ST-ExpertNet: A Deep Expert Framework for Traffic Prediction

Hongjun Wang®, Jiyuan Chen®, Zipei Fan®, Zhiwen Zhang®, Zekun Cai®, and Xuan Song®

Abstract—Recently, forecasting the crowd flows has become an important research topic, and plentiful technologies have achieved good performances. As we all know, the flow at a citywide level is in a mixed state with several basic patterns (e.g., commuting, working, and commercial) caused by the city area functional distributions (e.g., developed commercial areas, educational areas, and parks). However, existing technologies have been criticized for their lack of considering the differences in the flow patterns among regions since they want to build only one comprehensive model to learn the mixed flow tensors. Recognizing this limitation, we present a new perspective on flow prediction and propose an explainable framework named ST-ExpertNet, which can adopt every spatial-temporal model and train a set of functional experts devoted to specific flow patterns. Technically, we train a bunch of experts based on the Mixture of Experts (MoE), which guides each expert to specialize in different kinds of flow patterns in sample spaces by using the gating network. We define several criteria, including comprehensiveness, sparsity, and preciseness, to construct the experts for better interpretability and performances. We conduct experiments on a wide range of real-world taxi and bike datasets in Beijing and NYC. The visualizations of the expert’s intermediate results demonstrate that our ST-ExpertNet successfully disentangles the city’s mixed flow tensors along with the city layout, e.g., the urban ring road structure. Different network architectures, such as ST-ResNet, ConvLSTM, and CNN, have been adopted into our ST-ExpertNet framework for experiments and the results demonstrate the superiority of our framework in both interpretability and performances.

Index Terms—Crowd flow prediction, mixture of experts, neural networks, urban computing

1 INTRODUCTION

In recent years, with the maturity of perception technology and computing environment, various kinds of big data have quietly emerged in cities, such as traffic flow, weather data, road networks, points of interest, movement trajectories, and social media. Among them, the task of traffic flow prediction is of great significance to urban traffic management and public safety and has received extensive attention from academia in recent years. Existing efforts have mainly considered that traffic flow is affected by recent time intervals, daily periodicity, and weekly trend [1], [2], [3], [4], which can help the model distinguish functional regions. For instance, we can recognize between residential areas and office areas by recent traffic flows. Specifically, people go out to work on weekdays morning, so the outflow of residential regions will increase, and the inflow will grow in office areas. Furthermore, by observing periodic flows, we can find that both inflow and outflow remain stable during the week in the residential area, but will drop significantly on weekends in the office area.

However, in the literature, these factors are mainly considered as purely time-series signals rather than the unique features of each functional region. Moreover, they ignored the fact that the flows of the different functional areas only influence relevant parts rather than citywide. For instance, the entertaining region’s flow growing up on weekends will not affect the flow state in the office areas. This ignorance will finally result in the improper design of the previous citywide prediction model, which aims to use a single network to predict the flows of all functional regions, inevitably causing difficulties in thoroughly understanding the different flow patterns. Further, when an abnormal prediction is produced from the previous model, it can’t tell why this happens. This will be solved if we can see the knowledge of the area learned by the model. For instance, if the region is an entertaining district, the model considers it an office area and outputs tiny values on weekends. We will know if the misunderstanding of the model causes this.

Most of the previous work has successfully exploited the laws of human mobility in the flow tensor and provided a new perspective for researchers to understand the human flow pattern [5], [6], [7]. Recently, the crowd flow tensor has been demonstrated can be decomposed into several basic
patterns of human life [8], each of which has its law, such as the commuting pattern that focuses mainly on the morning and evening peaks on weekdays and the entertaining pattern that focuses on the entire day of the weekends. The flow pattern distribution is greatly affected by both time and the distribution of functional regions geographically. For example, the commuting pattern dominates highways all day while the working pattern only occupies office area till evenings. A functional region can be filled with many flow patterns at each timestamp, but only one or two of them is the main component.

Fig. 1 shows an example of using three same networks (experts) trained with attention mechanism on real-world taxi data in Beijing [1] during rush hours. The attention mechanism assigns weights to the experts so that each region of the $32 \times 32$ mesh grids (i.e., the Beijing city map) might receive predictions from a number of experts but with different weightings. In Fig. 1a, there are $32 \times 32$ points representing all regions of Beijing, and each region receives attention from three experts, whose relatively proportional weight can be viewed from the axis. We further cluster regions with similar relatively proportional weights of three experts using KMeans, and one of the classes has been represented in Fig. 1b, which is the primary commuting road section in Beijing, demonstrating that the experts have successfully mined the commuting flow patterns and the underneath Beijing ring road structure. However, as we can see in the Fig. 1a, the flow patterns are closely connected to each other, which suggests that a single attention mechanism without regularization can not fulfill our needs.

By summing up the conclusions proposed by previous work, this paper considers that the traffic flow prediction task could be divided into two-part: 1) building a set of expert models to automatically disentangle the mixed-flow tensors with several inherent patterns. 2) predicting the future flow state of each pattern. The task of training multiple functional experts could be considered as the MoE (mixture of experts) problem [9], [10], [11], which generates expert systems through differentiated learning strategies to improve the precision and robustness of the deep neural network. We propose the ST-ExpertNet framework based on MoE [12], which divides the sample space by a gating network. In ST-ExpertNet, the flow of a region is equal to the combination of multiple basic flow patterns, which enhances the robustness and certainty of the model compared to a single model. For example, we can see the reasonableness of the model’s predictions through the components of different flow patterns. Meanwhile, as mentioned above, the combination of multiple models improves the fault tolerance of the model compared to a traditional model, which is a kind of approach widely used in adversarial robustness[13].

For better performance and interpretability of ST-ExpertNet, this paper considers the following criteria:

**Comprehensiveness.** The data distributions learned by each expert should be as different as possible. To achieve that, each expert should have enough knowledge to distinguish different types of flow patterns caused by time and the characteristic distribution of functional regions. In our approach, the factors of hour, day, and week and metadata have been concatenated together to provide each expert the distinctive features for each region and flow patterns. This is different from the traditional strategy [1], [14], [15].

**Sparsity.** The expert should be discrepant in both spatial and temporal aspects. For instance, in terms of the temporal aspects, while one expert is in charge of the peaking hours in residential areas, the other one is responsible for the midnight periods. From spatial aspects, the experts should pay attention to different functional regions. Therefore, for each input, only a few “activated” experts are encouraged. In this paper, the penalty term Expert Inter Discrepancy Loss which bases on Determinant Point Process (DPP) has been proposed to punish the similarity between experts. Additionally, redundant experts should be avoided, as they add to the complexity of the explanation but do not add additional performance. Therefore, the number of experts should be controlled in small numbers to provide a clear explanation for users in ST-ExpertNet.

**Preciseness.** Each expert should assume full responsibility for its prediction. Therefore, Expert Responsibility Loss has been proposed to improve the prediction performance in the flow patterns charged by experts. Two types of gating networks have been proposed in terms of spatial and temporal aspects. The former intends to generate the attention for each expert and to provide the opportunity to peek inside the experts by attention mechanisms [16], and the latter aims to adjust the output signal amplitude from a global perspective.

