Artificial Intelligence 328 (2024) 104065

Contents lists available at ScienceDirect

Artificial Intelligence

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

Dual-track spatio-temporal learning for urban flow prediction with adaptive normalization

Xiaoyu Li [a], Yongshun Gong [a,*], Wei Liu [b], Yilong Yin [a], Yu Zheng [c], Liqiang Nie [d]

[a] School of Software, Shandong University, 27 Shanda South Road, Jinan, 250100, Shandong, China
[b] School of Computer, University of Technology, Sydney, 15 Broadway, Sydney, 2007, NSW, Australia
[c] JD Intelligent Cities Research, Beijing, 100176, China
[d] School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Shenzhen, 518055, China

ARTICLE INFO

Keywords:
Urban flow prediction
Spatio-temporal learning
Spatio-temporal normalization
Contrastive learning
Regional and global correlations

ABSTRACT

Robust urban flow prediction is crucial for transportation planning and management in urban areas. Although recent advances in modeling spatio-temporal correlations have shown potential, most models fail to adequately consider the complex spatio-temporal semantic information present in real-world scenarios. We summarize the following three primary limitations in existing models: a) The majority of existing models project overall time periods into the same latent space, neglecting the diverse temporal semantics between different time intervals. b) Existing models tend to capture spatial dependencies from a locale perspective such as surroundings but do not pay attention to the global influence factors. c) Beyond the spatio-temporal properties, the dynamics and instability of the data sequences introduce perturbations to the prediction results, potentially leading to model degradation. To address these issues, we propose a dual-track spatial-temporal learning module named DualST for accurate urban flow inference. To more effectively differentiate semantic information in the time dimension, we assign the overall time scales into closeness and periodicity. The dual-track module, which includes temporal causality inference and temporal contextual inference, simultaneously exploits the dynamic evolutionary trends and periodic traffic patterns, respectively. The proposed DualST captures global spatial features in a self-supervised manner which not only enriches the spatial semantics but also avoids introducing additional prior knowledge. To eliminate the instability caused by dynamics, we first adopt spatio-temporal adaptive normalization to learn appropriate data sequence normalization. We evaluate the proposed DualST on two typical urban flow datasets. The experiment results show that our model not only exhibits a consistent superiority over various state-of-the-art baselines but also has remarkable generalization capability.

1. Introduction

Accurate urban flow prediction (UFP) is of great significance for the development of intelligent transportation systems, which aims to optimize transportation operations and improve mobility. Forecasting urban flow has great potential for application in intelligent traffic management as it enables effective traffic control and can help prevent accidents caused by sudden traffic flow spike [1,2].

* Corresponding author.
E-mail address: yongshun2512@hotmail.com (Y. Gong).

https://doi.org/10.1016/j.artint.2024.104065
Received 19 April 2023; Received in revised form 13 November 2023; Accepted 2 January 2024

Available online 15 January 2024

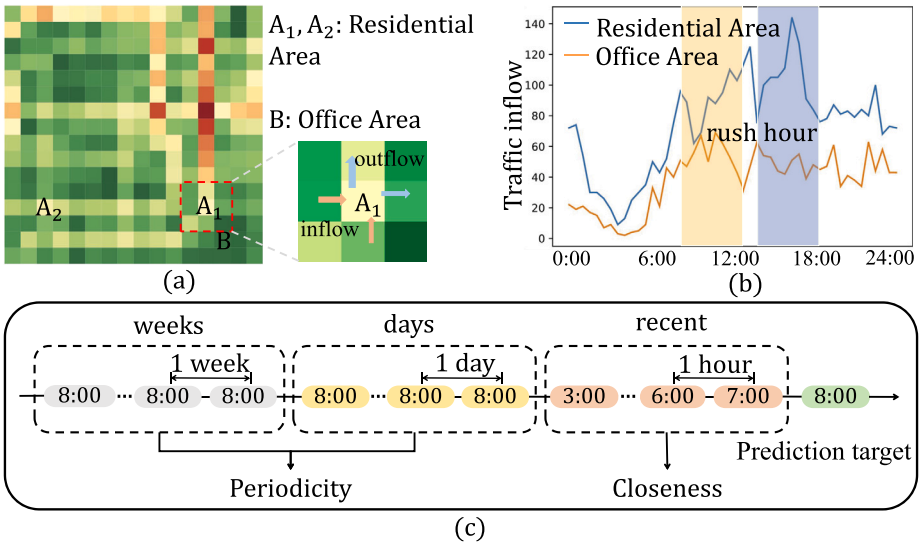


Fig. 1. (a): The traffic flow of a region is affected not only by its local surroundings but also by distant regions with similar functionality. (b): The traffic flow distributions of residential areas differ from office areas. (c): According to temporal semantics, weeks, days, and recent time scales are reassigned to periodicity and closeness.

Road accidents are the leading causes of death worldwide [3] and sadly, dozens of people have died in a deadly stampede during Halloween celebrations in an overcrowded nightlife district in Seoul [4]. With the overview of the future urban flow, city managers can deploy traffic control and related transport scheduling in advance to avoid such unnecessary traffic disasters [5,6]. Because of the significance and urgent need for urban flow prediction, recent advancements in deep learning techniques have greatly accelerated research. This study aims to predict the traffic volume of all regions in the city using knowledge mined from the historical urban flow sequence. Specifically, we focus on predicting the inflow and outflow of each region at a given time, which indicates the crowd flow entering and leaving a region, respectively, as shown in Fig. 1 (a). UFP is a typical spatio-temporal prediction task that requires modeling dependencies in both spatial and temporal dimensions.

First, in the context of temporal dependencies, the urban flow of a region is significantly influenced by its traffic status at the previous moment including corresponding times from previous days or weeks and just past moments. A common strategy is to divide historical time intervals of different lengths into multiple scales, such as weeks, days, and recent periods in Fig. 1 (c), and combine them in a designed model. Existing methods [7–10] process these time fragments in separate modules and merge the results to extract additional temporal correlations. However, modeling temporal dependencies separately may result in the loss of interaction between different time fragments. Although research [11] has realized the importance of global temporal information representation, the heterogeneity of temporal semantics, that is, different time scales showing different semantics, is still neglected. To better maintain the consistency of temporal semantics, as shown in Fig. 1 (c), we can reassign time intervals to periodicity (including weeks and days) and closeness (including recent periods), with periodicity reflecting similar traffic patterns at corresponding times from previous days or weeks while closeness indicating the traffic flow evolutionary trend of the urban flow of the next time steps. Therefore, capturing consecutive traffic patterns and dynamic changes exhaustively is a key challenge that needs to be addressed.

Second, modeling spatial dependencies is equally significant in urban flow prediction. On the urban flow map, the crowd flow of an area is likely to be affected by its surroundings, making it important to extract information from neighbors. To accomplish this, some previous studies [9,12,11] used convolutional neural networks (CNNs) to model the local spatial correlation, while others [13,1] applied graph neural networks (GNNs) to aggregate spatial features from the surroundings. However, most existing models primarily concentrate on capturing local features and disregarding global factors. For example, as shown in Fig. 1 (a), region A_1 and A_2 are both residential areas, which implies they may have similar functionalities and traffic volume even if they are far apart. Meanwhile, the urban flow distributions of residential areas may differ from those of other functional places, such as office areas, as shown in Fig. 1 (b). The relationship between these distant functionally similar areas and the spatial heterogeneity should be considered.

Additionally, external factors such as weather conditions can impact the urban flow to some extent and need to be incorporated as additional features in prediction models [9,14]. Despite the inherent spatial-temporal factors, the dynamics of traffic flow sequence can perturb the prediction results, and the performance can degenerate rapidly if the data are not appropriately normalized [15].

In summary, there are several investigations that need to be explored in urban flow prediction. Existing methods apply the same encoders separately to model three time scales which ignores the semantic differences in different temporal fragments. The temporal semantic differences lie in that the closeness indicates the short-term evolutionary trend and the periodicity reflects the long-term regular traffic patterns. Besides the local spatial dependencies, the effects of global factors are also essential. However, it is challenging to incorporate the global spatial factors without extra information such as the location of functional areas. Moreover, the dynamic disturbances to the urban flow are never analyzed. We list three research questions that are still to be addressed:

· How to distinguish the temporal semantics of periodicity and closeness time scales?
· How to incorporate global influence factors into spatial dependencies?
· How to reduce the perturbations in results from data dynamics?

In this study, we propose a dual-track spatial-temporal framework named DualST to address these challenges. First, we reassign the time scales into periodicity and closeness in terms of temporal semantics and present a dual-track inference structure to encode these time fragments. The dual-track inference structure comprises temporal contextual and temporal causality inferences, which exploit long-term traffic patterns from corresponding previous time intervals and obtain the dynamic evolutionary trend for the next time steps. We then introduce a comprehensive global semantic encoder to capture global spatial representations in a self-learning manner. It assigns semantic tokens to regions according to the similarity of spatial features. We further design a contrastive learning paradigm to enhance these learned tokens and combine them with local spatial embedding. To guarantee the stability of prediction, two auxiliary spatial and temporal adaptive normalization methods are proposed, which adaptively eliminate unstable factors and decrease the perturbations in the urban flow sequence. Finally, we integrate the learned traffic representations with external factors and complete the prediction. The main innovations and contributions of this study are as follows:

• We propose a novel spatial-temporal framework DualST for urban flow prediction. DualST can capture not only the periodic traffic patterns but also dynamic evolution trends based on the dual-track inference module.

• We exploit the potential of contrastive learning in spatial-temporal forecasting, that is, empowering the learned global spatial embedding and incorporating them into regional features.

• To our knowledge, we are the first to introduce the spatial and temporal adaptive normalization to urban flow prediction, which mitigates the instability caused by the dynamics of urban flow.

• Extensive experiments are conducted on two real-world datasets to demonstrate the effectiveness of our method in comparison to other state-of-the-art models. The evaluation results across diverse settings indicate that the proposed method exhibits superior performance.

