Interpretable Spatiotemporal Deep Learning Model for Traffic Flow Prediction based on Potential Energy Fields

Jiahao Ji†, Jingyuan Wang‡§*, Zhe Jiang¶, Jingtian Ma† and Hu Zhang†
†*School of Computer Science and Engineering, Beihang University, Beijing, China*
‡*State Key Laboratory of Software Development Environment, Beihang University, Beijing, China*
§*MOE Engineering Research Center of ACAT, Beihang University, Beijing, China*
¶*Department of Computer Science, The University of Alabama, Tuscaloosa, Alabama, USA*
{*jiahaoji, jywang*}*@buaa.edu.cn, *Corresponding author*

***Abstract*—Traffic flow prediction is of great importance in traffic management and public safety, but is challenging due to the complex spatial-temporal dependencies as well as temporal dynamics. Existing work either focuses on traditional statistical models, which have limited prediction accuracy, or relies on black-box deep learning models, which have superior prediction accuracy but are hard to interpret. In contrast, we propose a novel interpretable spatiotemporal deep learning model for traffic flow prediction. Our main idea is to model the physics of traffic flow through a number of latent Spatio-Temporal Potential Energy Fields (ST-PEFs), similar to water flow driven by the gravity field. We develop a Wind field Decomposition (WD) algorithm to decompose traffic flow into poly-tree components so that ST-PEFs can be established. We then design a spatiotemporal deep learning model for the ST-PEFs, which consists of a temporal component (modeling the temporal correlation) and a spatial component (modeling the spatial dependencies). To the best of our knowledge, this is the first work that make traffic flow prediction based on ST-PEFs. Experimental results on real-world traffic datasets show the effectiveness of our model compared to the existing methods. A case study confirms our model interpretability.**

***Keywords*-Potential Energy Fields; Spatiotemporal Model; Interpretable Prediction; Deep Learning;**

I. Introduction

Traffic flow prediction is of great importance in traffic and urban management. In general, an accurate traffic prediction model plays a critical role in many real-world applications. For example, traffic flow prediction on the vehicle can help the transportation department better understand and manage congestion [1]. Accurate crowd flow prediction can help event organizers maintain the safety of crowded people [2]. Besides, long-term population tracking prediction of a city is also very valuable for urban planners.

Extensive studies exist on traffic flow prediction over the last few decades. Early approaches for this problem are usually based on statistical models in time series analysis (e.g., auto-regressive integrated moving average, or ARIMA [3]). These models are easy to interpret but cannot capture complex non-linear and dynamic spatial-temporal dependencies. Recently, a series of studies are inspired to apply deep learning technique [4] to traffic flow prediction. These methods usually utilize recurrent neural networks (RNN) and its variants to model dynamic temporal dependencies [1], and employ convolutional neural networks (CNN) to extract spatial relationships from the whole city by modeling city-wise traffic as a grid heatmap [5]. When it comes to non-Euclidean structured data such as spatial networks, some studies use graph convolution networks (GCN) to capture spatial patterns [6].

Although spatial correlation and temporal dynamics have been considered in existing deep learning models, these models are often black-box with poor interpretability. It is hard to explain the rationale behind model predictions for decision making in real-world transportation applications. There are some studies on interpreting black-box deep learning models, e.g., explaining their working mechanism and decision-making process. However, in terms of spatiotemporal prediction, the explanation of the mechanism of urban dynamics with physical meaning is not well studied.

To fill the gap, we propose an interpretable spatiotemporal deep learning model for traffic flow prediction. Our main idea is to model the physics of traffic flow through a number of latent spatiotemporal potential energy fields (ST-PEFs), similar to water flows driven by the gravity field. Our framework first decomposes traffic flows into poly-tree components so that ST-PEFs can be established. We then design a spatiotemporal deep learning model for the ST-PEFs. The model consists of a temporal component (Gated Recurrent Units, GRU [7]) and a spatial component (graph attention networks, GAT [8]). The predicted potential energy fields can be used to predict future traffic flows as well as help interpret the predictions being made. Our main contributions are summarized as follows.

- We develop a flow graph decomposition algorithm to extract potential energy fields from traffic flow graphs.
- We design a spatiotemporal deep learning model for potential energy fields, which models temporal dynamics and spatial dependencies, respectively.
- Extensive experiments show that our model outperforms several state-of-the-art methods. A case study confirms out model interpretability.