To further evaluate the interpretability and performance of the ST-ExpertNet framework, we integrate ST-ResNet [1], convolutional LSTM (ConvLSTM) [3], and convolutional neural networks (CNN), which are popular baselines in the flow prediction task, into our framework. We test our explainable framework in real-world taxi and bike datasets: TaxiBJ, TaxiNYC, BikeNYC-I, and BikeNYC-II in Beijing and New York respectively. ST-ExpertNet can give embedded representations for each region or pattern in our city with spatial and temporal aspects. For temporality, we show the evolutionary process between the office and residential areas of Beijing. For spatiality, we show the knowledge distribution of each expert at the citywide level, which is well-aligned with the human cognition of our city.

We summarize our contributions to the field as follows:

1) We provide a new perspective of viewing the crowd flow prediction problem by decomposing the mixed...
flow pattern into several basic ones and training "specialized" experts to gain better performances. We propose a novel and efficient architecture based on the mixture of experts (MoE), which can adopt every ST-model by making it as an expert.

2) We design Experts Inter Discrepancy Loss and Experts Responsibility Loss for regularizing the expert learning paradigm.

3) We conduct experiments on several real-world traffic datasets. The popular baselines in flow prediction: ST-ResNet, ConvLSTM, CNN have been employed in our framework, and the results show that our model is consistently better and more explainable than other state-of-the-art methods.

The remainder of our paper is arranged as follows: Section 2 introduces the related work, including mixture of experts, crowd flow prediction and data mining in human mobility. Section 3 gives the primary definition of flows and the flow prediction problem. It also describes our basic idea to solve crowd flow prediction based on the general framework of mixture of experts. Section 4 introduces our ST-ExpertNet framework in detail. Section 5 conducts both the quantitative and qualitative experiments on our ST-ExpertNet. Section 6 discusses the conclusion and future works.

2 RELATED WORK

In this section, we will discuss some relevant works about mixture of expert system, traffic flow prediction, as well as data mining in human mobility.

2.1 Mixture of Experts

Mixture of experts (MoE) belongs to a type of combining method, which is similar to boosting and negative correlation learning methods. The basic ideas of MoE is to train multiple experts, and each expert is assigned to a different part of the subtask. Therefore, MoE is a good compromise between a single global model or multiple local models, but one of the most important questions for MoE is how to divide a dataset into different parts. Now, MoE can be roughly divided according to whether the data is divided in advance [17]. Based on the implicit problem space partitioning [18], [19], [20], the prior information is used to partition the dataset by clustering methods. The key issues for these technologies are the reliability of the prior information and how to balance the data after clustering. To overcome this challenge, a gating approach has been proposed to balance the learning strategy by guiding the experts to learn to handle different subsets of training data [9], [21], [22]. There are other kinds of work focusing on adopting different models such as Support Vector Machines (SVMs) [23], Gaussian Processes [24] and Dirichlet Processes [25], or different expert configurations such as a hierarchical structure [26], infinite numbers of experts [27], and adding experts sequentially [28] into the framework. Our work extends the application of MoE systems into the framework of flow prediction in urban computing by dividing the flow tensor space. Further, our design serves as a more explainable and efficient framework which can adopt every ST-models, which aims to improve their performance and interpretation.

2.2 Traffic Flow Prediction

Recently, machine learning is increasingly being used in intelligent transportation system, such as, ordinal classification [29], intrusion detection [30] and classify vehicle images [31]. Among the literature, Traffic flow prediction has become the hottest topic in transportation. Traffic flow owns the characteristics of highly nonlinear time correlation and uncertainty [32], [33], which brought huge challenges to traffic flow prediction. The basic assumption is to consider the traffic flow prediction as a time series problem. Among them, the autoregressive moving average model (ARMA) [34] as the fundamental time series prediction model has been widely used in traffic flow. The variant methods based on the wavelet neural network [35] and difference integration [36] have been proposed. However, these kinds of models lack the necessary modeling of spatial dependence [1]. To both consider the spatial and temporal factors, the deep learning model is first proposed [37] by combining the convolutional neural network (CNN) and Deep Neural Networks (DNN). The time series concepts of the hour, day, and week patterns are further defined [1] and the residual network (ResNet) [38] is used to process them respectively. A multitask adversarial ST-network is later proposed to infer the in-out flows and origin-destination (OD) simultaneously [39]. A deep irregular convolutional residual LSTM network is proposed [40] to model the challenges, such as, hybrid transportation lines, mixed traffic, transfer stations, and some extreme weathers. Multiple architectures have been designed to improve the performance based on the framework [1], such as using ConvGRU [4], Long Short-Term Memory (LSTM) [15], ConvLSTM [3] instead of CNN. Although these models achieved good performances, the criticisms are mainly focused on their lack of considering the discrimination of the flow patterns and try to use a single model to monitor them all. In this paper, we model the flow prediction problem from a new perspective and intend to design a set of experts specialized in different kinds of flow patterns.

2.3 Data Mining in Human Mobility

Recently, a branch of works focus on mining the inherent patterns from large-scale human mobility datasets. A series of approaches have been proposed for geographical topics (like beach, hiking, and sunset) [6], city functional region (e.g., areas of historic interests, diplomatic and embassy areas) [5], human trip purpose (e.g., shopping, work, eating out) [41], space-time structure [42]. Other part of works concentrates on using tensor factorization to explore the urban mobility patterns. The Non-negative Tensor Factorization (NTF) is utilized [8] to decompose the human flow tensor into several basic life patterns. The probabilistic tensor factorization framework [43] is introduced to discover the travel behavior and spatial configuration of the city in Singapore. A regularized non-negative Tucker decomposition approach is applied [7] to reveal the principal patterns from traffic networks evolving over time. These researches demonstrated the mixed state of human mobility and provide good theoretical motivation for using the MoE method, which is skilled in the handling of multiple sources of data. To the best of our knowledge, we are the first to extend the idea of decomposing the mixed-state flow tensor into the flow prediction.
3 PROBLEM FORMULATION

As Section 2.3 described, crowd flow at a citywide level is in a mixed state consisting of several basic patterns (e.g., commuting, working, and commercial). Therefore, a natural thought would be to disentangle the patterns and study them separately before merging them to give the final prediction. This section mainly describes some primary definitions of flows and flow prediction problems, as well as our idea to disentangle the mixed flow pattern into basic ones using the mixture of experts structure. Note that the detailed implementation of our ST-ExpertNet framework is in Section 4 and here we only express the idea of disentangling in an abstract manner. For better understanding, the symbols used through this paper is summerized in Table 1.

**Definition 3.1.** Region [37]. A city is partitioned into $h \times w$ equal-size grids and each grid is called a region. We use $(i, j)$ to represent a region that lies at the $i$th row and $j$th column of the city.

**Definition 3.2.** Functional Region [5]. Assuming the city is partitioned into equal-size grids and each grid is called a region. Then a functional region refers to a set of regions that share the same specific social or economic attribute (e.g., developed commercial areas, education, and science areas and parks). In this paper, we assume that at a given time interval, the dominant flow pattern of regions belonging to the same functional region is the same.

