mathematics

MDPI

Article

Traffic Prediction with Self-Supervised Learning: A Heterogeneity-Aware Model for Urban Traffic Flow Prediction Based on Self-Supervised Learning

Min Gao, Yingmei Wei *, Yuxiang Xie and Yitong Zhang

College of Systems Engineering, National University of Defense Technology, No. 137 Yanwachi Street, Changsha 410073, China; gaomin1103@nudt.edu.cn (M.G.); yxxie@nudt.edu.cn (Y.X.); zhangyitong23@nudt.edu.cn (Y.Z.)

* Correspondence: weiyingmei@nudt.edu.cn

Abstract: Accurate traffic prediction is pivotal when constructing intelligent cities to enhance urban mobility and to efficiently manage traffic flows. Traditional deep learning-based traffic prediction models primarily focus on capturing spatial and temporal dependencies, thus overlooking the existence of spatial and temporal heterogeneities. Heterogeneity is a crucial inherent characteristic of traffic data for the practical applications of traffic prediction. Spatial heterogeneities refer to the differences in traffic patterns across different regions, e.g., variations in traffic flow between office and commercial areas. Temporal heterogeneities refer to the changes in traffic patterns across different time steps, e.g., from morning to evening. Although existing models attempt to capture heterogeneities through predefined handcrafted features, multiple sets of parameters, and the fusion of spatial–temporal graphs, there are still some limitations. We propose a self-supervised learning-based traffic prediction framework called Traffic Prediction with Self-Supervised Learning (TPSSL) to address this issue. This framework leverages a spatial–temporal encoder for the prediction task and introduces adaptive data masking to enhance the robustness of the model against noise disturbances. Moreover, we introduce two auxiliary self-supervised learning paradigms to capture spatial heterogeneities and temporal heterogeneities, which also enrich the embeddings of the primary prediction task. We conduct experiments on four widely used traffic flow datasets, and the results demonstrate that TPSSL achieves state-of-the-art performance in traffic prediction tasks.

Keywords: deep learning; self-supervised learning; traffic prediction; heterogeneity modeling

MSC: 68T07

Citation: Gao, M.; Wei, Y.; Xie, Y.; Zhang, Y. Traffic Prediction with Self-Supervised Learning: A Heterogeneity-Aware Model for Urban Traffic Flow Prediction Based on Self-Supervised Learning. *Mathematics* **2024**, *12*, 1290. https://doi.org/10.3390/math12091290

Academic Editor: António Lopes

Received: 21 March 2024
Revised: 16 April 2024
Accepted: 19 April 2024
Published: 24 April 2024

1. Introduction

The importance of traffic prediction in urban planning and management is self-evident. Accurate traffic predictions enable effective traffic management, reduce congestion, and enhance the sustainability of urban transport systems. In particular, grid-based traffic flow prediction plays a crucial role in understanding and managing the dynamics of urban mobility. Dividing cities into manageable segments and predicting the traffic flow within each segment allows for a more detailed analysis of traffic patterns, thus facilitating targeted interventions and infrastructure planning.

Over the years, traffic prediction methodologies have evolved through three major stages: traditional statistical models, machine learning techniques, and deep learning methods. Each stage represents a leap forward regarding prediction accuracy and the ability to handle complex spatial–temporal data. Adopting deep learning in traffic prediction marks a significant milestone, thus offering unprecedented levels of accuracy by leveraging large datasets and capturing intricate patterns in traffic flow. This evolution underscores

the growing complexity of urban traffic systems and the increasing need for advanced predictive models to navigate the challenges of modern urban environments.

Traffic data inherently exhibit spatial and temporal heterogeneities, thus reflecting the variability of traffic flow across different regions and time steps. Figure 1a visualizes traffic flow data in Beijing, with (a) showing a heatmap of inflow at 9 a.m. on 1 March 2015 (Sunday). It vividly illustrates the differences in traffic volume between various areas, thus highlighting the concept of spatial heterogeneities. Spatial heterogeneities can arise from many factors, including road layouts, the positioning of transportation hubs (e.g., subway and train stations), the distribution of commercial and residential areas, and events specific to certain regions (e.g., sports events and concerts). Figure 1b shows the changes in inflow for two selected areas, A and B, on 1 March 2015 (Sunday), and 2 March 2015 (Monday). It reveals that traffic patterns in different areas change over time steps, e.g., from weekends to weekdays or from morning to midnight, thereby leading to temporal heterogeneities. These changes are crucial for understanding the dynamics of urban mobility and necessitate sophisticated prediction models capable of capturing such complexities. The spatial and temporal heterogeneities in traffic data not only challenge traditional prediction methods but also provide an opportunity to improve prediction accuracy by incorporating these heterogeneities into the modeling process. Therefore, acknowledging and modeling traffic data's spatial and temporal heterogeneities are crucial for developing accurate and reliable traffic prediction models.

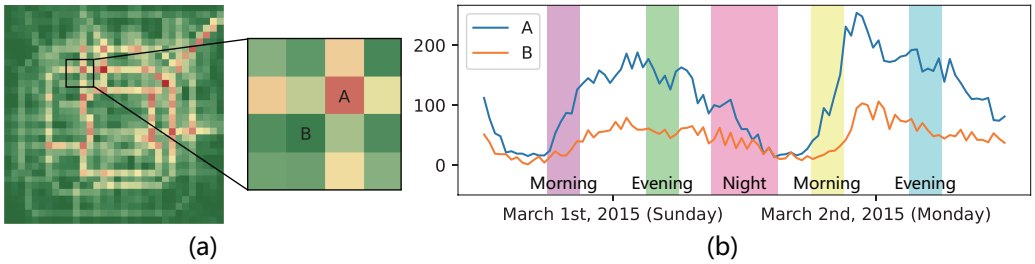

Figure 1. Visualization of spatial and temporal heterogeneities in traffic flow data in Beijing. (**a**) Heatmap of inflow at 9 a.m. on 1 March 2015 (Sunday). (**b**) Changes in inflow for two selected areas, A and B, on 1 March 2015 (Sunday), and 2 March 2015 (Monday).

We have reviewed many studies and found that current traffic prediction models need to be improved with respect to capturing spatial and temporal heterogeneities. Some models attempt to incorporate temporal features (e.g., periodicity and holidays) into the model [1,2] to capture temporal heterogeneities. Still, these are predefined features that may not fully capture the complexities of urban traffic patterns. Predefined spatial heterogeneities features are typically obtained by graph embedding based on an adjacency matrix [3], thus overlooking the complexities and diversities of regions. This reliance on handcrafted features limits the models' ability to adapt and generalize across various spatial regions and time scales. Models that overlook spatial heterogeneities tend to favor popular areas with heavy traffic flow [4], thereby leading to an incomplete understanding of urban traffic flow. Some studies attempt to capture spatial heterogeneities using different parameters in different regions. Still, this involves many parameters and may lead to suboptimal solutions in nonuniform urban environments [5,6]. Meta learning techniques have recently been introduced into traffic prediction to capture spatial–temporal heterogeneities, but the model's effectiveness depends on predefined spatial and temporal features [7,8]. Methods that adopt spatial–temporal graphs address temporal nonuniformity [9,10] but assume that temporal heterogeneities across the same period are static, which does not reflect reality.

Furthermore, attempts to actively capture spatial and temporal heterogeneities within models frequently encounter challenges in effectively balancing the granularity of representation with computational efficiency. Models struggle to balance oversimplified assumptions and enormous computational demands when dealing with complex traffic flows

that include large amounts of data [11]. This gap highlights the necessity for innovative approaches to inherently understand and the model spatial and temporal heterogeneities in traffic flow data.

To address the limitations above, we propose a novel self-supervised learning framework: **T**raffic **P**rediction with **S**elf-**S**upervised **L**earning (**TPSSL**). First, this framework leverages a spatial–temporal encoder to encode traffic data's spatial and temporal dependencies. Then, we introduce an adaptive data masking strategy to dynamically adjust the regions that need to be masked based on traffic data characteristics. Recognizing the complexities of capturing spatial and temporal heterogeneities in traffic data, we introduce two auxiliary self-supervised learning paradigms. The self-supervised learning paradigm based on soft clustering is responsible for exploring unique spatial patterns across different regions to learn spatial heterogeneities. It allows the model to identify and differentiate the unique traffic patterns across various urban areas without explicitly labeling, thereby inferring meaningful clusters of spatial regions from the natural distribution of traffic data. Moreover, we adopt a self-supervised learning paradigm based on positive and negative samples to incorporate temporal heterogeneities into the model's feature space. This paradigm is designed to maintain dedicated representations of traffic dynamics, thus adapting to the variability in traffic flow across different time steps in a day.

