Traffic Flow Prediction Based on Spatiotemporal Potential Energy Fields

Jingyuan Wang, Jiahao Ji, Zhe Jiang,and Leilei Sun

Abstract—Traffic flow prediction is a fundamental problem in spatiotemporal data mining. Most of the existing studies focuses on designing statistical models to fit historical traffic data, which are purely data-driven approaches and fail to reveal the underlying mechanisms of urban traffic. To address this issue, we propose the spatiotemporal potential energy field model (ST-PEF+), which applies the field theory for human mobility to interpret the underlying mechanisms of urban traffic, and introduces the theory into data-driven deep learning models. ST-PEF+ consists of a PEF extraction module and a data-driven module. Inspired by the field theory for human mobility, the PEF extraction module adopts an algorithm to decompose the grid-based traffic flow graph into several polytree-based potential energy fields (PEFs), where traffic flows from high potential locations to low potential locations, just as water is driven by the gravity field. We also provide a theoretical analysis to ensure that the polytree decomposition algorithm can decompose any traffic flow graph. In the data-driven module, ST-PEF+ learns a spatiotemporal deep learning model to predict the dynamics of PEFs. The model adopts correlation-adaptive neural network structures, which consists of a temporal component for temporal correlations and a spatial component for spatial correlations. The temporal component employs a GRU and DCN combined structure to capture both short-term autocorrelation and long-term repeating patterns of PEFs. The spatial component extends the GAT using weighted directed attention to model the asymmetric spatial structure in PEFs. The prediction results of traffic flow are finally derived from PEFs that are predicted by the spatiotemporal deep learning model. We conduct extensive evaluations on three real-world traffic datasets. The results show that our model outperforms the state-of-the-art baselines. In addition, case studies confirm that the PEFs learned in our framework can reveal the underlying mechanisms of urban traffic, thus improving the model interpretability.

Index Terms—Potential Energy Fields, Spatiotemporal Data Mining, Interpretable Traffic Prediction, Field Theory for Human Mobility.

1 Introduction

Traffic flow prediction is a fundamental problem in spatiotemporal data mining [1]. An accurate traffic flow prediction model plays a critical role in real-world applications, such as traffic congestion management and public safety maintenance. For example, vehicle flow prediction of a city can support transportation departments for better understanding and managing congestion [2] and crowd flow prediction can help event organizers maintain crowd safety [3].

In the literature, many statistical learning models are proposed for traffic flow prediction. Early approaches focus on shallow statistical models that treat traffic flow as time series and employ AutoRegressive Integrated Moving Average (ARIMA), Kalman filtering [4], *etc.* to address the prediction problem. In recent years, deep learning-based traffic prediction models have drawn significant attention from both industry and academia. These methods usually apply Recurrent Neural Network (RNN) and its variants to capture dynamic temporal dependencies of traffic flow [5], [6], and use Convolutional Neural Network (CNN) to extract spatial dependencies for traffic data with regular spatial structures, such as grids [7], [8]. When traffic data are defined on non-Euclidean structures such as road networks, Graph Neural Network (GNN) become a more popular approach to capture spatial dependencies [9], [10]. However, most of these statistical learning models follow a data-driven methodology (*i.e.,* directly using a statistical model to fit historical observation data) and lacks for revealing the mechanisms of traffic flow. This defect reduces the generalization of the prediction models and also makes the models poorly interpretable. Although these models have achieved impressive prediction performance in many practical applications, they still cannot be trustingly applied in some critical application areas, such as public safety and emergency management.

In statistical physics, many theories are proposed to interpret the mechanisms of traffic flow and human mobility in cities. For example, the gravity model analogizes the traffic flow between two locations as gravity that is proportional to the masses (populations) and inversely proportional to the distance between the two locations [11]. The radiation law extends the gravity model through modeling human mobility as radiation and absorption processes of energy [12], [13]. The field theory further assumes human mobilities are driven by a field where each location has a potential and human moves from high potential locations to low potential locations [14]. Compared with the data-driven models, the statistical physics theories can reveal the underlying mechanisms of human mobility and therefore have better interpretability and generalizability. However, for practical applications, the statistical physics theories usually cannot achieve satisfactory prediction performance since they only capture the primary mechanisms of human mobility but fail to model specific and localized correlations in particular scenarios.

- *Jingyuan Wang, Jiahao Ji and Leilei Sun are with the School of Computer Science and Engineering, Beihang University, Beijing, China. E-mail: {jywang, jiahaoji, leileisun}@buaa.edu.cn.*
- *Zhe Jiang is with the Department of Computer & Information Science & Engineering, The University of Florida. E-mail: zhe.jiang@ufl.edu.*

This paper proposes a hybrid model collaboratively driven by physical theory and real-world data, Spatio-Temporal Potential Energy Fields (ST-PEF+)[1], to bring the advantages of generalizability and interpretability of statistical physics methods in data-driven traffic flow prediction models. The ST-PEF+ model consists of a PEF extraction module and a data-driven module. Inspired by the field theory for human mobility [14], the PEF extraction module considers that traffic flow in a city is driven by several latent *Potential Energy Fields (PEFs)*. In each PEF, human mobility behavior is like water flow driven by the gravity field [15], [16], *i.e.,* flowing from a high potential location to a low potential location. To mine latent PEFs from traffic data, we propose a decomposition algorithm to decompose the grid-based traffic flow graph into a group of polytree-based subgraphs. Then, we define each polytree subgraph as a PEF where each node has a unique relative potential energy, and traffic only flows from high potential nodes to low potential nodes. In this way, a complex traffic flow graph is decomposed into multiple PEFs that follow a simple rule like water flow driven by the gravity field. The original traffic flow graph is interpreted as the combined action of multiple PEFs. We also provide a novel continuous bound scaling method to prove that the proposed algorithm can decompose any grid-based traffic flow graph into PEFs.

In the data-driven module, we adopt a spatiotemporal deep learning model, including a temporal component and a spatial component, to predict the future PEF states. For different types of correlations in PEF time series, the model adopts suitable deep neural network structures. Specifically, in the temporal component, the model adopts gated recurrent unit (GRU) to capture PEF short-term autocorrelation and adopts dilated causal convolution network (DCN) with a gating mechanism to capture long-term repeating patterns in PEF series. In the spatial component, our model extends graph attention networks (GAT) by weighted directed attention to adapt the directed structures of PEFs. Benefitting from this correlation-adaptive model structure, the data-driven module achieves accurate PEF state prediction. The future traffic flow prediction is finally derived from the predicted PEFs.

The effectiveness of the proposed model is verified by extensive experiments over three large-scale real-world datasets. The results show that the proposed model outperforms multiple state-of-the-art baselines. A case study also confirms that our model can reveal the underlying urban dynamic mechanisms and interpret the physical process of traffic flow.

We highlight the key contributions of this work as:

- ST-PEF+ (and its previous version ST-PEF [6]) introduces the field theory for human mobility into the data-driven traffic flow prediction, which has been rarely studied in previous research. We use the field theory in our model to reveal the underlying mechanisms of human mobility. It improves the interpretability of the prediction model.
- The proposed PEF decomposition algorithm can convert complex urban traffic flow data as regular PEFs. This can reduce the complexity of traffic flow data modeling and help our approach achieve improved prediction accuracy. Moreover, we conduct theoretical analysis on the proposed decomposition algorithm, which provided a novel continuous bound scaling method to guarantee that any traffic flow graph can be decomposed into a group of polytree-based PEFs.
- We propose a novel spatiotemporal deep learning network to model the PEF dynamics. The network adopts correlation-adaptive model structure. We believe this network can be applied to other urban spatiotemporal modeling applications.
- Benefitting from the novel PEF decomposition and efficient deep learning prediction model, our method outperforms several state-of-the-art baselines. A case study confirms that our model can reveal underlying urban dynamic patterns that are not apparent in present data-driven flow prediction models.

1. The previous version of our model was named as ST-PEF [6].

2 Preliminaries

Before we introduce our model, we first define the traffic flow prediction problem in our study. Given a city map, we divide it as a *spatial raster*.

Definition 1 (Spatial Raster). *A spatial raster $\boldsymbol{S} \in \mathbb{R}^{I \times J}$ is a tessellation of a spatial region with regular $I \times J$ grids. Each cell in the raster is a spatial zone, which is considered a minimum prediction unit in our study.*

Fig. 1(a) shows an example of a spatial raster with 4×4 zones. For each cell, there are eight traffic flow types, which are the inflow and outflow to its up, right, down, and left neighboring zones. For a spatial raster, we construct a *flow graph* to model the traffic flow between spatial zones, which is defined as follows.

Definition 2 (Flow Graph). *A flow graph $G(V_G, E_G)$ is a weighted directed graph, of which nodes are the spatial zones in a raster, and edges are the directed traffic flow volume between adjacent zones. We denote the node as $v_i \in V_G$ and the flow volume from v_i to v_j as $e_{ij} \in E_G$.*

Definition 3 (Flow Graph Sequence). *We divide time as regular time slices. For each time slice, a spatial raster has a flow graph $G^{(t)}$. The traffic flow time series for the spatial raster is expressed as a flow graph sequence $\mathcal{G}^{(1:t)} = \left(G^{(1)}, G^{(2)}, \ldots, G^{(t)}\right)$, which is named a flow graph sequence.*

In this work, we study a *traffic flow prediction* problem, which is defined as follows.

Definition 4 (Traffic Flow Prediction). *Given the historical flow graph sequence of a city from time $t - T$ to t, the traffic flow prediction problem is to build a function f to generate*

$$\hat{G}^{(t+1)} = f\left(\mathcal{G}^{(t-T:t)}\right), \tag{1}$$

where $\hat{G}^{(t+1)}$ is a predicted value of $G^{(t+1)}$.

Usually, the traffic flow prediction paradigm directly models correlations between $\mathcal{G}^{(t-T:t)}$ and $G^{(t+1)}$. We adopt an intermediate concept, *i.e.,* the *potential energy field*, to improve the performance, generalizability and interpretability of the traffic flow prediction models. Our model is based on the field theory for human mobility [14]. The theory considers that human mobility in cities is driven by a group

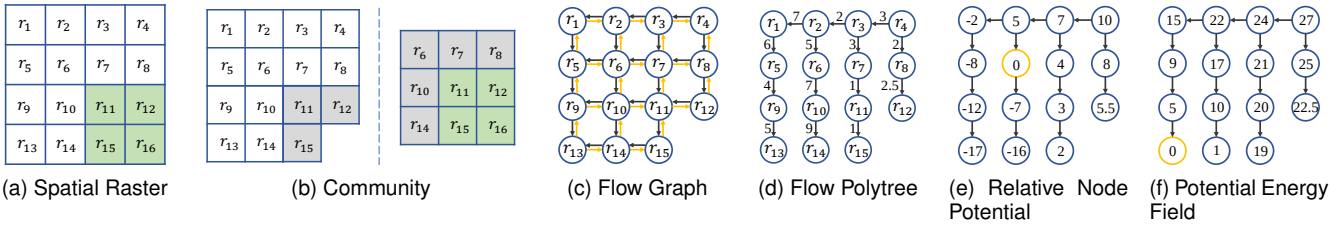


Fig. 1. Illustration of basic concepts. (a): Different colors denote different communities in a spatial raster. (b): Community partitioning results with consideration of the inter-community traffic flow. (c): The flow graph of the larger community can be decomposed into two subgraphs represented by different colors. (d): The flow polytree is a spanning tree of the corresponding flow graph. The number next to the edge denotes the flow volume. (e): The relative potential energy of each node can be calculated according to the given zero-potential node. (f): A standard potential energy field has to maintain all node potential to be positive.

of latent potential energy fields, which is similar to water flow driven by the gravity field.