2374-8486/20/$31.00 ©2020 IEEE
DOI 10.1109/ICDM50108.2020.00128

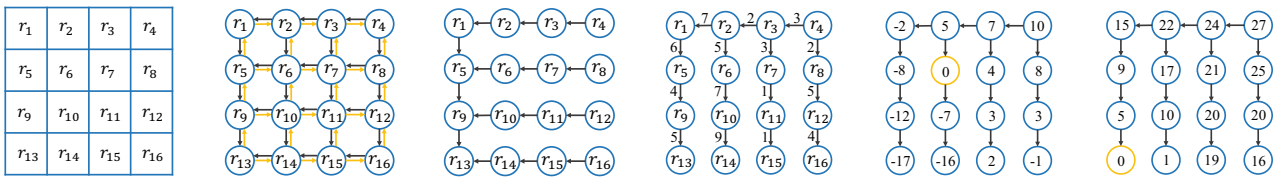

Figure 1. Illustration of basic concepts.

II. Preliminaries

A. Definitions

Definition 1 (Spatial Raster Framework). *A Spatial Raster Framework is a tessellation of a spatial region into a regular grid. Each cell in the framework is a Spatial Data Sample.*

Figure 1(a) shows an example of a spatial raster framework with 4×4 cells. Each cell contains inflow and outflow to neighboring cells in four different directions (up, right, down, and left).

Definition 2 (Flow Graph). *A Flow Graph $G(V_G, E_G)$ is a directed grid graph of which nodes are the cells in a raster framework and edges are the traffic flow volume and direction between pairs of adjacent cells.*

Figure 1(b) shows an example of a flow graph for the 4 by 4 raster framework. Note that there are two different color in edge of flow graph indicating the major flow graph (black) and the minor flow graph (yellow).

Definition 3 (Flow Poly-tree). *A Flow Poly-tree $T(V_T, E_T)$ is a directed spanning tree of the flow graph. In other words, it is a poly-tree whose nodes are the set of all nodes ($V_T = V_G$) and whose edges are a subset of all edges in the flow graph ($E_T \subset E_G$).*

For instance, Figure 1(c) and 1(d) are both flow poly-tree of the major flow graph in Figure 1(b), for the reason that they have the same nodes as the major flow graph and their edges are a subset of all edges in the major flow graph.

Definition 4 (Flow Graph Decomposition). *Flow Graph Decomposition is the process of decomposing a flow graph G into k flow poly-trees $T_1, T_2, ..., T_k$, such that $E_{T_i} \cap E_{T_j} = \emptyset$ for any i, j and $\bigcup_{i=1}^{k} E_{T_i} = E_G$.*

Definition 5 (Potential Energy Field). *A Potential Energy Field is a scalar field defined on a graph, whereby gradients between adjacent nodes represent traffic flow along poly-tree edges.*

Figure 1(d) to 1(f) provide an illustration. (d) Given a flow poly-tree consisting of 16 nodes and 15 edges with flow volume besides the corresponding edge and a randomly selected node as the zero-potential node. (e) We can calculate the potential energy value of all other nodes. (f) The potential energy field value can be readjusted based on any fixed node, which is assumed to be zero-potential.

B. Flow Prediction Problem

Given a spatial raster framework of a city, as well as the historical traffic flow between neighboring grid cells, the traffic flow prediction problem aims to predict the flow between neighboring cells at the next time step (or next few time steps).

III. Approach

In this section, we propose an interpretable deep learning framework for traffic flow prediction.

Our approach is based on a key idea that traffic flow is driven by a set of latent potential energy fields, similar to water flow driven by the gravity field. However, generalizing the idea of water flow modeling to traffic flow modeling is non-trivial. Because unlike water flows, traffic flows may contain cycles. The existence of cycles violates the concept of potential energy fields.

To address this challenge, we propose to decompose the traffic flow graph into a number of components, whereby each component is acyclic, then model the potential energy field for each component. The potential energy fields, after being learned, provide an opportunity to interpret model predictions on traffic flow. In order to maintain the good interpretability of potential energy fields (the effects of traffic flow and energy field are more complicated over a long distance), we propose to partition the region of study into a number of communities and apply our model to each individual community (divide and conquer strategy).