**Definition 3.3.** Inflow and outflow of regions [37]. Let $\mathbb{P}$ be the set of trajectories at the $t$th time interval. For a region $(i, j)$, the inflow and outflow at time interval $t$ can be defined as:

\[
\begin{align*}
\dot{g}_t^{(i,j)} &= \sum_{T \in \mathbb{P}} \{ k > 1 \mid v_{k-1} \notin (i,j) \land v_k \in (i,j) \} \\
\dot{g}_t^{out,(i,j)} &= \sum_{T \in \mathbb{P}} \{ k \geq 1 \mid v_k \in (i,j) \land v_{k+1} \notin (i,j) \}
\end{align*}
\]

where, $T_r \colon v_1 \rightarrow v_2 \rightarrow \cdots \rightarrow v_{|T_r|}$ is a single trajectory in $\mathbb{P}$; $v_k \in (i,j)$ is a geospatial coordinate within a region $(i,j)$ and $|\cdot|$ denotes cardinality of a set.

**Definition 3.4.** Flow prediction. Given the historical observations $X_t = \{g_{te} \mid t' \in [0, n-1]\}$, predict future flow $\hat{Y}_{t+1}$.

\[
\Phi : X_t \rightarrow \hat{Y}_{t+1}
\]

where $\Phi$ as neural network and $n$ denotes the length of the input time interval, $X_t \in \mathbb{R}^{2h \times w}$ and $h, w$ denote the height and width of the city of grids.

The Mixture of Experts (MoE) is a combination technique suitable for handling questions when the problem can be divided into a number of subtasks. Different from general neural networks, MoE separates and trains multiple expert models corresponding to each sub-task, and a gating module is used to assign weights to experts indicating their importances. The actual output of the model is the combination of the output of each expert, multiplying the weight from the gating module.

In this case, we apply the idea of MoE to our question and feed the history flow tensor $X_t$ to all the experts and the gating network simultaneously. The gating network will then assign weights to each expert, helping to ensure its localization at a certain flow pattern. As a result, the mixed crowd flow tensors are automatically divided and charged by different experts. Fig. 2 presents the idea with a graph. Formally, let the history flow tensor $X_t = \{g_{te} \mid t' \in [t-n+1, t]\}$ and the set of expert models $\Phi = \{E_1, E_2, \ldots, E_k\}$, the formulation of the output can be written as:

\[
\hat{Y}_{t+1} = \sum_{E_i \in \Phi} W_i \odot E_i(X_t),
\]

where $\odot$ denotes an element-wise product, $\hat{Y}_{t+1}$ denotes the future prediction flow tensor, $W_i \in \mathbb{R}^{2h \times w}$ is the weight of each model $E_i$ assigned by the gating network. Therefore, let $Y_{t+1}$ denote the ground-true future flow tensor, our first optimization goal can be written as:

\[
\min ||Y_{t+1} - \sum_{E_i \in \Phi} W_i \odot E_i(X_t)||^2.
\]

Since different flow patterns acts differently, similar output for each expert is unwanted. For instance, the working
pattern focuses on the peak of morning and evening on weekdays, but the entertaining pattern mostly concentrated on the weekends. Therefore, to force the knowledge learned by each expert to be different, this paper tries to maximize the discrimination for each model’s final output \( W_iE_t(X_t) \), which can be formulated as

\[
\max \sum_{E_i \in \Phi} \sum_{E_j \in \Phi} D_{eid}(W_iE_t(X_t), W_jE_t(X_t)), \quad E_i \neq E_j, \quad \forall E_i, E_j,
\]

where \( D_{eid} \) denotes the distance metric between a pairwise model’s final output. Moreover, to improve the interpretability and robustness of the experts, we hope each expert can be “responsible” for its own target pattern. The scenario is unwelcome when the gating network assigns the entertaining pattern to an expert but the expert itself tries to learn from the commuting pattern. We try to ensure this consistency between experts and the gating network by optimizing the following target which can be formulated as:

\[
\min \sum_{E_i \in \Phi} W_i D_{er}(Y_{t+1} - E_t(X_t)),
\]

where \( D_{er} \) denotes the distance between expert output and the future ground truth flow.

To sum up, our final optimization goals are threefold and described as Equations (3), (4) and (5). The detailed implementation of the optimization process is demonstrated in the next section.

4 METHODOLOGY

In this section, we introduce the architecture of ST-ExpertNet, which is the framework based on MoE and aims to solve the optimization target Equations (3), (4) and (5) in Section 3.

4.1 Overview

We here systematically summarized our ST-ExperNet framework shown in Fig. 3. The input of the expert network and gating network is composed of four parts: the historical observation of week, day, hour, and external information, which indicates the recent time intervals, daily periodicity, weekly trend, and metadata (e.g., weather conditions and events) respectively. While different expert networks produce their output, the spatial gating network further localizes them to their responsible patterns via the attention mechanism. The localized outputs of each expert are then aggregated together before the nonlinear transformation of Tanh. Inspired by PixelCNN [44], which uses a gated activation unit to control the magnitude of the output, we introduce the temporal gating network to do the same thing. The output of the temporal gating network will pass through a Sigmoid function before doing an element-wise multiplication with the aggregated experts’ output. There are two regularization strategies shown in the figure: Expert Responsibility Loss and Expert Inter Discrepancy Loss. The former intends to force the experts to take responsibility for their sub-tasks by calculating the loss between their sub-task prediction (e.g., the prediction of commuting pattern) and the ground truth of the sub-task, and the latter aims to reduce the overlap of the responsible sub-tasks for every pair of experts. We organize the framework description as follows: The ST-ExpertNet input is introduced in Section 4.2. The detailed implementation of ST-ExpertNet and its regularization are shown in Sections 4.3 and 4.4 respectively. Finally, we describe the training steps of our framework in Section 4.5.

4.2 Multi-Source Information Fusion

Fig. 3 shows the input of ST-ExpertNet, which consists of four main components that we adopt from [1]: closeness, period, trend, and external influence denoted as \( X_t^c, X_t^p, X_t^t, X_t^e \) respectively. However, unlike traditional operations in [1], [2], [3], [4], which feed these components into different models and hinder a comprehensive understanding of property of the flow patterns, our goal is to provide enough information for the experts to recognize the regional inter-discrepancy (e.g., if regions (1,3) and (2,4) are highly relevant in the commuting functional region). Formally, given the set of functional regions \( \Omega = \{ \Omega_1, \Omega_2, \ldots, \Omega_n \} \), where each region \( R_i \in \Omega \) and \( \Omega_i \in \Omega \) (e.g., \( \Omega_i \) can be a functional region like office area), we want each part of our model to have the ability to mine the pattern differences between functional regions \( \Omega_1 \) and \( \Omega_j \) or regions \( R_i \) and \( R_j \), under rich input information. To achieve this goal, we concatenate closeness, period, trend, and external information as an entirety and feed into the experts and gating networks, which can give a comprehensive and distinct feature space for each region. The FC layer shown in Fig. 3 denotes the multilayer perceptron, which intends to process external information (e.g., weather condition). After reshaping the FC’s output as \( X_t^{\text{FC}} \in \mathbb{R}^{n_w \times 1 \times 1 \times W} \), where \( n_w \) is the number of weather information, such as sunny, rainy, cloudy, we have the final ST-ExpertNet input \( X_t \) as:

\[
\begin{align*}
X_t^c &= \{ g_t \mid t' \in [t - (n_q + 1) \times q, t - q] \} \\
X_t^p &= \{ g_t \mid t' \in [t - (n_p + 1) \times p, t - p] \} \\
X_t^t &= \{ g_t \mid t' \in [t - (n_e + 1), t] \} \\
X_t &= \text{Concat}([X_t^{\text{FC}}, X_t^c, X_t^p, X_t^t])
\end{align*}
\]

Fig. 3. The framework of our ST-ExpertNet.
where $n_T$, $n_p$, $n_c$ represent the length of time slots of Trend, Period, and Closeness, and $q, p$ are the length of one week and day respectively. For example, suppose $\Delta t = 30$ min, then we have, $q = 24h/0.5h = 48$ and $p = q \times 7$. Here $\text{Concat}$ denotes the channel-wise concatenate operator.