2. Related work

Urban Flow Prediction. Urban flow prediction is one of the spatial-temporal prediction tasks in the urban computing field [16]. There is significant research on urban flow prediction owing to its wide applicability [17]. Traditional approaches, such as ARIMA [18] and its variations [19,20], treated the urban flow as time sequences and forecasted the flow of each area using linear time series models. The study [21] introduced a probabilistic model that incorporated flow conservation constraints and travel duration distributions between observed locations. However, these models neglected the spatial features between different regions. Recent deep learning studies have gained significant traction in the areas of traffic prediction [10,9,22–25,11]. In [10], they divided time intervals into closeness, period, and trend scales and applied CNNs to capture spatio-temporal correlations. ST-ResNet [9], which is based on residual structure [26], employed a residual framework to further model temporal properties in terms of closeness, period, and trend fragments. However, using the same separate modules for different time scales may result in a loss of the temporal interaction. DMVST-Net [22] proposed a unified multi-view model that jointly considered spatial, temporal, and semantic relations. STRN [23] jointly considered the spatial and temporal information to predict fine-grained urban flows and emphasized the influence of external factors. MF-STN [24] proposed a framework utilizing matrix factorization, aimed at enhancing existing deep ST models. These methods also failed to capture the temporal interaction between different time scales. PDformer [25] introduced a traffic delay-aware feature transformation module to empower the model in propagating time delay spatial information. Nevertheless, the PDformer was unable to leverage the temporal information from the previous days. STDN [27] introduced a flow gating module to learn the similarities between regions and incorporated a periodically shifted attention mechanism to process long-term periodic temporal shifting. However, the above methods did not consider the difference in temporal semantics between time scales. Although ST-GSP [11] utilized an attention mechanism to learn global temporal dependencies, it failed to capture the global spatial dependencies. ATFM [12] predicted urban flow by adaptively incorporating sequential and periodic representations; however, extraction of various temporal semantics is limited by the identical structures used for these two sequences.

In addition to CNN-based methods, GNNs have shown significant progress in traffic prediction [1,13,27,28]; however, these models inevitably inherit notorious inefficiency of GNNs [29]. Most of the existing GCN-based models are built based on a fixed adjacency matrix, which inherently limits the expressive power of the developed models especially when the raw graphs are often noisy or even incomplete [30]. AutoSTG+ [28] adopted neural architecture search for spatio-temporal graphs and employed meta-learning to study knowledge of the attributed graph. ST-MGCN [27] used multi-graph convolution and predefined global contextual information to model the temporal correlation. ST-GDN [13] learned the local regional spatial dependencies and semantics with a hierarchically structured graph neural architecture. In ST-SSL [1], adaptive spatio-temporal self-supervised learning is achieved to encode the information across space and time with temporal and spatial convolutions. In this study, we propose a dual-track inference structure to extract temporal semantics from different time scales and a global semantic encoder to enhance spatial embedding.

Contrastive Learning. Recently, contrastive representation learning has attracted significant attention in computer vision and natural language processing tasks [31–33]. In computer vision, the diverse augmentations of images such as rotation, flipping, color jitters, translation, and noise injection [34] are popular. In natural language processing, input space augmentations include token-level random augmentations such as synonym replacement, word swapping, word insertion, and deletion [35]. Various augmentation methods are employed to generate different views of the input space. Besides that, some studies have verified their excellent performance in learning unsupervised representations of graph data. GraphCL [36] and GCA [37] adopted augmentations of graph or node level to generate positive and negative pairs. IGSD [38] employed graph diffusion for the purpose of producing augmented

Table 1
Table of Notations.

Notation	Description
$X^c; X^p$	closeness and periodicity flow maps
X^t	ground truth urban flow at time step t
$\hat{X}^t$	predicted urban flow at time step t
E^c, E^p	external factors of closeness fragment; external factors of periodicity fragment
$H; W$	height and width of the flow map
$l_c; l_p$	time interval length from closeness fragment; time interval length from periodicity fragment
$N; M$	the number of flow patches; the number of global semantic tokens
$Enc^c; Enc^p$	closeness and periodicity encoder

perspectives, and utilizes a teacher-student framework. Inspired by these studies, efforts have been devoted to spatio-temporal prediction using contrastive learning. In UrbanSTC [39], they presented a self-supervision method considering correlated spatial and temporal contrastive patterns for fine-grained urban flow inference. STGCL [40] investigated the prospect of utilizing the additional signals from contrastive learning for mitigating data scarcity issues. It is concluded that the joint learning scheme of contrastive tasks can achieve better performance than pretrained schemes. In ST-SSL [1], two adaptive augmentations of traffic flow graph data were performed at attribute and structure levels. However, in ST-SSL [1], the influence of external factors is neglected. These above two methods use various augmentation schemes for traffic flow graphs. However, the node embedding obtained via the graph augmentations may be highly biased, which somewhat limits contrastive models in regards to learning discriminative features [41]. Our model generates positive and negative pairs from two temporal views with a joint learning scheme, which makes it end-to-end trainable and avoids the extra augmentations.

3. Preliminaries

This section outlines some fundamental definitions pertinent to urban flow prediction. Related notations used in this study are summarized in Table 1.

Grid flow maps: Given the urban flow maps X gathered from sensors or other signals, we divide them in both space and time dimensions. In the space dimension, the traffic network is divided into $H \times W$ non-overlapping grid regions based on latitude and longitude. In our task, each region in the city has its traffic states, namely, inflow, and outflow, which respectively represent crowd flow entering and leaving it. At any timestamp t, $X^t \in \mathbb{R}^{2\times H\times W}$ represents the flow map, and the two channels denote the inflow and outflow of a specific region.

External factors: Crowd flow is significantly affected by some external factors such as weather conditions and events [42,23]. For example, some extreme weather conditions cause the number of people going out may decrease, while the number of people going out may increase when the weather is sunny [16]. Therefore, it is important to consider external factors. Based on the previous study [11], we scale the temperature and wind speed into the range [0,1] using min-max linear normalization and encode them with other external factors. We transform timestamps, holidays, and weather conditions into binary vectors using one-hot encoding.

Urban flow prediction: We aim to predict the urban flow map at the next time step given the historical data. In this task, temporal embedding fusion is common because we have a consensus that different lengths of time intervals have an impact on prediction. However, most relevant works ignore the differences in temporal semantics. We notice that recent temporal scales represent the dynamic evolution trends, while the corresponding times of previous days or weeks indicate static traffic patterns existing in a specific region. Based on these observations, we summarize the urban flow prediction in a dual-track inference process. Specifically, we reassign the original flow maps into the closeness component X^c (including recent periods) and periodicity component X^p (including days and weeks) in terms of temporal semantics. Our main target is to predict the urban flow map in the next time step,

$$\hat{X}^t \longleftarrow \mathcal{F}(X^c, X^p), \tag{1}$$

where $\hat{X}^t \in \mathbb{R}^{2\times H\times W}$ denotes the predicted urban flow map and $\mathcal{F}(\cdot)$ is our objective function.

4. Methodology

In this section, we propose a model, referred to as DualST, to excavate the spatio-temporal evolution for urban flow prediction. DualST comprises four main components: dual-track temporal inference, global spatial correlation modeling, spatial-temporal adaptive normalization, and embedding fusion. The dual-track temporal inference consists of two components: temporal contextual inference and temporal causality inference, which aim to capture integral temporal dependencies. The global spatial correlation modeling extracts global spatial semantic features and enhances the learned spatial features with a contrastive learning method. The spatial-temporal adaptive normalization is employed to eliminate unstable factors and improve robustness. The embedding fusion merges two urban flow maps from different perspectives and generates the final prediction. Further details are presented in the following section, and the overall architecture is shown in Fig. 2.

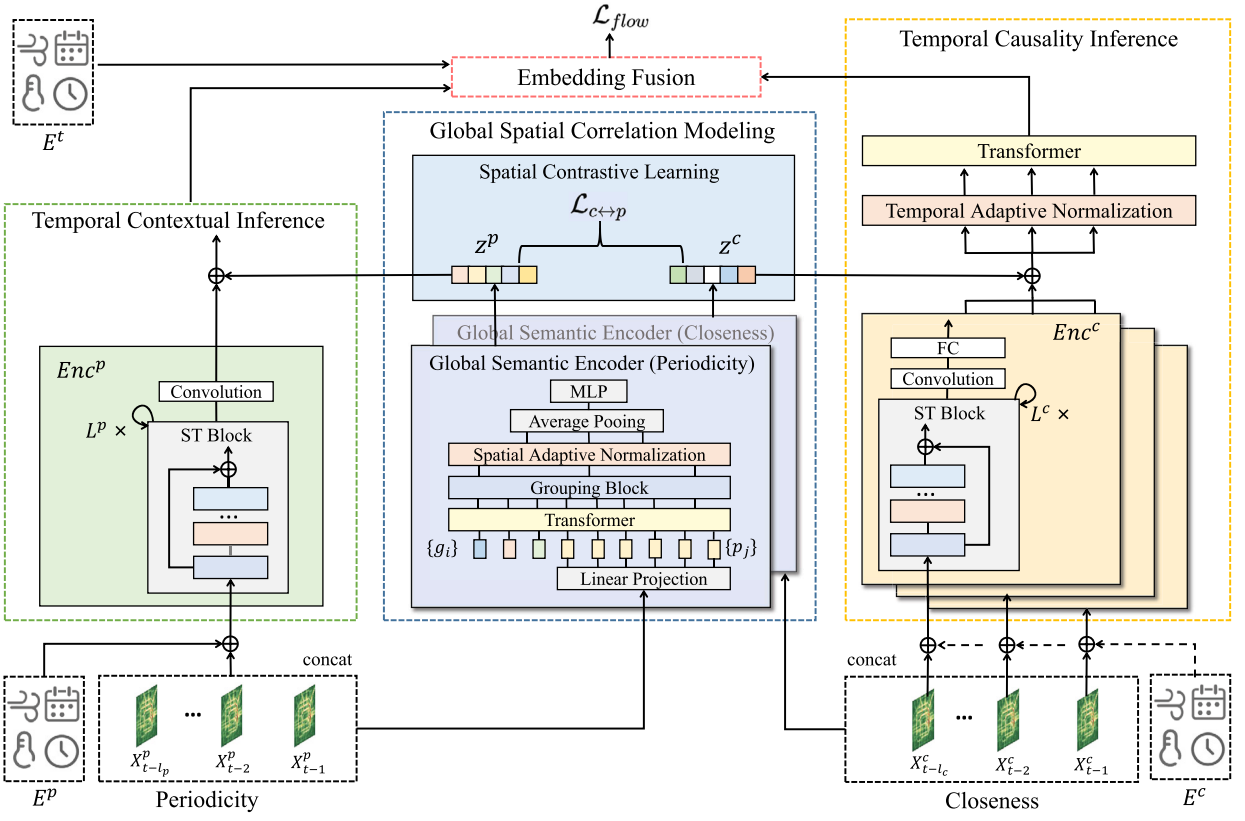


Fig. 2. Visualization of the overall model. The dotted arrows in the lower right from E^c indicate adding to the flow map separately and concat is the concatenate operation. z^p and z^c represent the global semantic embedding of periodicity and closeness. $\{g_i\}$ and $\{p_j\}$ are learnable spatial semantic tokens and flow patches embedding respectively.