Considering water flows on the Earth's surface, the flow speed and direction are determined by the gravity field on the Earth's surface (from high elevation to nearby lower elevation). Similarly, we can also assume that the traffic flow between different locations in a city is driven by some latent "potential energy fields" established on urban functions. For example, during the morning rush hours, residential areas have higher potential than working areas, which drives people to travel from home to the workplace. In contrast, during the evening rush hours, working areas have higher potential than residential areas, driving people to travel from the workplace back home.

To model the latent PEFs and exploit them to improve the performance of traffic flow prediction, the proposed ST-PEF+ model designs a PEF extraction module and a data-driven module. The PEF extraction module is a *Potential Field Extraction* algorithm, while the data-driven module is a *Traffic Flow Prediction* deep learning model.

3 Potential Field Extraction Module

3.1 Motivation of PEF Decomposition

In the potential field extraction module, we propose a direction decomposition and a polytree decomposition to convert a flow graph as several PEFs. We assume the traffic flow in cities is driven by latent PEFs, like water flow driven by the gravity field. However, traffic flow dynamic is much more complex than simple physical water flow. The most significant differences between water flow and traffic flow involve two aspects:

- First, water flow is unidirectional, but traffic flow is bidirectional. Water only flows from a high location (a high potential energy node) to a low location (a high potential energy node), but traffic flow between two locations is usually bidirectional. Given two connected nodes r_1 and r_2 (locations) in the flow graph, water can only flows either from r_1 to r_2 in a unidirectional way or from r_2 to r_1. However, in traffic flow, there could exist bidirectional flows between r_1 and r_2.
- Second, water flow is acyclic. Only driven by the gravity, water cannot flow a cycle such as $r_1 \rightarrow r_2 \rightarrow r_3 \rightarrow r_1$. However, cyclic flows always exist in traffic flow graphs.

The main challenge caused by the two differences is that we cannot determine the unique potential energy of each node as we can in the gravity field. For example, for two connected nodes r_1 and r_2 that contain bidirectional flows, the $r_1 \rightarrow r_2$ traffic flow means the potential of r_1 is higher than that of r_2, but the co-existing $r_2 \rightarrow r_1$ flow means the potential of r_2 is higher than that of r_1. Similarly, in the cyclic flow $r_1 \rightarrow r_2 \rightarrow r_3 \rightarrow r_1$, the potential of r_1 is higher than of r_3 due to $r_1 \rightarrow r_2 \rightarrow r_3$, but the potential of r_3 is also higher than that of r_1 due to $r_3 \rightarrow r_1$.

To fill the gap between water flow and traffic flow, our potential field extraction module considers a traffic flow graph as the compound of a group of PEFs. In each PEF, the traffic flow is unidirectional and acyclic. The complex dynamic of the original flow graph is the combined actions of multiple PEFs. To mine PEFs from flow graphs, we propose a potential field extraction algorithm consists of three steps: i) community partitioning, ii) direction decomposition to handle the bidirectional flow problem, and iii) polytree decomposition to handle the cyclic flow problem.

3.2 Community Partitioning

As the first step of PEF modeling, Community Partitioning (CP) aims to divide the study region into multiple communities, each with a smaller footprint (see the green community in Fig. 1(a)). In this way, we can reduce the size of the input flow graph and make it easier to model and clearer to interpret by the latent PEFs.

The object of the community partitioning is to maximize the intra-community traffic flow and minimize the inter-community traffic flow. Formally, given the historical flow graphs of a spatial raster from time slice $t-T$ to t, we build an accumulated flow graph as $\tilde{G} = \sum_{i=t-T}^{t} G^{(i)}$. Then, we partition $\tilde{G}$ into several connected subgraphs (*i.e.*, communities) as

$$\{G_{\boldsymbol{c}} | \boldsymbol{c} \in C\} = \mathrm{CP}(\tilde{G}), \tag{2}$$

where CP denotes the community partitioning, C is the list of communities, and $G_{\boldsymbol{c}}$ is the flow graph of community $\boldsymbol{c}$.

We propose using the improved Girvan-Newman (GN) algorithm to generate communities [17]. The algorithm progressively removes edges from a graph until the graph is divided into enough subgraphs. Edge removal is based on the edge betweenness, *i.e.*, the number of the shortest path between all node pairs that run through the given edge. To handle edge weights, the improved GN algorithm adopted by our model considers an edge with weight n as n unweighted edges.

To include cross-community flow into the PEFs, for each community, we extend grids that are neighbouring to its boundary but belong to other communities into the community. (see gray grids in Fig.1 (b)). This operation makes neighbouring communities overlap at the boundaries. The cross-community flow is included in the traffic flow between extended grids and original community grids, and thus can be modeled by PEFs.

For more convenient modeling and training, we choose a fixed community structure for a flow graph sequence. We assume that communities represent a long-term structure, and thus we consider a long-time historical data into the partitioning process. Of course, the current community partitioning could be extended to dynamic clustering [18].

3.3 Direction Decomposition

The aim of direction decomposition is to handle the problem caused by the bidirectional flow in traffic. The function of direction decomposition is to divide the flow graph of each community into two subgraphs that only contain unidirectional edges between connected nodes.

Given the aggregated historical flow graph $\widetilde{G}$ of a spatial raster, we denote the edges from the bottom to the top as a set $E_{\uparrow}$ and the edges from the top to the bottom as $E_{\downarrow}$. If the total traffic flow of $E_{\uparrow}$ is larger than $E_{\downarrow}$, we name $E_{\uparrow}$ the major direction and $E_{\downarrow}$ the minor direction, and vice versa. In the same way, we define the edge sets $E_{\leftarrow}$ and $E_{\rightarrow}$ and select a major direction from the two sets. As shown in Fig. 1(c), each flow graph in the sequence is decomposed into two subgraphs: a major flow graph (colored black) consisting of edges in major directions, and a minor flow graph (colored yellow) consisting of edges in minor directions. In the major and minor flow graphs, two adjacent nodes are only connected by unidirectional edges. Thus, the bidirectional flow problem is solved.

3.4 Polytree Decomposition

The aim the polytree decomposition is to overcome the cycle flow problem in major and minor flow graphs. Its function is to decompose a flow graph into a set of polytrees. A polytree is a directed acyclic graph whose underlying undirected structure is a tree [19], *i.e.,* it is structurally acyclic, so it can solve the cyclic flow problem.

3.4.1 Problem Definition

Definition 5 (Flow Polytree). *A Flow Polytree $T(V_T, E_T)$ is a directed spanning tree of the flow grid graph $G(V_G, E_G)$. In other words, it is a polytree, of which the node set contains all nodes of G,* i.e., *$V_T = V_{G_g}$, and the edge set is a subset of the flow graph's edge set,* i.e., *$E_T \subset E_{G_g}$.*

For instance, Fig. 1(d) is a flow polytree of the major flow graph (black) in Fig. 1(c). It is a tree having the same nodes as the major flow graph and its edges are a subset of edges in the major flow grid graph.

Definition 6 (Polytree Decomposition). *Polytree decomposition is the process of decomposing a flow graph G_g into k flow polytrees $T_1, T_2, ..., T_k$, such that $E_{T_i} \cap E_{T_j} = \emptyset$ for any i, j and $\bigcup_{i=1}^{k} E_{T_i} = E_{G_g}$.*

In the Definition 6 of polytree decomposition, we require the edge sets of different polytrees are strictly disjoint. This constraint may lead to the decomposition without a solution. To overcome this issue, we make a trick called the *edge-split technique*. It splits a directed edge as multiple edges in the same direction, and the sum of split edges' flow is equal to the total flow of the original edge. We partition the split edges into different polytrees in the decomposition. In this way, we can ensure the edge sets of different polytrees are disjoint. Benefitting from this trick, we can theoretically guarantee that the polytree decomposition always has solutions (see the following theoretical analysis for the proof).

3.4.2 Theoretical Analysis

Here we provide a theoretical analysis to discuss whether there is a solution to the polytree decomposition problem. First, we introduce a lemma.

Lemma 1. *A finite graph G has k edge-disjoint spanning trees if and only if $\Delta_G(V_G) = 0$ and $\Delta_G(X) \geq 0$ for every nonempty subset X of V_G, where $\Delta_G(X) = k\left(|X| - 1\right) - |E_X|$ and $|\cdot|$ is the element number of a set. [20]*

Based on the lemma, we propose a continuous bound scaling method to prove that the polytree decomposition problem is theoretically guaranteed to be solvable.

Theorem 1. *There exists a solution for the polytree decomposition problem when $k = 2$.*

Proof: According to lemma 1, there are two conditions that our flow graph has to satisfy: (a) $\Delta_G(X) \geq 0$ for every nonempty subset X of V_G and (b) $\Delta_G(V_G) = 0$.

Condition (a). From lemma 1, we know that $\Delta_G(X) = k(|X| - 1) - |E_X|$, where E_X is the edge set of graph X. The condition (a) can be transformed as

$$k \geq p(X) = \frac{|E_X|}{|X| - 1}. \tag{3}$$

This means that condition (a) can hold if we set k properly. It is nontrivial to set a proper k, because the edge number $|E_X|$ is undetermined for any grid graph with $|X|$ nodes.

To address this problem, we propose a continuous bound scaling method that scales the right term, *i.e.,* $p(X)$ of the inequation (3) to its upper bound twice to determine its value. Specifically, we find the upper bound for $|E_X|$, *i.e.,* the edge number of a grid graph as follows

$$|E_X| \leq |E_X^u| = 2|X| - \text{ceil}\left(2\sqrt{|X|}\right) \tag{4}$$

where $|E_X^u|$ is the upper bound and $\text{ceil}(\cdot)$ is a function that maps a real number to the least succeeding integer. Taking Eq. (4) into (3), we have

$$p(X) \leq \frac{2|X| - \text{ceil}\left(2\sqrt{|X|}\right)}{|X| - 1}. \tag{5}$$

This makes the original inequation almost solved. We continue to scale the upper bound $|E_X^u|$ as

$$|E_X^u| \leq 2|X| - 2, \tag{6}$$

because $\sqrt{|X|} \geq 1$. Combining the two scales of Eq. (4) and (6), we have

$$p(X) \leq \frac{2|X| - 2}{|X| - 1} = 2. \tag{7}$$

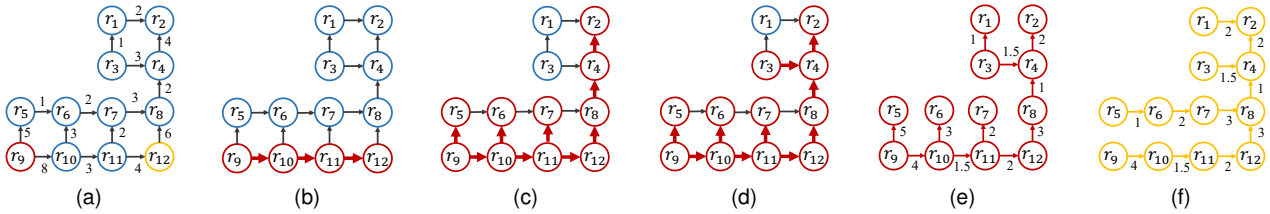


Fig. 2. Figures (a)-(f) denote the process of decomposing a flow graph G into two flow polytrees T_1 and T_2 by Algorithm 1. The number next to the edge indicates the flow volume, and the colors distinguish different polytree elements.