The main contributions of this paper are summarized as follows:

- We propose a novel self-supervised learning framework to model spatial and temporal heterogeneities in urban traffic flow data. We offer a detailed understanding and new insights for other spatial–temporal prediction tasks, e.g., weather forecasting.
- We introduce an adaptive data masking strategy that dynamically adjusts the regions that need to be masked based on traffic data characteristics, thereby enhancing the model's robustness against noise disturbances and ensuring that the learned representations are accurate and generalizable across different traffic conditions.
- Our framework incorporates two auxiliary self-supervised learning tasks, which aim to enrich the model's feature space, thus allowing for a deeper exploration of the underlying patterns of spatial and temporal heterogeneities to enhance the primary traffic prediction task.
- We conduct experiments on several real-world public datasets, thus demonstrating the superiority of TPSSL by achieving state-of-the-art results. We also conduct ablation studies to illustrate the importance of the adaptive data masking strategy and the two self-supervised learning paradigms. Furthermore, we explain the effectiveness of TPSSL through case studies.

2. Related Work

Traffic prediction has undergone several stages of development, from traditional statistical models to machine learning methods and then to deep learning techniques. The advancements in deep learning techniques have brought breakthroughs to traffic prediction, thus attracting many researchers' attention. Self-supervised learning, a highly effective unsupervised learning paradigm widely used in various fields, has been introduced into traffic prediction. This section reviews the following research: (1) deep learning in traffic prediction and (2) self-supervised learning in representation learning.

2.1. Deep Learning in Traffic Prediction

Accurate traffic prediction is crucial for urban planning and traffic management, and deep learning has emerged as a powerful tool in this domain. Deep learning techniques, e.g., convolutional neural networks (CNNs), recurrent neural networks (RNNs), graph neural networks (GNNs), and attention mechanisms, have been widely applied to traffic prediction tasks [12]. CNNs have been effectively applied to capture spatial dependencies in traffic data, thus offering significant improvements over traditional methods. Zhang et al. [1] introduced ST-ResNet, a deep spatial–temporal residual network that leverages CNNs to forecast citywide crowd flows, thus showcasing the capability of CNNs to model

complex spatial relationships within urban traffic systems. Traditional CNNs are unable to address sequence modeling problems, so Bai et al. [13] proposed the temporal convolutional network, which captures temporal dependencies in traffic data by introducing one-dimensional CNNs. RNNs and their variants, e.g., Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), have been widely adopted to model temporal dependencies in traffic data. Yao et al. [14] used LSTM to model the correlations between future traffic demand values and neighboring time steps. Li et al. [15] used GRU to model temporal dependencies and replaced matrix multiplication in GRU with diffusion convolution. GNNs have gained attention for their ability to model the graph-structured data commonly found in traffic networks. ChebNet is a spectral method, and Yu et al. [4] used Chebyshev first-order approximation graph convolution to obtain neighboring information for nodes. Due to the ability to model long-range dependencies without the sequential processing limitations of RNNs, attention mechanisms have been explored for traffic prediction. Inspired by attention-based models, Cai et al. [2] proposed a traffic transformer to parallelly predict traffic flow for multiple time steps in a nonautoregressive manner. Attention mechanisms can capture both temporal and spatial dependencies, so Zheng et al. [3] proposed multiple attention mechanisms to jointly act on traffic prediction tasks. Due to the advantages of CNNs, RNNs, GNNs, and attention mechanisms, most studies tend to combine them to improve the accuracy of traffic prediction [2,4,14–17]. In existing research, most models focus on capturing spatial and temporal dependencies in traffic data, thus often focusing on popular areas in the city and overlooking less popular areas.

Recent advancements in traffic flow prediction models have demonstrated significant improvements by integrating cutting-edge deep learning techniques. Naheliya et al. [18] introduced the MFOA-Bi-LSTM by utilizing a modified firefly optimization algorithm to enhance the predictive capabilities of Bi-LSTMs through optimal hyperparameter tuning. Similarly, Redhu et al. [19] employed a particle swarm optimization-enhanced Bi-LSTM model, thus showcasing the potential of swarm intelligence in refining neural network performance for traffic prediction. Zhang et al. [20] proposed a Multiattention Hybrid Convolution Spatial–Temporal Recurrent Network (MHSRN), which integrates multiattention mechanisms with hybrid convolutional layers to capture complex spatial–temporal patterns effectively. Moreover, Chen et al. [21] developed a Traffic Flow Matrix-Based Graph Neural Network (TFM-GCAM) that employs a novel graph convolution strategy enhanced with attention mechanisms to improve the accuracy of traffic flow prediction. He et al. [22] presented a 3D dilated dense neural network that leverages multiscale dilated convolutions to address the spatiotemporal variations in traffic data more dynamically. Lastly, Bao et al. [23] introduced the Spatial–Temporal Complex Graph Convolution Network (ST-CGCN), which uses a complex correlation matrix to model the intricate relationships between traffic nodes, thereby enhancing both the spatial and temporal feature extraction capabilities.

Recent research efforts have begun to explore how better to capture the spatial and temporal heterogeneities within traffic systems using deep learning methods. Bai et al. [5] introduced an adaptive module, i.e., a data-adaptive graph generation module, to automatically infer the interdependencies among different traffic series, thus avoiding predefined graph structures. Pan et al. [6] adopted a matrix factorization approach in neural networks to decompose region-specific parameters into learnable matrices, thereby modeling latent region functionality and inter-region correlations. Guo et al. [11] represented spatial heterogeneities features by assigning an additional embedding vector to each region and learning these vectors through model training. The above methods learn spatial heterogeneities by applying unique parameters to different areas. However, this strategy results in many parameters and may yield suboptimal results in nonuniformly distributed urban environments. Meta learning techniques have also been introduced into traffic prediction to capture spatial–temporal heterogeneities, but their effectiveness still depends on predefined external spatial and temporal attributes [7,8]. Li et al. [9] generated a temporal graph and fused it with a spatial graph to form a spatial–temporal fusion graph. Song et al. [10] captured spatial–temporal heterogeneities in traffic data by constructing a local spatial–

temporal graph. Although spatial–temporal graphs aim to capture heterogeneities, they often provide a relatively static representation. If heterogeneities in the traffic network change over time, these graphs may fail to capture dynamic heterogeneities. The above methods have made some progress in capturing spatial and temporal heterogeneities in traffic data, but there are still some limitations.

2.2. Self-Supervised Learning in Representation Learning

Self-supervised learning (SSL) is a technique used in representation learning [24], thereby allowing models to discover feature detection or classification representations in raw data automatically. Unlike supervised learning, which requires manually annotated labels, SSL uses inherent structures in the data to generate supervisory signals. This method enables models to learn rich data representations from any observed part of the input data by predicting any unobserved or hidden part. Self-supervised learning has been used in various fields, including Natural Language Processing (NLP) and Computer Vision (CV). In NLP, SSL has been used to learn word embeddings or language models from large, unannotated text corpora, such as BERT and GPT [25,26]. In CV, SSL techniques have been used to pretrain models on large image datasets, thus enabling them to recognize visual patterns and objects without relying on labeled datasets, such as SimCLR and MoCo [27,28].

Contrastive learning and generative models are the two prominent methods used in SSL [29]. Contrastive learning methods learn representations by contrasting positive and negative sample pairs, thus pulling similar samples closer in the representation space and pushing dissimilar samples further apart. On the other hand, generative models focus on learning to reconstruct or generate data, thereby capturing the data distribution and learning features.

However, the application of self-supervised learning in traffic prediction still needs to be improved. Researchers have explored using self-supervised learning in traffic prediction, and their work has shown promising results. Ji et al. [30] adopted a self-supervised learning paradigm based on temporal continuity to examine the context information of traffic data, thereby better understanding and predicting the dynamic changes in traffic flow. Another study by Ji et al. [31] proposed a contrastive learning-based traffic prediction framework and learned the representation of traffic data through auxiliary tasks to improve traffic prediction accuracy. Our approach differs from these studies because we spatially model traffic flow data as regular grids rather than as a graph. Consequently, our self-supervised learning tasks focus more on learning the spatial and temporal features of regular grid-based data.

3. Methodology

In this section, we first clarify the key concepts and problem definition of grid-based short-term traffic prediction tasks, then introduce the overall architecture of TPSSL that we propose, and finally describe the critical components of the framework in detail.

3.1. Problem Definition

In addressing the grid-based short-term traffic prediction task, it is essential to clearly define key concepts and the specific formulation of the problem.