A. Overall Framework

Figure 2 provides an overview of the proposed deep learning framework. The unique design in our framework is to conduct a spatiotemporal predictive model on potential energy fields instead of directly on the flow graph. The reason is that potential energy fields provide important latent information to explain the predicted traffic flow.

B. Community Partition

The goal of community partition is to divide the study region into smaller contiguous zones so that each zone has a smaller footprint and thus easier to explain based on its

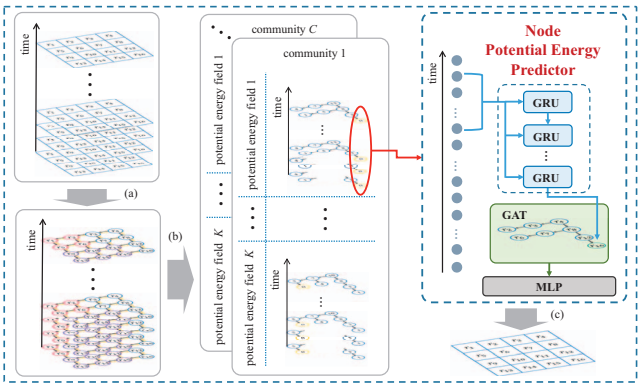

Figure 2. The overall architecture of our framework. (a) Community partition step divides the study region into small contiguous zones. (b) Flow graph decomposition step decomposes the traffic flow graph within a community into K flow poly-trees and then converts each poly-tree into a potential energy field. (c) Flow prediction step predicts the flow from potential energy fields of all communities.

Algorithm 1 Wind Field Decomposition Algorithm

Input: Flow graph $G(V_G, E_G)$.
Output: A set of two edge-disjoint flow poly-trees $\{T_1, T_2\}$.

Initialize backbone node set $B = \{\}$, backbone extension direction set $D = \{\}$.
for $i = 1$ to 2 **do**
  Initialize $T_i = \{\}$.
  Select backbone node b_i of T_i in V_G and add b_i to B.
  Decide the extension direction d_i of b_i and add d_i to D.
for $i = 1$ to 2 **do**
  Extend b_i along d_i to obtain the backbone subgraph G^b.
  T_i.add(G^b).
  Extend all nodes in V_{G^b} along the axial direction of d_i to obtain the tooth subgraph G^t.
  T_i.add(G^t).
  for $v^t \in V_{G^t}$ **do**
    Extend it along d_i to get node m.
    if m does not form loop in T_i **then**
      V_{G^b}.add(m) and jump to step 9.
for $e \in E_{T_1}$ **do**
  if $e \in E_{T_2}$ **then**
    Flow volume of e is halved in T_1 and T_2.
return $\{T_1, T_2\}$.

potential energy field. Given a study region as a regular grid of cells, as well as historical traffic flows between cells, the object of the zone partition is to maximize the intra-zone traffic flow and minimize the inter-zone traffic flow. The problem is similar to community partition in social network analysis. We propose to use the improved Girvan–Newman (GN) algorithm [9]. The main idea is to progressively remove edges from a network and then to partition the network based on a measure of modularity. Edge removal is based on the edge betweenness, i.e., the number of shortest paths between all node pairs that run through an edge.

C. Flow Graph Decomposition

The goal of the flow graph decomposition is to decompose the traffic flow graph within a community into four orthogonal flow poly-trees so that each flow poly-tree corresponds to a potential energy field.

The flow graph decomposition problem, however, is computational challenging in three aspects. First of all, it is not clear whether a solution exists for this graph problem. Moreover, it is not clear what is the number of poly-tree components we can produce in the decomposition. At last, it is harder to decompose a flow graph within a community because its spatial footprint is more irregular.

In order to prove the decomposition problem is solvable, we adopt the theorem from [10]. It describes as Theorem 1 that studies on what condition a graph can be decomposed into edge-disjoint spanning trees.

Theorem 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 non-empty subset X of V_G.*

Here, $\Delta_G(X) = k(|X|-1) - |E_X|$ and the order $|\cdot|$ of a set is the number of elements. There are only two constraints that need to be satisfied if a graph can be decomposed into k edge-disjoint spanning trees. In fact, our flow graph exactly satisfies both constraints when $k = 2$, making the flow graph decomposition problem solvable.