### 4.3 ST-ExpertNet

The structure of our ST-ExpertNet consists mainly of three parts: the Expert Layer, Spatial Gating Network, and Temporal Gating Network. In the following part, we will introduce them one by one.

**The Expert Layer.** According to Section 3, our goal is to build a set of expert models, where each expert can take charge of a distinctive flow pattern within the flow tensor. This without any prior consultation holds the same view as the motivation of Mixture of Experts (MoE), which involves decomposing predictive tasks into several subtasks, training the expert models corresponding to each subtask, developing a gating model that learns which expert to trust based on the input variables, and combining all the predictions from each expert together. The original Mixture-of-Experts (MoE) [12] can be formulated as:

$$y = \sum_{i=1}^{n} G(x)_i \odot E(x)_i,$$

where $x$ denotes the input of MoE and $G(x)_i$, representing the $i^{th}$ logit of the output of $G(x)$, indicates the proportional contribution of expert $E_i$. Note that $\sum_{i=1}^{n} G(x)_i = 1$. Here, $E_i, i = 1, \ldots, n$ are the $n$ expert networks and $G$ represents a gating network that helps to assign weights to the results of all experts. More specifically, the gating network $G$ produces a distribution over the $n$ experts based on the input and assigns a weight $G(x)_i$ to the output of the $i^{th}$ expert. The weight can also be interpreted as the prior probability that the expert $i$ could generate the desired prediction. The final output of the model is a weighted sum of the outputs of all experts.

Subsequently, the spatial and temporal gating network will be introduced. Similar to the structure of MoE, the ST-ExpertNet also consists of a set of $K$ "expert networks", $E_1, \ldots, E_K$, and two "gating networks" $G_s, G_t$. $G_s$ is to control the spatial attention and helps disentangle the mixed flow pattern for experts to localize and it is formally named as **Spatial Gating Network**. On the other hand, $G_t$ is used to adjust the expert temporal signal in a global perspective and formally named as **Temporal Gating Network**. Fig. 3 shows an overview of the ST-ExpertNet architecture. The experts are themselves neural networks, each with their own parameters.

**Spatial Gating Network.** Different from Equation (6) which generates and sets $G_s$ as simple weighting parameters of experts, however, the objective of the spatial gating network $G_s$ is to adjust the prediction of experts in spatial aspect. It can automatically disentangle mixed crowd flows into basic patterns and identify the current dominant pattern of each region. Then, different regions will be assigned to different specialized experts through the attention mechanism according to their dominant flow patterns. In our design, we try to avoid the ambiguous scenario in which expert $i$ outputs tiny values for region $a$, which means that it does not charge this region, but receives high attention values from $G_s$. To comprehensively handle the output of $G_s$ and experts, this paper deploys the attention model by combining $G_s$ and $E$ here to ensure the consistency of the gating and expert network, which can be written as

$$a_i = \frac{\exp(G_s(x)_i \odot E(x)_i)}{\exp(\sum_{i=1}^{n} G_s(x)_i \odot E(x)_i)}, \quad i = 1 \cdots n,$$

where $\exp$ denotes the exponential operation. Therefore, by adopting attention, the spatial gating network final output can be written as

$$e_i = a_i E(x)_i,$$

and $e_i$ presents the final output of expert $E_i$.

**Temporal Gating Network.** The task of crowd flow prediction is a problem of both spatial and temporal. Time is also an important factor that affects the flow of different patterns. For instance, on weekdays, the morning outflow of the working pattern in a residential area is significantly larger than that on weekends. However, the expert network, due to its self-characteristics, cannot always perceive this change and is likely to also produce high values for the morning outflow of working pattern on weekends. Thus, we consider it to be essential adding a module to help expert networks capture and make a smooth transition over these temporal changes. Inspired by PixelCNN [44] which uses a gated activation unit to control the final time series signal, we here design the temporal gating network $G_t$ to modify the amplitude of time series output of $y_t$. To sum up, the final output of ST-ExpertNet can be written as follows:

$$\hat{y}_{t+1} = \text{Tanh}\left(\sum_{i=1}^{K} e_i \odot \sigma(G_t(x)_i)\right),$$

where $\odot$ denotes an element-wise multiplication operator. The $\text{Tanh}$ localizes the input between the range of $(-1, 1)$ and $\sigma$ denotes the Sigmoid function to limit $G_t(x)_i$’s value into $(0, 1)$. Note that Equation (9) is only the general form of ST-ExpertNet’s output. For multichannel cases, for example, input with inflow and outflow channels, Equation (9) will act on them independently.

### 4.4 Experts’ Regularization

Considering the aforementioned criteria of sparsity and preciseness, here we introduce two regularization strategies: **Expert Responsibility Loss** and **Expert Inter Discrepancy Loss**. The former intends to force the experts taking responsibility for their own prediction, and the latter aims to drive the experts learning different flow patterns. **Expert Responsibility Loss.** An important issue of MoE is that we need to ensure the localization of each expert in their own subtasks. We try to avoid the scenario that $G_s$ assigns the entertaining area which is now hold by entertaining pattern to expert $i$, but expert $i$ refuses to focus on it and try to produce high value in the office area. This will lead to a great inconsistency between the expert’s output and the attention mechanism. To solve this problem, a punitive strategy named experts responsibility loss has been proposed to restrict and control the high confidence between $G_s(x)$, and $E(x)_i$, and experts should assume responsibility.
for their own behavior. We set up expert responsibility loss as $L_{er}$ intending to minimize the gap between expert output and their responsible ground-true flow. Suppose we have $K$ experts and their output $E(x)$, the general version of the error function is

$$L_{er} = \sum_{i=1}^{K} a_i(Y_{t+1} - H_i)^2,$$  \hspace{1cm} (10)

where $Y_{t+1}$ denotes the ground truth flow tensor, and $H_i = \sigma(G_i(x)) \odot \text{Tanh} (E_i(X_i))$. In this error function, the weights of each expert are updated based on their own error to the prediction target and thus, decouples from the influence of other experts. If we use a gradient descent method to train the network with Equation (10), the network will tend to dedicate a single expert to each basic flow pattern. Equation (10) works in practice, however, according to [12], if we further introduce negative log probability and mixture of Gaussian model into the function, it might give better performance. The new formulation of $L_{er}$ is defined as:

$$L_{er} = -\log \sum_{i=1}^{K} a_i \exp \left(-\frac{||Y_{t+1} - H_i||^2}{2}\right).$$  \hspace{1cm} (11)