Definition 1. *Spatial Region: A spatial region refers to a spatial area within a city designated for analysis. In grid-based traffic prediction models, the city is divided into numerous equally sized grids, each representing a spatial region. These regions are the basic units for collecting and analyzing traffic flow data within their boundaries.*

Definition 2. *Inflow/Outflow: Inflow denotes the quantity of traffic entering a spatial region within a specified time interval, which encompasses all forms of traffic movement, including vehicles, bicycles, or pedestrians. Conversely, outflow signifies the quantity of traffic exiting a spatial region within the same time interval.*

We have historical traffic flow data $X = [x_{t-T+1}, x_{t-T+2}, \ldots, x_t]$, where $x_t \in \mathbb{R}^{M \times N \times 2}$ represents the traffic flow matrix at time step t. M and N denote the city divided into M rows and N columns of grids. The value of 2 represents the number of channels, where channel 0 denotes inflow, and channel 1 denotes outflow. The objective of the short-term traffic flow prediction problem is to obtain the traffic flow matrix $y_{t+1} \in \mathbb{R}^{M \times N \times 2}$ at time step $t+1$. The problem can be formally described as

$$y_{t+1} = f(X) \tag{1}$$

$f(\cdot)$ represents the traffic prediction model that maps historical traffic flow data X to future traffic flow data y_{t+1} at the next time step.

3.2. Architecture

We propose a traffic prediction model called TPSSL. Its purpose is to improve the accuracy of traffic flow data prediction by capturing spatial and temporal heterogeneities through self-supervised learning. As seen in Figure 2, the overall architecture of TPSSL consists of four key modules: a spatial–temporal encoder, adaptive data masking, spatial heterogeneity modeling, and temporal heterogeneity modeling. The spatial–temporal encoder generates a similarity matrix and prediction embeddings while capturing the spatial–temporal dependencies in traffic flow data. Adaptive data masking enhances the model's robustness by dynamically selecting spatial regions to be masked. Spatial heterogeneity modeling and temporal heterogeneity modeling delve deeper into the complexity of traffic data, thus capturing spatial and temporal heterogeneities in traffic flow data and enriching the feature space of the model.

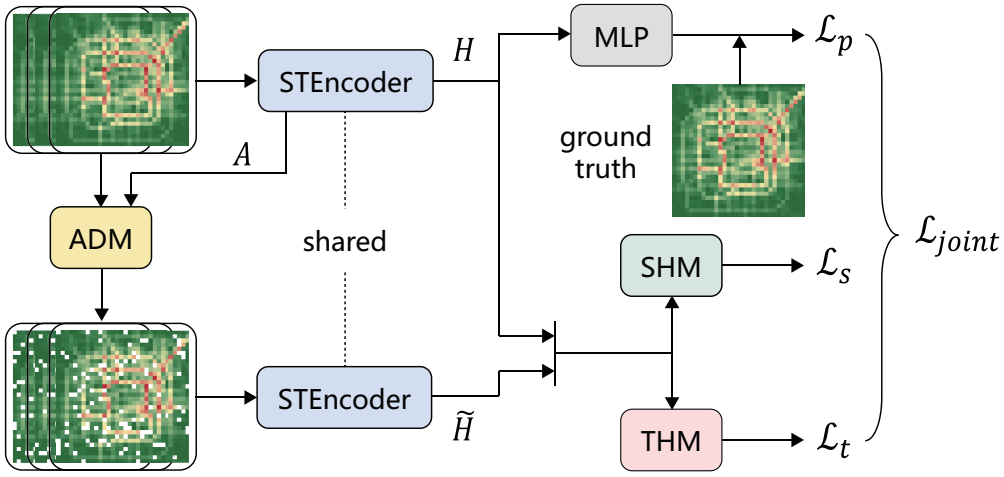

Figure 2. The architecture of TPSSL. There are four key modules: Spatial–Temporal Encoder (STEncoder), Adaptive Data Masking (ADM), Spatial Heterogeneity Modeling (SHM), Temporal Heterogeneity Modeling (THM).

3.3. Spatial–Temporal Encoder

The spatial–temporal encoder in our model is designed to effectively capture both spatial and temporal dependencies of traffic flow data, thus providing rich spatial–temporal embeddings for subsequent modules. The encoder is composed of several essential layers, each of which uniquely contributes to the overall ability of the model to process and interpret traffic flow data.

Initially, the traffic data undergoes processing through two 3D convolutional layers. The 3D convolutional layers handle data across spatial and temporal dimensions, thereby allowing interactions between neighboring regions and time steps to extract features that reflect traffic flow dynamics. The following formula can summarize this sequential processing:

$$X' = \mathrm{ReLU}(\mathrm{Conv3D}_2(\mathrm{ReLU}(\mathrm{Conv3D}_1(X)))) \tag{2}$$

where $X \in \mathbb{R}^{T\times M\times N\times 2}$ represents the input traffic flow data, $X' \in \mathbb{R}^{T\times M\times N\times D}$ denotes the embedding after processing by the convolutional layers, and D represents the embedding size.

Next, an essential aspect of the encoder is the computation of the similarity matrix $A \in \mathbb{R}^{T\times M\times N}$ derived from the embedding X'. This matrix is intended for use in adaptive data masking, thus facilitating the augmentation of the model's training data by emphasizing similarities between traffic patterns. The calculation of the similarity matrix is as follows:

$$A = \text{Softmax}(\text{AvgPool3D}(X')) \tag{3}$$

where AvgPool3D refers to the average pooling operation across the feature channels. Softmax is applied to normalize the values and emphasize the relative importance of different time steps in the traffic data.

Then, the core of the spatial–temporal encoder is the Convolutional LSTM (ConvLSTM) layer [32], which has been chosen for its proficiency in capturing spatial and temporal dependencies within the data. Unlike the standard LSTM, which processes temporal data, ConvLSTM extends its capability to spatial dimensions, thus making it particularly suitable for traffic prediction tasks where spatial relationships are crucial. The ConvLSTM layer effectively integrates spatial information with temporal dynamics, thus enhancing the model's predictive performance. Following processing by the ConvLSTM layer, we obtain a richer spatial–temporal embedding $H \in \mathbb{R}^{M\times N\times D}$, which is an important input for subsequent modules.

3.4. Adaptive Data Masking

The adaptive data masking module is pivotal with respec to enhancing our traffic prediction model's robustness and generalization capability. Unlike traditional random masking techniques, we design a targeted data masking strategy employing the similarity matrix A obtained from the spatial–temporal encoder. This strategy ensures that the augmentation focuses on the most informative parts of the traffic flow data, thereby challenging the model to learn to simulate natural and challenging traffic scenarios.

The similarity matrix A represents the normalized importance of each spatial region at each time step. We aim to mask a percentage of the data that is inversely proportional to its similarity score, meaning regions with lower similarity scores are more likely to be masked. This is achieved by calculating a masking probability distribution from A, where the probability of masking a given spatial region is higher if its corresponding similarity score is lower. Formally, the masking probability for each spatial region is determined as follows:

$$P_{t,i,j} = \frac{1 - A_{t,i,j}}{\sum_{t=1}^{T}\sum_{i=1}^{M}\sum_{j=1}^{N}(1 - A_{t,i,j})} \tag{4}$$

where $P_{t,i,j}$ represents the masking probability for spatial region $r_{i,j}$ at time step t, and $A_{t,i,j}$ denotes the corresponding element in the similarity matrix A.

The masking operation involves selecting regions to be masked based on P, and a predefined masking ratio determines the total number of masked regions. The inflow and outflow of the selected spatial regions are then set to zero, thus simulating the absence of traffic flow information in these regions. This approach challenges the model to make predictions without specific data but encourages it to leverage its understanding of spatial and temporal dependencies to fill in the missing information. The augmented data obtained through adaptive data masking are denoted as $\tilde{X}$. The embedding obtained after $\tilde{X}$ passes through the spatial–temporal encoder is denoted by $\tilde{H}$.