4.1. Dual-track temporal inference

Recently, the concept of formulating time series as a sum of trend, seasonal, and error variables [43,44] has achieved great success in long-term time series prediction. In this study, we induce such prior knowledge to closeness and periodicity for short-term spatio-temporal prediction. Given the historical urban flow maps X, we divide them into a closeness fragment X^c and periodicity fragment X^p and feed them to our inference modules. The temporal contextual inference exploits the regular traffic patterns from X^p and the temporal causality inference extracts the urban flow evolution trend from X^c.

4.1.1. Temporal contextual inference

Inputs of this module are periodicity flow maps X^p and external factors at the corresponding time. Given the inputs $X^p = [X^p_{t-l_p}, X^p_{t-(l_p-1)}, \dots, X^p_{t-1}]$, we first concatenate both inflow and outflow together, which yields $X^p_{concat} \in \mathbb{R}^{2l_p \times H \times W}$. Subsequently, we feed the X^p_{concat} to a convolution layer as:

$$X^{p_{(1)}} = f(W^{(1)} * X^p_{concat} + b^{(1)}), \tag{2}$$

where * denotes the convolutional operation; $f(\cdot)$ is an activation function; $W^{(1)}$ and $b^{(1)}$ are the learnable parameters; and $X^{p_{(1)}} \in \mathbb{R}^{d_p \times H \times W}$ denotes the preliminary flow embedding. To induce the influence of external factors, we use two linear layers to extract the features of E^p. Then we reshape the obtained features to $E^{p_{(1)}} \in \mathbb{R}^{d_p \times H \times W}$ and add them to the flow embedding as inputs to the periodicity encoder Enc^p. Enc^p is composed of stacked L^p ST blocks and captures contextual information from the perspective of periodicity so as to model the regular traffic patterns. The ST block uses convolution layers with residual structure as the backbone for spatio-temporal relational representation, as shown in Fig. 3. To cover more nearby regions, we adopt larger 7×7 convolutional kernels and depthwise convolutions like [45] rather than a 3×3 receptive field. At the bottom of the L^pth ST block, we append a transformation function to generate the intermediate flow embedding. The overall calculation process is as follows:

$$M^{p_{(1)}} = X^{p_{(1)}} + E^{p_{(1)}}, \tag{3}$$

$$M^{p_{(L+1)}} = M^{p_{(L)}} + \mathcal{ST}(M^{p_{(L)}}; \theta^{(L)}) \quad L = 1, \dots, L^p, \tag{4}$$

$$\tilde{M}^p = f(W^{(2)} * M^{p_{(L^p+1)}} + b^{(2)}), \tag{5}$$

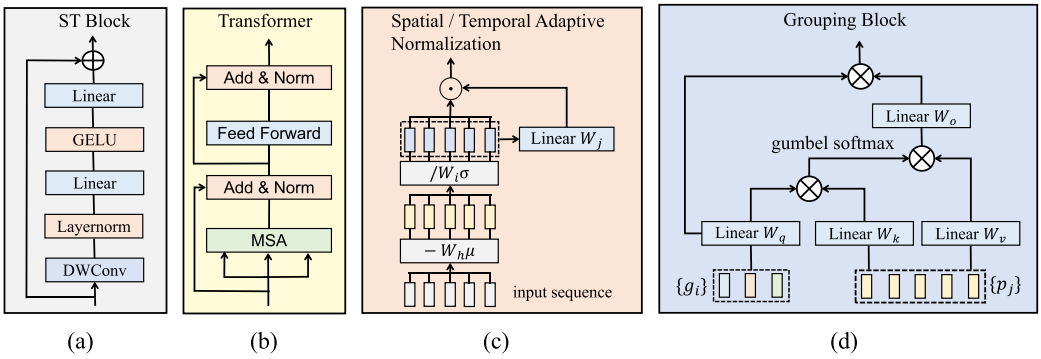


Fig. 3. (a): The structure of the ST Block. (b): Transformer in the temporal causality inference. (c): The design of spatial and temporal adaptive normalization. (d): Grouping block in the global semantic encoder.

where $\mathcal{ST}$ denotes the ST block (the structure of the ST block is shown in Fig. 3 (a)); $\theta^{(L)}$ includes all learnable parameters in the L-th ST block; and $\tilde{M}^p \in \mathcal{R}^{2\times H\times W}$ denotes the region-wise flow embedding. Drawing support from the temporal contextual inference, we can capture region-wise spatial correlations and regular patterns of traffic data across different time steps. Finally, we combine the region-wise flow embedding $\tilde{M}^p$ with the global spatial representations z^p (discussed in Section 4.2) to generate the intermediate urban flow map $\hat{X}^p \in \mathbb{R}^{2\times H\times W}$ over the periodicity view.

4.1.2. Temporal causality inference

Unlike the periodic patterns in a region, recent time periods project the evolution trend of traffic data to the next time step. The evolution trend of urban flow in an area is not only determined by the historical traffic conditions of this region but also affected by local events happening in adjoining areas. These causes and effects from different regions and periods, referred to as temporal causality in this paper, are important influencing factors in urban flow prediction. The traffic flow has dynamic interactions in each previous period; thus, modeling every time slice included in the closeness fragment appropriately and exploiting their relationship are both significant for prediction. Given the $X^c = [X^c_{t-l_c}, X^c_{t-(l_c-1)}, \dots, X^c_{t-1}]$, we combine each of them with the corresponding external factors after the initial feature mapping instead of concatenating them to get $M^{c_{(1)}} = [M^{c_{(1)}}_{t-l_c}, M^{c_{(1)}}_{t-(l_c-1)}, \dots, M^{c_{(1)}}_{t-1}]$. Then each case in $M^{c_{(1)}}$ is fed into a separate closeness encoder Enc^c. To implement the consistency of embedding space, the closeness encoder is designed with a similar structure to that of the periodicity encoder except for the depth and hidden size. Specifically, Enc^c has fewer ST blocks and a larger hidden size compared to Enc^p, such a design imparts a slow-fast [46] formation to our dual-track module, which yields excellent results in spatio-temporal feature extraction. Through a sequence of closeness encoders, we obtain the flow embedding for each time interval $\tilde{M}^c_{\hat{t}}, \hat{t} = t - l_c, \dots, t-1$. Subsequently, we flatten each of them to a vector and reduce the dimensions using a fully connected layer for subsequent temporal reasoning:

$$F_{\hat{t}} = flatten(\tilde{M}^c_{\hat{t}}) \cdot W_f + b_f, \tag{6}$$

where $F_{\hat{t}} \in \mathbb{R}^{d_{model}}$ denotes region-wise flow embedding at a specific time interval $\hat{t}$ and d_{model} is the embedding dimension; W_f and b_f are weights and bias.

We stack all $F_{\hat{t}}$ and incorporate $F \in \mathbb{R}^{l_c\times d_{model}}$ with global spatial representations z^c in the same process as above. To capture the dynamic temporal interaction across different time steps in the closeness fragment, we need to propagate and aggregate the learned flow embedding F. However, the joint distribution of the urban flow sequence changes rapidly over time, which may degrade the performance if the data are not appropriately normalized [15]. Inspired by study [15], we adjust the original normalization approach to spatial-temporal adaptive normalization (Section 4.3), which adapts to the traffic prediction task. Thus, we first feed F into the temporal adaptive normalization procedure to obtain a more stationary flow embedding as:

$$F^{norm} = TemporalAN(F), \tag{7}$$

where *TemporalAN* represents temporal adaptive normalization which does not change the attributes of the original F. Because of the impressive performance achieved by the transformer in time series forecasting [47,48], we employ multi-head a self-attention mechanism [49] (MSA) to capture dynamic temporal correlations. A standard self-attention(SA) operation projects the inputs into three potential spaces, including Q, K, and V, and maps them to outputs as:

$$Attn(Q,K,V) = Softmax\left(\frac{QK^T}{\sqrt{D}}\right)V, \tag{8}$$

where $Q, K, V \in \mathbb{R}^{S\times D}$ are length-S queries, keys, and values of D respectively. MSA is an extension of SA where k self-attention operations run in parallel and project their concatenated outputs. MSA enables the model to collectively attend to diverse represen-

tation subspaces across distinct positions. In our problem, we set $F^{norm} = [F^{norm}_{t-l_c}, \dots, F^{norm}_{t-1}]$ as inputs such that S and D are equal l_c and d_{model}/k respectively. The transformer consists of alternating layers of MSA and MLP blocks, which can be presented as:

$$\begin{aligned} U^0 &= [F^{norm}_{t-l_c}, \dots, F^{norm}_{t-1}], \\ U^n &= LN(MSA(F^{n-1}) + F^{n-1}) \quad n = 1, \dots, N^c, \\ U^n &= LN(MLP(F^n) + F^n) \quad n = 1, \dots, N^c, \end{aligned} \tag{9}$$

where LN denotes layer normalization and N^c is the depth of the transformer. Following transformer-based embedding aggregation, we reshape the learned representation U^n and transform it into intermediate urban flow maps $\hat{X}^c \in \mathbb{R}^{2\times H\times W}$ over the closeness view.

4.2. Global spatial correlation modeling

The flow of a region in a city is influenced not only by its surroundings but also by certain distant regions. Some functional places, such as markets and entertainment facilities, often produce comparable traffic patterns even if they are far apart. Thus, capturing these intrinsic properties of global spatial dependencies is of great significance. Nevertheless, convolutional layers may overlook the relationship between distant regions owing to the size limitations of the convolution kernel and receptive field. Previous study [50] uses the predefined metric of points of interest (POIs) to quantify the functional similarity as a measurement for long-range spatial dependencies; however, such a design introduces human prior knowledge and data noise. To mitigate this challenge, we propose a global semantic encoder to extract additional spatial correlations and augment learned semantic features with spatial contrastive learning.

