downstream tasks after pre-training, this paper uses soft prompts. Soft prompts are composed of a set of learnable continuous vectors that are continuously optimized during training. This approach provides greater flexibility and adaptability, allowing the model to better capture the nuanced differences of various tasks. By refining these soft prompts, the model can more effectively adapt to new and diverse downstream tasks while retaining the knowledge gained during pre-training.

2.5. Traffic Flow Prediction Prompt Network

The prompt network is a critical component of a traffic flow prediction model that combines Graph Convolutional Networks (GCN), Multi-Layer Perceptrons (MLP), and Prompt Learning. This module is essentially the specific implementation of Prompt Learning for the task of traffic flow prediction. This module integrates time series data, temporal information (such as hour of the day and day of the week), and spatial information (such as graph Laplacian positional encoding) by embedding them into low-dimensional spaces. These embedded features are then concatenated to form a comprehensive hidden state hidden. The module uses Graph Convolutional Network (GCN) layers to capture spatial dependencies in the traffic network and two Multi-Layer Perceptron (MLP) encoders to perform non-linear transformations on the GCN output, enhancing the feature representation. Finally, the processed hidden state is combined with the original input data's base features to generate a prompt "Prompt". The prompt is then L2-normalized to ensure that the norm of each time step's feature vector is 1. The prompt network structure is shown in Figure 3.

This module consists of two main parts: (1) Node embedding, time embedding, and time series embedding. Capturing time and space-related context from the dataset, enabling the model to learn from specific contexts within the data, thereby facilitating effective adaptation to a variety of spatiotemporal scenarios. (2) Temporal dependency encoder and spatial dependency encoder building.

To initialize the representation of spatiotemporal data, we use a projection layer with two steps: normalization using a Z-Score function and enhancement through a linear transformation.

$$\widetilde{X} = Z(X) = \frac{X - \mu}{\sigma} \tag{2}$$

where μ and σ represent the mean and standard deviation of the original spatiotemporal matrix $X \epsilon R$.

Time embedding. To capture the dynamic and periodic temporal patterns from different traffic flow data, this study utilizes multi-resolution temporal features, specifically the hour of the day and the day of the week. Given time step index $t_d \epsilon \{0, 1 \ldots, 288\}$ and $t_w \epsilon \{0, 1 \ldots, 7\}$ the embedding for time in day and time in a week can be calculated using the following formula:

$$E_{td} = \text{Embedding}(t_d, W_d), E_{tw} = \text{Embedding}(t_w, W) \tag{3}$$

$$E_t = \text{Concat}(E_{tw}, E_{td}) \tag{4}$$

where t_d is an integer tensor that represents the time step index for each sample, with a range of [0, 288] (assuming a time step of 5 min, resulting in 24 × 60/5 = 28,824 × 60/5 = 288 time steps in a day); t_w is an integer tensor that represents the day of the week for each sample, with a range of [0, 6] (0 corresponds to Monday and 6 corresponds to Sunday); W_d is the weight matrix of the embedding layer, which maps the time step index to a low-dimensional vector space; E_t is the Time embedding.

Time series embedding. The time series embedding can be calculated using the following formula:

$$E_{ts} = \text{Linear}(X, W_{ts}, b) \tag{5}$$

where W_{ts} is the weight matrix of the linear layer and b is the bias term of the linear layer.

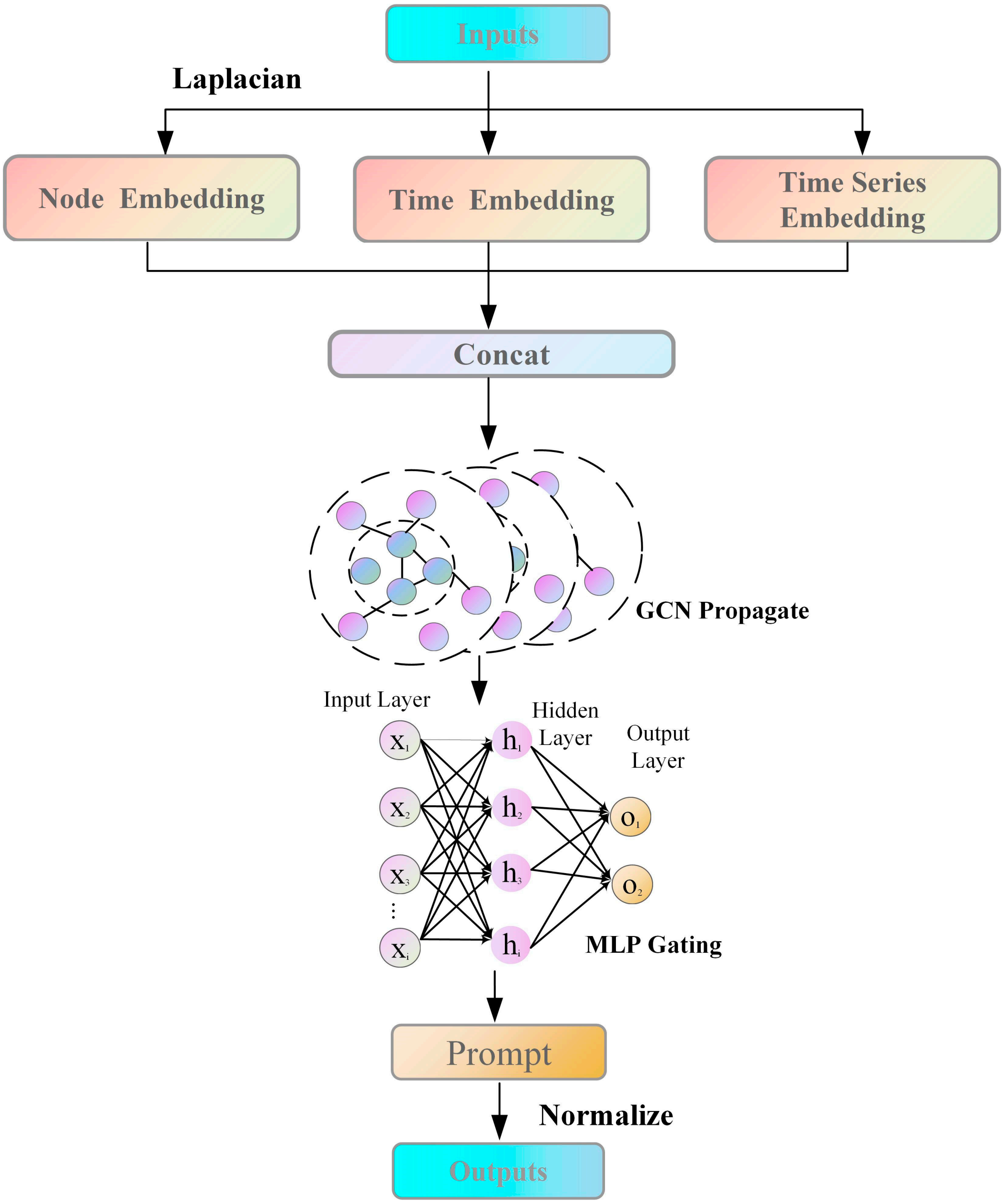

Figure 3. The structure of traffic flow prediction prompt network.

Node embedding. To enhance the context information related to spatial attributes, we incorporate the road network structure as encoding features that reflect spatial context. This process begins with the formulation of a normalized Laplacian matrix, defined as follows:

$$\mathrm{L} = \mathrm{I} - \mathrm{D}^{-1/2}\mathrm{A}\mathrm{D}^{-1/2} \tag{6}$$

where I, D and A denote the identity matrix, degree matrix, and adjacency matrix, respectively; L represents the Laplacian matrix. The adjacency matrix is computed based on the distances between nodes. Given the input Laplacian positional encoding L. The node embedding can be calculated using the following formula:

$$\mathrm{E}_{\mathrm{node}} = \mathrm{Linear}_2(\mathrm{ReLu}(\mathrm{Linear}_1(\mathrm{L}))) \tag{7}$$

where L is the input Laplacian positional encoding matrix, representing the position of each node in the graph; $\mathrm{Linear}_1()$ is the first linear layer, and $\mathrm{Linear}_2()$ is the second linear layer.

Subsequently, these features are concatenated via Concat to obtain the initial spatiotemporal embeddings.

$$\overset{\sim}{\mathrm{E}} = \mathrm{Concat}[\mathrm{E}_{\mathrm{ts}}, \mathrm{E}_{\mathrm{node}}, \mathrm{E}_{\mathrm{t}}] \tag{8}$$

where $\overset{\sim}{\mathrm{E}}$ represents the integrated embedding, where we combine the time embedding E_{t}, the temporal context embedding E_{ts}, and the node embedding $\mathrm{E}_{\mathrm{node}}$.

GCN is used to capture spatial correlations between nodes. Spatial Dependency Encoder: Inspired by the use of graph neural networks in capturing spatial correlations between geographic locations, we utilize message passing based on graph convolutions to encode spatial correlations. The adjacency matrix A, defined in the equation, serves as the connectivity matrix within the graph network framework. The formal process of spatial encoding is as follows:

$$\mathrm{h}_{\mathrm{gcn}} = \mathrm{ReLU}(\mathrm{A} \times \mathrm{Conv2D}(\mathrm{h})) + \mathrm{h} \tag{9}$$

where $\mathrm{h}_{\mathrm{gcn}}$ is the output feature matrix after applying the GCN layer; A is the normalized adjacency matrix; h is a tensor that contains the hidden state after concatenating the time series embedding, spatial embedding, and temporal embedding. Additionally, this paper adopts a residual network to mitigate potential over-smoothing effects caused by multiple layers of GCN.

Temporal Dependency Encoder. To capture the dependencies between different time slots and retain the patterns of temporal changes in the data, we introduce an MLP encoder. The formal operation of this mechanism is as follows:

$$\mathrm{h}_{\mathrm{m}} = \mathrm{Conv2D}(\mathrm{Dropout}(\mathrm{ReLU}(\mathrm{Conv2D}(\mathrm{h}_{\mathrm{input}})))) \tag{10}$$

where h_{m} is the output after the MLP transformation, and it has the same shape as $\mathrm{h}_{\mathrm{input}}$; Conv2D is a convolution operation used to perform linear transformations; and Dropout is a regularization technique used to prevent overfitting.

$$\mathrm{h}_1 = \mathrm{h}_{\mathrm{m}}(\mathrm{h}_{\mathrm{gcn}}(\mathrm{h})),\ \mathrm{prompt} = \mathrm{Normalize}(\mathrm{h}_{\mathrm{m}}(\mathrm{h}_{\mathrm{gcn}}(\mathrm{h}_1))) \tag{11}$$

where Normalize() represents the normalization of L2. By applying two GCN layers and two MLP layers to generate the prompt, the model can abstract more complex features at different levels. The residual connections help maintain information flow and alleviate the vanishing gradient problem. This design enables Prompt Network to better handle complex spatiotemporal data, capturing multi-scale spatial relationships between nodes and temporal dynamics.

3. Results

3.1. Experimental Results

To evaluate the effectiveness of our model, we select five advanced spatiotemporal prediction models as baseline models. These include RNN-based models, attention-based models, and spatiotemporal prediction methods based on GNN.

- AGCRN [36]: This method combines RNN with learnable node embeddings to capture personalized spatiotemporal patterns of regions.
- MTGNN [37]: This method uses temporal convolutional networks combined with skip connections to capture temporal dependencies and uses a learnable graph network to model spatial correlations.
- STWA [38]: This approach extends the standard attention mechanism by incorporating specialized node features and temporal dynamic parameters.

- STGCN [39]: This method encodes temporal dependencies using gated convolutional networks and captures local spatial relationships using GCNs.
- TGCN [40]: This model utilizes a combination of RNN and GCN to model temporal dependencies and spatial correlations, respectively.

The experimental results are shown in Table 1 and Figure 4, presenting the prediction error metrics for each baseline model and the proposed model on the PEMS03, PEMS04, PEMS07M, and Chengdu Didi datasets.

Table 1. Prediction Error Metrics for Baseline Models and our Model on Various Datasets.