3.5. Spatial Heterogeneity Modeling

As illustrated in Figure 2, spatial heterogeneity modeling is a crucial component of our traffic prediction framework. We designed a self-supervised learning task based on soft clustering to capture the underlying spatial heterogeneities in traffic data through self-supervised signals, as shown in Figure 3. Specifically, we mapped the embeddings

of different spatial regions to prototypes corresponding to different urban functions (e.g., residential areas, office areas, transportation hubs). We obtained the embeddings of the original data and the augmented data through the spatial–temporal encoder, which are denoted as H and $\tilde{H}$, respectively. We will refer to H and $\tilde{H}$ as the original and augmented embeddings. The original embedding and the augmented embedding of the region $r_{i,j}$ are denoted as $h_{i,j}$ and $\tilde{h}_{i,j}$, respectively. The prototypes representing the K clusters are denoted as $\{c_1, \ldots, c_K\}$. The following formula achieves the clustering results of the augmented embedding:

$$\tilde{z}_{i,j,k} = c_k^\top \tilde{h}_{i,j} \tag{5}$$

where $\tilde{z}_{i,j,k}$ represents the similarity score between the augmented embedding $\tilde{h}_{i,j}$ of region $r_{i,j}$ and the prototype c_k. Thus, the clustering assignment of region $r_{i,j}$ can be represented as $\tilde{z}_{i,j} = (\tilde{z}_{i,j,1}, \ldots, \tilde{z}_{i,j,K})$. Similarly, $\hat{z}_{i,j,k}$ is the similarity score between the original embedding $h_{i,j}$ and the prototype c_k: $\hat{z}_{i,j,k} = c_k^\top h_{i,j}$. We designed the learning task to maximize the similarity of the original embedding $h_{i,j}$ and the augmented embedding $\tilde{h}_{i,j}$ in the clustering space. The following formula can express the optimization process:

$$\mathcal{L}(\tilde{z}_{i,j}, \hat{z}_{i,j}) = -\sum_{k=1}^{K} \log\left(\frac{\exp(\tilde{z}_{i,j,k}/\tau)}{\sum_{l=1}^{K}\exp(\tilde{z}_{i,j,l}/\tau)}\right) - \sum_{k=1}^{K} \log\left(\frac{\exp(\hat{z}_{i,j,k}/\tau)}{\sum_{l=1}^{K}\exp(\hat{z}_{i,j,l}/\tau)}\right) \tag{6}$$

where τ is the temperature parameter, which controls the sharpness of the distribution output by the Softmax function. The sum of the loss functions for all regions is used as the final loss of the model, i.e.,

$$\mathcal{L}_s = \sum_{i=1}^{M}\sum_{j=1}^{N} \mathcal{L}(\tilde{z}_{i,j}, \hat{z}_{i,j}) \tag{7}$$

By minimizing the crossentropy of the original embedding and the augmented embedding in the clustering space, these two types of embeddings are made as close as possible regarding clustering assignments.

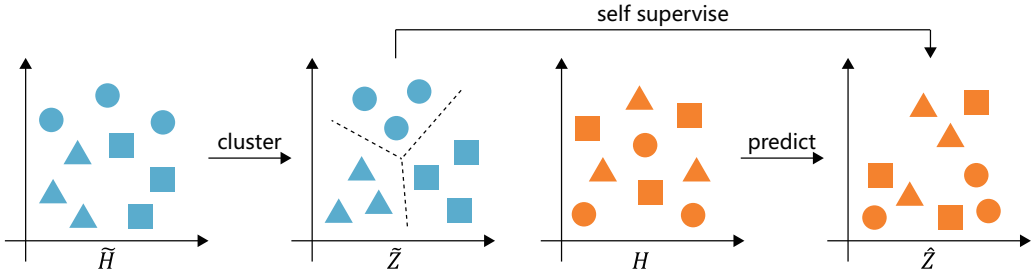

Figure 3. Spatial heterogeneity modeling in TPSSL. Different shapes of embeddings represent different prototypes. Blue embeddings are generated from the original data, and orange embeddings are generated from the augmented data. This module is implemented based on soft clustering, thus using the similarity of original and augmented embeddings in the clustering space to guide learning spatial heterogeneities.

In the above approach, we generated the clustering assignment matrices $\tilde{Z} \in \mathbb{R}^{M\times N\times K}$ and $\hat{Z} \in \mathbb{R}^{M\times N\times K}$ to serve as self-supervised signals for spatial heterogeneity modeling. We must address two issues to ensure that the regional features conform to the proper distribution of urban space. First, we need to ensure that the sum of the clustering assignment matrices for each region is 1. Second, we must avoid situations where all areas receive the same assignment. We introduced the Sinkhorn algorithm [33], which is a regularization-based optimization method to address these two issues. It was used to adjust the clustering assignment matrices to satisfy certain normalization conditions, i.e., the sum of the assignments for each spatial region over all clusters is 1, and the sum for each cluster over all spatial regions is also 1. By alternately normalizing over the spatial region and cluster dimensions, the Sinkhorn algorithm can achieve a balanced clustering

assignment strategy. Using Equation (7), we applied the Sinkhorn algorithm to $\tilde{Z}$ and $\hat{Z}$ and replaced the original assignment matrices with the results of the algorithm.

3.6. Temporal Heterogeneity Modeling

To inject temporal heterogeneities into TPSSL, we designed a self-supervised learning task based on contrastive learning, as shown in Figure 4. This task aims to identify and capture changes in traffic patterns at different time steps through contrastive learning, thereby enhancing the model's dynamic understanding of time. First, we fused the original embedding $h_{i,j}$ and the augmented embedding $\tilde{h}_{i,j}$ of region $r_{i,j}$ at time step t to obtain the region-level embedding $u_{t,i,j}$:

$$u_{t,i,j} = w_1 \odot h_{t,i,j} + w_2 \odot \tilde{h}_{t,i,j} \tag{8}$$

where w_1 and w_2 are learnable weights, and $\odot$ denotes elementwise multiplication. Then, we generated the city-level embedding s_t based on $u_{t,i,j}$. Specifically, we averaged $u_{t,i,j}$ across its spatial dimensions and applied a sigmoid activation function to obtain s_t:

$$s_t = \sigma\left(\frac{1}{M \times N}\sum_{i=1}^{M}\sum_{j=1}^{N} u_{t,i,j}\right) \tag{9}$$

Subsequently, we used the city-level embedding s_t as the summary information, the region-level embedding $u_{t,i,j}$ as the positive sample, and the region-level embedding $u_{t',i,j}$ at other time steps as the negative sample. We introduced a bilinear discriminator to evaluate the congruence of the summary information s_t with the positive and negative samples. The congruence score of the summary information s_t with the positive sample $h_{t,i,j}$ obtained through the discriminator can be calculated using the following formula:

$$g(h_{t,i,j}, s_t) = h_{t,i,j}^{\top} W s_t + b \tag{10}$$

where $W \in \mathbb{R}^{D \times D}$ is a learnable weight matrix, and b is a bias term. To optimize temporal heterogeneity modeling, we contrasted the congruence scores of the summary information s_t with the positive sample $h_{t,i,j}$ and the negative sample $h_{t',i,j}$.

$$\mathcal{L}(s_t, h_{t,i,j}, h_{t',i,j}) = -\Big(\log \sigma(g(h_{t,i,j}, s_t)) + \log(1 - \sigma(g(h_{t',i,j}, s_t)))\Big) \tag{11}$$

The sum of the loss functions for all regions is used as the final loss:

$$\mathcal{L}_t = \sum_{i=1}^{M}\sum_{j=1}^{N} \mathcal{L}(s_t, h_{t,i,j}, h_{t',i,j}) \tag{12}$$

This positive and negative sample contrastive learning mechanism ensures that the prediction results are consistent with the traffic pattern at a specific time step while distinguishing other traffic patterns at different time steps and learning temporal heterogeneities.

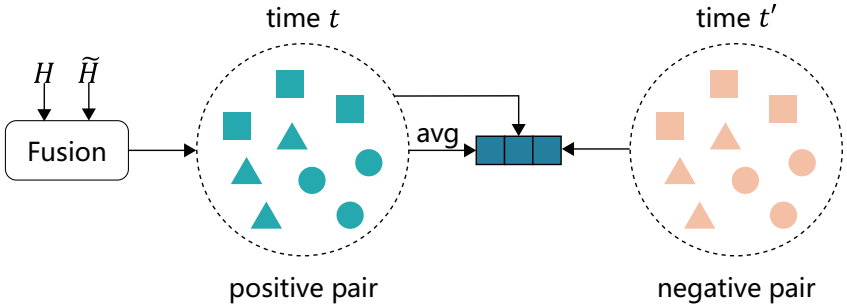

Figure 4. Temporal heterogeneity modeling in TPSSL. This module is implemented based on contrastive learning, thus capturing changes in traffic patterns at different time steps through the congruence of the summary information of spatial regions with positive and negative samples.