Till now, we have determined the upper bound of $p(X)$ as 2, indicating $k \geq 2$. This means that if we want condition (a) to hold, we can set k as any integer greater than 2. More precisely, we can decompose a grid graph into at least 2 edge-disjoint spanning trees.

Condition (b). For a grid graph G, the condition (b) $\Delta_G(V_G) = 0$ can be transformed into $k = E_G/(|V_G| - 1)$. From the continuous bound scaling method (Eq. (4) and (6)), we know that $|E_G| \leq 2|V_G| - 2$, *i.e.*,

$$k = \frac{|E_G|}{(|V_G| - 1)} \leq 2. \tag{8}$$

This means that the edges of the entire graph G can cover edges of 2 edge-disjoint spanning trees at most.

Combining the results of condition (a) and (b), we know the polytree decomposition problem is solvable when $k = 2$. In particular, the condition (a) $\Delta_G(X) = 2(|X| - 1) - |E_X| \geq 0$ always holds because of Eq. (7). For the condition (b) $\Delta_G(V_G) = 2(|V_G| - 1) - |E_G| = 0$, when there are not enough edges in G, *i.e.*, $|E_G| < 2(|V_G| - 1)$, we can make more edges using the edge-split technique until the obtained graph satisfies the condition. □

It is worth noting that we introduce the continuous bound scaling method to find solutions to the polytree decomposition problem. The method can also be generalized to other grid graph based theoretical analysis.

3.4.3 Algorithm

Based on Theorem 1, we design a Polytree Decomposition (PTD) algorithm. In the Direction Decomposition step, our model decomposes a flow graph into a major and a minor subgraph. The PTD algorithm further decomposes each subgraph into two polytrees with a comb-like structure (see Fig. 1(d)). For each polytree, the algorithm first determines the *backbone direction* and the *tooth direction* of the comb-like structure. Then, the algorithm alternately extends nodes along the two directions to generate the polytree. Algorithm 1 gives the pseudocode of PTD. In the following, we explain the algorithm step by step. To facilitate understanding, we also exhibit a toy example in Fig. 2, corresponding to each step of the algorithm. (We also visualize a real example of polytree decomposition results on a community of the Beijing Taxi dataset in Sec. V of the Supplementary Materials.)

Fig. 2(a): In Lines 1-2 of the pseudocode, PTD determines the backbone direction and the tooth direction of each polytree. For the polytree T_1, we set the direction of the edge with the largest flow volume as its backbone direction, and set the vertical direction of the backbone as the tooth direction. For the polytree T_2, we set its backbone and tooth

Algorithm 1 Polytree Decomposition

Input: Flow graph $G(V_G, E_G)$.
Output: A set of two flow polytrees $\{T_1, T_2\}$.
1: Set the backbone direction and the tooth direction of T_1.
2: Set the backbone direction and the tooth direction of T_2 as the orthogonal directions of T_1.
3: **for** $i = 1$ to 2 **do**
4: Initialize $T_i = \{\}$.
5: Set the origin node of the greatest flow edge as b_i.
6: Bidirectionally extend b_i along the backbone direction to obtain the backbone subgraph G^b.
7: T_i.add(G^b).
8: Bidirectionally extend all nodes in V_{G^b} along the tooth direction to obtain the tooth subgraph G^t.
9: T_i.add(G^t).
10: **if** $V_{T_i} \neq V_G$ **then**
11: **for** $v^t \in V_{G^t}$ **do**
12: Bidirectionally extend it along the backbone direction to get node m.
13: **if** m does not form loop in T_i **then**
14: V_{G^b}.add(m).
15: Set b_i as m and jump to step 6.
16: **for** $e \in E_{T_1}$ **do**
17: **if** $e \in E_{T_2}$ **then**
18: Flow volume of e is halved in T_1 and T_2.
19: **return** $\{T_1, T_2\}$.

direction are orthogonal to T_1. Corresponding to Fig. 2(a), since the edge $e_{9,10}$ has the greatest flow volume in the flow graph, we set the direction $r_9 \rightarrow r_{10}$ as the backbone direction, *i.e.*, horizontally from west to east.

Fig. 2(b): Next, we focus on the construction of T_1 (T_2 is similar). In Lines 5-7, we set the origin node of the greatest flow edge as b_i, which is the r_9 in Fig. 2(b). The algorithm bidirectionally extends b_i along the backbone direction to obtain the backbone subgraph G^b. Here, the term "extend" means recursively adding the nodes that are connected with the newly added nodes into V_{G^b}. In Fig. 2(b), we extend the backbone subgraph as $V_{G^b} = \{r_9, r_{10}, r_{11}, r_{12}\}$.

Fig. 2(c): In Lines 8-9, the algorithm bidirectionally extends all nodes of G^b along the tooth direction to obtain a subgraph G^t. In Fig. 2(c), we vertically extend the node r_9 to obtain node r_5, extend r_9 to obtain r_6, and repeat until all nodes in V_{G^b} are visited. Then, we obtain $V_{G^t} = \{r_5, r_6, r_7, r_8, r_4, r_2\}$.

Fig. 2(d): In Lines 11-15, the algorithm extends new nodes into the backbone subgraph. Before it, the algorithm checks at Line 10 whether the current subgraph contains all nodes, which means its growth terminates. If not, the algorithm orderly checks nodes in V_{G^t}. For each checking node, the algorithm tries to once extend one node along the backbone direction, and check whether the extended node introduces a loop into T_1. If there is no loop, we add the new node into V_{G^b}, and jump to Line 6. In Fig. 2(d), the algorithm

orderly checks the nodes $\{r_5, r_6, r_7, r_8, r_4, r_2\}$ one by one. If we extend the nodes r_5 to r_6, there will be a loop. The same situation occurs for r_6, r_7 and r_8. Therefore, the first node can be extended is r_4 to r_3, so we add r_3 into V_{G^b}.

Fig. 2(e): When V_{G^b} has new nodes, we return to Lines 8-9 to extend the tooth subgraph. In the example, we extend the node r_3 to r_1 along the tooth direction. Then, we move to Line 10, the current subgraph contains all nodes, meaning the growth of T_1 terminates.

Fig. 2(f): All steps in Lines 3-15 are repeated to obtain T_2, which is colored yellow. After the above steps, in Lines 16-18, we half the flow volume of edges that lie in both polytrees.

In the PTD algorithm, we select the origin node of the greatest-flow-volume edge as start of our algorithm and make it a criterion. However, we can start with other nodes as well. It will generate different polytrees, but impact a little to our later deep learning models.

3.5 Potential Field Generation

Based on the flow polytree, we define the node potential energy and the potential energy field as follows.

Definition 7 (Node Potential Energy). *In a flow polytree, the potential energy of node v_i is defined as $p_{v_i} = p_{v_j} + e_{ij}$, where p_{v_i} is the node potential of v_i and e_{ij} denotes the traffic flow from v_i to v_j. By defining a node in a polytree as a zero-potential node, the relative potential energy of all nodes can be determined.*

Definition 8 (Potential Energy Field). *A potential energy field $P(T)$ is a scalar field defined on the nodes of the flow polytree T, whereby each node v_i has its potential energy p_{v_i}. We ensure that the minimum potential energy of $P(T)$ is zero.*

Fig. 1(d) and 1(e) provide illustrations of how to generate a basic potential energy field from a flow polytree. Fig. 1(f) ensures that the minimum potential energy is zero and generates a potential energy field.

3.6 Potential Field for Inter-community

The methods described above mainly deal with potential fields within communities, but cannot model potential fields across several communities. Therefore, in this subsection, we build inter-community PEFs. First, we define a macro-flow graph to describe the inter-community traffic flow.

Definition 9 (Macro-Flow Graph). *A macro-flow graph $MG(V_{MG}, E_{MG})$ is a directed graph, where the node in V_{MG} is a community, and the directed edges in E_{MG} describe the aggregated traffic flow between adjacent communities.*

In the macro-flow graph, the bidirectional and cyclic flow problems (see Section 3.1) also exist. Therefore, we propose a major-minor partition and a general polytree decomposition algorithm to overcome the two problems respectively.

Given a bidirectional edge of the macro-flow graph, the major-minor partition divide its direction with higher flow into a major subgraph, while the other one into a minor subgraph. In this way, the macro-flow graph is partitioned as two sub-graphs. Each sub-graph contains all nodes of the macro-flow graph and unidirectional edges. The bidirectional problem is solved.

For the macro-flow graph without bidirectional edges, *i.e.*, the major subgraph or the minor subgraph, we propose a General Polytree Decomposition (GPTD) algorithm to decompose it into several edge-disjoint polytrees (see Algorithm 2). The core idea of the algorithm is to adopt a modified Kruskal's minimum spanning tree algorithm [21] to obtain polytrees from the unidirectional macro-flow subgraphs. The Kruskal's algorithm includes four steps (in Lines 4-9 of Algorithm 2): i) considering each node of a macro-flow subgraph as an independent tree, ii) iteratively selecting the smallest-weight edge that connected two independent trees from the macro-flow subgraph, iii) merging the two independent trees connected by the selected edge, iv) repeating steps ii and iii until all nodes in the macro-flow subgraph belong to the same tree, and we can obtain a spanning tree. Since the selected edges for tree merging come from different trees, the Kruskal's algorithm ensures that the resulting spanning tree is acyclic. Therefore, the cyclic issue could be solved. In the algorithm, we set the edge weighs as the inverse of traffic flow, *i.e.*, $w_{uv} = 1/e_{uv}, \forall e_{uv} \in E_{MG}$.

The GPTD algorithm repeatedly run the Kruskal's algorithm to extract multiple polytrees until all edges of the input graph are included into the polytrees (the line 13). However, if we keep the edge weights of the macro-flow graph fixed, each run of the Kruskal's algorithm will produce the same spanning trees. To address this problem, we propose to dynamically update edge weights of the macro-flow graph after each run of Kruskal's algorithm (in Lines 10-12 of GPTD). Specifically, we denote edges of the ploytree generated by the i-th run of Kruskal's algorithm as E_T. We increase the weight $w_{uv}^{(i)}$ of E_T after the i-th run as

$$w_{uv}^{(i+1)} = w_{uv}^{(i)} \times \phi^{(i)} \times 2^{t_{uv}^{(i)}}, \tag{9}$$

where $\phi^{(i)} = w_{max}^{(i)}/w_{min}^{(i)} + 1$ is the expansion coefficient, $w_{max}^{(i)}$ and $w_{min}^{(i)}$ are the maximum and minimum edge weights of the macro-flow graph in the i-th run. This coefficient expands the weights $w_{uv}^{(i)}$ of the edges selected by the i-th run as the biggest-weight edges in the $(i+1)$-th run (to reduce the probability of being selected). Term $2^{t_{uv}^{(i)}}$ is a scaling factor which exponentially increases the weights of edges that are selected by previous polytrees. In this way, we can reduce the selection probability of the edges that have been selected multiple times in the previous runs. Both the expansion coefficient and the scaling factor can reduce the number of overlap edges among polytrees.

We repeat the Kruskal's algorithm until all edges are included by the polytrees, *i.e.*, $t_{uv} > 0, \forall e_{uv} \in E_{MG}$ (Line 13). Finally, in Line 14, GPTD evenly assigns traffic flow of overlapped edges to all involved polytrees to generate the final polytree set. Having polytrees of the macro-flow graph, we can generate potential fields for inter-community flows based on Section 3.5. In this way, our PEF extraction module are more complete, including the intra- and inter-community potential fields that interpret traffic flows from the micro- and macro-view respectively.

4 Traffic Flow Prediction Module

Traffic flow prediction aims to predict the future traffic flow states using the decomposed historical PEF sequences.

Algorithm 2 General Polytree Decomposition

Input: Macro-Flow graph $MG(V_{MG}, E_{MG})$.
Output: A set of edge-disjoint flow polytrees MT.
1: Initialize the visit count as $t_{uv} = 0, \forall e_{uv} \in E_{MG}$.
2: Initialize the edge weight as $w_{uv} = 1/e_{uv}, \forall e_{uv} \in E_{MG}$.
3: Calculate the expansion coefficient as $\phi = w_{max}/w_{min} + 1$.
4: Consider each node of MG is an independent tree.
5: Take the smallest-weight edge e_{uv} from E_{MG}.
6: **if** node u and v come from different trees **then**
7: Merge the two trees and add e_{uv} to the merged tree.
8: Repeat 5 to 7 until all nodes belong to the same tree T.
9: MT.add(T).
10: **for** $e_{uv} \in E_T$ **do**
11: $t_{uv} \leftarrow t_{uv} + 1$.
12: $w_{uv} \leftarrow w_{uv} \times \phi \times 2^{t_{uv}}$.
13: Repeat 3 to 12 until $t_{uv} > 0, \forall e_{uv} \in E_{MG}$.
14: Evenly assign traffic flow to polytrees containing edge e_{uv} according to t_{uv}.
15: **return** MT.

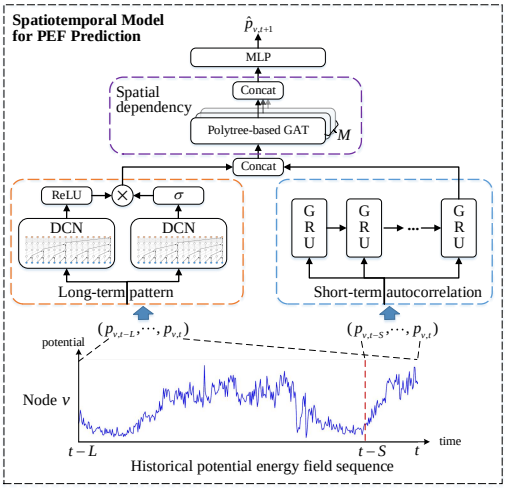


Fig. 3. Spatiotemporal model for PEF prediction in ST-PEF+.

We adopt a two-stage method to achieve this goal. Specifically, we first design a spatiotemporal deep learning model to predict the future PEFs and then derive the future traffic flow based on the predicted PEFs.

The spatiotemporal deep learning model takes decomposed PEFs from historical time series as input and predicts the future PEFs in the next time step. Designing such a predictive model is nontrivial due to the PEF series contain multiple types of spatiotemporal correlations. We need to design different neural network structures to capture different types of correlations. To address these challenges, we design a correlation-adaptive spatiotemporal deep learning model that consists of a temporal modeling component and a spatial modeling component. In the temporal modeling component, we also adopt two types of neural network structures to capture short-term and long-term temporal correlations. The proposed temporal model is illustrated in Fig. 3. In this way, the diversity of correlations in PEF series can be fully exploited by the data-driven module.

4.1 Temporal Modeling Component

In the temporal modeling component, we classify temporal correlations as two categories. The first is the short-term correlations, in which the autoregressive features of the series are dominant. The second is the long-term correlations, in which the periodic patterns are dominant. According to the analysis in the study [22], short-term correlations and long-term periodicity have different natures. Compared with short-term correlations, the long-term periodicity knowledge is more generalizable, while the short-term correlations becomes more important when the time interval is reduced. Therefore, we adopt an RNN network without the attention mechanism to capture short-term correlations in traffic data since RNNs recursively take the recent data as model inputs. In contrast, we employ the DCN to capture long-term periodicity since CNNs are good at extracting repeated patterns from sequence data.

Short-term Correlation Capturing. We denote the historical potential energy of a node v as a time series $\boldsymbol{p}_v^S = (p_{v,t-S}, \cdots, p_{v,t})$. To model the temporal dynamics of $\boldsymbol{p}_v^S$, we employ GRU [23] to recursively encode the historical potential energy series as

$$\boldsymbol{h}_{v,t} = \mathrm{GRU}(\boldsymbol{h}_{v,t-1}, p_{v,t}), \tag{10}$$

where $\boldsymbol{h}_{v,t} \in \mathbb{R}^{K_S}$ is a hidden temporal representation of a node v at time step t. The GRU models the representation at t as a function of previous representations, so is very suitable to capture the autoregressive features of short-term correlations.

Long-term Correlation Capturing. In urban traffic flow, there are many long-term temporally repeating patterns. For example, the rush hours of every weekday are similar to the days before but showing a significant difference from the neighbor time steps such as an hour ago. To identify such repeating patterns, we need to enable the network to model long-term historical data. However, RNN-based methods suffer from vanishing and exploding gradient while modeling long sequences. To address this challenge, we introduce dilated causal convolutional network (DCN) [24], [25] to capture the long-term repeating patterns [26], [27].

DCN is based on 1D convolution but allows an exponentially large receptive field with an increase in the number of hidden layers. DCN preserves the causal order by padding zeros in the inputs so that predictions of the current time step only involve historical information. Specifically, given long-term historical data of a PEF node, *i.e.*, $\boldsymbol{p}_v^L = (p_{v,t-L}, \cdots, p_{v,t}) \in \mathbb{R}^L$ and a filter $\boldsymbol{\theta} \in \mathbb{R}^r$, the dilated causal convolution operator is defined as

$$\mathcal{D}(\boldsymbol{\theta}, \boldsymbol{p}_v^L) = \sum_{i=0}^{r-1} \theta_i \cdot p_{v,t-d\cdot i}, \tag{11}$$

where θ_i is the i-th element of $\boldsymbol{\theta}$ and d denotes the dilation factor. We stack the dilated operator $\mathcal{D}$ with an increasing dilated factor so that the receptive field grows exponentially. This enables our DCN to capture longer dependencies with fewer layers. By adopting multiple filters in parallel, different long-term dependencies can be captured by DCN simultaneously, which enhances the modeling capacity.

In the implementation, we feed the potential energy sequence of the past 3 days into DCN for long-term modeling. Such a long-term sequence usually contains different patterns in different time periods. For example, when modeling rush hours, the patterns of regular times should be ignored,

and vice versa. To implement it, we introduce a gating mechanism to switch patterns in DCN as

$$\boldsymbol{c}_{v,t} = \mathrm{ReLU}(\mathrm{DCN}(\boldsymbol{\Theta}_1, \boldsymbol{p}_v^L)) \odot \sigma(\mathrm{DCN}(\boldsymbol{\Theta}_2, \boldsymbol{p}_v^L)), \tag{12}$$

where $\boldsymbol{c}_{v,t} \in \mathbb{R}^{K_L}$ is the long-term representation of PEF node v at time step t. σ is the sigmoid activation function while $\odot$ represents element-wise production. $\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2 \in \mathbb{R}^{N_L \times K_L \times r}$ denote the learnable convolution filters and N_L is the number of convolutional layers in DCN. The term $\sigma(\mathrm{DCN}(\boldsymbol{\Theta}_2, \boldsymbol{p}_v^L))$ plays a gating mechanism to control the information flow $\mathrm{ReLU}(\mathrm{DCN}(\boldsymbol{\Theta}_1, \boldsymbol{p}_v^L))$ of DCN.

Compared to RNN-based methods, DCN handles sequences nonrecursively, which alleviates the vanishing and exploding gradient problem. The stacked dilated operators enable the model to handle very long history inputs. The gating mechanism enables the model to switch patterns for different conditions in a long-term sequence. All these features make our gated DCN model very suitable for handling long-term correlations in urban traffic. Compared to the long-term traffic modeling methods that constructs multiscale historical time series [7], [8], DCN requires fewer human efforts and makes the temporal modeling more automatic and efficient.

We concatenate the long-term representation $\boldsymbol{c}_{v,t}$ with the autocorrelation representation $\boldsymbol{h}_{v,t}$ generated by Eq. (10), *i.e.,* $\boldsymbol{n}_v = [\boldsymbol{h}_{v,t}, \boldsymbol{c}_{v,t}]$, as the final representation of polytree node v at time step t.

4.2 Spatial Modeling Component

We utilize the graph attention mechanism called GAT [28] to capture spatial dependencies. We denote $\boldsymbol{n}_{v_k}$ as the representation of node v_k of a PEF. For the time step t, the GAT iteratively updates the matrix $\boldsymbol{N} \in \mathbb{R}^{K_G \times |V_T|}$, which consists of $\boldsymbol{n}_{v_k}$ for all nodes in the PEF, as

$$\boldsymbol{N}^{(z)} = \mathrm{GAT}\left(\boldsymbol{N}^{(z-1)}|t\right), \tag{13}$$

where z is iteration number. To involve the temporal information in spatial modeling, we initialize the node representation as $\boldsymbol{n}_v^{(0)} = [\boldsymbol{h}_{v,t}, \boldsymbol{c}_{v,t}]$ for each node v_k.

For the specific update process, the GAT sets the weight between two nodes v_i and v_j as

$$\alpha_{(v_i,v_j)} = \frac{\exp\left(\boldsymbol{w}^\top \left(\boldsymbol{W}_1 \boldsymbol{n}_{v_i} + \boldsymbol{W}_2 \boldsymbol{n}_{v_j}\right)\right)}{\sum_{v_k \in \mathcal{N}_{v_i}} \exp\left(\boldsymbol{w}^\top \left(\boldsymbol{W}_1 \boldsymbol{n}_{v_i} + \boldsymbol{W}_2 \boldsymbol{n}_{v_k}\right)\right)}, \tag{14}$$

where $\alpha(\cdot)$ is the attention score, $\boldsymbol{w}$ and $\boldsymbol{W}_{(\cdot)}$ are the learnable parameters, and $\mathcal{N}_{v_i}$ denotes the neighbors of the polytree node v_i [28].