4.2.1. Global semantic encoder

The global semantic encoder captures spatial features from all regions in a self-learning manner. Drawing inspiration from [33], we introduce the grouping mechanism to urban flow prediction, which assigns the spatial tokens automatically according to their embedding spaces. We consider the learned spatial tokens as the global feature representations for urban flow maps and impose them on the regional flow embedding for integral spatial dependencies. Taking closeness flow maps X^c as an example, we first split them into N non-overlapping patches and linearly project each patch into a latent space following the previous work [51] and get $\{p_j\}^N_{j=1}$. Subsequently, we propose a set of learnable spatial semantic tokens $\{g_i\}^M_{i=1}$, which are treated as input with learned flow patches $\{p_j\}^N_{j=1}$ to the transformer layers (as shown in equation (9)):

$$\{\hat{g}_i\}, \{\hat{p}_j\} = Transformer([\{g_i\}; \{p_j\}]), \tag{10}$$

where [;] denotes the concatenation operator. Through the transformer layers, the global semantic tokens can incorporate flow embedding from all regions on the maps. The grouping block then assigns the spatial semantic tokens to each region based on the similarity between $\{\hat{g}_i\}$ and $\{\hat{p}_j\}$ and Fig. 3 (d) shows the structure of grouping block. We measure the similarity using a attention matrix A, which is computed by cross-attention with a Gumbel-Softmax [52,53] operation as:

$$A_{i,j} = \frac{\exp(W_q\hat{g}_i \cdot W_k\hat{p}_j + \gamma_i)}{\sum^M_{k=1}\exp(W_q\hat{g}_i \cdot W_k\hat{p}_j + \gamma_k)}, \tag{11}$$

where W_q and W_k are the weights of the learned linear projections for the semantic tokens and flow embedding, respectively and $\{\gamma_i\}$ are random samples drawn from the Gumbel (0,1) distribution. We generate the assignment matrix from A as:

$$\hat{A} = one - hot(A_{argmax}) + A - sg(A), \tag{12}$$

where sg is the stop gradient operator used to avoid the indifferentiable facts of argmax. Subsequently, all the flow embedding belonging to the same spatial semantic token are merged to form the new tokens as:

$$\tilde{p}_i = \hat{g}_i + W_o\frac{\sum^M_{j=1}\hat{A}_{i,j}W_v\hat{s}_j}{\sum^M_{j=1}\hat{A}_{i,j}}, \tag{13}$$

where W_o and W_v are the learned weights used to project the merged features. Next, we obtain the spatial semantic tokens $\{\tilde{p}_i\}$ which merge the flow embedding from all regions.

To eliminate unstable factors in the spatial dimension, we feed the learned spatial semantic tokens to the spatial adaptive normalization module and average the outputs to obtain the global spatial representations z^c as:

$$\begin{aligned} \{\tilde{p}^{norm}_i\} &= SpatialAN(\{\tilde{p}_i\}), \\ z^c &= MLP(AvgPool(\{\tilde{p}^{norm}_i\})), \end{aligned} \tag{14}$$

where $SpatialAN$ denotes the spatial adaptive normalization. We obtain z^p for the periodicity flow maps following the same process. Towards this end, we propose a spatial contrastive learning method to excavate the interrelation of z^c and z^p for subsequent inference procedures.

4.2.2. Spatial contrastive learning

To leverage the additional signals from the spatial dimension, we employ a contrastive learning framework to enhance the spatial correlations. In contrast to existing studies, which apply sophisticated data augmentations to generate positive pairs and contrasts, we ingeniously use the learned global representations to construct these pairs. The closeness and periodicity fragments divided from the same X in a mini-batch will be projected in a consistent embedding space. Therefore, we consider all matched closeness-periodicity pairs in a mini-batch as positive pairs and all other unmatched pairs as negative ones. Our objective is to maximize the agreement between the representations of flow maps with similar semantics (i.e., positive pairs) while minimizing those with unrelated spatial semantic information (i.e., negative pairs).

Through the spatial global semantic encoder, we can obtain a batch of B closeness-periodicity pairs $\{(z_i^c, z_i^p)\}_{i=1}^B$, where z_i^c and z_i^p are the closeness and periodicity global spatial representations, of the i-th pair, respectively. We then compute their dot product to measure their similarity. The total closeness-periodicity contrastive loss [33] is defined as:

$$\mathcal{L}_{c\leftrightarrow p} = \mathcal{L}_{c\to p} + \mathcal{L}_{c\leftarrow p}, \tag{15}$$

which is composed of a closeness-to-periodicity contrastive loss, defined as follows:

$$\mathcal{L}_{c\to p} = -\frac{1}{B}\sum_{i=1}^{B} \log \frac{\exp(z_i^c \cdot z_i^p/\tau)}{\sum_{j=1}^{B}\exp(z_i^c \cdot z_j^p/\tau)},$$

and a periodicity-to-closeness contrastive loss defined as

$$\mathcal{L}_{c\leftarrow p} = -\frac{1}{B}\sum_{i=1}^{B} \log \frac{\exp(z_i^p \cdot z_i^c/\tau)}{\sum_{j=1}^{B}\exp(z_i^p \cdot z_j^c/\tau)},$$

where τ is a learnable temperature parameter to scale the logits.

4.3. Spatio-temporal adaptive normalization

Recent research [15] has found that the non-stationary and multimodal nature of data sequence poses significant challenges and severely affects the performance. For real-world traffic datasets, the spatio-temporal dynamics can perturb the prediction results. To address this issue, we adjust the original normalization [15] to spatio-temporal adaptive normalization to better adapt to urban flow prediction. Unlike in a fixed normalization scheme, our method enables the model to learn how to perform normalization for urban flow prediction. We make it a learnable process when the model normalizes the data sequence. For temporal adaptive normalization, given the sequence $F = [F_{t-l^c}, \ldots, F_{t-1}]$ from Enc^c, we first obtain the $\hat{\mu} = W_h\mu$ and $\hat{\sigma} = W_i\sigma$, where W_h and W_i are learnable parameters, and μ is the mean of F; σ is the variance of $\hat{F}$ after subtracting $\hat{\mu}$. Following these operations, we get the initial stationary representations $\tilde{F}$. Subsequently, an adaptive gating layer is applied on $\tilde{F}$ to suppress features that are not relevant or useful:

$$\tilde{F}^{norm} = \tilde{F} \odot sigm(W_j\tilde{\mu} + b_j), \tag{16}$$

where $\odot$ is the Hadamard multiplication operator; sigm$(\cdot)$ is the sigmoid function; W_j and b_j are the parameters of the gating layer; $\tilde{\mu}$ is the mean of $\tilde{F}$. Thus, we obtain $\tilde{F}^{norm}$ for the transformers for more robust temporal inference. Spatial adaptive normalization follows a calculation similar to that of the flow embedding sequence $\{\tilde{p}_i\}$.

4.4. Embedding fusion

Embedding fusion aims to effectively combine the predicted flow maps from different time fragments and replenish results with external factors at the next time step. We achieve this by fusing $\hat{X}^c$ and $\hat{X}^p$ as follows:

$$\hat{X}_t^{cp} = W_c \odot \hat{X}_t^c + W_p \odot \hat{X}_t^p, \tag{17}$$

where W_c and W_p are learnable parameters that adjust the degrees affected by the closeness and periodicity. We employ one layer MLP to generate the representation of external factors E_t at a future corresponding time. Finally, we merge the $\hat{X}_t^{cp}$ with E_t and obtain the predicted urban flow as follows:

$$\hat{X}_t = tanh(\hat{X}_t^{cp} + E_t), \tag{18}$$

where $tanh$ is an activation function and $\hat{X}_t \in \mathbb{R}^{2\times H\times W}$ is the predicted flow map at time step t.

We choose Huber loss [54] as our loss function for urban flow prediction:

$$\mathcal{L}_{flow} = \begin{cases} \frac{1}{2}(\hat{X}_t - X_t), & |\hat{X}_t - X_t| \le \delta \\ \delta|\hat{X}_t - X_t| - \frac{1}{2}\delta^2, & |\hat{X}_t - X_t| > \delta \end{cases}, \tag{19}$$

where δ is a hyperparameter that controls the sensitivity of the squared error loss. Combined with our contrastive loss, the overall loss function is shown as:

Algorithm 1: Training of our model.

Input: Historical urban flow data: $\{X_0, \dots, X_{n-1}\}$;
External factors: $\{E_0, \dots, E_{n-1}\}$;
Length of closeness and periodicity: l_c, l_p.
Output: Our model $\mathcal{M}$.
// construct training instances
$\mathcal{D} \leftarrow \emptyset$;
for *all available time intervals* $t (1 \le t \le n-1)$ **do**
  $X_c = [X^c_{t-l_c}, X^c_{t-(l_c-1)}, \dots, X^c_{t-1}]$;
  $X_p = [X^p_{t-l_p}, X^p_{t-(l_p-1)}, \dots, X^p_{t-1}]$;
  $E_c = [E^c_{t-l_c}, E^c_{t-(l_c-1)}, \dots, E^c_{t-1}]$;
  $E_p = [E^p_{t-l_p}, E^p_{t-(l_p-1)}, \dots, E^p_{t-1}]$;
  // X_t, E_t are target and external factors at time t;
  put an training instance $(\{X_c, X_p, E_c, E_p\}, X_t, E_t)$ into $\mathcal{D}$.
initialize the parameters θ;
// train the model
repeat
  randomly select a batch of instances $\mathcal{D}_c$ from $\mathcal{D}$;
  find θ by minimizing objective (20) with $\mathcal{D}_c$.
until *stopping criteria is met*;
output the learned model $\mathcal{M}$.

Table 2
Statistics of datasets.

Dataset	TaxiBJ	BikeNYC
Time span	7/1/2013 - 10/30/2013 3/1/2014 - 6/30/2014 3/1/2015 - 6/30/2015 11/1/2015 - 4/10/2016	4/1/2014 - 9/30/2014
Location	Beijing	New York
Time interval	30 minutes	1 hour
Grid map size	32×32	16×8
# Available time interval	22,459	4,392
# Number of taxis/bikes	34,000+	6,800+
External Factor (holidays and meteorology)		
# Holidays	41	20
Weather conditions	16 types	\
Temperature / °C	$[-24.6, 41.0]$	\
Wind Speed / mph	$[0, 48.6]$	\

$$\mathcal{L} = \mathcal{L}_{flow} + \mathcal{L}_{c \leftrightarrow p}, \tag{20}$$

and the complete training process is represented by Algorithm 1.

5. Experiments

In this section, we evaluate the performance of our proposed model on two real-world datasets. The experiments show that our model outperforms all baseline methods, achieving significant improvements in accuracy and robustness.

5.1. Experimental settings

5.1.1. Datasets

We perform experiments on two representative datasets: the TaxiBJ and the BikeNYC. The summaries of these two widely used datasets are shown in Table 2.

TaxiBJ. This dataset is generated from the taxicab GPS data and meteorology data in Beijing. It contains a total of 22,459 available traffic flow maps from four different periods. Each flow map reveals inflow and outflow for every half hour interval with a size of 32×32. The external factors contain weather conditions, temperature, wind speed, and 41 categories of the holiday. We chose the data of the last four weeks as testing data and all data before that as training data.

BikeNYC. This dataset is generated from the NYC Bike system of GPS trajectory data. It contains a total of 4,392 available traffic flow maps. Each flow map reveals inflow and outflow hourly with a size of 16×8. We use the data of the last 10 days as testing data and the remainder as training data.

5.1.2. Baselines

We compare our model with the following 11 end-to-end baselines including traditional time series prediction methods, state-of-the-art traffic prediction approaches with graph neural networks, and convolution-based networks for urban flow forecasting.

Traditional time series prediction methods:

HA:. HA predicts the future inflow and outflow by averaging historical data in the corresponding periods. For instance, to predict flow from 10:00 am to 10:30 am on a Wednesday, the corresponding periods are all considered historical time intervals from 10:00 am to 10:30 am on all historical Wednesdays.

ARIMA [55]: Auto-Regressive Integrated Moving Average (ARIMA) is a well-known model used in time series prediction. It uses past time series data to predict future trends.

SARIMA [56]: Seasonal ARIMA (SARIMA) considers seasonal terms when predicting.

VAR [57]: Vector Auto-Regression (VAR) is a famous stochastic process model that can capture the pairwise relationships among all flows.

Traffic prediction with graph neural networks:

ST-SSL [1]: Spatio-Temporal Self-Supervised Learning (ST-SSL) traffic prediction framework applies auxiliary self-supervised learning paradigms to learn traffic pattern representations of both spatial and temporal heterogeneity. Experimental results only include time spans from 03/01/2015 to 06/30/2015 for TaxiBJ owing to data limitation.

AGCRN [58]: Adaptive Graph Convolutional Recurrent Network (AGCRN) enhances the traditional graph convolution with adaptive modules and combines them into a recurrent neural network to capture fine-grained spatio-temporal correlations.

ASTGCN [59]: ASTGCN is a spatio-temporal graph convolutional network based on the attention mechanism, which integrates spatio-temporal attention and convolutional operations to capture dynamic spatio-temporal features.

Convolution-based Network for Traffic Forecasting:

DeepST [10]: DeepST predicts crowd flows using diverse temporal dependencies and external factors.

ST-ResNet [9]: ST-ResNet uses convolution networks with a residual structure to capture spatial dependencies and incorporates the closeness, period, and trend data for the temporal dimension.

ATFM [12]: ATFM is composed of two progressive Convolutional Long Short-Term Memory (ConvLSTM) units. These structures aim to capture spatial attention map inference and dynamic spatial-temporal representations.

ST-GSP [11]: Spatio-Temporal Global Semantic representation learning for urban flow prediction (ST-GSP) uses CNNs to capture the spatial dependencies and the transformer to learn the temporal dependencies.

5.1.3. Implementation details

The proposed model is implemented with PyTorch on NVIDIA RTX 3090, and the same parameter configurations are applied to two datasets. Following the previous study [9], we use the Min-Max normalization method to scale the data into the range $[-1, 1]$ and rescale the predicted value back to the normal values in the evaluation. We apply 4 and 12 ST blocks, with the hidden size of 96 and 16 for temporal contextual and causality inference, respectively. The size of the mini-batch is 32. We set the maximum epochs to 300 and use an early stop strategy with a patience of 50. The initial learning rate is $2e^{-4}$, which is cut in half at epoch $[50, 70, 80, 90]$. The lengths of time intervals for the closeness and periodicity are set to 3 and 2, respectively. Temperature parameter τ is 0.04 and δ is 0.02 for Huber loss. We employ Adam optimization [60] to optimize the parameters of our network.

5.1.4. Evaluation metrics

We evaluate the different methods in terms of Mean Absolute Error (MAE) and Root Mean Square Error (RMSE).

$$\mathrm{MAE} = \frac{1}{\mathrm{N}} \sum_{t=1}^{\mathrm{N}} |\hat{\mathrm{X}}_t - \mathrm{X}_t|,$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{\mathrm{N}} \sum_{t=1}^{\mathrm{N}} (\hat{\mathrm{X}}_t - \mathrm{X}_t)^2},$$

where $\hat{\mathrm{X}}_t$ is the predicted flow map and X_t is the ground truth; N indicates the number of all samples used for validation.

5.2. Comparison with baselines

We present the experimental results in Table 3. It lists the performances of our proposed model and 11 different end-to-end models on the TaxiBJ and BikeNYC. Notably, a lower value of RMSE or MAE indicates better predictive ability. We provide the relative improvement of our method to the second-best model in the last row. The optimal results in each column have been emphasized in **bold**. We summarize the tables with several key observations:

(1) The results of our model demonstrate superior performance to all competing methods in terms of RMSE and MAE, across both datasets. The proposed model achieves an improvement of 3.18% and 3.79% for RMSE and MAE, respectively, on TaxiBJ compared to the second-best model and yields improved results on BikeNYC as well.

(2) Our model exhibits a more stable and robust performance. We apply the same model parameter settings on both datasets when training and obtain satisfactory results. Thus, our method yields promising results with different data distributions.

Table 3
Performance comparison of different methods on TaxIBJ and BikeNYC datasets. Our method outperforms the existing state-of-the-art methods on two datasets.

Method	TaxiBJ		BikeNYC	
	RMSE	MAE	RMSE	MAE
HA	57.69±0.00	29.94±0.00	21.57±0.00	9.13±0.00
SARIMA	25.85±0.00	14.94±0.00	10.34±0.00	5.48±0.00
VAR	21.86±0.00	12.67±0.00	9.72±0.00	5.31±0.00
ARIMA	21.37±0.00	11.52±0.00	9.87±0.00	5.32±0.00
ASTGCN	22.08±0.22	12.66±0.35	7.27±0.23	3.67±0.11
AGCRN	19.78±0.28	10.70±0.16	6.80±0.04	4.18±0.02
ST-SSL	18.63±0.15	11.43±0.07	7.61±0.08	5.10±0.03
DeepST	18.36±0.00	10.52±0.00	7.02±0.00	3.94±0.00
ST-ResNet	16.78±0.35	10.24±0.11	6.38±0.04	3.04±0.02
ATFM	15.38±0.06	9.10±0.03	5.89±0.10	2.97±0.05
ST-GSP	15.09±0.08	8.97±0.05	5.86±0.04	2.86±0.02
Ours	**14.61±0.05**	**8.63±0.02**	**5.69±0.05**	**2.68±0.02**
(Improve)	**+3.18%**	**+3.79%**	**+2.90%**	**+6.29%**

Table 4
Experimental results of multi-step prediction.

Dataset	Method	Step 1		Step 2		Step 3		Step 4		Step 5		Step 6	
		RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
TaxiBJ	ST-ResNet	16.78	10.24	19.21	10.81	20.39	11.38	21.40	11.86	22.29	12.29	24.12	13.25
	ATFM	15.38	9.10	17.06	10.00	18.54	10.67	19.76	11.23	20.89	11.61	21.98	12.04
	ST-GSP	15.09	8.97	16.70	9.63	17.51	10.01	18.58	10.33	19.67	10.77	20.48	11.05
	Ours	**14.61**	**8.63**	**16.24**	**9.30**	**17.23**	**9.87**	**18.21**	**10.08**	**19.09**	**10.43**	**20.07**	**10.72**
	(Improve)	**+3.2%**	**+3.8%**	**+2.8%**	**+3.5%**	**+1.6%**	**+1.4%**	**+2.0%**	**+2.4%**	**+2.9%**	**+3.2%**	**+2.0%**	**+2.9%**
BikeNYC	ST-ResNet	6.38	3.04	7.37	3.53	7.79	3.45	8.21	3.62	8.55	3.74	8.57	3.76
	ATFM	5.89	2.97	6.58	3.16	7.42	3.47	7.60	3.43	7.82	3.60	7.94	3.69
	ST-GSP	5.86	2.86	6.53	2.89	7.26	3.23	7.58	3.37	7.62	3.47	7.76	3.54
	Ours	**5.69**	**2.68**	**6.28**	**2.76**	**7.04**	**3.05**	**7.37**	**3.24**	**7.48**	**3.26**	**7.63**	**3.29**
	(Improve)	**+2.9%**	**+6.3%**	**+3.8%**	**+4.5%**	**+3.0%**	**+5.6%**	**+2.8%**	**+3.9%**	**+1.8%**	**+6.1%**	**+1.7%**	**+7.1%**

The above results prove the validity of our DualST. The statistical methods (e.g. HA and ARIMA) perform worse than the deep learning models. Using only the historical average data oversimplifies the problem, which leads to HA obtaining the worst results. Although ARIMA, SARIMA, and VAR consider the linear relationship over the time series, ignoring spatial dependencies still leads to poor performance. The GCN-based models AGCRN and ASTGCN (implemented on the platform LibCity [61]) show passable results, and we speculate that the graph neural networks may decrease the capacity with a sparse adjacency matrix. The convolutional-based methods, such as DeepST, ST-ResNet, and AFTM, can effectively capture local spatial dependencies and extract temporal features using different methods; hence, they perform adequately. Our method obtains the best performance on the two datasets as it further considers the temporal semantic difference between the closeness and periodicity fragments and incorporates the global spatial features with regional ones.

5.3. Multi-step prediction

To further improve the robustness of our model, we present extra experiments on a more challenging task, multi-step predictions, that use historical data to predict the urban flow maps of the next multi-steps. It requires a higher prediction capacity of the model, which is the ability to capture precise long-range dependency coupling between output and input efficiently [62]. We replace the prediction target at the corresponding time as the new ground truth and obtain the long-term prediction results for state-of-the-art ST-ResNet, ATFM, ST-GSP, and the proposed model. Table 4 lists the prediction results of the next six time steps on the two datasets. There is a performance drop when predicting long-term flow maps for all models, but our approach still yields the best predicted accuracy.

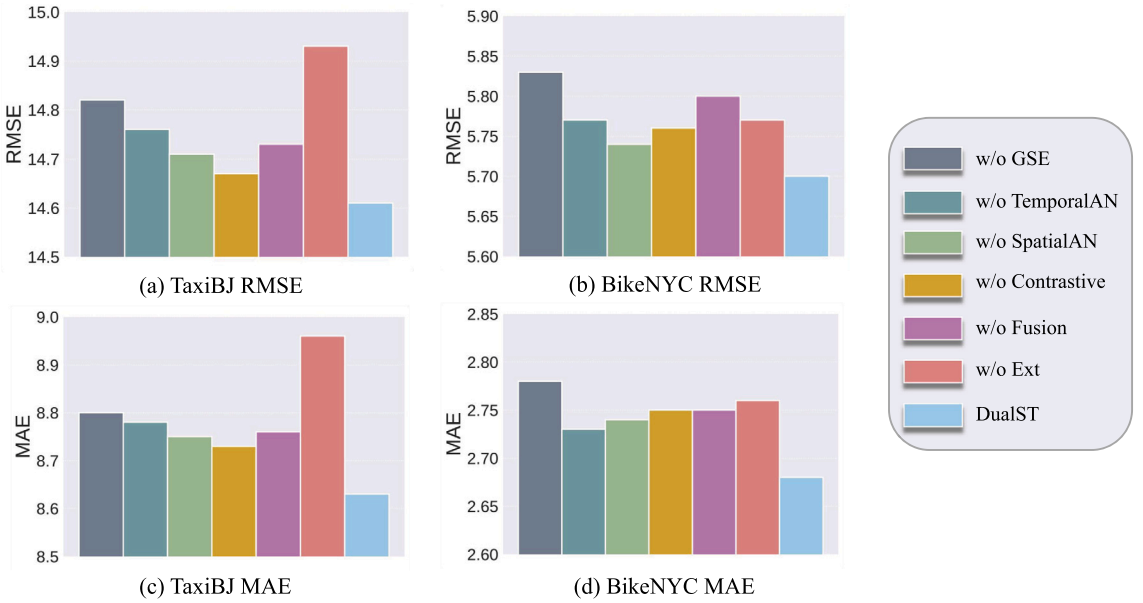


Fig. 4. Ablation studies. We visualize the results of different variants on the TaxiBJ and BikeNYC datasets.