3.7. Model Training

In TPSSL, we used a Multilayer Perceptron (MLP) to predict traffic flow, which can be expressed by the following formula:

$$\hat{y}_{t+1,i,j} = \mathrm{MLP}(h_{t,i,j}) \tag{13}$$

where $\hat{y}_{t+1,i,j}$ represents the predicted traffic flow value for region $r_{i,j}$ at time step $t+1$. The prediction loss $\mathcal{L}_p$ is calculated using the mean absolute error:

$$\mathcal{L}_p = \sum_{i=1}^{M}\sum_{j=1}^{N} \lambda|\hat{y}^0_{t+1,i,j} - y^0_{t+1,i,j}| + (1-\lambda)|\hat{y}^1_{t+1,i,j} - y^1_{t+1,i,j}| \tag{14}$$

where λ is a hyperparameter used to balance the traffic flow prediction values of different channels, and $y_{t+1,i,j}$ represents the true traffic flow value for region $r_{i,j}$ at time step $t+1$. Finally, the overall loss function $\mathcal{L}$ of TPSSL is the weighted sum of the three loss functions:

$$\mathcal{L} = \alpha\mathcal{L}_s + \beta\mathcal{L}_t + \gamma\mathcal{L}_p \tag{15}$$

where α, β, and γ are weights. We adopted a dynamic weight adjustment mechanism to accommodate the varying scales and complexities of different tasks, i.e., the dynamic weight averaging (DWA) technique. Initially, the weights α, β, and γ were set to $[1, 1, 1]$, thus providing equal importance to each loss. The DWA technique recalibrates the weights based on the relative learning progress of each task, thus ensuring a balanced optimization among different modules.

The training process of TPSSL can be summarized as follows: First, the original traffic flow data X is input into the spatial–temporal encoder, thus obtaining the original embedding H and the similarity matrix A. Then, the adaptive data masking module utilizes the similarity matrix A to generate augmented data $\tilde{X}$. $\tilde{X}$ is input into the spatial–temporal encoder, thereby obtaining the augmented embedding $\tilde{H}$. Next, H and $\tilde{H}$ are fed into the spatial heterogeneity modeling, temporal heterogeneity modeling, and MLP to obtain the final loss function $\mathcal{L}$. Finally, we optimize the model's parameters using the backpropagation algorithm to minimize the loss function $\mathcal{L}$.

4. Experiment

In this section, we first introduce four datasets and evaluation metrics used in the experiments and then describe the baseline models and the details of the implementation of TPSSL. Finally, we evaluate the performance of TPSSL through comparative experiments, ablation studies, and case studies.

4.1. Data Description

We utilized four publicly available traffic flow datasets: BJTaxi [1], NYCBike1 [1], NYCBike2 [34], and NYCTaxi [34]. The NYCBike1 and NYCBike2 datasets are based on the

bike rental systems of New York City, while the BJTaxi and NYCTaxi datasets are based on the taxi systems of Beijing and New York City, respectively. A detailed overview of each dataset, including the number of grids, time intervals, start and end dates, and the number of bikes or taxis, is provided in Table 1. These datasets differ in geographical location, time span, and traffic volume, which enables our model to be comprehensively evaluated across various traffic conditions.

These datasets were constructed using a sliding window strategy to generate input–output pairs. The input data comprise traffic flow data for the four hours preceding the predicted time step, traffic flow data from the same time step on the previous three days, and the two hours before and after that time step. After the generation of input–output pairs, which preserve the continuous chronological order, the dataset was divided into training, validation, and testing sets with a ratio of 7:1:2. Specifically, the initial 70% of the sequentially ordered data was allocated for training, thus ensuring that the validation and testing sets representing the subsequent 10% and 20%, respectively reflect the original temporal order to maintain the inherent time series structure and prevent data leakage.

Table 1. Detialed information of datasets.

Dataset	# Regions	Time Interval	Start Date	End Date	# Bikes/Taxis
BJTaxi [1]	32 × 32	30 min	1 March 2015	30 June 2015	34k+
NYCBike1 [1]	16 × 8	1 h	1 April 2014	30 September 2014	6.8k+
NYCBike2 [34]	10 × 20	30 min	1 July 2016	29 August 2016	2.6m+
NYCTaxi [34]	10 × 20	30 min	1 January 2015	1 March 2015	22m+

Note: # Regions represents the number of spatial regions in the dataset. # Bikes/Taxis represents the number of bikes or taxis. The symbol + indicates the actual number is greater than the displayed value.

4.2. Evaluation Metrics

To evaluate the accuracy of TPSSL, we used two widely accepted metrics: Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE). Both metrics are essential to assess the performance of traffic flow predictions, with lower values indicating better predictive performance. The MAE measures the average magnitude of prediction errors and is calculated as follows:

$$\mathrm{MAE} = \frac{1}{M}\frac{1}{N}\sum_{i=1}^{M}\sum_{j=1}^{N}\left|y_{i,j} - \hat{y}_{i,j}\right| \tag{16}$$

where $y_{i,j}$ and $\hat{y}_{i,j}$ represent the true and predicted values, respectively. The MAPE provides a percentage measure of predictive accuracy, which is particularly useful for understanding the magnitude of prediction errors relative to the true values. It is defined as

$$\mathrm{MAPE} = \frac{100\%}{M \times N}\sum_{i=1}^{M}\sum_{j=1}^{N}\left|\frac{y_{i,j} - \hat{y}_{i,j}}{y_{i,j}}\right| \tag{17}$$

where $y_{i,j}$ and $\hat{y}_{i,j}$ have the same meaning as in Equation (16).

4.3. Baselines

To evaluate the performance of TPSSL, we compared it against a series of baseline models encompassing traditional time series models, machine learning algorithms, and deep learning models. These models have been categorized as follows:

Traditional Models:

- Autoregressive Integrated Moving Average (ARIMA) [35]: It is a classic model in time series forecasting that combines autoregressive, differencing, and moving average components to model various time series data.
- Support Vector Regression (SVR) [36]: It provides a powerful mechanism for capturing linear relationships in data by using support vector machines for regression tasks.

Dependency-Aware Traffic Prediction Models:

- Spatiotemporal Residual Network (ST-ResNet) [1]: It captures the spatial and temporal dependencies of traffic data through residual connections and convolutional operations.
- Spatiotemporal Graph Convolutional Network (STGCN) [4]: It integrates graph convolutional networks with temporal convolutional networks, thus simultaneously modeling spatial and temporal dependencies in traffic data.
- Graph Multiattention Network (GMAN) [3]: It introduces multiple attention mechanisms, thus allowing the model to dynamically adjust its focus on different regions and time steps of the traffic network.

Heterogeneity-Aware Traffic Prediction Models:

- Adaptive Graph Convolutional Recurrent Network (AGCRN) [5]: It combines node-adaptive parameter learning and data-adaptive graph generation modules to capture fine-grained spatial and temporal correlations without predefined graphs automatically.
- Spatial–Temporal Synchronous Graph Convolutional Network (STSGCN) [10]: It captures complex local spatial–temporal correlations through a synchronous modeling mechanism and the heterogeneities of local spatial–temporal graphs through multiple modules at different time periods.
- Spatial–Temporal Fusion Graph Neural Network (STFGNN) [9]: It generates a time graph and fuses it with the spatial graph to parallelly process data from different periods, thus effectively learning hidden spatial–temporal dependencies.

These baseline models provide a wide range of approaches to traffic flow prediction, from traditional methods to state-of-the-art models that integrate complex spatial and temporal dependencies and heterogeneities. The heterogeneity-aware traffic prediction models capture the complexity and diversity of traffic data by assigning different parameters to different regions and time steps, which makes them particularly useful for traffic prediction tasks.

4.4. Implementation Details

The TPSSL model was built using the PyTorch framework, and we carried out all experiments on a single GeForce RTX 4090 GPU. The model has an embedding size of 64, and all convolution operations adopt a kernel size of three, which balances model complexity and computational efficiency. We used an adaptive data masking strategy with a masking rate 0.1 to introduce variations into the training data without significant information loss. For efficient convergence to the optimal solution, the training process leverages the adaptive learning rate capabilities of the Adam optimizer. Some hyperparameters were set: the learning rate was 0.001, the weight decay was 0, the batch size was 32, and the number of training epochs was 100. We used an early stopping strategy, which terminates the training process early if the loss value on the validation set does not improve for 15 consecutive epochs.

4.5. Results

In this study, we evaluated the performance of TPSSL on four widely used public traffic flow datasets: BJTaxi, NYCBike1, NYCBike2, and NYCTaxi. We compared TPSSL against a diverse set of baseline models, including traditional models such as ARIMA and SVR; dependency-aware traffic prediction models such as ST-ResNet, STGCN, and GMAN; and heterogeneity-aware traffic prediction models such as AGCRN, STSGCN,

and STFGNN. Additionally, we included the backbone network ConvLSTM of the spatial–temporal encoder as a baseline to demonstrate the effectiveness of our self-supervised learning paradigms. To ensure fairness, we trained ConvLSTM and TPSSL with five different seeds, just like the baseline models whose results come from Ji et al. [31].