Method	PEMS03			PEMS04			PEMS07M			Chengdu Didi		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
MTGNN	15.52	26.67	18.12%	19.12	31.09	12.97%	2.63	5.32	6.58	2.33	3.55	10.62%
AGCRN	16.55	27.1	17.51%	20.8	32.64	14.25%	2.87	5.69	7.27%	2.63	3.98	12.2%
STWA	16.76	26.3	17.57%	19.15	30.63	12.88%	2.75	5.57	6.92%	2.36	3.57	10.74%
STGCN	18.63	30.59	20.56%	20.8	33.04	13.98%	3.17	6.39	8.32%	2.54	3.88	11.8%
TGCN	17.95	28.19	18.32%	22.56	35.62	16.35%	5.05	8.56	13.68%	3.14	4.56	14.3%
Ours	15.42	26.5	17.09%	18.95	29.99	12.73%	2.62	5.27	6.60%	2.33	3.56	10.73

The x-axis represents the time steps, and the y-axis represents the traffic flow. From the results, it is evident that our model achieves the best performance metrics across all datasets. On the temporal dimension, the model can effectively predict long-term cyclical trends and short-term sudden changes in traffic flow. During periods of sharp increases or decreases in traffic flow across different sensors, the model consistently predicts the onset of abrupt changes and their states. The model proposed in this paper significantly outperforms the AGCRN, STWA, STGCN, and TGCN models on the PEMS03, PEMS04, PEMS07M, and Chengdu Didi datasets. Specifically, our model achieves better MAE (Mean Absolute Error) scores on all four datasets compared with MTGNN, with improvements of 0.1, 0.17, and 0.05 respectively. Among them, TGCN shows the worst overall metrics, likely because TGCN uses a combination of RNN and GCN to model temporal dependencies and spatial correlations separately, without integrating the temporal and spatial relationships. The prediction performance of AGCRN is limited by RNN, resulting in weaker performance compared with most spatiotemporal graph convolution methods. The graph convolution operations in STGCN typically only consider the direct neighbors of each node, which makes it difficult to capture broader global spatial dependencies. For tasks that require considering the mutual influence between distant nodes (e.g., long-distance traffic flow propagation in a city), STGCN may miss important information. STWA uses wavelet transforms to capture multi-scale features in time series data. However, wavelet transforms rely on predefined wavelet basis functions (such as Haar, Daubechies, etc.), which are fixed and cannot adaptively adjust to the characteristics of different datasets. If the temporal patterns in the data do not match the selected wavelet basis functions, the model may miss important features, leading to a decline in prediction performance. To enhance the model’s generalization capability, we conducted pre-training on multiple city datasets. This approach allows the model to share common traffic patterns across different cities while retaining the unique characteristics of each city. By doing so, we not only improve the model’s robustness but also reduce its reliance on large amounts of labeled data. During the training process, we use data augmentation techniques and regularization methods such as Dropout and L2 regularization to prevent overfitting. These techniques help the model maintain good generalization performance when faced with new data, especially when the distribution of test data differs from that of the training data. Because of the reasons mentioned above, our model demonstrates superior performance when faced with diverse datasets.

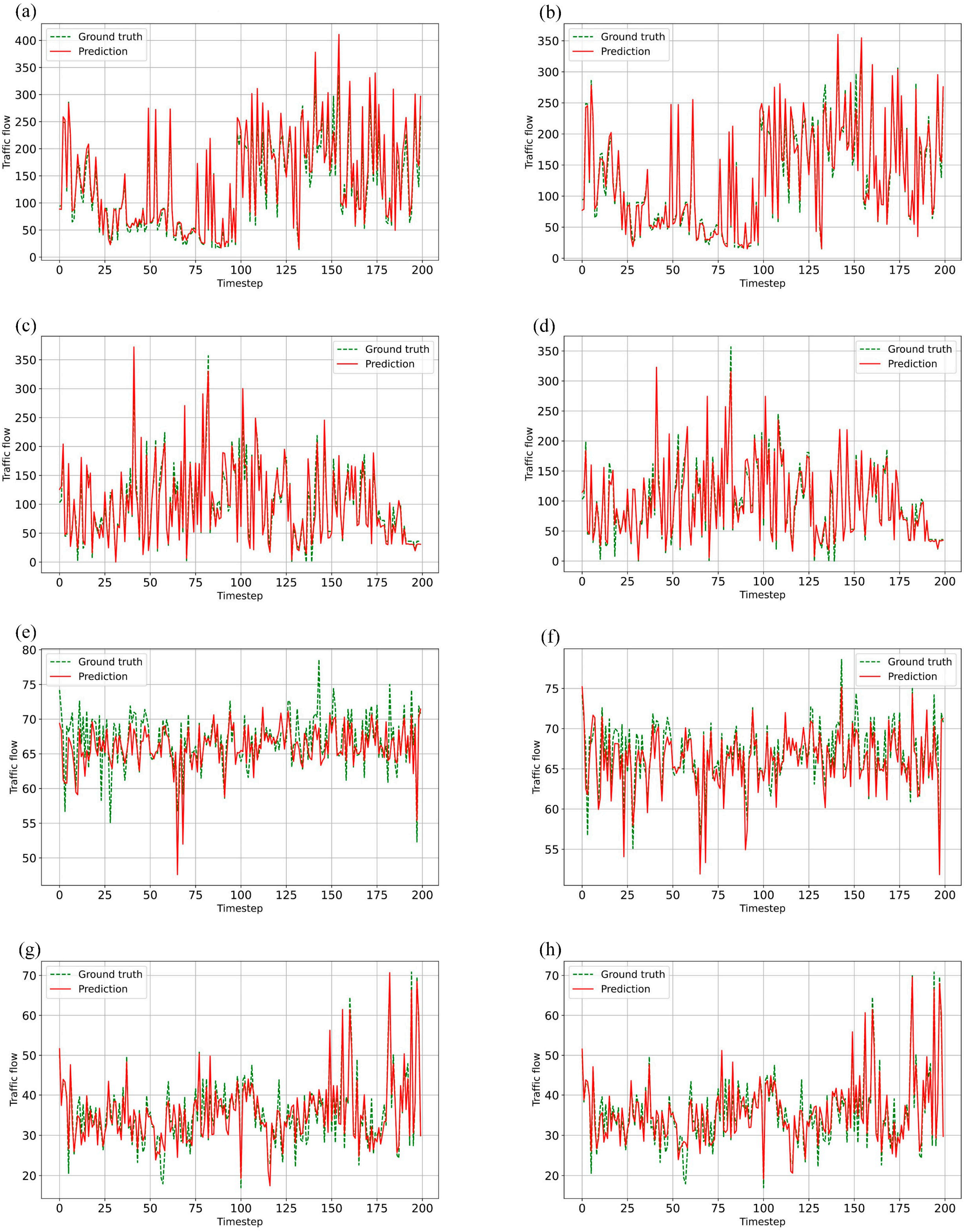

Figure 4. Prediction Performance of the STGCN Model and our Model on Various Datasets: (**a**,**c**,**e**,**g**) describe the performance of the STGCN model on various datasets; (**b**,**d**,**f**,**h**) describe the performance of our model on various datasets.