To evaluate these two functions, we start by analyzing their deviations with respect to the $i$-th expert. From Equation (10), we get

$$\frac{\partial L_{er}}{\partial E_i(X_i)} = -2a_iH'_i(Y_{t+1} - H_i),$$  \hspace{1cm} (12)

where $H'_i = \sigma(G_i(x)) \odot (1 - \text{Tanh}^2(E_i(X_i)))$, while from Equation (11), we deduce

$$\frac{\partial L_{er}}{\partial E_i(X_i)} = -H'_i \left[ \frac{a_i e^{-\frac{||Y_{t+1} - H_i||^2}{2}}}{\sum_j a_j e^{-\frac{||Y_{t+1} - H_j||^2}{2}}} \right] (Y_{t+1} - H_i).$$  \hspace{1cm} (13)

It is obvious from the deviations that both functions update an expert’s weights based on their individual error. However, in Equation (12), we use term $a_i$ to be the weight update factor of each expert, while in Equation (13), the weighting term further considers the ratio of an expert’s error value to the total error. This unique feature allows network trained with the new error function to find the most suitable expert for a specific subtask more quickly, especially in early training stages [17]. Therefore, in this paper, we choose Equation (11) to be the Expert Responsibility Loss.

Experts Inter Discrepancy Loss. Dividing the original task into several subtasks and localizing each expert to their corresponding subspaces are the key idea of MoE. Previous work has also shown that promoting the diversity of experts can improve the performance of the network[45]. Thus, we intend to punish the experts’ inter discrepancies and try forcing the experts to learn from non-overlapping problem spaces. Inspired by the mathematical theory of the determinant point process[46] and the design of the ADP regularizer [13], we proposed Experts Inter Discrepancy Loss to increase the gap between experts. The loss can be defined as

$$L_{red} = -\det (V^TV).$$  \hspace{1cm} (14)

According to the variable definition of Equations (7) and (8), we can unfold $V$ as

$$V = [\bar{g}_1\bar{e}_1, \bar{g}_2\bar{e}_2, \ldots, \bar{g}_n\bar{e}_n].$$  \hspace{1cm} (15)

Here $\bar{g}_i$ represents the mean value of $i$-th $G_i(x)$ and $\bar{e}_i$ refers to the flattened one-dimensional vector of the product of $i$-th $G_i(x)$ and $i$-th $E(x)$ under $L_2$ normalization. For sparsity, we only choose the experts among the top $n$ of attention values.

It can be noticed that different from the ADP regularizer [13] where all the columns of $V$ should be normalized vectors, we make an improvement by including the mean value of the corresponding attention to make each column vector has its own “length”. This design further ensures the sparsity of experts and is more suitable for our model because we hope that each expert can have a similar mean value of attention, which means that every expert has its own subtask instead of leaving a few experts with nothing to do.

The theory about matrix [47] has stated that if $A \in \mathbb{R}^{n \times m}$ is a matrix and $\text{rank}(A) = m$, we have

$$\text{volume}(\delta) = |\det(V^TV)|^{1/2},$$  \hspace{1cm} (16)

from which $\delta$ denotes the geometrical body spanned by $V$’s column vector set. Therefore, we can have a geometrical illustration of the expert inter discrepancy loss at Fig. 4. The volume spanned by the vector set indicates the divergence between experts. We intend to maximize the volume to force each expert to pay attention to different flow patterns of the city. The histogram in Fig. 4 shows that before we punish the expert discrepancy, different experts might make nearly equal contributions to the same flow pattern. However, after applying the punishment, each expert now has its own pattern to charge, and this agrees with our initial motivation.

4.5 Training Steps

To sum up the previous discussion in Sections 3 and 4, we solve the problem in Equation (2) by proposing the ST-ExpertNet with $K$ experts and 2 gating networks from spatial and temporal aspects, solve the problem in Equation (4) by proposing Experts Responsibility Loss, and solve the problem in Equation (5) by proposing Experts Inter Discrepancy Loss. We need to train $K$ expert models and two gating networks, which take input from multisource information $X_t$. Our ST-ExpertNet can be trained to predict $Y_{t+1}$ by minimizing the
mean square error between the predicted flow tensor and the ground truth flow tensor. The total loss function is written as

$$\mathcal{L}(\theta) = (1 - \lambda_{er} - \lambda_{eid}) \| Y_{t+1}^i - \hat{Y}_{t+1}^i \|^2 + \lambda_{er} L_{er} + \lambda_{eid} L_{eid},$$

where $\theta$ are all the learnable parameters in ST-ExpertNet, $L_{er}$ and $L_{eid}$ are simply designed as the same structure of pure CNN-ExpertNet, please see supplementary materials, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TKDE.2022.3196936.

For all different architectures of ST-ExpertNet, the $G_s$ and $G_i$ are simply designed as the same structure of pure CNN with 64 filters of 3 × 3 kernel.

5.3 Methods for Comparison

We compared our model with the following methods and tuned the parameters for all the methods. Then we report the best performances in the validation set. Additionally, to test ST-ExpertNet’s general applicability, we apply the architect of CNN, ConvLSTM, and ST-ResNet into our model. Here we follow the experiment setting on the benchmark work [48] in traffic flow prediction, and the detailed implements for each ST-ExpertNet are listed as follows:

- Historical average (HA): Historical average predicts the future flow by averaging the value of previous flow at the same region in the same relative time interval.
- Pure CNN. It is a basic and popular deep learning baseline in flow prediction constructed with three CNN layers with 64 filters of the 3 × 3 kernel. The Batch Normalization operation are added between two consecutive layers. And the ReLU operation is set at the last layer.
- ConvLSTM. The convolutional LSTM [49] extends the fully connected LSTM (FC-LSTM) [50] and maintains the advantage of FC-LSTM. Further, the convolutional operations staid in input-to-state and state-to-state transitions provide opportunities to capture both spatial and temporal dependency. The ConvLSTM layers maintain 32 filters of 3×3 kernel size. Four ConvLSTM layers are used and the ReLU operation is set in the final output.
- ST-ResNet [1]: ST-ResNet extends the ResNet [38] into the framework of traffic prediction. Moreover, the three blocks of ResNet have been designed to process the hour, day, and week patterns, respectively. Each block in ST-ResNet owns 3 residual units with the 32 filters of 3×3 kernels.
- DMVST-Net. DMVST-Net [15] combines spatial and temporal information using local CNN and LSTM. The local CNN only captures spatial feature among nearby grids and LSTM only processes the recent time intervals.
- PCRN. Periodic-CRN [4] focuses on capturing the periodic patterns in spatio-temporal data by using the ConvGRU model [51]. The pyramidal architecture created by stacking three convolutional RNN layers has been proposed to learn the spatial and temporal features simultaneously.

5.4 Preprocessing

Here, we scale all the value of flow datasets into [0,1] using the Min-Max normalization method. For MSE and RMSE, the
values are re-scaled back to the normal values. Then, we use the one-hot coding method to encode external factors of discrete value in Beijing, such as DayOfWeek, and Weekend/Weekday into binary vectors. For other continuous values in Beijing, like temperature and wind speed, we also use the Min-Max normalization method. Finally, external factors in New York are set to be holidays and time information.