Our results show that TPSSL outperformed all other models on all datasets, whether from the perspective of MAE or MAPE. Bolded numbers represent the best results, and underlined numbers represent the second-best results. This success was mainly due to our choice of an appropriate backbone model for the spatial–temporal encoder. ConvLSTM also perforeds well when making spatial–temporal predictions of traffic data alone, as seen from the underlined data in Tables 2 and 3. However, the two self-supervised learning tasks introduced in TPSSL further improved the predictive performance of ConvLSTM.

Moreover, we observed some interesting phenomena from Tables 2 and 3. Deep learning-based traffic prediction models were found to be far superior to traditional time series and machine learning methods regarding prediction accuracy. Additionally, there was no strict distinction between dependency-aware and heterogeneity-aware models regarding their predictive performance. They exhibited different strengths on different datasets. On the BJTaxi dataset, the predictive performance of heterogeneity-aware models was worse than that of the dependency-aware models. We believe heterogeneity-aware models introduce additional parameter space, thus affecting the model's judgment of dependencies while attempting to capture heterogeneities.

Table 2. Predictive performance of each model on inflow for the four datasets.

Dataset	**BJTaxi**		**NYCBike1**		**NYCBike2**		**NYCTaxi**	
Metric	**MAE**	**MAPE**	**MAE**	**MAPE**	**MAE**	**MAPE**	**MAE**	**MAPE**
ARIMA	21.48	23.12	10.66	33.05	8.91	28.86	20.86	21.49
SVR	52.77	65.51	7.27	25.39	12.82	46.52	52.16	65.10
ST-ResNet	12.12 ± 0.11	15.50 ± 0.26	5.53 ± 0.06	25.46 ± 0.20	5.63 ± 0.14	32.17 ± 0.85	13.48 ± 0.14	24.83 ± 0.55
STGCN	12.34 ± 0.09	16.66 ± 0.21	5.33 ± 0.02	26.92 ± 0.08	5.21 ± 0.02	27.73 ± 0.16	13.12 ± 0.04	21.01 ± 0.18
GMAN	13.13 ± 0.43	18.67 ± 0.99	6.77 ± 3.42	31.72 ± 12.29	5.24 ± 0.13	27.38 ± 1.13	15.09 ± 0.61	22.73 ± 1.20
AGCRN	12.30 ± 0.06	15.61 ± 0.15	5.17 ± 0.03	25.59 ± 0.22	5.18 ± 0.03	27.14 ± 0.14	12.13 ± 0.11	18.78 ± 0.04
STSGCN	12.72 ± 0.03	17.22 ± 0.17	5.81 ± 0.04	26.51 ± 0.32	5.25 ± 0.03	29.26 ± 0.13	13.69 ± 0.11	22.91 ± 0.44
STFGNN	13.83 ± 0.04	19.29 ± 0.07	6.53 ± 0.10	32.14 ± 0.23	5.80 ± 0.10	30.73 ± 0.49	16.25 ± 0.38	24.01 ± 0.30
ConvLSTM	11.70 ± 0.11	16.05 ± 0.49	5.15 ± 0.04	24.80 ± 0.35	5.05 ± 0.01	22.61 ± 0.07	12.05 ± 0.12	17.69 ± 0.38
TPSSL	**11.28 ± 0.02**	**15.07 ± 0.15**	**4.96 ± 0.02**	**23.38 ± 0.12**	**5.00 ± 0.02**	**22.15 ± 0.12**	**11.85 ± 0.06**	**16.39 ± 0.26**

Bolded numbers represent the best results, and underlined numbers represent the second-best results.

Table 3. Predictive performance of each model on outflow for the four datasets.

Dataset	**BJTaxi**		**NYCBike1**		**NYCBike2**		**NYCTaxi**	
Metric	**MAE**	**MAPE**	**MAE**	**MAPE**	**MAE**	**MAPE**	**MAE**	**MAPE**
ARIMA	21.60	20.67	11.33	35.03	8.70	28.22	16.80	21.23
SVR	52.74	65.51	7.98	27.42	11.48	41.91	41.71	64.06
ST-ResNet	12.16 ± 0.12	15.57 ± 0.26	5.74 ± 0.07	26.36 ± 0.50	5.26 ± 0.08	30.48 ± 0.86	10.78 ± 0.25	24.42 ± 0.52
STGCN	12.41 ± 0.08	16.76 ± 0.22	5.59 ± 0.03	27.69 ± 0.14	4.92 ± 0.02	26.83 ± 0.21	10.35 ± 0.03	20.78 ± 0.16
GMAN	13.20 ± 0.43	18.84 ± 1.04	7.17 ± 3.61	34.74 ± 17.04	4.97 ± 0.14	26.75 ± 1.14	12.06 ± 0.39	21.97 ± 0.86
AGCRN	12.38 ± 0.06	15.75 ± 0.15	5.47 ± 0.03	26.63 ± 0.30	4.79 ± 0.04	26.17 ± 0.22	9.87 ± 0.04	18.41 ± 0.21
STSGCN	12.79 ± 0.03	17.35 ± 0.17	6.10 ± 0.04	27.56 ± 0.39	4.94 ± 0.05	28.02 ± 0.23	10.75 ± 0.17	22.37 ± 0.16
STFGNN	13.89 ± 0.04	19.41 ± 0.07	6.79 ± 0.08	32.88 ± 0.19	5.51 ± 0.11	29.98 ± 0.46	12.47 ± 0.25	23.28 ± 0.47
ConvLSTM	11.78 ± 0.10	16.15 ± 0.47	5.45 ± 0.02	25.46 ± 0.31	4.72 ± 0.03	21.37 ± 0.25	9.84 ± 0.15	18.27 ± 0.42
TPSSL	**11.38 ± 0.03**	**15.21 ± 0.17**	**5.27 ± 0.02**	**24.26 ± 0.08**	**4.65 ± 0.02**	**21.14 ± 0.12**	**9.65 ± 0.14**	**16.77 ± 0.14**

Bolded numbers represent the best results, and underlined numbers represent the second-best results.

In contrast, the proposed TPSSL framework uses independent modules to capture dependencies and heterogeneities without affecting each other. This indicates that the self-supervised learning paradigms in TPSSL are very effective in traffic flow prediction tasks. It also suggests that incorporating self-supervised learning into traffic prediction models could be a promising direction for future research.

In a broader comparison across baseline models, TPSSL outshined traditional models like ARIMA and SVR, which, while robust in simpler scenarios, struggled with the complex spatial and temporal dynamics that are typical of urban traffic data. Such observations underscore the limitations of models that fail to integrate advanced spatial–temporal mechanisms.

Among the deep learning approaches, TPSSL showed clear advantages over models such as ST-ResNet, STGCN, GMAN, AGCRN, STSGCN, and STFGNN. Unlike these models, which may excel in spatial or temporal settings but not uniformly across both, TPSSL's architecture allows it to adeptly manage and synthesize these dimensions. The effectiveness of TPSSL was particularly notable in environments with intricate spatial–temporal interactions, where it maintained high accuracy and robustness, thus suggesting a superior ability to generalize across varied traffic conditions.

Each model brings certain strengths to traffic prediction: ST-ResNet and STGCN are praised for their spatial and temporal resolution; GMAN is known for its attention mechanisms that finely tune its focus across the network; AGCRN adapts well to dynamic graph structures; STSGCN synchronizes spatial–temporal elements effectively; and STFGNN explores novel graph fusion techniques for enhanced prediction. Unlike STGCN, GMAN, AGCRN, and STFGNN, which utilize complex graph-based approaches, ST-ResNet and TPSSL employ grid-based data structures. TPSSL differentiates itself by integrating adaptive data masking and heterogeneity-aware modules that optimize spatial and temporal dependencies within this grid framework. The integration of these features reduces the computational demands compared to graph-based models. It improves prediction accuracy, thereby enabling TPSSL to consistently excel in head-to-head comparisons on inflow and outflow predictions across all listed datasets.

The distinct modular approach of TPSSL, which independently but cohesively handles both spatial and temporal data variances, sets it apart from other models. This dual capability positions it as a benchmark model in traffic flow prediction and a highly adaptive framework suitable for the evolving demands of urban traffic management and planning.

4.6. Ablation Study

To analyze the impact of each submodule on the performance of TPSSL, we conducted ablation studies. We proposed three variants for the ablation study, i.e., TPSSL-SHM, TPSSL-THM, and TPSSL-RM. TPSSL-SHM disables the temporal heterogeneity modeling module in TPSSL, while TPSSL-THM disables the spatial heterogeneity modeling module in TPSSL. TPSSL-RM uses a random data masking strategy to replace the adaptive one.