3.2. Ablation Study Results

To analyze the effectiveness of the various components of the model, we design four variants for ablation experiments on the PEMS03, PEMS04, and PEMS08 datasets and analyze the results. All models have the same settings and parameters as the Baseline Model.

(1) "w/o TC" refers to removing temporal embeddings, and "w/o SC" refers to removing spatial embeddings.

(2) "w/o TE" refers to removing the temporal correlation encoder, and "w/o SE" refers to removing the spatial correlation encoder.

The results of the ablation experiments are shown in Table 2 and Figure 5. It is clear that our model performs significantly better than the other variants, proving the effectiveness of each component in our model. As shown in Table 2 and Figure 4, removing the temporal embeddings has the greatest impact on model performance. On the PEMS03 dataset, the MAE increased from 2.62 to 2.98, which is a rise of 13.74%. This indicates that temporal embeddings contribute the most to the model, possibly because the temporal embedding module captures the characteristic increase in traffic flow during peak hours. In urban areas, the traffic volume during morning and evening rush hours is much higher than during other periods, making this feature generalizable across all traffic datasets.

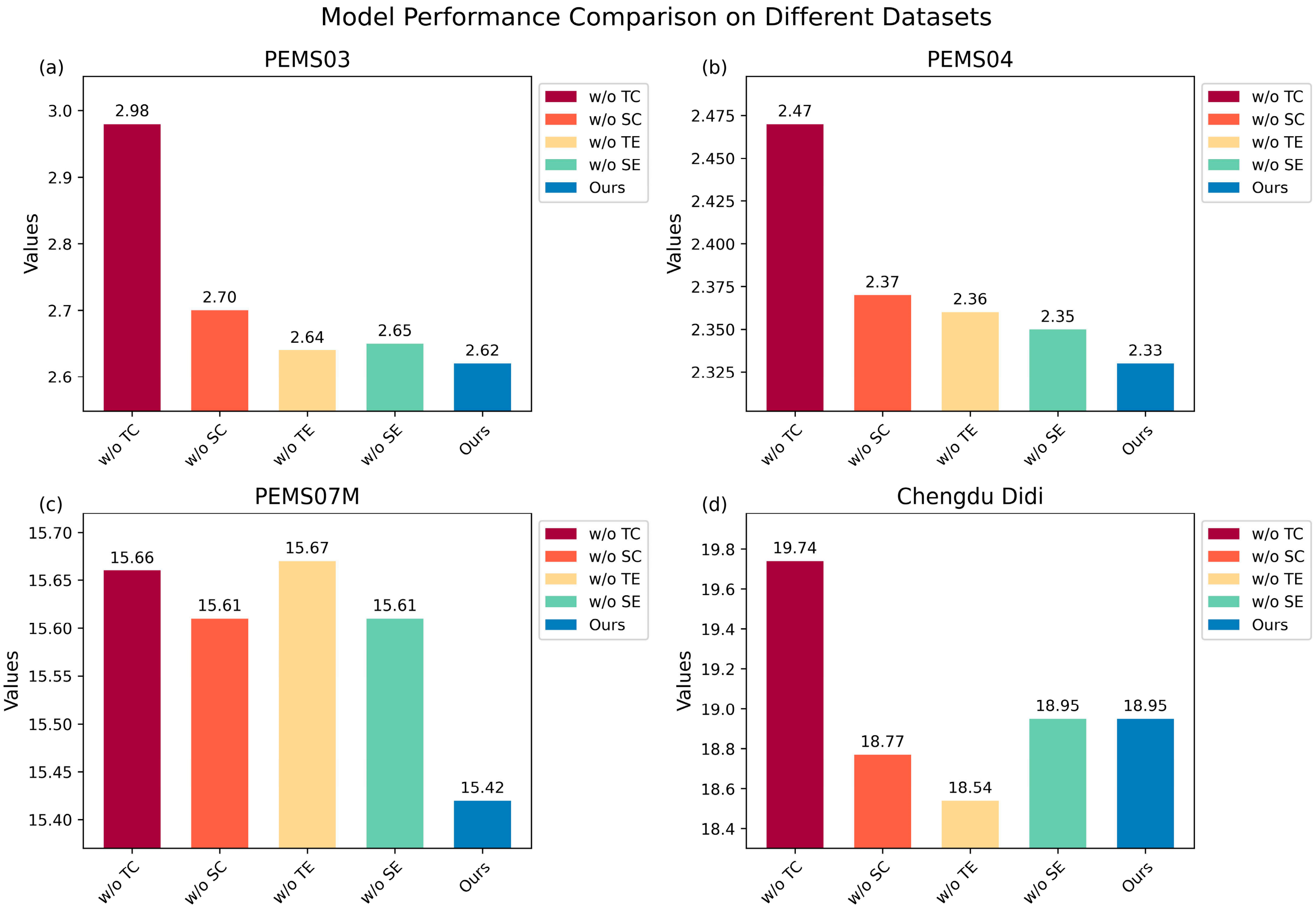

Figure 5. Ablation Experiment Results for the Baseline Model and Four Variants. (**a**) Ablation Study on the PEMS03 Dataset. (**b**) Ablation Study on the PEMS04 Dataset. (**c**) Ablation Study on the PEMS07M Dataset. (**d**) Ablation Study on the Chengdu Didi Dataset.

Table 2. Evaluation Metrics for the Baseline Model and Four Variants on the PEMS03, PEMS04, PEMS07M, and Chengdu Didi Datasets.

Method	**PEMS03**			**PEMS04**			**PEMS07M**			**Chengdu Didi**		
	MAE	**RMSE**	**MAPE**	**MAE**	**RMSE**	**MAPE**	**MAE**	**RMSE**	**MAPE**	**MAE**	**RMSE**	**MAPE**
w/o TC	15.66	26.29	15.1%	19.74	31.44	13.52%	2.98	5.69	7.51%	2.47	3.75	11.45%
w/o SC	15.61	27.23	15.65%	18.77	30.15	12.95%	2.7	5.31	6.87%	2.37	3.6	10.9%
w/o TE	15.71	27.68	15.84%	18.62	30.46	12.74%	2.64	5.27	6.65%	2.36	3.59	10.76%
w/o SE	15.67	26.43	15.89%	18.54	30.19	12.54%	2.65	5.32	6.61%	2.35	3.59	10.79%
Ours	15.42	26.5	17.09%	18.95	29.99	12.73%	2.62	5.27	6.60%	2.33	3.56	10.73

The removal of the spatial correlation module shows significant differences on the PEMS07M and Chengdu Didi datasets. On the Chengdu Didi dataset, the performance is even better than our model, possibly due to the reduction in model complexity. However, performance on the PEMS07M dataset is the worst, indicating that spatial correlation encoding significantly enhances the model's generalization capability and reduces the impact of distribution shifts during training. These results demonstrate the necessity of spatiotemporal embeddings and spatiotemporal correlation encodings, with temporal embeddings contributing the most to the model's performance.

3.3. Model Performance

To further verify the performance of our model, first, we use it to conduct an experiment without early stopping for 100 epochs on the same dataset, PEMS07M. We compare training loss and validation loss to demonstrate the model's learning effect and whether underfitting or overfitting occurred. The experimental results are shown in Figure 6.

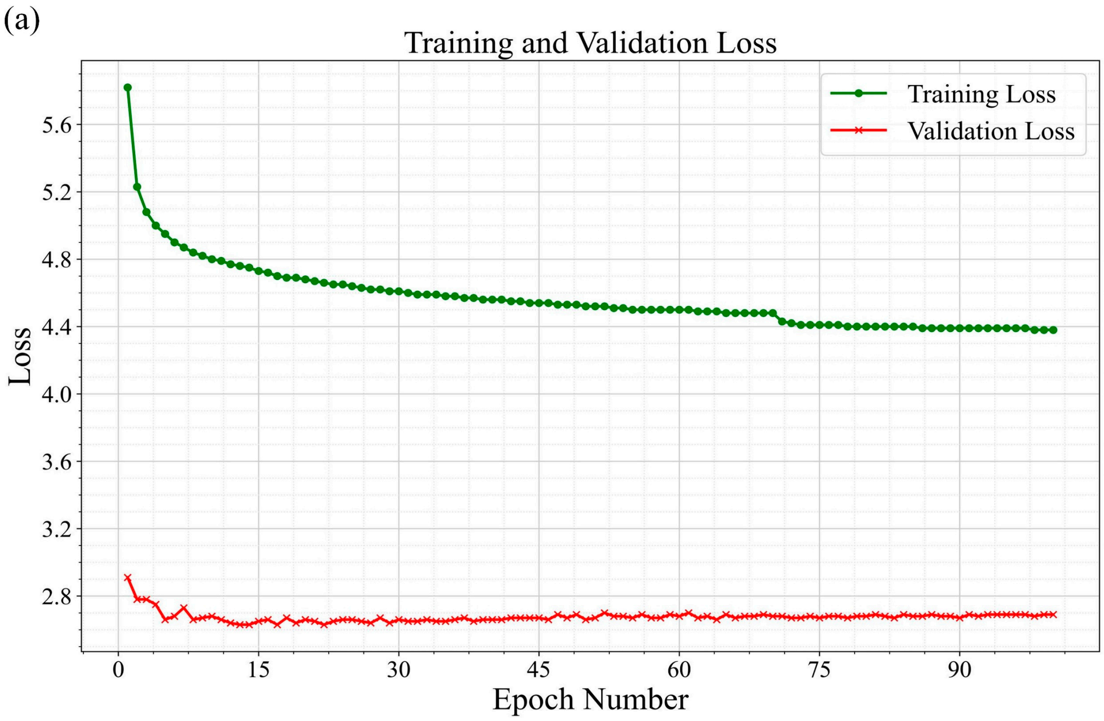

Figure 6. *Cont.*

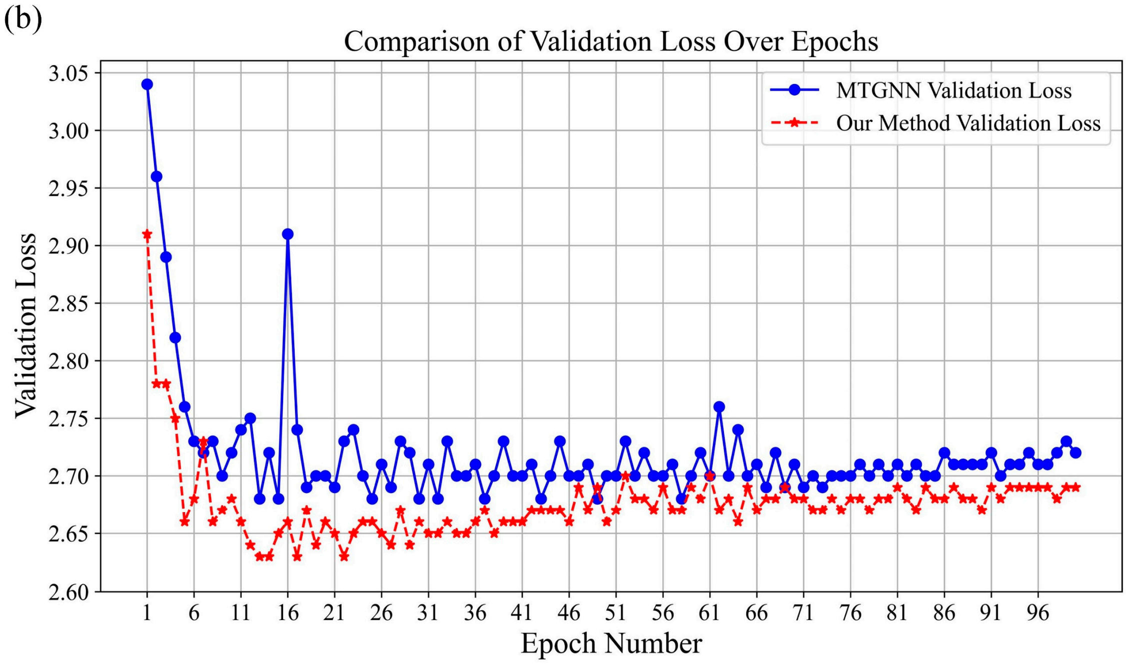

Figure 6. Analysis of model performance. (**a**) Comparison of Training Loss and Validation Loss for our Model; (**b**) Comparison of Validation Loss between our Model and MTGNN Model.

Secondly, we study the convergence speed of the model on the same dataset, PEMS07M, and compare it with the MTGNN model. The results are shown in Figure 6.

From Figure 6, it can be observed that both the training loss and validation loss decrease during the first 20 epochs as the model was trained. However, after the initial 20 epochs, the training loss continues to decrease while the validation loss remains nearly unchanged or even shows an upward trend. This indicates that the model learns effectively during the first 20 epochs and reaches an optimal state around the 20th epoch, highlighting the importance of implementing an early stopping mechanism. The validation loss being lower than the training loss might be due to the use of Dropout and Z-Score normalization, which can increase the training loss but does not apply regularization terms during validation, potentially leading to lower validation loss

As shown in Figure 6, both our model and the MTGNN model converge rapidly, but the MTGNN model stabilizes around a value of approximately 2.7, whereas our model reaches a value of around 2.6, demonstrating significantly higher accuracy than the MTGNN model.

4. Discussion

Distribution shift is a significant challenge in the field of machine learning, particularly in deep learning, as it can lead to a marked decline in model performance and weakening of generalization ability. When the distribution of training data differs from that of test data or real-world application scenarios, models may fail to effectively adapt to new situations, thereby affecting the accuracy and reliability of decision-making. To address this issue, various strategies have been proposed in recent years, including but not limited to domain adaptation, domain generalization, and data augmentation [41–44]. Domain adaptation techniques improve model performance on target domains by leveraging the relationship between source and target domains; domain generalization aims to enable models to perform well on unseen domains; and data augmentation enhances model robustness by transforming existing data to simulate different data distributions. This paper aims to explore and propose an effective framework that combines the advantages of these methods to mitigate or eliminate the impact of distribution shift on the performance of deep learning models, enhancing their applicability and stability in increasingly diverse and complex real-world applications.

In this study, we focus on the temporal-spatial traffic flow prediction using prompt learning. The traffic flow prediction model based on prompt learning builds upon graph convolutional neural networks by incorporating prompt learning and a prompt network. This approach addresses distribution shifts in some datasets and improves the accuracy of traffic flow predictions. It also provides a scalable framework that aligns multi-source spatiotemporal data, enabling its extension to other prediction domains in urban traffic.

Prompt learning introduces specific prompt vectors that enable the model to better adapt to downstream tasks. Experimental results show that using soft prompts can significantly enhance the performance of the model. The prompt vectors integrate historical data and various embeddings (such as temporal sequence embeddings, spatial embeddings, and temporal embeddings) to guide the training of the model, thereby improving its generalization capabilities across different datasets and tasks. Additionally, the use of the infoNCE loss function further enhances the robustness and discriminative power of the model, helping it learn more stable and effective feature representations.

Compared with existing traffic flow prediction models, our model demonstrates higher accuracy and robustness across multiple datasets. Specifically, compared with the MTGNN model, our model exhibits lower validation loss and faster convergence. These results indicate that the introduction of prompt learning and the prompt network significantly enhances model performance. Despite the significant achievements of this study, there are still some limitations. First, the complexity of the model is relatively high, and the training time is longer, which may limit its application in large-scale real-time traffic flow prediction. Second, while prompt learning significantly improves the model's generalization ability, designing more effective prompt vectors remains an area worthy of further research. Finally, the model's performance in handling special conditions such as extreme weather and unexpected events still needs further verification and improvement.

Moreover, this study has widely practical applications. The proposed model provides significant theoretical and technical supports for improving traffic management systems in both urban and highway settings. It facilitates dynamic adjustment of traffic signals to optimize traffic flow, improves accident prevention and emergency response through precise risk assessment, and alleviates congestion by providing real-time traffic guidance and supporting infrastructure planning. For instance, in the event of a traffic accident, the model can rapidly predict the extent and duration of the impact on surrounding traffic. This capability enables emergency departments to devise optimal rescue routes and traffic diversion plans, thereby shortening response times and minimizing the overall disruption to traffic flow [45]. Moreover, the model promotes the integration and coordination of connected and autonomous vehicles (CAVs), connected vehicles (CVs), and regular vehicles (RVs), fostering smarter and safer multi-modal transportation. By integrating the car-following model of connected autonomous vehicles (CAVs), it can also reduce energy consumption and traffic emissions during car-following maneuvers, thereby making traffic behavior more environmentally friendly and safer [46]. On highways, the short-term traffic flow prediction model plays a crucial role in maintaining smooth and safe traffic conditions. Our model can optimize ramp metering. By predicting traffic volumes at on-ramps and off-ramps, the model can dynamically adjust ramp metering rates to prevent congestion from spilling over into the main highway, ensuring a steady flow of traffic. The model can quickly identify potential incidents, such as accidents or stalled vehicles, by detecting abnormal traffic patterns, enhancing incident detection and response. This early detection allows for faster deployment of emergency services and implementation of traffic control measures, reducing the likelihood of secondary accidents and minimizing delays. In urban traffic management, traffic flow prediction enables intelligent transportation systems (ITS) to dynamically adjust signal timings based on forecasted traffic volumes. This reduces vehicle waiting times and improves road throughput efficiency. Additionally, by predicting passenger demand and traffic flows, public transportation routes and schedules can be optimized to ensure that public transit vehicles arrive at stops on time, thereby reducing passenger wait times.

Overall, the robust adaptability and high predictive accuracy of the model render it an invaluable tool for data-driven decision-making in traffic management, contributing to more efficient, safe, and sustainable urban mobility solutions.

5. Conclusions

This paper introduces a spatiotemporal traffic flow prediction model based on prompt learning, targeting the significant complexity and dynamics present in traffic flows, as well as the notable differences in spatiotemporal distribution among datasets. The traffic flow prediction model uses a prompt learning approach to generate an optimal pre-trained model during the pre-training phase. By using a prompting network module to acquaint itself with the characteristics of the dataset, it aims to enhance the generalization ability of the traffic flow prediction model across various forecasting scenarios. Extensive experiments were carried out on the PEMS03, PEMS04, PEMS07M, and Chengdu Didi datasets. The results indicate that the spatiotemporal traffic flow prediction model incorporating a spatiotemporal prompt learning module exhibits superior predictive performance compared to mainstream baseline models. The model demonstrates excellent learning performance, good convergence speed, and iteration efficiency, thus saving training time and reducing the impact of distribution shifts. It is capable of adapting to spatiotemporal traffic flow data under diverse spatiotemporal contexts.

In summary, the proposed traffic flow prediction model has superior performance in handling distribution shifts and adapting to varying spatiotemporal contexts. This model can enhance urban traffic management by providing more accurate and reliable predictions, leading to better traffic control and reduce congestion. Future work will focus on extending the model to other cities with different traffic patterns and exploring its integration into real-time traffic management systems.

Author Contributions: Siteng Cai: conceptualization, writing, formal analysis, methodology. Gang Liu: conceptualization, methodology, funding acquisition, review and editing, supervision. Jing He: review, formal analysis. Yulun Du: software. Zhichao Si: data curation. Yunhao Jiang: investigation. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Natural Science Foundation of China (grant numbers 42371418).

Data Availability Statement: Data is available according to link “http://pems.dot.ca.gov/” (accessed on 15 June 2024).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lieberthal, E.B.; Serok, N.; Duan, J.; Zeng, G.; Havlin, S. Addressing the urban congestion challenge based on traffic bottlenecks. *Philos. Trans. A* **2024**, *382*, 20240095. [CrossRef] [PubMed]
2. Qin, Y.; Luo, Q.; Wang, H. Stability analysis and connected vehicles management for mixed traffic flow with platoons of connected automated vehicles. *Transp. Res. Part C Emerg. Technol.* **2023**, *157*, 104370. [CrossRef]
3. Li, Y.; Yu, R.; Shahabi, C.; Liu, Y. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv* **2017**, arXiv:1707.01926.

4. Kumar, S.V.; Vanajakshi, L. Short-term traffic flow prediction using seasonal ARIMA model with limited input data. *Eur. Transp. Res. Rev.* **2015**, *7*, 21. [CrossRef]
5. Emami, A.; Sarvi, M.; Bagloee, S.A. Short-term traffic flow prediction based on faded memory Kalman Filter fusing data from connected vehicles and Bluetooth sensors. *Simul. Model. Pract. Theory* **2020**, *102*, 102025. [CrossRef]
6. Ata, A.; Khan, M.A.; Abbas, S.; Khan, M.S.; Ahmad, G. Adaptive IoT empowered smart road traffic congestion control system using supervised machine learning algorithm. *Comput. J.* **2021**, *64*, 1672–1679. [CrossRef]
7. Wang, Y.; Gao, Z.; Long, M.; Wang, J.; Philip, S.Y. Predrnn++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; PMLR: New York, NY, USA; pp. 5123–5132.
8. Lin, Z.; Li, M.; Zheng, Z.; Cheng, Y.; Yuan, C. Slf-attention convlstm for spatiotemporal prediction. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 11531–11538.
9. Ren, X.; Gu, H.; Wei, W. Tree-RNN: Tree structural recurrent neural network for network traffic classification. *Expert Syst. Appl.* **2021**, *167*, 114363. [CrossRef]
10. Zhao, Z.; Chen, W.; Wu, X.; Chen, P.C.; Liu, J. LSTM network: A deep learning approach for short-term traffic forecast. *IET Intell. Transp. Syst.* **2017**, *11*, 68–75. [CrossRef]
11. Fu, R.; Zhang, Z.; Li, L. Using LSTM and GRU neural network methods for traffic flow prediction. In Proceedings of the 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), Wuhan, China, 11–13 November 2016; IEEE: New York, NY, USA, 2016; pp. 324–328.
12. Zhang, Q.; Zhou, L.; Su, Y.; Xia, H.; Xu, B. Gated Recurrent Unit Embedded with Dual Spatial Convolution for Long-Term Traffic Flow Prediction. *ISPRS Int. J. Geo-Inf.* **2023**, *12*, 366. [CrossRef]
13. Hussain, B.; Afzal, M.K.; Ahmad, S.; Mostafa, A.M. Intelligent traffic flow prediction using optimized GRU model. *IEEE Access* **2021**, *9*, 100736–100746. [CrossRef]
14. Ranjan, N.; Bhandari, S.; Zhao, H.P.; Kim, H.; Khan, P. City-wide traffic congestion prediction based on CNN, LSTM and transpose CNN. *IEEE Access* **2020**, *8*, 81606–81620. [CrossRef]
15. Zhang, W.; Yu, Y.; Qi, Y.; Shu, F.; Wang, Y. Short-term traffic flow prediction based on spatio-temporal analysis and CNN deep learning. *Transp. A Transp. Sci.* **2019**, *15*, 1688–1711. [CrossRef]
16. Bogaerts, T.; Masegosa, A.D.; Angarita-Zapata, J.S.; Onieva, E.; Hellinckx, P. A graph CNN-LSTM neural network for short and long-term traffic forecasting based on trajectory data. *Transp. Res. Part C Emerg. Technol.* **2020**, *112*, 62–77. [CrossRef]
17. Ali, A.; Zhu, Y.; Zakarya, M. Exploiting dynamic spatio-temporal graph convolutional neural networks for citywide traffic flows prediction. *Neural Netw.* **2022**, *145*, 233–247. [CrossRef]
18. Roy, A.; Roy, K.K.; Ahsan Ali, A.; Amin, M.A.; Rahman, A.M. SST-GNN: Simplified spatio-temporal traffic forecasting model using graph neural network. In Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, Virtual, 11–14 May 2021; Springer International Publishing: Cham, Switzerland, 2021; pp. 90–102.
19. Jiang, W.; Luo, J. Graph neural network for traffic forecasting: A survey. *Expert Syst. Appl.* **2022**, *207*, 117921. [CrossRef]
20. Kashyap, A.A.; Raviraj, S.; Devarakonda, A.; Nayak K, S.R.; KV, S.; Bhat, S.J. Traffic flow prediction models–A review of deep learning techniques. *Cogent Eng.* **2022**, *9*, 2010510. [CrossRef]
21. Chen, C.; Liu, Y.; Chen, L.; Zhang, C. Bidirectional spatial-temporal adaptive transformer for urban traffic flow forecasting. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *34*, 6913–6925. [CrossRef]
22. Jiang, J.; Han, C.; Zhao, W.X.; Wang, J. Pdformer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; Volume 37, pp. 4365–4373.
23. Vaswani, A. Attention is all you need. *arXiv* **2017**, arXiv:arXiv:1706.03762.
24. Niu, Z.; Zhong, G.; Yu, H. A review on the attention mechanism of deep learning. *Neurocomputing* **2021**, *452*, 48–62. [CrossRef]
25. Li, M.; Wang, Y.; Wang, Z.; Zheng, H. A deep learning method based on an attention mechanism for wireless network traffic prediction. *Ad Hoc Netw.* **2020**, *107*, 102258. [CrossRef]
26. Hospedales, T.; Antoniou, A.; Micaelli, P.; Storkey, A. Meta-learning in neural networks: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 5149–5169. [CrossRef]
27. Vanschoren, J. Meta-learning. In *Automated Machine Learning: Methods, Systems, Challenges*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 35–61.
28. Weiss, K.; Khoshgoftaar, T.M.; Wang, D. A survey of transfer learning. *J. Big Data* **2016**, *3*, 1345–1459. [CrossRef]
29. Zhuang, F.; Qi, Z.; Duan, K.; Xi, D.; Zhu, Y.; Zhu, H.; Xiong, H.; He, Q. A comprehensive survey on transfer learning. *Proc. IEEE* **2020**, *109*, 43–76. [CrossRef]
30. Ding, N.; Hu, S.; Zhao, W.; Chen, Y.; Liu, Z.; Zheng, H.T.; Sun, M. Openprompt: An open-source framework for prompt-learning. *arXiv* **2021**, arXiv:2111.01998.

31. Khattak, M.U.; Rasheed, H.; Maaz, M.; Khan, S.; Khan, F.S. Maple: Multi-modal prompt learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 19113–19122.
32. Zhou, K.; Yang, J.; Loy, C.C.; Liu, Z. Conditional prompt learning for vision-language models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 16816–16825.
33. Zhou, K.; Yang, J.; Loy, C.C.; Liu, Z. Learning to prompt for vision-language models. *Int. J. Comput. Vis.* **2022**, *130*, 2337–2348. [CrossRef]
34. Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Scialom, T.; et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv* **2023**, arXiv:2307.09288.
35. Song, C.; Lin, Y.; Guo, S.; Wan, H. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 9–11 February 2020; Volume 34, pp. 914–921.
36. Bai, L.; Yao, L.; Li, C.; Wang, X.; Wang, C. Adaptive graph convolutional recurrent network for traffic forecasting. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 17804–17815.
37. Wu, Z.; Pan, S.; Long, G.; Jiang, J.; Chang, X.; Zhang, C. Connecting the dots: Multivariate time series forecasting with graph neural networks. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual, 6–10 July 2020; pp. 753–763.
38. Cirstea, R.G.; Yang, B.; Guo, C.; Kieu, T.; Pan, S. Towards spatio-temporal aware traffic time series forecasting. In Proceedings of the 2022 IEEE 38th International Conference on Data Engineering (ICDE), Kuala Lumpur, Malaysia, 9–12 May 2022; IEEE: New York, NY, USA, 2022; pp. 2900–2913.
39. Yu, B.; Yin, H.; Zhu, Z. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv* **2017**, arXiv:1709.04875.
40. Zhao, L.; Song, Y.; Zhang, C.; Liu, Y.; Wang, P.; Lin, T.; Deng, M.; Li, H. T-GCN: A temporal graph convolutional network for traffic prediction. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 3848–3858. [CrossRef]
41. Wiles, O.; Gowal, S.; Stimberg, F.; Alvise-Rebuffi, S.; Ktena, I.; Dvijotham, K.; Cemgil, T. A fine-grained analysis on distribution shift. *arXiv* **2021**, arXiv:2110.11328.
42. Koh, P.W.; Sagawa, S.; Marklund, H.; Xie, S.M.; Zhang, M.; Balsubramani, A.; Hu, W.; Yasunaga, M.; Phillips, R.L.; Liang, P.; et al. Wilds: A benchmark of in-the-wild distribution shifts. In Proceedings of the International Conference on Machine Learning, Virtual, 18–24 July 2021; PMLR: New York, NY, USA, 2021; pp. 5637–5664.
43. Chen, M.; Goel, K.; Sohoni, N.S.; Poms, F.; Fatahalian, K.; Ré, C. Mandoline: Model evaluation under distribution shift. In Proceedings of the International Conference on Machine Learning, Virtual, 18–24 July 2021; PMLR: New York, NY, USA, 2021; pp. 1617–1629.
44. Gibbs, I.; Candes, E. Adaptive conformal inference under distribution shift. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 1660–1672.
45. Louati, A.; Louati, H.; Li, Z. Deep learning and case-based reasoning for predictive and adaptive traffic emergency management. *J. Supercomput.* **2021**, *77*, 4389–4418. [CrossRef]
46. Qin, Y.; Liu, M.; Hao, W. Energy-optimal car-following model for connected automated vehicles considering traffic flow stability. *Energy* **2024**, *298*, 131333. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.