5.5 Hyperparameters
For all methods with convolution filters, we set the kernel size to be $3 \times 3$. For the length of input $X^c$, $X^p$ and $X^{tr}$, we follow the classical setting $X^c = 3$, $X^p = 1$, and $X^{tr} = 1$, which is demonstrated to be a good choice in common datasets [1], [52]. There are three additional hyperparameters: the number of experts $K$, the penalty terms $\lambda_c$ and $\lambda_r$ for $L_{eid}$ and $L_{tr}$, respectively. We performed a grid search for these three hyperparameters regarding different datasets in Section 5.7, and choose the best combination to use for experiments. We select 80% of the data as training data and 20% of the training data as a validation set, which help to early-stop the model based on the best validation score. The batch size is set as 32 and Adam [53] is used for optimization with learning rate $10^{-3}$.

5.6 Result Analysis
We first compare our model with seven other models on TaxiBJ, TaxiNYC, BikeNYC-I, and BikeNYC-II, as shown in Tables 2 and 3. We give three variants of ST-ExpertNet by adopting different architectures: CNN, ConvLSTM and ST-ResNet, which are often deployed in crowd flow prediction task. The results in Tables 2 and 3 show that these baseline models have been greatly improved by adapting them into ST-ExpertNet framework. By observing during experiments, we find that for weak base learners like CNN, the network can be set with more experts, and each weak learner can learn a basic pattern from input, which may be easier to train and achieve higher improvements than using the same number of complicated learners (might cause overfitting). Further, we discover that datasets will also have influences on different architectures of ST-ExpertNet. For example, CNN-ExpertNet gets 137%, 144%, and 304% improved in NYC, but only gets 62% in TaxiBJ. We consider the reason is that the time span in TaxiBJ is large and the relationship between regions and patterns processes higher nonlinearity, which may be more unpredictable for weak learners. On the contrary, the TaxiNYC and BikeNYC datasets only contains data in less than three months, which may be suitable to run simple experts on. For ConvLSTM and ST-ResNet in ST-ExpertNet, they tend to get satisfying performances with proper choice of numbers of experts. For more detailed information on model running time and parameter scales, please refer to supplementary materials, available online.

5.7 Parameter Analysis
Statistical Analysis. To statistically analyze the discrepancy between experts, the Quade test [54] is used, which is a non-parametric test to see if $k$ experiment outcomes have

### Table 2: Effectiveness Evaluation on TaxiBJ and TaxiNYC

| Model                  | TaxiBJ          | TaxiNYC         |
|------------------------|-----------------|-----------------|
| HistoricalAverage      | 2025.328        | 463.763         |
| CopyYesterday          | 1998.375        | 1286.035        |
| CNN                    | 554.615         | 280.262         |
| ConvLSTM               | 370.448         | 147.447         |
| ST-ResNet              | 349.754         | 133.479         |
| DMVST-Net              | 415.739         | 185.601         |
| PCRN                   | 355.511         | 181.091         |
| CNN-ExpertNet          | 342.737         | 114.717         |
| ConvLSTM-ExpertNet     | 328.250         | 109.445         |
| ST-Res-ExpertNet       | 330.258         | 111.445         |

### Table 3: Effectiveness Evaluation on BikeNYC-I and BikeNYC-II

| Model                  | BikeNYC-I      | BikeNYC-II      |
|------------------------|----------------|-----------------|
| HistoricalAverage      | 245.743        | 23.757          |
| CopyYesterday          | 241.681        | 54.054          |
| CNN                    | 145.549        | 20.351          |
| ConvLSTM               | 43.765         | 10.076          |
| ST-ResNet              | 37.279         | 10.182          |
| DMVST-Net              | 63.849         | 12.397          |
| PCRN                   | 130.878        | 10.893          |
| CNN-ExpertNet          | 36.068         | 8.577           |
| ConvLSTM-ExpertNet     | 35.338         | 8.574           |
| ST-Res-ExpertNet       | 36.722         | 8.806           |

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
identical effects. The null hypothesis of the test assumes an identical distribution for all test cases. In our paper, we set \( k = 2 \) and compute the pairwise p-value for all the experts. Formally, given two expert outputs \( Y_1, Y_2 \in \mathbb{R}^{n \times 2 \times h \times w} \), where \( n \) denotes the number of samples (batch size), we calculate the p-value between two experts as

\[
p - \text{value} = \frac{1}{2 \times h \times w} \sum_{c=1}^{2} \sum_{i=0}^{h} \sum_{j=0}^{w} \text{Quade}(Y_{1,c,h,w}, Y_{2,c,h,w}).
\]

The Quade function outputs the significance level of the Quade test and \( c \) denotes the index of in-out flow channel. The final p-value is calculated as the arithmetic mean of significance levels from two experts’ regional outputs, and a smaller p-value indicates higher variance between the two experts. Fig. 6 shows the pairwise p-values of the test results for all 10 experts in CNN-ExpertNet under \( \lambda_{vid} = 0 \) and \( \lambda_{vid} = 0.1 \). It is obvious that pairwise p-values are much lower after using Experts Inter Discrepancy Loss by setting \( \lambda_{vid} = 0.1 \). It is also noted that the max pairwise p-value under \( \lambda_{vid} = 0.1 \) is less than 0.05, indicating it’s statistically significant to show the diversity of experts.

**Ablation Study.** We also conduct the ablation experiments on TaxiBJ and TaxiNYC with \( G_s \) and \( G_t \) being masked separately. Removing \( G_s \) and using the expert’s output directly to deploy self-attention in the model can mask \( G_s \). For masking \( G_t \), simply throwing away \( G_t \) will be fine. The experiment includes CNN, ResNet and ConvLSTM as base experts and the result is shown in Table 4 with comparison to the original models. We can observe that \( G_s \) and \( G_t \) all play a crucial role in both datasets and improve performance significantly.

**Parameters Search.** We examine the sensitivities of three important hyperparameters: the number of experts \( K \), the punishment intensity \( \lambda_{er} \) and \( \lambda_{vid} \) in CNN-ExpertNet trained on four datasets. Each experiment will repeat three times, and the mean and variance have been calculated, and depicted in Fig. 9 as the format of bar plot. The best results are marked in red color. For the sake of fairness, the experiments of searching optimal \( K \) will first be run, then, the optimal \( K \) will be set as the default number of expert in the

| Model                                  | TaxiBJ  | TaxiNYC |
|-----------------------------------------|---------|---------|
| CNN-ExpertNet                           | 342.737 | 114.717 |
| CNN-ExpertNet—\( G_s \)                 | 470.824 | 131.002 |
| CNN-ExpertNet—\( G_t \)                 | 444.237 | 153.528 |
| ST-Res-ExpertNet                        | 330.258 | 111.445 |
| ST-Res-ExpertNet—\( G_s \)              | 357.201 | 126.173 |
| ST-Res-ExpertNet—\( G_t \)              | 343.534 | 155.113 |
| ConvLSTM-ExpertNet                      | 329.250 | 109.445 |
| ConvLSTM-ExpertNet—\( G_s \)            | 344.776 | 110.318 |
| ConvLSTM-ExpertNet—\( G_t \)            | 331.034 | 117.772 |

(\( = \) denotes the mask operation).
experiments of $\lambda_{cr}$ and $\lambda_{eid}$. We briefly summarize several conclusions from Fig. 9:

- $K$: The number of experts is consistent with the complexity of the dataset. For example, we observe that the order of estimation error for all baselines in Tables 2 and 3 is TaxiBJ > TaxiNYC > BikeNYC-I > BikeNYC-II, and the optimal $K$ are 9, 7, 4, 2 respectively. Therefore, when the dataset is hard to fit, increasing the number of experts is encouraged.