In the original GAT's attention defined in Eq. (14), the input graph is assumed to be undirected. However, in our model, the PEF has a directed graph structure in the corresponding polytree, and the spatial influence between two polytree nodes (two different locations in reality) is usually asymmetric. To model this feature, we introduce a weighted directed attention mechanism into GAT. Specifically, we set two kinds of attention score for node v_i as

$$\begin{aligned}
\alpha_{(v_i,v_j)} &= \frac{\exp\left(e_{ij}\boldsymbol{w}^\top \left(\boldsymbol{W}_1 \boldsymbol{n}_{v_i} + \boldsymbol{W}_2 \boldsymbol{n}_{v_j}\right)\right)}{\sum_{v_k \in \mathcal{LO}_{v_i}} \exp\left(e_{ik}\boldsymbol{w}^\top \left(\boldsymbol{W}_1 \boldsymbol{n}_{v_i} + \boldsymbol{W}_2 \boldsymbol{n}_{v_k}\right)\right)}, \\
\alpha_{(v_j,v_i)} &= \frac{\exp\left(e_{ji}\boldsymbol{w}^\top \left(\boldsymbol{W}_1 \boldsymbol{n}_{v_i} + \boldsymbol{W}_2 \boldsymbol{n}_{v_j}\right)\right)}{\sum_{v_k \in \mathcal{HI}_{v_i}} \exp\left(e_{ki}\boldsymbol{w}^\top \left(\boldsymbol{W}_1 \boldsymbol{n}_{v_i} + \boldsymbol{W}_2 \boldsymbol{n}_{v_k}\right)\right)},
\end{aligned} \tag{15}$$

where $\alpha_{(\cdot)}$ is the attention score, $e_{(\cdot)}$ is the edge weight, $\boldsymbol{w}$ and $\boldsymbol{W}_{(\cdot)}$ are the learnable parameters, and $\mathcal{LO}_{v_i}$ and $\mathcal{HI}_{v_i}$ denote the neighbors of node v_i with lower and higher potentials, respectively.

Then, we employ the multi-head attention mechanism to generate the representation of each polytree node using the attention score defined in Eq. (15), *i.e.,*

$$\begin{aligned}
\boldsymbol{n}_{v_i}^{(z)} = \Big\|_{m=1}^{M} \mathrm{ReLU}\Bigg(&\sum_{v_k \in \mathcal{LO}_{v_i}} \alpha_{(v_i,v_k)}^{(m)} \boldsymbol{W}^{(m)} \boldsymbol{n}_{v_k}^{(z-1)} \\
&+ \sum_{v_k \in \mathcal{HI}_{v_i}} \alpha_{(v_k,v_i)}^{(m)} \boldsymbol{W}^{(m)} \boldsymbol{n}_{v_k}^{(z-1)} \Bigg),
\end{aligned} \tag{16}$$

where $\|$ represents concatenation, ReLU is rectified linear unit activation function, and M is the number of attention heads. $\alpha_{(v_i,v_k)}^{(m)}$ and $\alpha_{(v_k,v_i)}^{(m)}$ are the normalized attention scores computed by the m-th attention mechanism, and $\boldsymbol{W}^{(\cdot)}$ are learnable parameters. We name the extended GAT network as polytree-based graph attention network (PB-GAT) since its attention involves edge directions and weights of the corresponding polytree.

4.3 Model Training and Flow Prediction

Finally, we predict the potential energy of polytree node v at time step $t+1$ as $\hat{p}_{v,t+1} = \mathrm{MLP}(\boldsymbol{n}_{v,t})$, where MLP is a multilayer perceptron predictor. We train the spatiotemporal model using a mean squared error loss function as

$$\mathcal{L} = \sum_{v \in V_T} \left(p_{v,t+1} - \hat{p}_{v,t+1}\right)^2. \tag{17}$$

Repeating the process on all communities, we can obtain the predicted potential energy fields of every community.

In the PEF extraction module, we decompose a flow graph into several polytrees. Therefore, a direct training strategy is to train a deep learning model for each polytree, but it is time-demanding. To improve runtime efficiency, we adopt a mixed training strategy. Specifically, in the temporal component of the deep learning model, the model parameters are shared by all nodes in different communities because the temporal traffic correlations are similar for different locations. In the spatial component, the parameters are shared by the polytrees with the same backbone direction in different communities. In this way, the PEFs with similar spatial structure can share the same parameters.

After the PEF prediction, we derive the flow graph of each community from the predicted PEF and combine them to generate the city-wise traffic flow graph as the final predictions (see Section III of Supplementary Materials for more details). We also adopt the recursive strategy for multistep predictions [29], *i.e.,* the predicted value of the t-th step is recursively used as the input of the prediction at the $(t+1)$-th step.

In the experiments, we only focus on the prediction of the micro-view since it is related to the problem in Definition 4. The macro-view are used for interpreting inter-community and long-distance commuting.

TABLE 1
Datasets Statistics

City	Xi'an	Beijing	Porto
Trajectory	3,784,063	60,828,387	1,891,921
GPS Interval	3s	60s	15s
Duration	31 days	181 days	365 days
Zone Size	$550m \times 500m$	$800m \times 600m$	$500m \times 500m$
Flow Interval	5 min	30 min	60 min

5 Experiment

5.1 Datasets

In the experiments, we evaluate the performance of ST-PEF+ over three real-world datasets: the Xi'an, Beijing, and Porto taxi datasets. The three datasets contains raw taxi GPS trajectories with different sampling intervals (GPS Interval) collected from different cities. We split the urban areas of the cities into zones with different sizes and mapped taxi trajectories as traffic flow among zones. To increase diversity of experiment setups, we set different traffic flow sampling time intervals (Flow Interval) for different datasets. We summarized the dataset statistics in Table 5.1 (See Section I.A of Supplementary Materials for more details).

5.2 Evaluation Metrics & Baselines

We evaluate our method and baselines by four metrics: mean absolute error (MAE), root mean square error (RMSE), coefficient of determination (R^2) and explained variance score (EVAR). We compare ST-PEF and ST-PEF+ with the following five branches of baselines.

Traditional Time Series Prediction Approaches:
• Historical Average (HA): it models the traffic as a seasonal process. We average the data of the same time step from previous 4 weeks as prediction.
• ARIMA: it is a classical time series prediction model. We set the orders as (3, 0, 1) according to the validation set.

Recurrent Prediction Method:
• FC-LSTM [30]: it is an encoder-decoder framework with fully connected LSTM as hidden units. Both the encoder and decoder have one layer of fully connected network with 256 hidden units.

Spatiotemporal Traffic Prediction Model:
• ST-ResNet [8]: it constructs multiple traffic time series to capture the temporal dependencies and utilize residual convolution to model the spatial correlations.
• STDN [5]: it learns the spatial-temporal dependency by integrating LSTM, local-CNN and semantic network embedding.

Traffic Prediction with Graph Neural Networks:
• STGCN [31]: it is a pure convolution-based model that combines graph convolution and 1D convolution to capture spatial and temporal correlations, respectively.
• AGCRN [32]: it enhances the traditional graph convolution by adaptive modules and combines them into recurrent networks to capture spatial-temporal correlations.

Attentive Traffic Prediction Models:
• STAWnet [27]: it applies a self-attention network to learnable node embedding to capture spatial dependencies and employs dilated convolution for temporal correlations.
• STFGNN [2]: it fuses a data-generated temporal graph and a predefined spatial graph to extract the hidden spatial-temporal dependencies.

For the last three branches of deep learning baselines, we use the same model parameter settings as the open source code of the original paper. The optimization parameters are tuned that performed best on the validation set.

In ST-PEF+, the traffic flow is defined on edges. However, traffic flow prediction baselines are mainly node-level methods, of which the output layer is a regression predictor with two outputs, *i.e.*, one for inflow prediction and the other for outflow prediction. In our experiments, we extend the output layer as four outputs, respectively corresponding to the outflow to a node's up, down, left, and right neighbors. Since the outflow of one node is in the inflow of another node, in this way the node-level baselines become applicable to the edge-level prediction.

5.3 Overall Results

Table 2 demonstrates the result comparison. We implement and run our baselines on the platform LibCity [33], a unified and extensible library for traffic prediction. More details on the experimental setups are given in Section I.B of Supplementary Materials. We replicate each experiment 15 times and report the average results.

Performance Superiority of ST-PEF+. ST-PEF+ significantly outperforms other baselines according to the paired t-test at level 0.01. This demonstrates the effectiveness of ST-PEF+ in the joint use of potential energy fields and the correlation-adaptive deep learning. Fig. 4 plots specific prediction cases to analyze potential reasons of our performance improvement. As shown in the figure, ST-PEF+ captures the trend of rush hours more accurately than other baselines. Since the dynamic range of traffic flow in the rush hours is very large, accurate prediction of this part can significantly improve the prediction performance. In ST-PEF+, the traffic flow in different directions are decomposed into different PEFs, and are modeled by different deep-learning models. In this way, the traffic flows in rush hours, which are usually dominated by traffic in one direction, could be modelled more precisely by the deep-learning part of our model. We think this is a possible reason of why our method has significant performance improvement compared to the baselines.

Performance Improvement over ST-PEF. ST-PEF+ improves ST-PEF by 12.4% in RMSE on average. This result verifies that the advanced deep learning module, *e.g.*, the PB-GAT with directed attention and gated DCN capturing the long-term patterns, contributes a lot to the excellent performance of ST-PEF+. Without the improved spatiotemporal deep learning module, *i.e.*, only using ST-PEF, the proposed model is difficult to surpass the latest baselines, *i.e.*, STFGNN and STAWnet, which were proposed after the ST-PEF model.

Performance Comparison between Baselines. The spatiotemporal-based methods tend to have better performance than the traditional time series approaches and recurrent methods, which suggests the effectiveness of spatiotemporal dependencies modeling. Among various baselines,

TABLE 2
Performance comparison using four metrics on three datasets. "m" and "h" denote minute and hour, respectively. The results are better with smaller (↓) MAE and RMSE, but larger (↑) $\mathbf{R^2}$ and EVAR.

Metric	**MAE ↓**									**RMSE ↓**								
Dataset	Xi'an Taxi			Beijing Taxi			Porto Taxi			Xi'an Taxi			Beijing Taxi			Porto Taxi		
Length	5m	10m	20m	30m	1h	2h	1h	2h	4h	5m	10m	20m	30m	1h	2h	1h	2h	4h
HA	2.032	2.032	2.032	5.241	5.241	5.241	2.610	2.610	2.610	3.840	3.840	3.840	10.587	10.587	10.587	4.672	4.672	4.672
ARIMA	1.987	2.253	2.586	5.125	5.831	6.631	2.573	2.862	3.288	3.763	4.258	4.861	10.301	11.735	13.160	4.606	5.209	6.017
FC-LSTM	1.602	1.975	2.324	4.548	5.112	6.043	2.014	2.469	3.131	2.884	3.653	4.323	9.141	10.478	12.283	3.473	4.572	5.793
ST-ResNet	1.102	1.208	1.322	2.583	3.162	3.975	1.594	1.705	1.853	1.725	1.943	2.208	5.159	6.784	8.826	2.664	2.842	3.168
STDN	1.252	1.326	1.439	3.415	4.342	5.512	1.748	1.946	2.181	2.018	2.147	2.325	7.210	9.392	11.578	2.903	3.242	3.806
STGCN	1.141	1.255	1.422	3.374	4.340	5.418	1.785	1.950	2.179	1.898	2.025	2.314	7.122	9.380	11.322	2.936	3.251	3.803
AGCRN	1.105	1.208	1.373	3.018	3.839	5.152	1.720	1.856	2.087	1.854	1.969	2.271	6.763	8.512	10.793	2.886	3.127	3.642
STAWnet	1.008	1.120	1.269	2.602	3.034	3.866	1.625	1.701	1.814	1.766	1.933	2.181	5.316	6.691	8.545	2.793	2.870	3.061
STFGNN	0.987	1.093	1.233	2.563	2.730	2.945	1.604	1.679	1.761	1.708	1.847	2.009	5.169	5.793	6.566	2.726	2.834	2.975
ST-PEF	1.012	1.110	1.207	2.572	2.731	2.986	1.612	1.751	1.826	1.783	1.867	1.968	5.173	5.762	6.647	2.786	2.910	3.106
ST-PEF+	**0.859**	**0.921**	**1.003**	**2.186**	**2.375**	**2.518**	**1.408**	**1.513**	**1.610**	**1.595**	**1.663**	**1.759**	**4.590**	**4.941**	**5.326**	**2.449**	**2.592**	**2.758**

Metric	**$\mathbf{R^2}$ ↑**									**EVAR ↑**								
Dataset	Xi'an Taxi			Beijing Taxi			Porto Taxi			Xi'an Taxi			Beijing Taxi			Porto Taxi		
Length	5m	10m	20m	30m	1h	2h	1h	2h	4h	5m	10m	20m	30m	1h	2h	1h	2h	4h
HA	0.618	0.618	0.618	0.635	0.635	0.635	0.532	0.532	0.532	0.616	0.616	0.616	0.634	0.634	0.634	0.531	0.531	0.531
ARIMA	0.621	0.607	0.584	0.643	0.616	0.587	0.539	0.514	0.440	0.620	0.606	0.584	0.642	0.615	0.588	0.539	0.514	0.442
FC-LSTM	0.706	0.625	0.595	0.670	0.641	0.603	0.626	0.551	0.479	0.708	0.625	0.596	0.671	0.640	0.601	0.624	0.552	0.480
ST-ResNet	0.810	0.795	0.782	0.853	0.814	0.751	0.745	0.711	0.670	0.811	0.797	0.780	0.854	0.816	0.753	0.745	0.711	0.668
STDN	0.797	0.790	0.778	0.805	0.687	0.622	0.701	0.653	0.598	0.798	0.790	0.779	0.804	0.687	0.620	0.702	0.654	0.597
STGCN	0.809	0.796	0.781	0.809	0.686	0.627	0.695	0.650	0.598	0.810	0.796	0.782	0.809	0.687	0.625	0.695	0.651	0.596
AGCRN	0.811	0.801	0.785	0.821	0.745	0.640	0.700	0.674	0.632	0.811	0.802	0.785	0.820	0.746	0.640	0.699	0.673	0.630
STAWnet	0.816	0.810	0.793	0.860	0.823	0.742	0.730	0.705	0.680	0.816	0.809	0.794	0.861	0.823	0.741	0.731	0.704	0.680
STFGNN	0.821	0.813	0.800	0.867	0.851	0.832	0.738	0.718	0.689	0.822	0.811	0.798	0.867	0.852	0.833	0.739	0.717	0.688
ST-PEF	0.818	0.809	0.798	0.865	0.855	0.834	0.731	0.697	0.676	0.819	0.810	0.798	0.864	0.854	0.836	0.730	0.696	0.675
ST-PEF+	**0.840**	**0.829**	**0.818**	**0.904**	**0.883**	**0.870**	**0.802**	**0.770**	**0.735**	**0.842**	**0.829**	**0.819**	**0.905**	**0.885**	**0.871**	**0.800**	**0.769**	**0.735**

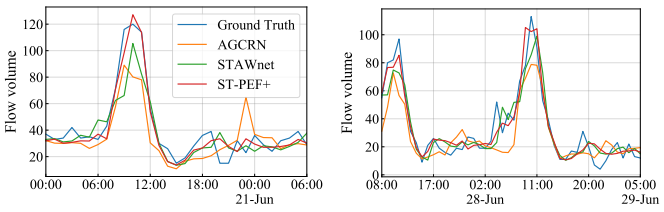


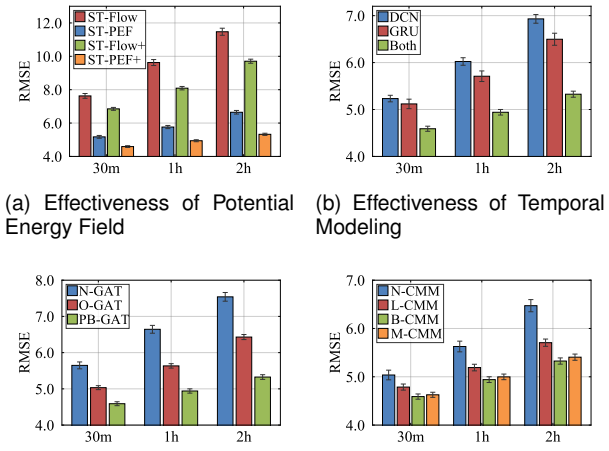
Fig. 4. Flow prediction cases in rush hours of Porto taxi dataset.

GNN-based methods have better performance than others, indicating the rationality of aggregating graph information into traffic flow prediction. Moreover, the attention mechanism is proven to be effective in traffic prediction.

5.4 Quantitative Analysis

Here, we perform a series of detailed quantitative analysis on our model to further evaluate the effectiveness of the major modules. We report the results of metric RMSE on the Beijing dataset. Results of other datasets are given in Section II.E of Supplementary Materials.

Effectiveness of Potential Energy Field. We first evaluate the potential energy field decomposition, which is a core component of ST-PEF+ and ST-PEF. The PEF eliminates the bidirectional patterns and cyclic patterns involved in traffic flow graph so that the traffic patterns are easier to capture. To evaluate the effectiveness of PEF, we compare ST-PEF+ and ST-PEF with two special baselines: *ST-Flow* and *ST-Flow+*, which directly run the spatiotemporal deep learning module of ST-PEF+ and ST-PEF over the original traffic flow graphs without converting them into PEFs. As shown in Fig. 5(a), the performance of both *ST-Flow* and *ST-Flow+* are worse than ST-PEF, indicating that PEF decomposition is very necessary for our model.



(a) Effectiveness of Potential Energy Field

(b) Effectiveness of Temporal Modeling

(c) Effectiveness of Spatial Modeling

(d) Effectiveness of Community Partitioning

Fig. 5. The Quantitative analysis on Beijing dataset using RMSE. The mean and standard derivation of the 15 runs are plotted in the figure.

Effectiveness of Temporal Modeling. We consider three temporal module variants: (1) *DCN* only uses dilated casual convolution networks, capturing the long-term temporal pattern and ignoring the short-term autocorrelation. (2) *GRU* only considers the short-term part in the temporal module. (3) *Both* combines both DCN and GRU for temporal modeling. Fig. 5(b) shows the comparison results, where the performance rank follows $DCN < GRU < Both$. We can observe that the performance of *GRU* is better than that of *DCN* because short-term autocorrelation generally

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2022.3221183

plays a more important role than long-term pattern. The best performance is obtained by the *Both* variant, indicating the effectiveness of simultaneously considering short-term and long-term temporal dependencies.

Effectiveness of Spatial Modeling. We fix the temporal module as its optimal setting and compare the performance of the following three variants: (1) *N-GAT* denotes ST-PEF+ without the attention mechanism, where we keep the node representation transformation but use the same attention weights to integrate the neighbors' information. (2) *O-GAT* uses the original implementation of GAT, designed for undirected graphs, to capture the spatial information. (3) *PB-GAT* is our improved version of GAT by incorporating the polytree context relations and edge information. As shown in Fig. 5(c), their performance rank is *N-GAT* $<$ *O-GAT* $<$ *PB-GAT*. *N-GAT* performs worse than variants utilizing attention mechanisms, which indicates the necessity of assigning different weights to different neighbors for better capturing the spatial dependencies. Moreover, *PB-GAT* consistently outperforms *O-GAT*. This demonstrates the effectiveness of incorporating the edge information of polytrees into the spatial dependency modeling.

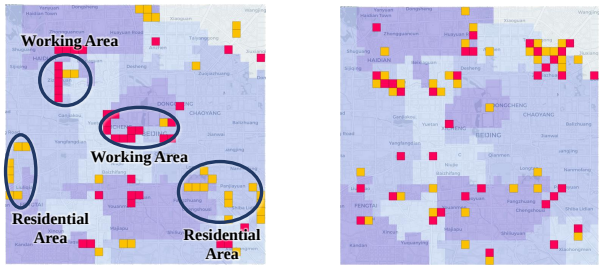
Effectiveness of Community Partitioning. We prepare four variants to examine the effectiveness of the community partitioning component. (1) *N-CMM* denotes no partition of the city (*i.e.,* one community). (2) *L-CMM* uses fewer communities than the best setting ($C = 12$) to divide the whole city, where $C = 6$. (3) *B-CMM* uses the best setting to conduct community partitioning (ie $C = 12$), while (4) *M-CMM* uses more communities, $C = 15$ specifically. In Fig. 5(d), the performance rank is *N-CMM* $<$ *L-CMM* $<$ *M-CMM* $<$ *B-CMM*. *N-CMM* performs worst because the urban dynamic of the whole city is too complex to model, indicating the effectiveness of the divide and conquer strategy. For the remaining three variants using this strategy, it can be observed that the community partitioning with the best setting (*B-CMM*) results in the best performance on flow prediction. *L-CMM* may couple multiple real communities into one, which could couple different spatiotemporal patterns and make them hard to model. *M-CMM* divides one community into multiple incomplete communities, resulting in a slight decrease in the model performance.

5.5 Case Study

Next, we visualize the potential energy fields learned by ST-PEF+ using a case study on the Beijing taxi dataset. The showcase highlights two advantages of our model: PEF can i) identify functional areas with distinct patterns and ii) interpret the physical process of traffic flow.

5.5.1 Identification for Functional Area

Fig. 6 shows the spatial patterns identified by the PEFs that are mined by our method. Here, we use the evening rush hours (*i.e.,* 17:30-19:30) in a weekday as a showcase. In Fig. 6(a), we calculate the total potential energy of all PEFs for each node, and mark the nodes with top 4% energy as pink as well as the bottom 4% as yellow on the map. As a comparison, we also calculate the total $outflow - inflow$ for each node, and mark the top and bottom 4% nodes



(a) Potential Energy (b) Flow Volume

Fig. 6. Spatial distribution of key zones identified by potential energy and flow volume. The background color is used to distinguish different communities. Each small square is a zone in reality. Pink and yellow represent high and low values, respectively.

in Fig. 6(b). The raw distribution of PEF and flow are given in Sec. II.F of Supplementary Materials.

We can observe that potential energy can identify functional areas with distinct patterns (*e.g.,* working areas and residential areas) whereas the flow volume cannot. The reason is that the potential energy can naturally characterize the source and sink of a dynamical system. In urban dynamics, influenced by human mobility patterns, the working area is a sink area in the morning and a source area at night, while the residential area is the opposite. On the contrary, the patterns detected by the flow volume ($outflow - inflow$) are much more fragmented. A possible reason is that traffic components in PEFs are purer than the original flow volume. The feature makes pattern identification for PEFs easier, while for original flow volume, these patterns are hidden in mixed traffic data.

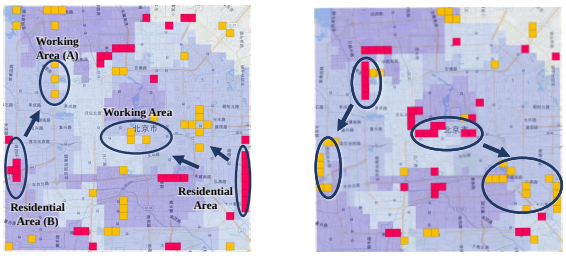
The patterns shown in Figure 6 of the mainbody may also be described by other deep-learning methods, such as CNN, RNN or GNN. Nevertheless, as shown in Fig. **??**(a) and (b), the patterns revealed by PEF have clear physical meaning, *i.e.,* the potential energy that drives people to move from high potential locations to low potential locations. Oppositely, the patterns mined by deep-learning-based methods, such as CNN, RNN or Graph-based methods, do not have such a physical interpretation.

5.5.2 Interpretation for Traffic Flow

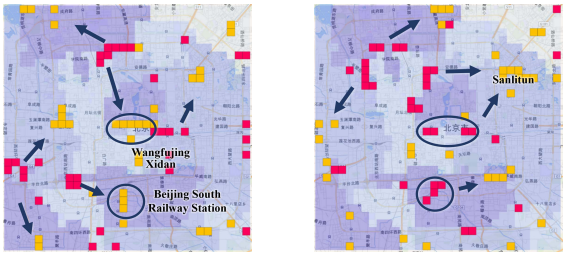
Here, we show how the PEF interprets the physical process of traffic flow. In our model, we divide the whole city as several communities and build PEFs over two levels, *i.e.,* intra-community and inter-community, so we also demonstrate interpretation of our model for both intra-community and inter-community traffic.

For Intra-community Traffic. In Fig. 7, we calculate the total potential energy of all PEFs for each node on weekdays and weekends. The high potential nodes and the low potential nodes are marked using pink and yellow color respectively. Fig. 7(a) and 7(b) demonstrate the *potential energy tidal effect* revealed by PEFs in the Beijing city. Here, we use circles to mark clusters whose potential energy is overturned at the morning peak and evening peak, and use arrows to imply the flow direction driven by the potential energy, which points from high potential zones to low potential zones. As shown in Fig. 7(a), on the weekday morning, driven by the potential energy, people usually come from

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2022.3221183

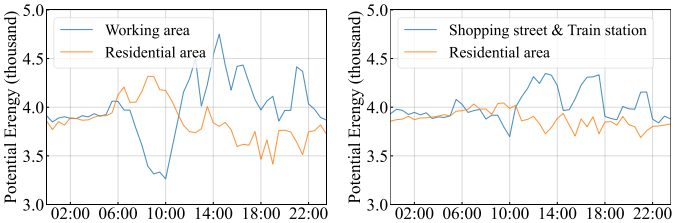


(a) Weekday, 7:30-9:30 (b) Weekday, 17:30-19:30



(c) Weekend, 7:30-9:30 (d) Weekend, 17:30-19:30

Fig. 7. The spatial distribution of potential energy in each community. The arrows mark the flow direction. The blue circles mark the clusters whose potential energy is overturned in the morning and evening.

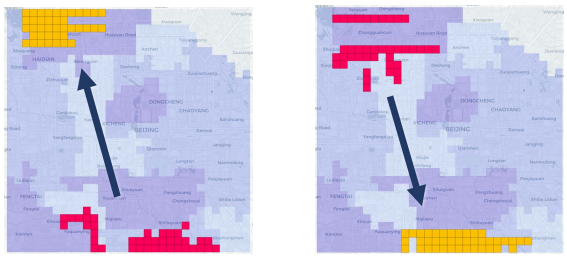


(a) Weekday (b) Weekend

Fig. 8. The temporal dynamics of potential energy on weekdays and weekends.

the suburban areas where they live (pink zones with high potential) and go to the downtown areas where they work (yellow zones with low potential). However, at the evening peak in Fig. 7(b), the flow trend reverses. People leave working areas for residential areas. Regarding the weekend, the phenomenon does not exist over the whole city but in two areas, *i.e.,* shopping streets and train stations. In the morning (Fig. 7(c)), the potential energy drives people to shopping streets (*e.g.,* Xidan and Wangfujing) and train stations (*e.g.,* Beijing South Railway Station) because these areas are under low potential. In the evening in Fig. 7(d), the potential energy of these areas become high and drive people to leave. Some go to residential areas and some go to entertainment areas such as the Sanlitun Bar Street.

In Fig. 8, we further demonstrate the temporal dynamics of potential energy in some key areas on weekdays and weekends. As shown in Fig. 8(a), on weekdays, the working area (the area (A) in Fig. 7(a)) has lower potential energy than the residential area (the area (B) in Fig. 7(a)) in the morning rush hours and higher potential energy in the afternoon. In Fig. 8(b), on weekends, the potential energy of shopping streets (Wangfujing, Xidan) and railway stations (areas labeled on the Fig. 7(c)) is low in the morning and then fluctuates three times from 10:00 to 22:00 (antemeridiem, afternoon, and evening). For residential areas, the potential energy remains relatively low, indicating that



(a) Weekday, 7:30-9:30 (b) Weekday, 17:30-19:30

Fig. 9. The spatial patterns of potential energy for inter-community traffic.

home is always an attractive place for people on weekends.

For Inter-community Traffic. In Fig. 9, for each node, we calculate total inter-community potential energy, which is the potential sum of intra-community PEFs and inter-community PEFs. In Fig. 9, the top and bottom 4% nodes are marked using pink and yellow color. During the morning peak, we can see a low potential energy area located at northwest region of Beijing, which is Zhongguancun Industrial Park, *i.e.,* the China's Silicon Valley. This area attracts many inter-community commuters from the southern region, which is a growing residential area of Beijing. During the evening rush hours, the southern region becomes a low potential area and attracts inter-community commuters home from the northwest region. The phenomenon indicates that the tidal effect also exists in long-distance commuting.

6 Related Work

Data-driven Traffic Flow Prediction. Existing works on data-driven traffic flow prediction can be divided into two categories: shallow statistical models and deep learning models. Shallow statistical models include time series analysis (*e.g.,* ARIMA family [34]), spatial models (*e.g.,* Markov random field [35]) and spatiotemporal models (*e.g.,* STAR [36]). These approaches assume traffic data follow certain distributions and may not capture the nonlinear and dynamic relationships in traffic data. Deep learning models for traffic prediction start from modeling temporal dynamics by RNN and its variants [37], [38]. However, these models treat traffic data of different roads as independent sequences. To fully exploit spatial information, the spatiotemporal model came into being, which has two branches: grid-based and graph-based. For the grid-based branch, eRCNN [39] first introduces CNN model into traffic prediction. ST-ResNet [8] treats a traffic flow grid as an image and utilizes residual CNN to explore the spatial information, while the temporal dependencies are explicitly considered in three views. Graph-based approaches believe that traffic networks usually have a non-Euclidean structure [40]. Thus, graph neural networks have been proposed for traffic flow prediction. Examples include original graph convolution [10] and graph convolution with an attention mechanism [41]. Recently, some work [27], [32], [42] focused on generating the graph structure purely from traffic data rather than using a predefined road/sensor network with context information. Deep learning models have shown promising prediction accuracy as compared to traditional

statistical models. However, they are not easily interpretable and lack revealing the mechanism of traffic flows.

These are some literatures employ traffic flow graph decompositions for gathering event detection [43], [44]. They extract k-dominant directed acyclic graphs (DAGs) from the traffic flow graph to efficiently and effectively capture important gathering events. These works demonstrate that decomposing a traffic flow graph into subgraphs is valuable for traffic data analysis. Along this direction, we further decompose the traffic flow graph as a stricter version of DAGs, *i.e.,* polytrees-based PEF. Compared with the DAG-based works, our model has tighter physical constraints, and thus is suitable to more general traffic flow analyzing tasks, such as traffic flow prediction.

Statistical Physics Model for Human Mobility. The literature on statistical physics theories for traffic flow and human mobility understanding includes the gravity model, the radiation model, and the field theory. The gravity model assumes that the traffic flow volume between two locations increases with the locations' populations while decreases with the distance between them [45], [46]. It draws an analogy with Newton's law of universal gravitation, so it is named as the gravity model [11]. Inspired by the radiation and absorption processes of energy, the radiation model [12], [13] considers extra factors that may affect the traffic flow between two locations, such as job opportunity [47]. Recently, along the gravity model, the field theory further assumes that traffic flows are driven by a field where each location has a potential and people move from high potential locations to low potential locations [14]. Compared to deep learning models, statistical physics models can capture the underlying mechanisms of human mobility and thus have better interpretability and generalization. However, they usually deliver unsatisfactory prediction performances because they fail to capture the dynamic spatiotemporal dependencies across different locations and time periods.

Urban Computing. Our work also fall into the research category of urban computing. Urban computing aims to address the issues caused by rapid urbanization, *e.g.,* urban flow prediction [48], public safety maintenance [49], [50], urban functional region recognition [51], urban dynamics understanding [52], epidemic control and prevention [53].

7 Conclusion and Future Work

In this paper, we introduced the statistical physics theories for human mobility into data-driven deep learning to design high performance and interpretable traffic flow prediction model. The proposed model, namely ST-PEF+, consisted of a PEF extraction module and a data-driven module. The PEF extraction module decomposed complex traffic flow data as polytree-based potential energy fields. In the data-driven module, we designed correlation-adaptive deep neural network structures. Extensive evaluations on three real-world traffic datasets showed that the proposed model outperforms multiple state-of-the-art baselines. Case studies confirmed that our interpretable model can reveal underlying urban dynamic patterns. There are several future works, *e.g.,* performing dynamic community partitioning and generalizing our framework from a grid graph to a realistic road network graph, *etc.* Of course, using the PEF is just one way of interpreting traffic flow, but not necessarily the only way, *e.g.,* just simply decomposing the traffic flow into several flow graphs without the constraint of "PEF" will also be interpretable. Studying other decomposition methods is also a direction of the future works.

References

[1] J. Ji, J. Wang, Z. Jiang, J. Jiang, and H. Zhang, "STDEN: Towards physics-guided neural networks for traffic flow prediction," *AAAI*, vol. 36, no. 4, pp. 4048–4056, 2022.

[2] M. Li and Z. Zhu, "Spatial-temporal fusion graph neural networks for traffic flow forecasting," in *AAAI*, 2021.

[3] J. Zhang, Y. Zheng, J. Sun, and D. Qi, "Flow prediction in spatio-temporal networks based on multitask deep learning," *IEEE TKDE*, vol. 32, no. 3, pp. 468–478, 2019.

[4] J. Wang, N. Wu, X. Lu, W. X. Zhao, and K. Feng, "Deep trajectory recovery with fine-grained calibration using kalman filter," *IEEE TKDE*, vol. 33, no. 3, pp. 921–934, 2019.

[5] H. Yao, X. Tang, H. Wei, G. Zheng, and Z. Li, "Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction," in *AAAI*, 2019.

[6] J. Ji, J. Wang, Z. Jiang, J. Ma, and H. Zhang, "Interpretable spatiotemporal deep learning model for traffic flow prediction based on potential energy fields," in *ICDM*, 2020.

[7] J. Zhang, Y. Zheng, D. Qi, R. Li, and X. Yi, "DNN-based prediction model for spatio-temporal data," in *SIGSPATIAL*, 2016.

[8] J. Zhang, Y. Zheng, D. Qi, R. Li, X. Yi, and T. Li, "Predicting city-wide crowd flows using deep spatio-temporal residual networks," *Artificial Intelligence*, vol. 259, pp. 147–166, 2018.

[9] N. Wu, X. W. Zhao, J. Wang, and D. Pan, "Learning effective road network representation with hierarchical graph neural networks," in *KDD*, 2020, pp. 6–14.

[10] W. Li, X. Wang, Y. Zhang, and Q. Wu, "Traffic flow prediction over muti-sensor data correlation with graph convolution network," *Neurocomputing*, vol. 427, pp. 50–63, 2021.

[11] F. Simini, G. Barlacchi, M. Luca, and L. Pappalardo, "A deep gravity model for mobility flows generation," *Nature Communications*, vol. 12, no. 1, pp. 1–13, 2021.

[12] F. Simini, M. C. González, A. Maritan, and A.-L. Barabási, "A universal model for mobility and migration patterns," *Nature*, vol. 484, no. 7392, pp. 96–100, 2012.

[13] Y. Ren, M. Ercsey-Ravasz, P. Wang, M. C. González, and Z. Toroczkai, "Predicting commuter flows in spatial networks using a radiation model based on temporal ranges," *Nature Communications*, vol. 5, no. 1, pp. 1–9, 2014.

[14] M. Mazzoli, A. Molas, A. Bassolas, M. Lenormand, P. Colet, and J. J. Ramasco, "Field theory for recurrent mobility," *Nature Communications*, vol. 10, no. 1, pp. 1–10, 2019.

[15] Z. Jiang, M. Xie, and A. M. Sainju, "Geographical hidden markov tree," *IEEE TKDE*, vol. 33, no. 2, pp. 506–520, 2021.

[16] A. M. Sainju, W. He, and Z. Jiang, "A hidden markov contour tree model for spatial structured prediction," *IEEE TKDE*, vol. 34, no. 4, pp. 1530–1543, 2020.

[17] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical review E*, 2004.

[18] L. Chen, D. Zhang, L. Wang, D. Yang, X. Ma, S. Li, Z. Wu, G. Pan, T.-M.-T. Nguyen, and J. Jakubowicz, "Dynamic cluster-based over-demand prediction in bike sharing systems," in *ACM UbiComp*, 2016, pp. 841–852.

[19] G. Rebane and J. Pearl, "The recovery of causal polytrees from statistical data," in *UAI*, 1987.

[20] C. S. J. Nash-Williams, "Edge-disjoint spanning trees of finite graphs," *Journal of the London Mathematical Society*, vol. 1, no. 1, pp. 445–450, 1961.

[21] J. B. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem," *Proceedings of the American Mathematical society*, vol. 7, no. 1, pp. 48–50, 1956.

[22] L. Wang, D. Chai, X. Liu, L. Chen, and K. Chen, "Exploring the generalizability of spatio-temporal traffic prediction: meta-modeling and an analytic framework," *IEEE TKDE*, 2021.

[23] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *EMNLP*, 2014, pp. 1724–1734.

[24] N. Wu, J. Wang, W. X. Zhao, and Y. Jin, "Learning to effectively estimate the travel time for fastest route recommendation," in *CIKM*, 2019, pp. 1923–1932.

[25] V. K. Shaojie Bai, J. Zico Kolter, "Convolutional sequence modeling revisited," in *ICLR*, 2018.

[26] J. Wang, N. Wu, and X. Zhao, "Personalized route recommendation with neural network enhanced A* search algorithm," *IEEE TKDE*, no. 01, pp. 1–1, 2021.

[27] C. Tian and W. K. Chan, "Spatial-temporal attention wavenet: A deep learning framework for traffic prediction considering spatial-temporal dependencies," *IET Intelligent Transport Systems*, vol. 15, no. 4, pp. 549–561, 2021.

[28] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *ICLR*, 2018.

[29] S. B. Taieb and A. F. Atiya, "A bias and variance analysis for multistep-ahead time series forecasting," *IEEE TNNLS*, vol. 27, no. 1, pp. 62–76, 2015.

[30] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *NeurIPS*, 2014, pp. 3104–3112.

[31] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *IJCAI*, 2018, pp. 3634–3640.

[32] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," in *NeurIPS*, 2020.

[33] J. Wang, J. Jiang, W. Jiang, C. Li, and W. X. Zhao, "LibCity: An open library for traffic prediction," in *ACM SIGSPATIAL*, 2021, pp. 145–148.

[34] S. V. Kumar and L. Vanajakshi, "Short-term traffic flow prediction using seasonal arima model with limited input data," *European Transport Research Review*, vol. 7, no. 3, pp. 1–9, 2015.

[35] Y. Cai and L. Yu, "Sensor network traffic load prediction with markov random field theory," in *ICCSNT*, 2015.

[36] R. K. Pace, R. Barry, J. M. Clapp, and M. Rodriquez, "Spatiotemporal autoregressive models of neighborhood effects," *The Journal of Real Estate Finance and Economics*, vol. 17, no. 1, pp. 15–33, 1998.

[37] Y. Tian and L. Pan, "Predicting short-term traffic flow by long short-term memory recurrent neural network," in *Proceedings of the 2015 IEEE international conference on smart city/SocialCom/SustainCom*, 2015.

[38] D. Yang, K. Chen, M. Yang, and X. Zhao, "Urban rail transit passenger flow forecast based on lstm with enhanced long-term features," *IET Intelligent Transport Systems*, 2019.

[39] J. Wang, Q. Gu, J. Wu, G. Liu, and Z. Xiong, "Traffic speed prediction and congestion source exploration: A deep learning method," in *ICDM*. IEEE, 2016, pp. 499–508.

[40] J. Wang, N. Wu, W. X. Zhao, F. Peng, and X. Lin, "Empowering A* search algorithms with neural networks for personalized route recommendation," in *KDD*, 2019, pp. 539–547.

[41] X. Yin, G. Wu, J. Wei, Y. Shen, H. Qi, and B. Yin, "Multi-stage attention spatial-temporal graph networks for traffic prediction," *Neurocomputing*, vol. 428, pp. 42–53, 2021.

[42] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," in *IJCAI*, 2019, pp. 1907–1913.

[43] X. Zhou, A. V. Khezerlou, A. Liu, Z. Shafiq, and F. Zhang, "A traffic flow approach to early detection of gathering events," in *SIGSPATIAL*, 2016, pp. 1–10.

[44] A. V. Khezerlou, X. Zhou, L. Li, Z. Shafiq, A. X. Liu, and F. Zhang, "A traffic flow approach to early detection of gathering events: Comprehensive results," *ACM TIST*, vol. 8, no. 6, pp. 1–24, 2017.

[45] G. K. Zipf, "The p 1 p 2/d hypothesis: on the intercity movement of persons," *American Sociological Review*, vol. 11, no. 6, pp. 677–686, 1946.

[46] H. Barbosa, M. Barthelemy, G. Ghoshal, C. R. James, M. Lenormand, T. Louail, R. Menezes, J. J. Ramasco, F. Simini, and M. Tomasini, "Human mobility: Models and applications," *Physics Reports*, vol. 734, pp. 1–74, 2018.

[47] A. P. Masucci, J. Serras, A. Johansson, and M. Batty, "Gravity versus radiation models: On the importance of scale and heterogeneity in commuting flows," *Physical Review E*, vol. 88, no. 2, p. 022812, 2013.

[48] J. Wang, Y. Lin, J. Wu, Z. Wang, and Z. Xiong, "Coupling implicit and explicit knowledge for customer volume prediction," in *AAAI*, 2017.

[49] J. Wang, C. Chen, J. Wu, and Z. Xiong, "No longer sleeping with a bomb: a duet system for protecting urban safety from dangerous goods," in *KDD*, 2017, pp. 1673–1681.

[50] J. Ji, J. Wang, J. Wu, B. Han, J. Zhang, and Y. Zheng, "Precision CityShield against hazardous chemicals threats via location mining and self-supervised learning," in *KDD*, 2022.

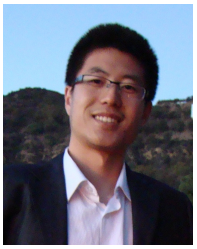
[51] J. Wang, X. He, Z. Wang, J. Wu, N. J. Yuan, X. Xie, and Z. Xiong, "CD-CNN: A partially supervised cross-domain deep learning model for urban resident recognition," in *AAAI*, 2018.

[52] J. Wang, J. Wu, Z. Wang, F. Gao, and Z. Xiong, "Understanding urban dynamics via context-aware tensor factorization with neighboring regularization," *IEEE TKDE*, vol. 32, no. 11, pp. 2269–2283, 2019.

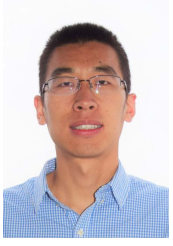
[53] J. Wang, X. Wang, and J. Wu, "Inferring metapopulation propagation network for intra-city epidemic control and prevention," in *KDD*, 2018, pp. 830–838.

Jingyuan Wang received the Ph.D. degree from the Department of Computer Science and Technology, Tsinghua University. He is currently a Professor of School of Computer Science and Engineering, Beihang University. His is also the head of Beihang Interest Group on SmartCity (BIGSCity). His general area of research is data mining and machine learning, with special interests in smart cities and spatiotemporal data analytics.

Jiahao Ji is working toward the PhD in the School of Computer Science and Engineering, Beihang University, Beijing. His research focuses on spatio-temporal data mining, interpretable machine learning and urban computing.



Zhe Jiang is an assistant professor in the Department of Computer & Information Science & Engineering at the University of Florida. He is also affiliated with the Center for Coastal Solutions. He received his Ph.D. in Computer Science from the University of Minnesota, Twin Cities in 2016, and B.E. in Electrical Engineering and Information Science from the the University of Science and Technology of China in 2010. His research interests include data mining, deep learning, and spatiotemporal data mining, interdisciplinary applications in earth science, transportation, public health, etc. He is a senior member of IEEE.



Leilei Sun is with the State Key Laboratory of Software Development Environment(SKLSDE), School of Computer Science and Engineering, Beihang University, Beijing, China. His research interests include machine learning and data mining. He has published a series papers on IEEE Transactions on Data and Knowledge Engineering (TKDE), ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), etc.