Figure 5 shows the results of the ablation study. The results indicate that each submodule plays a significant role in the model's performance. Specifically, the TPSSL-SHM variant, which lacks temporal heterogeneity modeling, tended to perform worse than the full TPSSL model, with increases in both the MAE and MAPE across all datasets. Thus was particularly evident in the outflow predictions for NYCTaxi, thus underscoring the importance of temporal heterogeneity modeling in traffic prediction tasks. When spatial heterogeneity modeling was removed from TPSSL, there was a decline in performance. This effect was observed across all datasets for inflow and outflow predictions, which underscores the significance of spatial heterogeneity modeling in understanding the complex patterns of urban traffic. The TPSSL-RM variant, which employs a random masking strategy, showed an inferior performance compared to the adaptive strategy used in TPSSL. This was consistent across all datasets, thus reinforcing the value of the adaptive data masking strategy in improving prediction accuracy.

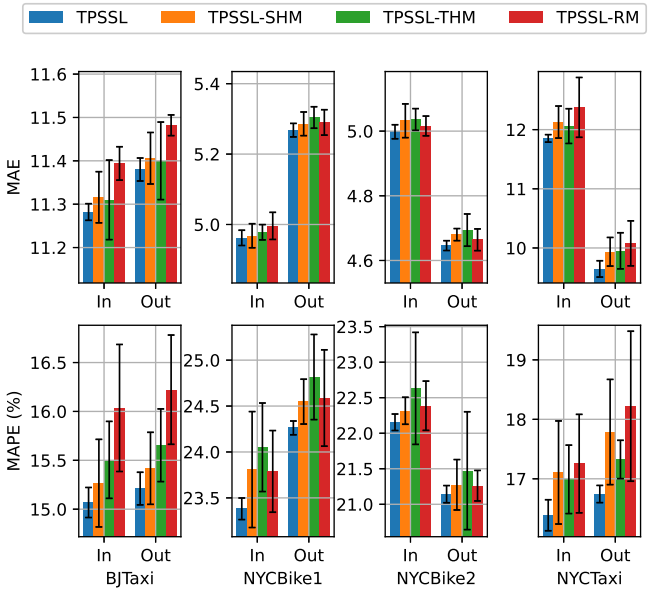

Figure 5. Ablation Study of TPSSL. We compared TPSSL with its three variants: TPSSL-SHM, TPSSL-THM, and TPSSL-RM. The results demonstrate that each submodule plays a significant role in the model's performance.

Despite the performance improvements of TPSSL over its variants, the error rates may still appear relatively high. This can be attributed to the inherent complexity and variability of urban traffic data across different datasets and channels (inflow/outflow). The datasets used in our studies represent a range of urban environments and traffic conditions that can influence the predictability and, hence, the resulting error metrics. Furthermore, the auxiliary self-supervised tasks of spatial and temporal heterogeneity modeling and the data augmentation strategy random masking are designed to enhance the primary prediction task's generalization and robustness. Still, they do not independently determine the model's overall predictive accuracy. In practice, the primary predictive performance of TPSSL during testing is derived from its core component, the spatial–temporal encoder, without the involvement of the auxiliary tasks.

4.7. Case Study

To further validate the performance of TPSSL, we conducted case studies on the BJTaxi dataset. The BJTaxi dataset, comprising detailed geotagged taxi trajectories within Beijing, provides a pertinent example due to its extensive coverage of densely populated urban areas and less congested suburban zones. This diversity makes it an exemplary case for testing the spatial–temporal modeling prowess of TPSSL.

Figure 6a shows the grid segmentation of the BJTaxi dataset, with the underlying map taken from Google Maps. With the t-SNE algorithm's help, the two models' hidden embeddings were projected into the 2D space. As shown in Figure 6b,c, we used the k-means clustering algorithm [37] to cluster the 2D embeddings. Furthermore, we visualized the clustering results in the grid space, as shown in Figure 6d,e.

We can see from Figure 6b,c that the hidden embeddings of TPSSL are more compact in space. At the same time, we can see from Figure 6d,e that TPSSL could accurately identify different types of areas, e.g., the traffic hub area marked in red and the suburbs marked in brown and green. Not all green and brown grids denote suburban areas; some represent central residential districts with lower taxi flow, like the Hutongs in Beijing. The lower taxi flow in the Hutongs can be attributed to their narrow alleyway configurations, which restrict vehicle access and discourage heavy traffic. This precision in classification demonstrates TPSSL's superior capability in discerning complex urban traffic structures compared to ConvLSTM.

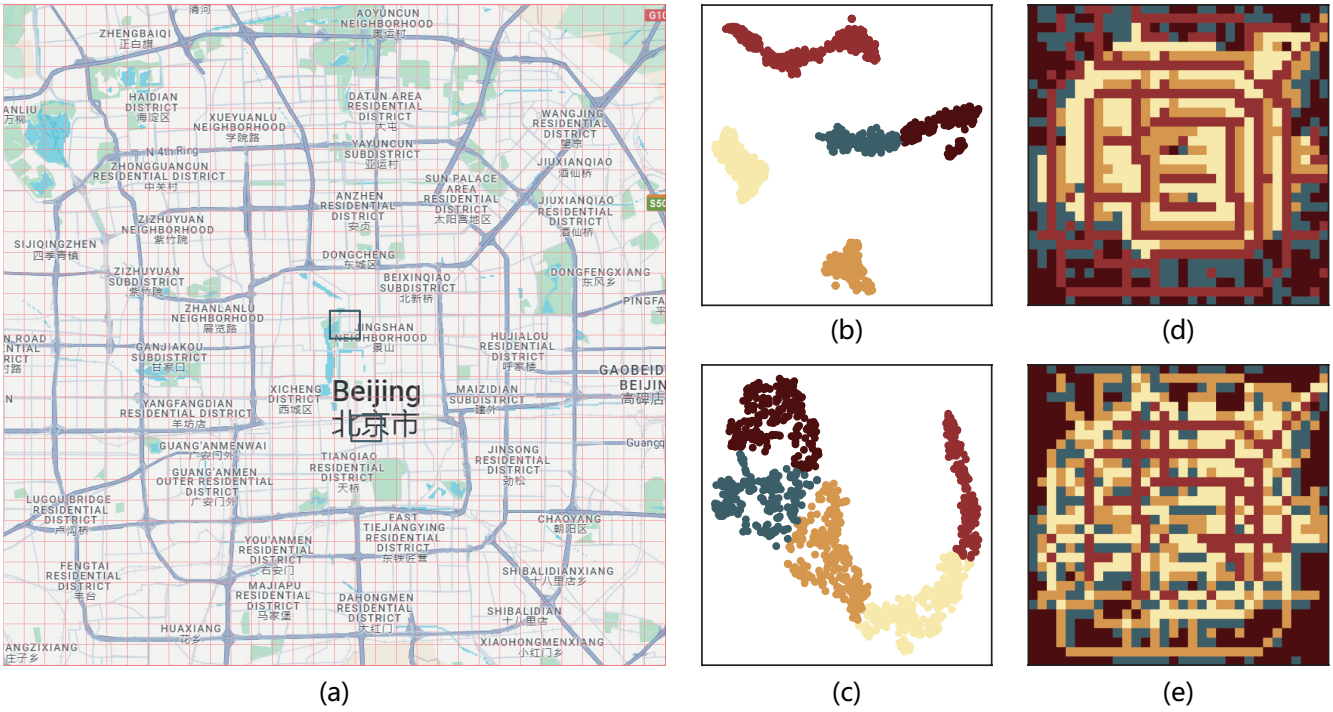

Figure 6. Visualization of the case studies of TPSSL and ConvLSTM. (**a**) is the grid segmentation of the BJTaxi dataset. (**b**,**c**) are the t-SNE projections of the hidden embeddings of TPSSL and ConvLSTM in the 2D space, respectively. (**d**,**e**) are the reconstructed visualizations of (**b**,**c**) in the grid space, respectively.

All these insights confirm that TPSSL excels at capturing the spatial heterogeneities inherent in urban traffic more effectively than ConvLSTM. The expanded case study validates TPSSL's enhanced performance and underscores its potential applicability in real-world urban planning and traffic management scenarios.

5. Conclusions

In this paper, we proposed a new self-supervised learning framework to improve the performance of traffic prediction models. TPSSL uses a spatial–temporal encoder and two self-supervised learning tasks to capture the dependencies and heterogeneities of traffic data, respectively. The generation of augmented data based on the adaptive data masking strategy can enhance the robustness and generalization of the model while providing more information for subsequent self-supervised tasks. The self-supervised paradigm based on soft clustering and positive–negative sample pairs can capture traffic data's spatial and temporal heterogeneities separately without negatively affecting the model's predictive performance. We conducted experiments on four public datasets, and the results show that TPSSL achieved the best predictive performance on all datasets. We also conducted ablation and case studies, thus verifying the accuracy and effectiveness of TPSSL and providing further explanations for the model's outstanding performance.

In the future, we will explore incorporating self-supervised learning techniques into other traffic prediction models to improve the predictive accuracy further. Additionally, we aim to investigate the application of TPSSL to real-time traffic data from area traffic control sensors, such as induction loops. This will enable us to leverage current data for learning and prediction, thereby enhancing model validation with actual traffic conditions observed over extended periods. At the same time, we will also study how to apply TPSSL to spatial–temporal data prediction tasks in other fields.

Author Contributions: Conceptualization, M.G. and Y.W.; methodology, M.G.; software, M.G. and Y.Z.; validation, M.G.; formal analysis, M.G.; investigation, M.G.; resources, Y.W. and Y.Z.; data curation, M.G. and Y.Z.; writing—original draft preparation, M.G.; writing—review and editing, M.G., Y.W., and Y.X.; visualization, M.G. and Y.Z.; supervision, Y.W. and Y.X.; project administration, Y.W. and Y.X.; funding acquisition, Y.W. and Y.X. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China (NSFC) under grant number 61873274 and the Hunan Provincial Natural Science Foundation Project (No. 2023JJ30082).

Data Availability Statement: The data supporting the findings of this study are publicly available and sourced from the ST-SSL Dataset https://github.com/Echo-Ji/ST-SSL_Dataset on GitHub (accessed on 1 March 2024).

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

References

1. Zhang, J.; Zheng, Y.; Qi, D. Deep spatio-temporal residual networks for citywide crowd flows prediction. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; Volume 31.
2. Cai, L.; Janowicz, K.; Mai, G.; Yan, B.; Zhu, R. Traffic transformer: Capturing the continuity and periodicity of time series for traffic forecasting. *Trans. GIS* **2020**, *24*, 736–755. [CrossRef]
3. Zheng, C.; Fan, X.; Wang, C.; Qi, J. Gman: A graph multi-attention network for traffic prediction. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 1234–1241.
4. Yu, B.; Yin, H.; Zhu, Z. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv* **2017**, arXiv:1709.04875.
5. 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.
6. Pan, Z.; Wang, Z.; Wang, W.; Yu, Y.; Zhang, J.; Zheng, Y. Matrix factorization for spatio-temporal neural networks with applications to urban flow prediction. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 2683–2691.
7. Pan, Z.; Liang, Y.; Wang, W.; Yu, Y.; Zheng, Y.; Zhang, J. Urban traffic prediction from spatio-temporal data using deep meta learning. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 1720–1730.
8. Ye, X.; Fang, S.; Sun, F.; Zhang, C.; Xiang, S. Meta graph transformer: A novel framework for spatial–temporal traffic prediction. *Neurocomputing* **2022**, *491*, 544–563. [CrossRef]
9. Li, M.; Zhu, Z. Spatial-temporal fusion graph neural networks for traffic flow forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021; Volume 35, pp. 4189–4196.
10. 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, 7–12 February 2020; Volume 34, pp. 914–921.
11. Guo, S.; Lin, Y.; Wan, H.; Li, X.; Cong, G. Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting. *IEEE Trans. Knowl. Data Eng.* **2021**, *34*, 5415–5428. [CrossRef]
12. Yin, X.; Wu, G.; Wei, J.; Shen, Y.; Qi, H.; Yin, B. Deep learning on traffic prediction: Methods, analysis, and future directions. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 4927–4943. [CrossRef]
13. Bai, S.; Kolter, J.Z.; Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv* **2018**, arXiv:1803.01271.
14. Yao, H.; Wu, F.; Ke, J.; Tang, X.; Jia, Y.; Lu, S.; Gong, P.; Ye, J.; Li, Z. Deep multi-view spatial-temporal network for taxi demand prediction. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
15. Li, Y.; Yu, R.; Shahabi, C.; Liu, Y. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv* **2017**, arXiv:1707.01926.
16. Wang, C.; Tian, R.; Hu, J.; Ma, Z. A trend graph attention network for traffic prediction. *Inf. Sci.* **2023**, *623*, 275–292. [CrossRef]
17. Li, F.; Feng, J.; Yan, H.; Jin, G.; Yang, F.; Sun, F.; Jin, D.; Li, Y. Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution. *ACM Trans. Knowl. Discov. Data* **2023**, *17*, 1–21. [CrossRef]
18. Naheliya, B.; Redhu, P.; Kumar, K. MFOA-Bi-LSTM: An optimized bidirectional long short-term memory model for short-term traffic flow prediction. *Phys. A Stat. Mech. Its Appl.* **2024**, *634*, 129448. [CrossRef]
19. Bharti; Redhu, P.; Kumar, K. Short-term traffic flow prediction based on optimized deep learning neural network: PSO-Bi-LSTM. *Phys. A Stat. Mech. Its Appl.* **2023**, *625*, 129001.

20. Zhang, X.; Wen, S.; Yan, L.; Feng, J.; Xia, Y. A hybrid-convolution spatial–temporal recurrent network for traffic flow prediction. *Comput. J.* **2024**, *67*, 236–252. [CrossRef]
21. Chen, J.; Zheng, L.; Hu, Y.; Wang, W.; Zhang, H.; Hu, X. Traffic flow matrix-based graph neural network with attention mechanism for traffic flow prediction. *Inf. Fusion* **2024**, *104*, 102146. [CrossRef]
22. He, R.; Zhang, C.; Xiao, Y.; Lu, X.; Zhang, S.; Liu, Y. Deep spatio-temporal 3D dilated dense neural network for traffic flow prediction. *Expert Syst. Appl.* **2024**, *237*, 121394. [CrossRef]
23. Bao, Y.; Huang, J.; Shen, Q.; Cao, Y.; Ding, W.; Shi, Z.; Shi, Q. Spatial–temporal complex graph convolution network for traffic flow prediction. *Eng. Appl. Artif. Intell.* **2023**, *121*, 106044. [CrossRef]
24. Ericsson, L.; Gouk, H.; Loy, C.C.; Hospedales, T.M. Self-supervised representation learning: Introduction, advances, and challenges. *IEEE Signal Process. Mag.* **2022**, *39*, 42–62. [CrossRef]
25. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
26. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving Language Understanding by Generative Pre-Training. 2018. Available online: https://paperswithcode.com/paper/improving-language-understanding-by (accessed on 10 March 2024)
27. Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G. A simple framework for contrastive learning of visual representations. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 13–18 July 2020; pp. 1597–1607.
28. He, K.; Fan, H.; Wu, Y.; Xie, S.; Girshick, R. Momentum contrast for unsupervised visual representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 9729–9738.
29. Liu, X.; Zhang, F.; Hou, Z.; Mian, L.; Wang, Z.; Zhang, J.; Tang, J. Self-supervised learning: Generative or contrastive. *IEEE Trans. Knowl. Data Eng.* **2021**, *35*, 857–876. [CrossRef]
30. Ji, J.; Yu, F.; Lei, M. Self-Supervised Spatiotemporal Graph Neural Networks With Self-Distillation for Traffic Prediction. *IEEE Trans. Intell. Transp. Syst.* **2022**, *24*, 1580–1593. [CrossRef]
31. Ji, J.; Wang, J.; Huang, C.; Wu, J.; Xu, B.; Wu, Z.; Zhang, J.; Zheng, Y. Spatio-temporal self-supervised learning for traffic flow prediction. In Proceedings of the AAAI Conference on Artificial Intelligence, Salt Lake City, UT, USA, 8–12 October 2023; Volume 37, pp. 4356–4364.
32. Shi, X.; Chen, Z.; Wang, H.; Yeung, D.Y.; Wong, W.K.; Woo, W.c. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 1–9.
33. Cuturi, M. Sinkhorn distances: Lightspeed computation of optimal transport. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 2292–2300.
34. Yao, H.; Tang, X.; Wei, H.; Zheng, G.; Li, Z. Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 5668–5675.
35. 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]
36. Castro-Neto, M.; Jeong, Y.S.; Jeong, M.K.; Han, L.D. Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions. *Expert Syst. Appl.* **2009**, *36*, 6164–6173. [CrossRef]
37. Krishna, K.; Murty, M.N. Genetic K-means algorithm. *IEEE TRansactions Syst. Man, Cybern. Part B (Cybern.)* **1999**, *29*, 433–439. [CrossRef]