- $\lambda_{cr}$: Fig. 9 shows that in TaxiNYC, BikeNYC-I, and BikeNYC-II, the best choice of $\lambda_{cr}$ is 0.2, while in TaxiBJ it rises to 0.4. We think this also correlates with the complexity of dataset and it is easy to understand since more complex data requires more regularization during training. Aside from providing more accuracy, $\lambda_{cr}$ also helps in adding to the interpretability by eliminating the expert's output in unassigned regions.

- $\lambda_{eid}$: Intuitively, the more experts we use, there can be more overlaps between experts naturally and thus requires more punishment in discrepancies. We can observe that in TaxiBJ TaxiNYC, BikeNYC-I, BikeNYC-II, the optimal $K$ are 9, 7, 4, 2 and the optimal $\lambda_{eid}$ are 0.3, 0.3, 0.1, 0.1. A noteworthy point is that a proper choice of $\lambda_{eid}$ is important, otherwise the model performances will vary a lot, because Expert Inter Discrepancy Loss itself does not drive the model to be more accurate. So it might corrupt training if used improperly.

### 5.8 Case Study With Gating Network $G_s$ in TaxiBJ

We evaluate $G_s$'s spatial self-adaption and interpretability by visualizing the attention values for each expert $E_i$. The spatial attention values of CNN-ExpertNet at 8:00 a.m. are computed and we choose the top six experts’ attention values to show on Fig. 5. We can observe that the attention of each expert varies spatially. For instance, expert 5 concentrates on the commuting pattern of Beijing’s 2nd, 3rd, and 4th ring roads, China’s oldest fully enclosed, full interchange, no-traffic-light urban express ring roads. On the other hand, expert 2 is responsible for the 4th and 5th ring roads in Beijing. Furthermore, expert 3 takes charge of the residential area in the Haidian and Fengtai districts, which

![Fig. 7. Visualization of the attention for different experts in TaxiBJ.](image)

![Fig. 8. The ST-ExpertNet $G_s$ attention visualization with temporal aspect. A. The residential area outflow changes within one day in Beijing. B. The office area expert attention values. C. The residential area expert attention values. D. The office area inflow changes within one day in Beijing.](image)
contain multiple kinds of places of interest, such as Summer Palace, Xiangshan Park, Beijing Garden Expo Park, etc. Expert 4 mainly charges the Capital Airport Expressway, an expressway connecting downtown Beijing and Beijing Capital International Airport, and contains huge flow values throughout the day. Since multiple patterns may exist in one region simultaneously, expert 1 co-manages the flow state with experts 2 and 5 in some places. Expert 6 mainly controls the suburban area in the southwest area of Beijing, where many villages and towns exist, and the model considers this area’s flow pattern is different from others. Therefore, we can conclude that ST-ExpertNet successfully learns the different flow patterns in our city. A more detailed and vivid visualization is shown in Fig. 7.

We also select two types of functional regions from TaxiBJ. One is the office area, and the other is the residential area. There are two conspicuous patterns for the office and residential areas. The inflow pattern for the previous one has morning and evening peaks every day. On the contrary, people often leave their homes and go to work on workdays. Therefore, the outflow of residential in peak hours tends to be higher than other timestamps. The Figs. 8 A and 8 B present the change of flows at coordinate (2,31) and (11,23) on 5/20/2015, respectively. It can be seen that in different time periods, the regional traffic flow is mainly charged by a certain expert. Fig. 8 C presents the experts’ attention values for office area.

We can observe that the duty of $E_3$ is mainly to handle the flow pattern for the midnight, and the morning and evening peak flow patterns are mainly managed by $E_5$. In Fig. 8 D, the residential area pattern is mainly charged by $E_1$ and $E_3$. The $E_1$ controls the midnight patterns, and the $E_3$ pays attention to the morning peak patterns.

### 5.9 Case Study With Gating Network $G_t$ in TaxiNYC

We investigate whether $G_t$ can be extended to learn meaningful temporal patterns and modify the amplitude of time series output in the real-world large scalar dataset. Therefore, the ablation study has been conducted on the TaxiNYC dataset to demonstrate that $G_t$ helps adjust the expert temporal signal. Fig. 10 shows an example of the predictions with ST-ExpertNet with $G_t$ and without $G_t$. The ground truth data of the figure is from the inflow of TaxiNYC at coordinate (0,2), where an office area locates. It can be viewed from the graph that on weekdays the inflow peaks in the morning and approaches nearly zero at night, whereas on weekends, the inflow has a much lower peak in the morning and reaches almost zero in the evening. However, we can see from the results that the network without $G_t$ tends to have unstable predictions, especially in the morning and late night. Besides, it can’t handle the temporal information well because every weekend, when there is a sudden shift of predictive pattern from weekdays to weekends, it will have trouble dealing with the late-night prediction. As a comparison, the predictive result of the network with $G_t$ is much more satisfying. The Gating Network $G_t$ learns the temporal pattern and helps the network adjust its output based on the input data, which consists of historical information. When it comes to weekends, $G_t$ will help the network switch to the “predictive weekend pattern” to gain a more accurate performance.

### 6 Conclusion

This paper introduced a new deep learning-based framework for city-wide crowd flow prediction using several...
experts based on historical spatial-temporal data, weather, and holiday events. The ST-ExpertNet can decompose the crowd flow tensor into several patterns (e.g., working, entertaining, commuting, commercial, etc.) and localizes different experts into different patterns. The final output of the framework is a weighted sum of all the experts via the attention mechanism, which can be visualized and, to some extent, increases the framework’s interpretability. Furthermore, experts in our framework can be set as any suitable model by users, such as CNN, ResNet, and ConvLSTM. We comprehensively investigate our framework’s performance on four crowd flows dataset in Beijing and NYC compared with the other seven baseline methods. The experiment result demonstrates that ST-ExpertNet can achieve better predictive accuracy on these datasets and is more applicable to crowd flow prediction.

As a limitation of this study, state-of-the-art models are hard to be directly employed into the ST-ExpertNet framework due to their complex architectures, and straightforward models are encouraged to be used as experts for basic flow patterns. In future work, we might consider further refining our framework to reduce its complexity and the number of hyperparameters. And experts can be designed specifically for certain special days or places (e.g., mayday or neighborhood of universities). Man-made rules can also be considered to cooperate with experts in the learning process. Another interesting direction is to view the traffic flow as a time series with trends, seasonality, and residual components, and build our expert models accordingly.

ACKNOWLEDGMENTS
Hongjun Wang and Jiuyuan Chen contribute equally to this work and serve as co-first author.