Based on Theorem 1, we design Algorithm 1 (*Wind field Decomposition*, WD) to solve the flow graph decomposition problem. The intuition is that we can project the flow graph within a community into four basic directions: north to south, south to north, east to west, and west to east. The projected flow graph in each basic direction forms a poly-tree with a comb-like structure. Such a decomposition has two advantages. First, it is simple and can be easily generalized to an arbitrary flow graph within a community. Second, the basic directions are orthogonal and easy to explain based on our common knowledge.

D. Spatiotemporal Model for Potential Energy

Here we aim to predict the potential energy of a poly-tree node v at time step t, a.k.a., $p_{v,t}$. Because the estimation process can be repeated on all the poly-tree nodes in parallel.

Temporal Modeling. The historical potential energy data of the node can be formed as time series, $\{p_{v,1}, \cdots, p_{v,t-1}\}$. Given the current time step t, we consider a closeness relation of t_S time steps in a short period, i.e., from $t - t_S$ to $t-1$. To model the temporal dynamics, we employ GRU to encode a subsequence of the near recent potential energy. At time step t, we update the hidden state of a poly-tree node v according to the GRU networks as

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

where $\boldsymbol{h}_{v,t} \in \mathbb{R}^{K_S}$ encodes the necessary temporal information, will be the input of the spatial module.

Spatial Modeling. To capture the spatial dependencies, we utilize the novel GAT. Formally, the update of GAT can be

given as

$$\boldsymbol{N}^{(z)} = \mathrm{GAT}(\boldsymbol{N}^{z-1}|t). \quad (2)$$

The v_k-th column $\boldsymbol{n}_{v_k} \in \mathbb{R}^{K_G}$ of $\boldsymbol{N}^{(z)}$ denotes the graph representation of node v_k, where v_k is a node in set V. Note that z indicate the number of iteration for a specific time step t. The GAT module aims to learn a dynamic representation of a poly-tree node at time step t. In detail, we initialize the node representation by the temporal representations at time step t, $\boldsymbol{n}_{v_k}^{(0)} = \boldsymbol{h}_{v,t}$. Note that the time index of $\boldsymbol{n}_{v_k}$ is omitted for simplicity.

A key point in GAT is the setting of attention weights between two nodes. Here we set the weights between two nodes v_i and v_j as

$$\alpha_{(v_i,v_j)} = \frac{\exp\left(\boldsymbol{w}^\top \cdot \left(\boldsymbol{W}_1 \boldsymbol{n}_{v_i} + \boldsymbol{W}_2 \boldsymbol{n}_{v_j}\right)\right)}{\sum_{v_k \in \mathcal{N}_{v_i}} \exp\left(\boldsymbol{w}^\top \cdot \left(\boldsymbol{W}_1 \boldsymbol{n}_{v_i} + \boldsymbol{W}_2 \boldsymbol{n}_{v_k}\right)\right)}, \quad (3)$$

where α is the attention weight, $\boldsymbol{w}$ and $\boldsymbol{W}_{(\cdot)}$ are learnable parameters, $\cdot^\top$ represents transposition, and $\mathcal{N}_{v_i}$ denotes the node neighbours of poly-tree node v_i. To capture the complex spatial influence and stabilize the learning process of self-attention, we employ the multi-head attention proposed in [8]. Here we use M independent attention mechanisms, resulting in the following output:

$$\boldsymbol{n}_{v_i}^{(z)} = \Big\|_{m=1}^{M} \mathrm{ReLU}\left(\sum_{v_k \in \mathcal{N}_{v_i}} \alpha_{(v_i,v_k)}^{(m)} \boldsymbol{W}^{(m)} \boldsymbol{n}_{v_k}^{(z-1)}\right), \quad (4)$$

where $\|$ represents concatenation, $\alpha_{(v_i,v_k)}^{(m)}$ is the normalized attention coefficients computed by the m-th attention mechanism, and $\boldsymbol{W}^{(m)}$ is the corresponding input linear transformation's weight matrix.

Finally, we use a MLP-based predictor to estimate the potential energy of poly-tree node v at time step t, a.k.a., $\hat{p}_{v,t}$. To train the spatial-temporal model, we compute the mean squared error loss by

$$\mathcal{L} = \left(p_{v,t} - \hat{p}_{v,t}\right)^2. \quad (5)$$

E. Flow Pediction from Potential Energy

After a potential energy field is predicted in every community, the future flow prediction component will derive flow poly-tree from the potential energy fields and predict the city-wise flow graph by combining the flow poly-trees of each community. A detailed algorithm can be found in Algorithm 2.

Algorithm 2 Potential Energy to Flow Algorithm

Input: List of potential energy field P, list of community C.
Output: City-wise flow graph G.

```
Initialize flow graph G.
for c ∈ C do
    Initialize flow graph G_c of community c.
    for p ∈ P do
        Initialize temporary flow graph G_t.
        for (u, v) ∈ p do
            if p_u > p_v then
                G_t(u, v) = p_u − p_v.
            else
                G_t(v, u) = p_v − p_u.
        Add G_t to G_c.
    Get the border of c.
    Add G_c to G according to the border.
return G.
```

IV. Experiment

In this section, we conduct extensive experiments on three real-world trajectory datasets to evaluate our ST-PEF.

A. Experimental Setup

Datasets. Xi'an taxi dataset comes from GAIA Open Dataset. It contains 3,784,063 trajectories of taxicab sampled every 3 seconds. Beijing taxi dataset contains 60,828,387 taxicab GPS trajectories sampled every minute. Porto taxi dataset is originally released for a Kaggle trajectory prediction competition with a sampling period of 15 seconds. The dataset contains 1,891,921 trajectory. For each dataset, we choose the previous 80% as training data and the rest as testing data.

Preprocessing. We split city area as 18×18, 32×32, and 12×8 regions for Xi'an, Beijing, and Porto, making the real-world size of each region is about $500m \times 500m$ in all three cities. In order to evaluate the performance on different prediction lengths, we set the time interval as 5, 30, and 60 minutes for Xi'an, Beijing, and Porto datasets, respectively. The sliding window method is utilized for sample generation on both training and testing data.

Evaluation Metrics & Baselines. We evaluate our method and baselines by Root Mean Square Error (RMSE) and Coefficient of Determination (R^2). We compare ST-PEF with the following baselines: (1) Historical Average (HA), (2) ARIMA, (3) Spatio-Temporal Auto-Regressive (STAR) [11], (4) LSTM, (5) XGBoost, (6) Convolutional LSTM (CLSTM), (7) Spatio-Temporal Residual Convolutional Network (STResNet) [2], (8) Spatial-Temporal Dynamic Network (STDN) [1].

Task Setting. For each dataset, we use the previous 90% of the training data to train the model and the remaining 10% to optimize the model. On all the test datasets, we generate three types of tasks using the next 1, 2, and 4 time steps. Then we set the hyperparameters based according to the performance on the validation set. The batch size is set as 100. The number of community C is 4, 12, and 8 for Xi'an, Beijing, and Porto datasets. A flow graph in a community has $K = 4$ flow poly-trees components. For temporal information, we set the length of short-term GRU

Table I
PERFORMANCE COMPARISON USING TWO METRICS ON THREE DATASETS. "M" AND "H" DENOTE MINUTE AND HOUR, RESPECTIVELY. THE RESULTS ARE BETTER WITH SMALLER RMSE, BUT LARGER $\mathbf{R^2}$.

Metric	**RMSE**									$\mathbf{R^2}$								
Datasets	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	3.845	3.858	3.778	10.052	10.547	11.741	4.580	4.689	4.755	0.256	0.252	0.283	0.610	0.571	0.469	0.245	0.209	0.186
ARIMA	3.040	3.040	3.040	11.490	11.490	11.490	3.870	3.870	3.870	0.570	0.570	0.570	0.480	0.480	0.480	0.460	0.460	0.460
STAR	3.440	3.320	3.430	11.220	11.380	12.730	4.470	4.580	4.930	0.410	0.450	0.410	0.500	0.490	0.360	0.280	0.240	0.130
LSTM	1.990	2.040	2.140	7.960	9.150	11.220	**2.770**	3.200	4.440	0.800	0.790	0.770	0.750	0.670	0.470	0.720	0.630	0.290
XGBoost	1.970	2.000	2.140	5.370	6.880	8.960	2.930	3.370	3.960	0.810	0.800	0.770	0.860	0.810	0.680	0.690	0.590	0.440
CLSTM	1.801	1.903	1.988	5.682	6.414	7.880	2.847	3.173	3.693	**0.837**	0.818	**0.813**	0.875	0.841	0.761	0.709	0.638	0.509
STResNet	1.885	1.956	1.989	5.199	5.884	6.726	2.896	2.942	3.128	0.815	0.797	0.792	0.883	0.856	0.841	0.728	0.683	0.611
STDN	1.905	1.984	2.106	7.199	9.384	11.578	2.838	3.242	3.675	0.836	**0.822**	0.800	0.819	0.692	0.531	0.702	0.611	0.499
ST-PEF	**1.774**	**1.897**	**1.926**	**5.134**	**5.782**	**6.543**	2.814	**2.910**	**3.115**	0.821	0.809	0.803	**0.886**	**0.859**	**0.843**	**0.734**	**0.697**	**0.637**

Table II
SUMMARY OF PERFORMANCE COMPARISONS.

Metric	HA	ARIMA	STAR	LSTM	ST-PEF
Win times	0	0	0	1	**14**
AVG arithmetic ranking	8.444	7.056	8.167	5.278	**1.333**
AVG geometric ranking	8.408	6.964	8.151	4.923	**1.220**
Metric	XGBoost	CLSTM	STResNet	STDN	ST-PEF
Win times	0	2	0	1	**14**
AVG arithmetic ranking	4.889	2.778	2.722	4.222	**1.333**
AVG geometric ranking	4.794	2.587	2.559	3.909	**1.220**

(a) Flow volume (b) Potential energy

Figure 3. Spatial distribution of key regions on flow volume and potential energy. The background color are used to distinguish different community. Each small block is a region in reality. Pink and yellow denotes high and low value, respectively. When two colors are overlapped, we color it blue.

as 4, and the dimension of the hidden representation is 128. We set $M = 8$ in spatial modeling. MLP consists of one hidden layer with 128 hidden units. All the representations are initialized by a truncated normal distribution with zero mean and 0.01 variance, the biases are initialized as zero as well. Here we set the max epoch number as 100, then choose the Adaptive Moment Estimation (Adam) optimizer with learning rate 0.001 and Early-Stopping technique to train the model until convergence.

B. Results and Analysis

Table I shows the performance comparison of our ST-PEF and other baselines, with the summarized information listed in Table II. Note that the best performance is highlighted in bold. From Table I, we can see that the traditional autoregression methods don't perform well, including HA, ARIMA, and STAR. They are too simple to catch the temporal dynamics in traffic flow forecasting. However, with the benefit of deep learning, LSTM performs much better. By comparing the performance of LSTM and CLSTM, short for ConvLSTM, the spatial factor plays an important role in this task. So the models with good performance are all spatiotemporal ones. In Table II, among all the competitors, ST-PEF achieves the best performance in terms of both the largest number of wins (the best in 14 out of 18 experiments) and the smallest average ranking. Our model mainly differs from other spatiotemporal models in its PEF, which indicates that the PEF we proposed makes great sense.

By summarizing these results, we can see deep learning models perform much better than the statistical ones. Both temporal and spatial factors play an important role in traffic flow prediction. Our proposed model ST-PEF, which combines the potential energy field and deep learning technique, achieves good performance and interpretability.

C. Case Study

Next, we show the potential energy learned by our approach and demonstrate a case study on the Beijing taxi dataset.

Figure 3 shows the spatial distribution of key regions on flow volume and potential energy. For a flow graph, we can calculate the inflow and outflow of each region by easily adding the flow leaving from this region and flow coming to this region in each direction. Then we choose the top 4% outflow regions of each community as outflow key regions and color them pink in Figure 3(a), while inflow key regions are colored yellow. If the key regions of inflow and outflow are overlapped, we color them blue. The same operations are repeated on potential energy fields to get Figure 3(b). It can be observed that there exist many overlapping regions in the flow graph but no one in the potential energy field, indicating potential energy is easier to understand. For example, a region that has a higher outflow at the evening peak may be defined as an office area in many studies. But we argue that if the inflow is high as well, the region is more likely

to be a transportation station. However, the potential energy can define a functional region clear, where the high potential energy region in the evening is an office area and low means a residential area. On the other hand, the key regions of the flow graph scatter on the whole city, while those of potential energy tend to gather and form a contiguous zone which may be a functional area. In a word, potential energy fields show better patterns and are easier to understand.

V. Related work

Existing works on traffic flow prediction can be divided into two categories: traditional statistical models and deep learning models. Traditional statistical models, such as ARIMA and STAR [11], assume traffic data to follow certain distributions and may not capture the complex, non-linear, and dynamic relationships in traffic flow data. Deep learning models include purely temporal models, purely spatial models, and spatiotemporal models (e.g., ST-ResNet [2] based on residual network, STDN [1] based on LSTM, ITRCN [12] based on GRU). Besides, graph neural networks have also been proposed for traffic flow prediction that uses graph convolution instead of 2D image convolution, such as DCRNN [13]. Deep learning models have shown promising prediction accuracy compared with traditional statistical models. However, the prediction model is not easily interpretable.

There exist some research on interpretable machine learning [14] (or explainable AI [15]). Related work can be divided into intrinsic interpretation and post-hoc interpretation according to the time when the interpretation is obtained. Intrinsic interpretation aims to construct a self-explainable model to help a human understand the overall logic behind the model and its internal working mechanism [16], such as decision tree and capsules [17]. Post-hoc interpretation requires creating a second model to explain an existing model, such as LIME [18], SHAP [19], and Anchor [20]. These methods assume that the predictions around the neighborhood of a given input can be approximated by an interpretable white-box model. Then they identify the contributions of each feature in the input towards a specific prediction made by a model. Our proposed model belongs to intrinsic interpretation since we develop a deep learning model that is partly self-explainable. Contrast by existing self-explainable models and deep learning models with post-hoc explanation methods (such as ST-ResNet plus SHAP), we directly model the underlying physics of traffic flows based on potential energy fields.

VI. Conclusion

In this paper, we propose a spatiotemporal deep learning model for the ST-PEFs, which consists of GRU for temporal dynamics as well as graph attention networks for spatial dependencies. Experimental results on three real-world traffic datasets show the effectiveness of our model compared to the state-of-the-art methods. Furthermore, a case study confirms the ability of our interpretable model to reveal some hidden urban dynamic patterns that are not apparent in present flow prediction models.

Acknowledgment

Dr. Jingyuan Wang's work was partially supported by the National Key Research & Development Program of China (Grant No. 2019YFB2102100), the National Natural Science Foundation of China (Grant No. 71531001, 61572059), the Fundamental Research Funds for the Central Universities (Grant No. YWF-20-BJ-J-839) and CCF-DiDi Gaia Collaborative Research Funds for Young Scholars.

References

[1] 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.
[2] J. Zhang, Y. Zheng, D. Qi, R. Li, X. Yi, and T. Li, "Predicting citywide crowd flows using deep spatio-temporal residual networks," *AAAI*, 2018.
[3] B. L. Smith, B. M. Williams, and R. K. Oswald, "Comparison of parametric and nonparametric models for traffic flow forecasting," *Transportation Research Part C: Emerging Technologies*, 2002.
[4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, 2015.
[5] J. Zhang, Y. Zheng, D. Qi, R. Li, and X. Yi, "Dnn-based prediction model for spatio-temporal data," in *Proceedings of the 24th ACM SIGSPATIAL*, 2016.
[6] X. Geng, Y. Li, L. Wang, L. Zhang, Q. Yang, J. Ye, and Y. Liu, "Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting," in *AAAI*, 2019.
[7] K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," in *EMNLP*, 2014.
[8] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *ICLR*, 2018.
[9] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical review E*, 2004.
[10] C. S. J. Nash-Williams, "Edge-disjoint spanning trees of finite graphs," *Journal of the London Mathematical Society*, 1961.
[11] 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*, 1998.
[12] X. Cao, Y. Zhong, Y. Zhou, J. Wang, C. Zhu, and W. Zhang, "Interactive temporal recurrent convolution network for traffic prediction in data centers," *IEEE Access*, 2017.
[13] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *ICLR*, 2018.
[14] M. Du, N. Liu, and X. Hu, "Techniques for interpretable machine learning," *Communications of the ACM*, 2019.
[15] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins *et al.*, "Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI," *Information Fusion*, 2020.
[16] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *CSUR*, 2018.
[17] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *NIPS*, 2017.
[18] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?" explaining the predictions of any classifier," in *KDD*, 2016.
[19] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *NIPS*, 2017.
[20] M. T. Ribeiro, S. Singh, and C. Guestrin, "Anchors: High-precision model-agnostic explanations," in *AAAI*, 2018.