5.4. Ablation study

To evaluate the contribution of each component in our model, we conduct an ablation study in this section. We evaluated six variants of the proposed model on the TaxiBJ dataset and BikeNYC. We presented the results in Fig. 4.

•**w/o-GSE(Global Semantic Encoder)**: remove the global semantic encoder and only use the local flow embedding to accomplish prediction.

•**w/o-TemporalAN**: delete the temporal adaptive normalization when capturing the temporal dependencies.

•**w/o-SpatialAN**: delete the spatial adaptive normalization when capturing the spatial dependencies.

•**w/o-Spatial contrastive learning**: only use $\mathcal{L}_{flow}$ as the final loss and directly add learned global semantic representations to the local flow embedding without joint optimization.

•**w/o-Fusion**: do not use the learnable parameters W_c and W_p at the fusion step but directly add the flow maps from the two views.

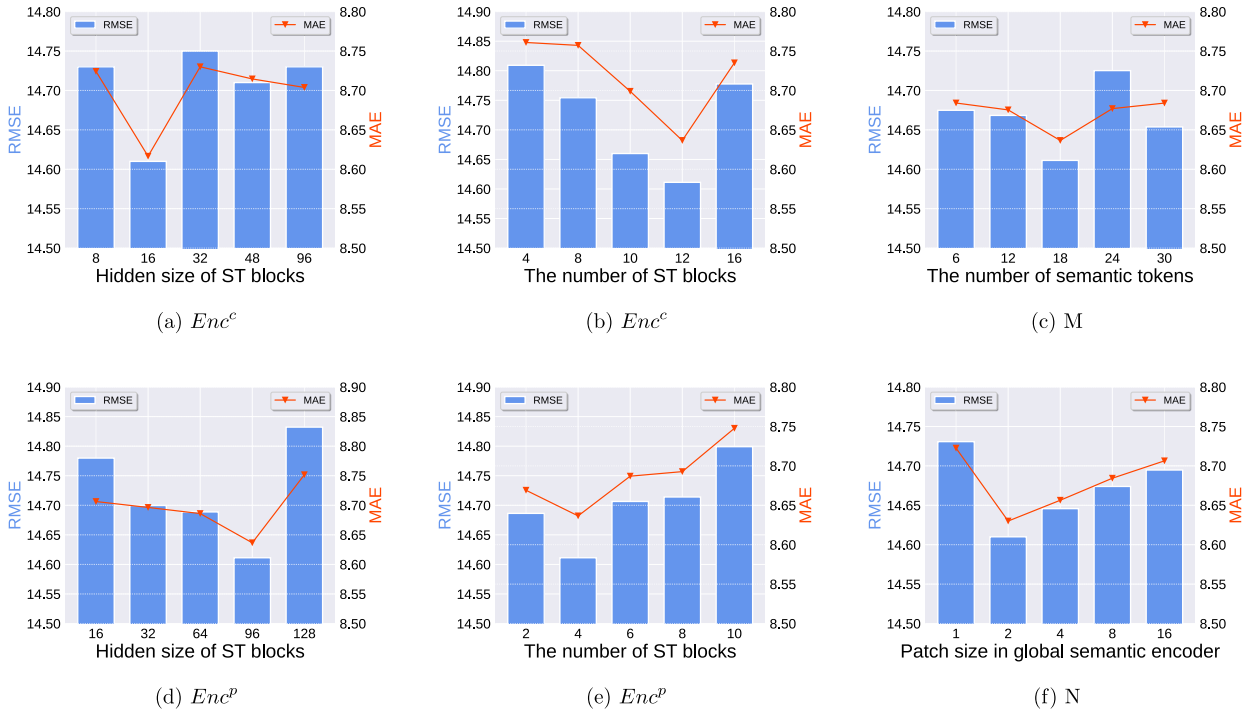
•**w/o-Ext**: ignore the influence of external factors and delete all layers about feature extraction of external factors.

•**Ours**: integral model for urban flow prediction.

The results in Fig. 4 show that each component of our model has played a significant role, and there is considerable performance degradation when any component is removed. As we expected, there is a noticeable performance drop if the global spatial features are neglected. Moreover, the removal of temporal/spatial adaptive normalization demonstrates that normalizing traffic data appropriately can effectively reduce perturbations in the results and improve accuracy. Without joint optimization by spatial contrastive learning, the predictive potential of the model can not be fully realized. External factors are indispensable for urban flow prediction, and we obtain the worst performance when all the parts related to external factors are removed. The last fusion process with the learnable parameters can better maintain the effects from two temporal views. In summary, our model outperforms all variants by jointly using all components, and experimental results meet the expectations of structural design and intuition.

5.5. Evaluation on hyper-parameters settings

In our dual-track inference module, we adopt different parameter settings for closeness and periodicity encoders and we investigate their influence in this section. It can be observed that periodic traffic properties change slowly in a region but the evolution trends usually change rapidly. Thus we apply contrasting training strategies to extract this subtle difference. Inspired by study [46] and to better capture the spatio-temporal correlations, we employ fewer ST blocks with a larger hidden size in Enc^c for dynamic evolutionary features and Enc^p is on the contrary. Therefore, we conduct related experiments on the number and hidden size of ST blocks to examine the robustness of our model as shown in Fig. 5. We observe that lower dimensions (16 in Fig. 5 (a)) and more ST blocks (12 in Fig. 5 (b)) are appropriate for closeness scales to perform temporal reasoning, while higher dimension (96 in Fig. 5 (d)) and fewer ST blocks (4 in Fig. 5 (e)) are better for periodicity scales to capture periodic information. The number of semantic tokens (M) denotes the diversity of spatial semantics in the city. It is advisable to set a suitable number to show the functional diversity in the city and avoid overfitting. The model gets the highest accuracy when the number of semantic tokens is set to 18. The number of spatial patches (N) is determined by the patch size and the size of the flow map. We applied the semantic tokens to the divided



(a) Enc^c (b) Enc^c (c) M

(d) Enc^p (e) Enc^p (f) N

Fig. 5. Studies on hyperparameters.

Table 5
Experimental results of three suboptimal model variants. The model with * denotes the version that adopts dual-track inference patterns and we have marked the improvements.

Models	RMSE	MAE
ST-ResNet	16.78	10.24
ST-ResNet*	16.35 (↑ 0.43)	9.86 (↑ 0.38)
ATFM	15.38	9.10
ATFM*	15.10 (↑ 0.28)	9.00 (↑ 0.10)
ST-GSP	15.09	8.97
ST-GSP*	14.92 (↑ 0.17)	8.87 (↑ 0.10)

regions to capture global representations. The model performs best when the number of patch size in the global semantic encoder is 2.

5.6. Temporal inference analysis

In this section, we design additional experiments on TaxiBJ to verify the effectiveness of the proposed temporal dual-track inference. We choose the three best-performing models ST-ResNet, ATFM, and ST-GSP, and update their structures to our dual-track format. For ST-ResNet, it uses three identical modules to capture the temporal dependencies for closeness, period, and trend time scales respectively. We consider the last two parts as a unified representation of the periodicity and model them with one encoder. Although ATFM has a separate treatment for sequential and periodic time fragments, it is not optimal to apply similar temporal reasoning structures with complicated designs for both. We remove the redundant temporal processing modules for periodicity to better extract regular traffic patterns, which improves the performance and reduces computation. Additionally, we analogously simplify the inputs to the temporal encoder of STGSP. As shown in Table 5, these simple modifications can make significant improvements to all three variants. Overall, extensive experiments demonstrate the generality and efficiency of our dual-track framework in urban flow prediction.

5.7. Investigation of spatio-temporal adaptive normalization

We visualize the series of traffic inflow on TaxiBJ in Fig. 6, where Truth, Prediction, and Without Norm represent the ground truth, the prediction of DualST, and the variant without adaptive normalization, respectively. It is apparent that the urban flow is

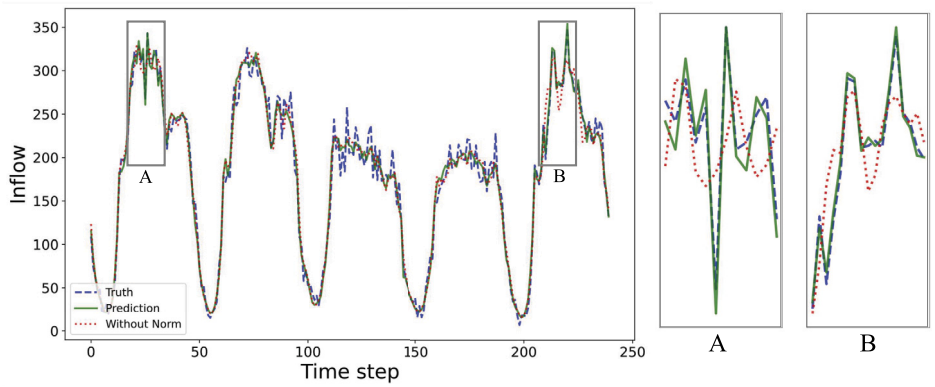


Fig. 6. Visualization of the urban flow of the ground truth, the prediction results of DualST, and the prediction results without adaptive normalization on TaxiBJ.

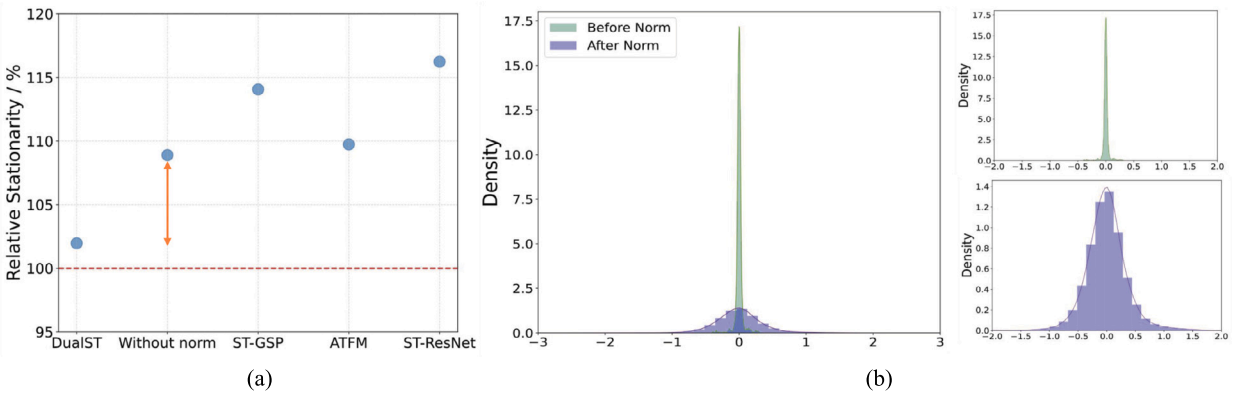


Fig. 7. (a): Relative stationarity between the model predictions and ground truth of different models on TaxiBJ. (b): Visualization of the feature distribution of a time step on the test set before and after the temporal adaptive normalization.

not always stationary in real traffic scenarios. It may change rapidly over a period of time due to traffic events, such as rush hour congestion or accidents. Such dynamic disturbances to the regular traffic patterns, referred to as urban flow perturbations, usually cause performance degradation. As shown in Fig. 6, the urban inflow has significant fluctuations in time periods A and B and the prediction errors exhibit a notable increase of the variant without adaptive normalization. DualST learns how to normalize the traffic data in different traffic states and capture the traffic dynamics in an adaptive manner. It can be seen that our model makes accurate predictions in these two periods and mitigates performance degradation caused by the urban flow perturbations.

5.8. Stationarity analysis

To verify the effectiveness of the spatial-temporal adaptive normalization, we have conducted two experiments about stationarity and feature distribution. We calculated the ratio of ADF test statistics between the model predictions and ground truth of different models and considered the value as the model relative stationarity. The closer the relative stationarity is to 100%, the more stable it is relative to the true data. As shown in Fig. 7 (a), our DualST gets a smaller degree of series stationarity than other advanced models. Different model structures and training strategies cause the discrepancy of the degree of stationarity. The variant removing the spatial-temporal adaptive normalization of DualST only holds a weak advantage compared to other methods. The adaptive normalization brings significant improvement of the model stability to DualST. We continue to explore the reasons for the improvement in stability. We show the feature distribution before and after the temporal adaptive normalization in Fig. 7 (b) and the experiments about spatial adaptive can draw similar conclusions. We find that the feature distribution before the adaptive normalization has high kurtosis which means the features are not stable and there may be some extreme values. Through the adaptive normalization, the model learns how to normalize the temporal features adaptively and can get a stable feature distribution. These experiments show that the spatial-temporal adaptive normalization is an effective component in improving prediction accuracy and model stability.

5.9. Visualization of global flow semantics

Fig. 8 depicts the semantic token assignments for each region of periodicity and closeness in TaxiBJ, with different colors representing different semantic categories. Regions with similar functionalities are assigned to the same category (e.g. green indicates areas on the edges, which typically have less traffic flow in suburban areas) according to our global spatial correlation modeling.

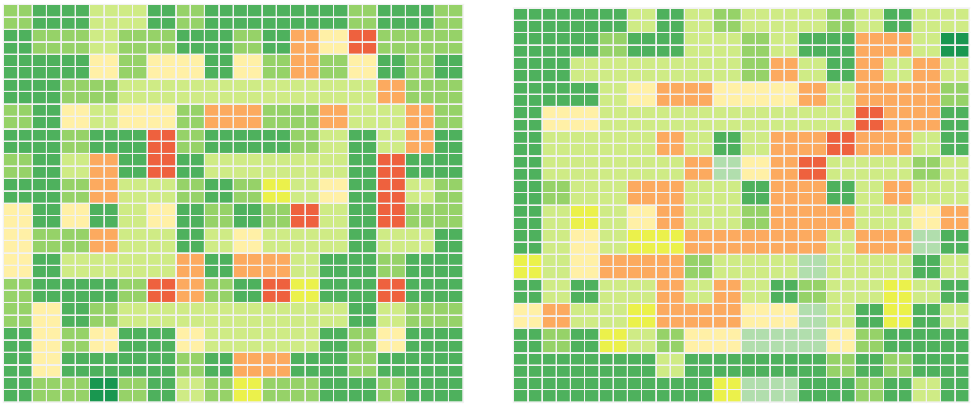


Fig. 8. Visualization of global flow semantics. Different colors indicate different spatial semantics learned by the global semantic encoder. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

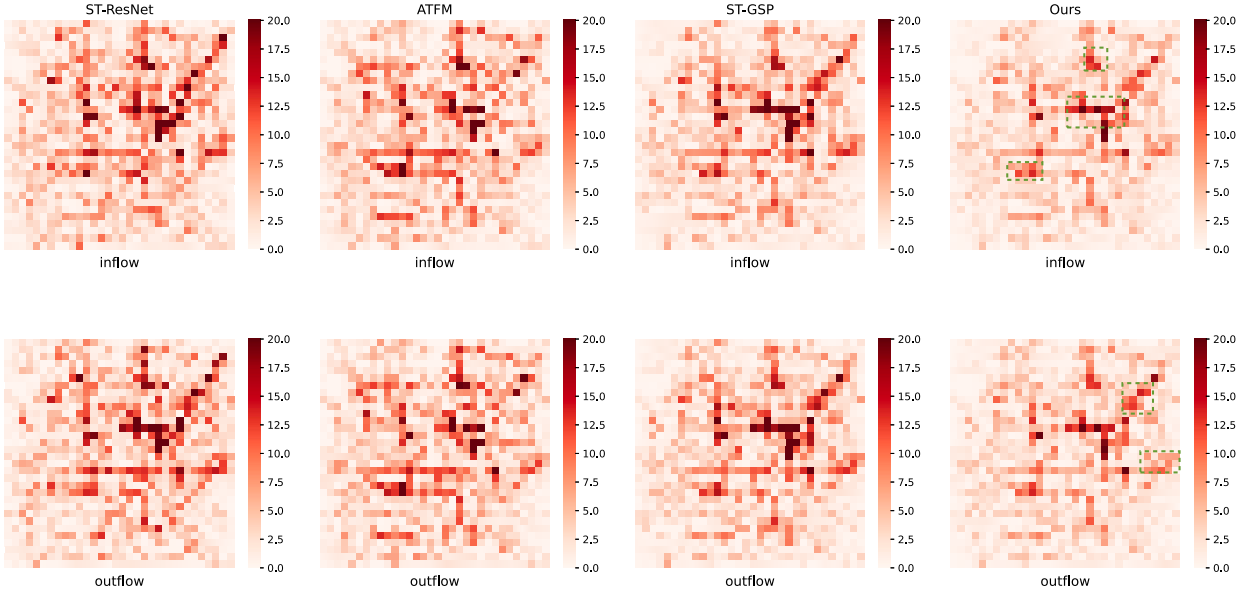


Fig. 9. Visualization for traffic prediction errors.

Disparate from some methods that inject time-invariant additional information (e.g. POIs) to enrich global spatial features, our approach makes global spatial modeling a self-learning process so that it can extract dynamic changes of all time periods. It can be observed that closeness and periodicity time fragments may represent different semantic information in some areas.

5.10. Visualization of urban flow prediction

Fig. 9 visualizes the prediction errors ($|\hat{X}_t - X_t|$) of the proposed DualST and three other best-performing baselines on the TaxiBJ dataset, where darker pixels indicate larger errors. Each column illustrates the inflow and outflow prediction errors. ST-ResNet performs worst with considerable mistakes in so many regions and ATFM makes some progress. And there is still room for improvement in certain challenging areas of the ST-GSP. Our model demonstrates its superiority over the baselines, and the regions of salient improvement are marked in green. DualST can not only reduce overall errors but also conduct outstanding performance in some difficult-to-predict areas. Additionally, regions with large errors are relatively consistent, suggesting that the intrinsic properties of these regions may impact the accuracy of predictions. Although significant progress has been made, it should be noted that the gap between the predicted results and ground truth still exists.

6. Conclusion and future work

This study proposed an intuitive approach with spatio-temporal normalization for urban flow prediction by distinguishing the temporal semantics of different time fragments and incorporating global spatial features with regional dependencies. Specifically, we assigned overall time scales to closeness and periodicity to process temporal causality inference and temporal contextual inference according to their semantics. Furthermore, we explored the application potential of contrastive learning in capturing global spatial correlations. We adopted adaptive spatio-temporal normalization to reduce the perturbations from the data. Comprehensive experiments on two typical urban flow datasets demonstrated the robustness of our model.

In the future, we will pay attention to the more challenging and difficult prediction tasks in urban construction. Concerned that many new smart cities are emerging and there is not enough historical data to refer to in the early construction stage of the city, we will make further efforts to predict urban flow with this model of the few-shot data sequence.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 62202270, 62176139, in part by the Natural Science Foundation of Shandong Province, China, under Grant ZR2021QF034, in part by the Shandong Excellent Young Scientists Fund (Oversea) under Grant 2022HWYQ-044, and in part by the Taishan Scholar Project of Shandong Province under Grant tsqn202306066, and in part by the Major Basic Research Project of Natural Science Foundation of Shandong Province under Grant ZR2021ZD15, and in part by the Open Fund of Beijing Key Laboratory of Traffic Data Analysis and Mining.

References

[1] J. Ji, J. Wang, C. Huang, J. Wu, B. Xu, Z. Wu, J. Zhang, Y. Zheng, Spatio-temporal self-supervised learning for traffic flow prediction, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 37, 2023, pp. 4356–4364.
[2] Y. Gong, Z. Li, J. Zhang, W. Liu, Y. Zheng, Online spatio-temporal crowd flow distribution prediction for complex metro system, IEEE Trans. Knowl. Data Eng. 34 (2) (2020) 865–880.
[3] A. Rosenfeld, O. Maksimov, S. Kraus, When security games hit traffic: a deployed optimal traffic enforcement system, Artif. Intell. 289 (2020) 103381.
[4] O. Analytica, Deadly stampede in Seoul will leave long legacy, in: Emerald Expert Briefings (oxan-es), 2022.
[5] Y. Gong, Z. Li, J. Zhang, W. Liu, J. Yi, Potential passenger flow prediction: a novel study for urban transportation development, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 4020–4027.
[6] Y. Gong, Z. Li, J. Zhang, W. Liu, Y. Zheng, C. Kirsch, Network-wide crowd flow prediction of Sydney trains via customized online non-negative matrix factorization, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, 2018, pp. 1243–1252.
[7] S. Guo, Y. Lin, S. Li, Z. Chen, H. Wan, Deep spatial–temporal 3d convolutional neural networks for traffic data forecasting, IEEE Trans. Intell. Transp. Syst. 20 (10) (2019) 3913–3926.
[8] Z. Xu, Y. Wang, M. Long, J. Wang, M. KLiss, Predcnn: predictive learning with cascade convolutions, in: IJCAI, 2018, pp. 2940–2947.
[9] J. Zhang, Y. Zheng, D. Qi, Deep spatio-temporal residual networks for citywide crowd flows prediction, in: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, 2017, pp. 1655–1661.
[10] J. Zhang, Y. Zheng, D. Qi, R. Li, X. Yi, Dnn-based prediction model for spatio-temporal data, in: Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, 2016, pp. 1–4.
[11] L. Zhao, M. Gao, Z. Wang, St-gsp: spatial-temporal global semantic representation learning for urban flow prediction, in: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, 2022, pp. 1443–1451.
[12] L. Liu, J. Zhen, G. Li, G. Zhan, Z. He, B. Du, L. Lin, Dynamic spatial-temporal representation learning for traffic flow prediction, IEEE Trans. Intell. Transp. Syst. 22 (11) (2020) 7169–7183.
[13] X. Zhang, C. Huang, Y. Xu, L. Xia, P. Dai, L. Bo, J. Zhang, Y. Zheng, Traffic flow forecasting with spatial-temporal graph diffusion network, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, 2021, pp. 15008–15015.
[14] Z. Li, J. Zhang, Q. Wu, Y. Gong, J. Yi, C. Kirsch, Sample adaptive multiple kernel learning for failure prediction of railway points, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 2848–2856.
[15] N. Passalis, A. Tefas, J. Kanniainen, M. Gabbouj, A. Iosifidis, Deep adaptive input normalization for time series forecasting, IEEE Trans. Neural Netw. Learn. Syst. 31 (9) (2019) 3760–3765.
[16] P. Xie, T. Li, J. Liu, S. Du, X. Yang, J. Zhang, Urban flow prediction from spatiotemporal data using machine learning: a survey, Inf. Fusion 59 (2020) 1–12.
[17] D.A. Tedjopurnomo, Z. Bao, B. Zheng, F.M. Choudhury, A.K. Qin, A survey on modern deep neural network for traffic prediction: trends, methods and challenges, IEEE Trans. Knowl. Data Eng. 34 (4) (2020) 1544–1561.
[18] C. Moorthy, B. Ratcliffe, Short term traffic forecasting using time series methods, Transp. Plann. Technol. 12 (1) (1988) 45–56.
[19] L. Moreira-Matias, J. Gama, M. Ferreira, J. Mendes-Moreira, L. Damas, Predicting taxi–passenger demand using streaming data, IEEE Trans. Intell. Transp. Syst. 14 (3) (2013) 1393–1402.
[20] S. Shekhar, B.M. Williams, Adaptive seasonal time series models for forecasting short-term traffic flow, Transp. Res. Rec. 2024 (1) (2007) 116–125.
[21] Y. Tanaka, T. Iwata, T. Kurashima, H. Toda, N. Ueda, T. Tanaka, Time-delayed collective flow diffusion models for inferring latent people flow from aggregated data at limited locations, Artif. Intell. 292 (2021) 103430.

[22] H. Yao, F. Wu, J. Ke, X. Tang, Y. Jia, S. Lu, P. Gong, Z. Li, J. Ye, D. Chuxing, Deep multi-view spatial-temporal network for taxi demand prediction, in: 32nd AAAI Conference on Artificial Intelligence, AAAI 2018, AAAI Press, 2018, pp. 2588–2595.
[23] Y. Liang, K. Ouyang, J. Sun, Y. Wang, J. Zhang, Y. Zheng, D. Rosenblum, R. Zimmermann, Fine-grained urban flow prediction, in: Proceedings of the Web Conference 2021, 2021, pp. 1833–1845.
[24] Z. Pan, Z. Wang, W. Wang, Y. Yu, J. Zhang, Y. Zheng, 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, 2019, pp. 2683–2691.
[25] J. Jiang, C. Han, W.X. Zhao, J. Wang, Pdformer: propagation delay-aware dynamic long-range transformer for traffic flow prediction, arXiv preprint arXiv: 2301.07945, 2023.
[26] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[27] H. Yao, X. Tang, H. Wei, G. Zheng, Z. Li, Revisiting spatial-temporal similarity: a deep learning framework for traffic prediction, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 5668–5675.
[28] S. Ke, Z. Pan, T. He, Y. Liang, J. Zhang, Y. Zheng, Autostg+: an automatic framework to discover the optimal network for spatio-temporal graph prediction, Artif. Intell. 318 (2023) 103899.
[29] X. Liu, Y. Liang, C. Huang, H. Hu, Y. Cao, B. Hooi, R. Zimmermann, Do we really need graph neural networks for traffic forecasting?, arXiv preprint arXiv: 2301.12603, 2023.
[30] K. Yao, J. Liang, J. Liang, M. Li, F. Cao, Multi-view graph convolutional networks with attention mechanism, Artif. Intell. 307 (2022) 103708.
[31] K. He, H. Fan, Y. Wu, S. Xie, R. Girshick, Momentum contrast for unsupervised visual representation learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 9729–9738.
[32] A. Radford, J.W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al., Learning transferable visual models from natural language supervision, in: International Conference on Machine Learning, PMLR, 2021, pp. 8748–8763.
[33] J. Xu, S. De Mello, S. Liu, W. Byeon, T. Breuel, J. Kautz, X. Wang, Groupvit: semantic segmentation emerges from text supervision, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 18134–18144.
[34] S. Yang, W. Xiao, M. Zhang, S. Guo, J. Zhao, F. Shen, Image data augmentation for deep learning: a survey, arXiv preprint arXiv:2204.08610, 2022.
[35] J. Wei, K. Zou Eda, Easy data augmentation techniques for boosting performance on text classification tasks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 6382–6388.
[36] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, Y. Shen, Graph contrastive learning with augmentations, Adv. Neural Inf. Process. Syst. 33 (2020) 5812–5823.
[37] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, L. Wang, Graph contrastive learning with adaptive augmentation, in: Proceedings of the Web Conference 2021, 2021, pp. 2069–2080.
[38] H. Zhang, S. Lin, W. Liu, P. Zhou, J. Tang, X. Liang, E.P. Xing, Iterative graph self-distillation, IEEE Trans. Knowl. Data Eng. (2023).
[39] H. Qu, Y. Gong, M. Chen, J. Zhang, Y. Zheng, Y. Yin, Forecasting fine-grained urban flows via spatio-temporal contrastive self-supervision, IEEE Trans. Knowl. Data Eng. 01 (2022) 1–17.
[40] X. Liu, Y. Liang, C. Huang, Y. Zheng, B. Hooi, R. Zimmermann, When do contrastive learning signals help spatio-temporal graph forecasting?, in: Proceedings of the 30th International Conference on Advances in Geographic Information Systems, 2022, pp. 1–12.
[41] Y. Zhang, H. Zhu, Z. Song, P. Koniusz, I. King Costa, Covariance-preserving feature augmentation for graph contrastive learning, in: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022, pp. 2524–2534.
[42] J. Zhang, Y. Zheng, D. Qi, R. Li, X. Yi, T. Li, Predicting citywide crowd flows using deep spatio-temporal residual networks, Artif. Intell. 259 (2018) 147–166.
[43] S.L. Scott, H.R. Varian, Bayesian variable selection for nowcasting economic time series, in: Economic Analysis of the Digital Economy, University of Chicago Press, 2015, pp. 119–135.
[44] G. Woo, C. Liu, D. Sahoo, A. Kumar, S. Hoi Cost, Contrastive learning of disentangled seasonal-trend representations for time series forecasting, in: International Conference on Learning Representations, 2021.
[45] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, S. Xie, A convnet for the 2020s, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 11976–11986.
[46] C. Feichtenhofer, H. Fan, J. Malik, K. He, Slowfast networks for video recognition, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 6202–6211.
[47] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, W. Zhang, Informer: beyond efficient transformer for long sequence time-series forecasting, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, 2021, pp. 11106–11115.
[48] H. Wu, J. Xu, J. Wang, M. Long, Autoformer: decomposition transformers with auto-correlation for long-term series forecasting, Adv. Neural Inf. Process. Syst. 34 (2021) 22419–22430.
[49] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017, pp. 6000–6010.
[50] X. Geng, Y. Li, L. Wang, L. Zhang, Q. Yang, J. Ye, Y. Liu, Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 3656–3663.
[51] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: transformers for image recognition at scale, in: International Conference on Learning Representations, 2020.
[52] E. Jang, S. Gu, B. Poole, Categorical reparameterization with gumbel-softmax, in: International Conference on Learning Representations, 2016.
[53] C. Maddison, A. Mnih, Y. Teh, The concrete distribution: a continuous relaxation of discrete random variables, in: Proceedings of the International Conference on Learning Representations, International Conference on Learning Representations, 2017.
[54] M. Li, Z. Zhu, Spatial-temporal fusion graph neural networks for traffic flow forecasting, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, 2021, pp. 4189–4196.
[55] G.E. Box, G.M. Jenkins, G.C. Reinsel, G.M. Ljung, Time Series Analysis: Forecasting and Control, John Wiley & Sons, 2015.
[56] B.M. Williams, P.K. Durvasula, D.E. Brown, Urban freeway traffic flow prediction: application of seasonal autoregressive integrated moving average and exponential smoothing models, Transp. Res. Rec. 1644 (1) (1998) 132–141.
[57] S. Johansen, Estimation and hypothesis testing of cointegration vectors in gaussian vector autoregressive models, Econometrica (1991) 1551–1580.
[58] L. Bai, L. Yao, C. Li, X. Wang, C. Wang, Adaptive graph convolutional recurrent network for traffic forecasting, Adv. Neural Inf. Process. Syst. 33 (2020) 17804–17815.
[59] S. Guo, Y. Lin, N. Feng, C. Song, H. Wan, Attention based spatial-temporal graph convolutional networks for traffic flow forecasting, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 922–929.
[60] D. Kingma, Adam: a method for stochastic optimization, in: Int Conf Learn Represent, 2014.
[61] J. Wang, J. Jiang, W. Jiang, C. Li, W.X. Zhao, Libcity: an open library for traffic prediction, in: Proceedings of the 29th International Conference on Advances in Geographic Information Systems, 2021, pp. 145–148.
[62] H. Zhou, J. Li, S. Zhang, S. Zhang, M. Yan, H. Xiong, Expanding the prediction capacity in long sequence time-series forecasting, Artif. Intell. 318 (2023) 103886.