REFERENCES
[1] J. Zhang, Y. Zheng, and D. Qi, “Deep spatio-temporal residual networks for citywide crowd flows prediction,” in Proc. 31st AAAI Conf. Artif. Intell., 2017, pp. 1655–1661.
[2] H. Yao, X. Tang, H. Wei, G. Zheng, and Z. Li, “Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction,” in Proc. AAAI Conf. Artif. Intell., 2019, pp. 5668–5675.
[3] Z. Lin, J. Feng, Z. Lu, Y. Li, and D. Jin, “DeepSTN+: Context-aware spatio-temporal neural network for crowd flow prediction in metropolitan,” in Proc. AAAI Conf. Artif. Intell., 2019, pp. 1020–1027.
[4] A. Zonoozi, J.-J. Kim, X.-L. Li, and G. Cong, “Periodic-CN: A convolutional recurrent model for crowd density prediction with recurring periodic patterns,” in Proc. 27th Int. Joint Conf. Artif. Intell., 2018, pp. 3732–3738.
[5] J. Yuan, Y. Zheng, and X. Xie, “Discovering regions of different functions in a city using human mobility and POIs,” in Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2012, pp. 186–194.
[6] Z. Yin, L. Cao, J. Han, C. Zhai, and T. Huang, “Geographical topic discovery and comparison,” in Proc. 20th Int. Conf. World Wide Web, 2011, pp. 247–256.
[7] J. Wang, F. Gao, P. Cui, C. Li, and Z. Xiong, “Discovering urban spatio-temporal structure from time-evolving traffic networks,” in Proc. Asia-Pacific Web Conf., 2014, pp. 93–104.
[8] Z. Fan, X. Song, and R. Shibasaki, “CitySpectrum: A non-negative tensor factorization approach,” in Proc. ACM Int. Joint Conf. Persuasive Ubiquitous Comput., 2014, pp. 213–222.
[9] R. A. Jacobs, M. I. Jordan, and A. G. Barto, “Task decomposition through competition in a modular connectionist architecture: The what and where vision tasks,” Cogn. Sci., vol. 15, no. 2, pp. 219–250, 1991.
[10] R. A. Jacobs, “Bias/variance analyses of mixtures-of-experts architectures,” Neural Comput., vol. 9, no. 2, pp. 369–383, 1997.
[38] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 770–778.

[39] S. Wang, H. Miao, H. Chen, and Z. Huang, “Multi-task adversarial spatial-temporal networks for crowd flow prediction,” in Proc. ACM Int. Conf. Inf.Knowl. Manage., 2020, pp. 1555–1564.

[40] B. Du et al., “Deep irregular convolutional residual LSTM for urban traffic passenger flows prediction,” IEEE Trans. Intell. Transp. Syst., vol. 21, no. 3, pp. 972–985, Mar. 2020.

[41] X. Zhao, Z. Li, Y. Zhang, and Y. Lv, “Discover trip purposes from cellular network data with topic modelling,” IEEE Intell. Transp. Syst. Mag., vol. 14, no. 4, pp. 37–46, Jul./Aug. 2022.

[42] A. Pozdnoukhov and C. Kaiser, “Space-time dynamics of topics in streaming text,” in Proc. 3rd ACM SIGSPATIAL Int. Workshop Location-Based Social Netw., 2011, pp. 1–8.

[43] L. Sun and K. W. Axhausen, “Understanding urban mobility patterns with a probabilistic tensor factorization framework,” Transp. Res. Part B: Methodol., vol. 91, pp. 511–524, 2016.

[44] A. V. D. Oord, N. Kalchbrenner, O. Vinyals, L. Espeholt, A. Graves, and K. Kavukcuoglu, “Conditional image generation with PixelCNN decoders,” Proc. Adv. Neural Inform. Process. Syst., vol. 29, 2016.

[45] A. Krogh and J. Vedelsby, “Validation, and active learning,” in Proc. Int. Conf. Neural Inf. Process. Syst., 1995, Art. no. 231.

[46] A. Kulesza and B. Taskar, “Determinantal point processes for machine learning,” Found. Trends Mach. Learn., vol. 5, no. 2–3, pp. 123–286, 2012.

[47] D. S. Bernstein, Matrix Mathematics: Theory, Facts, and Formulas With Application to Linear Systems Theory. Princeton, NJ, USA: Princeton Univ. Press, 2005.

[48] R. Jiang et al., “DL-Traff: Survey and benchmark of deep learning models for urban traffic prediction,” in Proc. 30th ACM Int. Conf. Inf. Knowl. Manage., 2021, pp. 4515–4525.

[49] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, “Convolutional LSTM network: A machine learning approach for precipitation nowcasting,” Proc. Adv. Neural Inform. Process. Syst., vol. 29, 2015.

[50] J. Zhao, F. Deng, Y. Cai, and J. Chen, “Long short-term memoryfully connected (LSTM-FC) neural network for PM2.5 concentration prediction,” Chemosphere, vol. 220, pp. 486–492, 2019.

[51] N. Ballas, L. Yao, C. Pal, and A. Courville, “Delving deeper into convolutional networks for learning video representations,” 2015, arXiv:1511.06432.

[52] J. Zhang, Y. Zheng, D. Qi, R. Li, X. Yi, and T. Li, “Predicting citywide crowd flows using deep spatio-temporal residual networks,” Artif. Intell., vol. 259, pp. 147–166, 2018.

[53] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, arXiv:1412.6980.

[54] D. Quade, “Using weighted rankings in the analysis of complete blocks with additive block effects,” J. Amer. Statist. Assoc., vol. 74, no. 367, pp. 680–683, 1979.

Hongjun Wang received the BE degree from the Nanjing University of Posts and Telecommunications, China, in 2019. He is currently working toward the MS degree in computer science and technology with the Southern University of Science and Technology. His research interests include broadly in machine learning with urban computing, explainable AI, data mining, and data visualization.

Jiuyuan Chen is currently working toward the BS degree in computer science and technology from the Southern University of Science and Technology. His major research fields include artificial intelligence, deep learning, urban computing, and data mining.

Zipei Fan received the PhD degree in civil engineering from the University of Tokyo, Japan, in 2014 and 2017, respectively. He became project researcher and project Assistant Professor in 2017 and 2019, and he has promoted to project lecturer with the Center for Spatial Information Science, University of Tokyo, in 2020. His research interests include ubiquitous computing, machine learning, spatio-temporal data mining.

Zhiwen Zhang received the BE and MS degrees in artificial intelligence from Nankai University, China, in 2016 and 2019, respectively. From 2019, he is currently working toward the PhD degree with the Department of Socio-Cultural Environmental Studies, University of Tokyo. His current research interests include urban computing and data visualization.

Zekun Cai received the BS degree in computer science and technology from the University of Electronic Science and Technology of China, in 2018, and the master degree from the Department of Socio-Cultural Environmental Studies, University of Tokyo. His research interests include mainly focused on artificial intelligence, deep learning, ubiquitous computing, and spatio-temporal data analysis.

Xuan Song received the PhD degree in signal and information processing from Peking University, in 2010. In 2017, he was selected as Excellent Young researcher of Japan MEXT. In the past ten years, he led and participated in many important projects as principal investigator or primary actor in Japan and China. His main research interests include artificial intelligence, data science and their related fields, and he served as associate editor, guest editor, area chair, Program Committee member or reviewer for many famous journals and top-tier conferences.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdl.