STDEN: Towards Physics-guided Neural Networks for Traffic Flow Prediction

Jiahao Ji, [1] **Jingyuan Wang,** [1,2*] **Zhe Jiang,** [3] **Jiawei Jiang,** [4] **Hu Zhang** [4]

[1] State Key Laboratory of Software Development Environment,
School of Computer Science & Engineering, Beihang University, Beijing, China
[2] Peng Cheng Laboratory, Shenzhen, China
[3] Department of Computer & Information Science & Engineering, The University of Florida
[4] MOE Engineering Research Center of Advanced Computer Application Technology,
School of Computer Science & Engineering, Beihang University, Beijing, China
{jiahaoji, jywang, jwjiang, zhanghu2006}@buaa.edu.cn, zhe.jiang@ufl.edu

Abstract

High-performance traffic flow prediction model designing, a core technology of Intelligent Transportation System, is a long-standing but still challenging task for industrial and academic communities. The lack of integration between physical principles and data-driven models is an important reason for limiting the development of this field. In the literature, *physics-based* methods can usually provide a clear interpretation of the dynamic process of traffic flow systems but are with limited accuracy, while *data-driven* methods, especially deep learning with black-box structures, can achieve improved performance but can not be fully trusted due to lack of a reasonable physical basis. To bridge the gap between purely data-driven and physics-driven approaches, we propose a physics-guided deep learning model named Spatio-Temporal Differential Equation Network (STDEN), which casts the physical mechanism of traffic flow dynamics into a deep neural network framework. Specifically, we assume the traffic flow on road networks is driven by a latent potential energy field (like water flows are driven by the gravity field), and model the spatio-temporal dynamic process of the potential energy field as a differential equation network. STDEN absorbs both the performance advantage of data-driven models and the interpretability of physics-based models, so is named a *physics-guided* prediction model. Experiments on three real-world traffic datasets in Beijing show that our model outperforms state-of-the-art baselines by a significant margin. A case study further verifies that STDEN can capture the mechanism of urban traffic and generate accurate predictions with physical meaning. The proposed framework of differential equation network modeling may also cast light on other similar applications.

Introduction

Rapid urbanization has brought about the growth of urban population, and presented huge transportation and sustainability challenges to modern cities. Intelligent Transportation System (ITS) has become an active research area because of its potential to improve transportation efficiency and solve the sustainability problem of cities (Snyder and Do 2019). As the core technology of the ITS, *traffic flow prediction*, aiming at forecasting the future status of traffic systems given historical observations, plays a crucial role in many important urban applications, such as public safety, congestion management and navigation (Li et al. 2020).

In the literature, traffic flow prediction methods mainly fall into two categories: *physics-based* and *data-driven*. The former one usually relies on traffic flow theory (Ni 2015), which represents traffic system as coupled Differential Equations (DEs). Traffic flow prediction is then achieved through conducting system simulation governed by these DEs. The physics-based models are able to guarantee the simulation results consistently represent the traffic dynamics over the entire domain, not only where it was calibrated by observation data. However, these models usually make strong assumptions about traffic flow with a small set of parameters (Mo, Shi, and Di 2021), which may not be able to capture the complex human behaviors and the uncertain factors in real-world traffic. Besides, the simulation process relies on solving DEs through numerical differentiation and integration techniques, which requires a lot of computing resources (Wang et al. 2020).

The second category is data-driven methods, which usually utilize historical observational data to train a statistical learning model, and then use the trained model to generate predictions. Among the data-driven methods, the most representative branch is traffic flow prediction based on deep learning. For example, using recurrent neural networks (Li et al. 2018) or temporal convolution (Wu et al. 2019) to model temporal dependencies, using convolutional neural networks (Tang et al. 2020) to capture spatial correlations, and using graph convolutional (Song et al. 2020; Tian and Chan 2021) to introduce road network information into traffic prediction. In recent years, with a huge volume of traffic data becoming available, the deep learning-based data-driven methods have drawn great attention from both industry and academia, and achieved great success in many real-world applications. However, these methods also have defects. First, without the physical knowledge to guarantee generalization ability, the data-driven models are very likely to lose effectiveness in scenarios that are not sampled by the training data. Second, the "black-box" structure of deep learning models introduces unknown risks in the ITS, which may cause potential threats to urban safety.

*Corresponding author: jywang@buaa.edu.cn

To bridge the gap between data-driven and physics-based approaches, we raise a hybrid modeling paradigm, *Physics-guided Deep Learning*, for traffic flow prediction. Specifically, we propose a Spatio-Temporal Differential Equation Network (STDEN[1]) that combines the physical mechanism of traffic dynamics and end-to-end deep learning into a whole framework. Our idea is based on the key assumption that traffic flow on road networks is driven by a latent potential energy field (like water flows are driven by the gravity field). The latent potential energy field follows physical constraints that reflect the transport of energy (Lienhard and Lienhard 2008). These constraints take the form of differential equations (DEs) commonly used in physics. To capture complex functions in the potential energy field DE in a learnable manner, we extend the existing ordinary DE network (Chen et al. 2018) by replacing the differential operator with graph Laplacian and design a graph neural network for spatio-temporal road networks. The overall framework of STDEN consists of an encoder that maps traffic flow into latent potential energy fields, a DE network that predicts the dynamics of potential energy fields continuously over time, and a decoder that generates traffic flow predictions from the latent potential energy fields. Evaluations on real-world traffic datasets show that STDEN consistently outperforms state-of-the-art traffic prediction baselines by a large margin in terms of prediction accuracy. Moreover, the learned potential energy field can potentially reveal the evolution of urban dynamics, thereby explaining changes in traffic flow. In summary, our contribution is three-fold:

- We model the fundamental physical mechanism of urban dynamics using potential energy fields, and introduce the physical mechanism into a data-driven deep learning model for traffic prediction. To the best of our knowledge, this is the first work that proposes a physics-based and data-driven mixed, *i.e.,* physics-guided, deep learning model for traffic flow prediction on road networks.
- We propose a novel hybrid deep learning model, STDEN, that unifies the traffic potential energy field DE and neural networks into one framework. This modeling paradigm may cast light on other DE-based applications, such as weather forecasting and epidemic prediction.
- We conduct extensive experiments on three real-world traffic datasets, and the proposed method achieves significant improvement over state-of-the-art baselines. Moreover, a case study confirms that the learned potential energy field can reveal the physical mechanism of traffic flow and provide interpretability for the deep traffic prediction model.

Preliminaries

We study the traffic flow prediction problem over urban road networks. A list of major symbols is in Tab. 1.

Definition 1 (Road Network)**.** *A* Road Network *is a directed graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$*, where* $\mathcal{V} = \{v_1, \ldots, v_n\}$ *is a set of nodes,* $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ *is a set of edges, and* $W \in \mathbb{R}^{n \times n}$ *is a weighted adjacency matrix. A node* $v_i \in \mathcal{V}$ *represents a road junction or a road end, while an edge* $e_{ij} \in \mathcal{E}$ *represents a directed road segment from node* v_i *to node* v_j*.*

[1]The code is available at https://github.com/Echo-Ji/STDEN

Table 1: List of major symbols and descriptions.

Sym.	Domain	Descriptions
n	$\mathbb{R}$	Number of nodes in a road network
W	$\mathbb{R}^{n \times n}$	Adjacency matrix of a road network
$\boldsymbol{f}^{(t)}$	$\mathbb{R}^{\lvert\mathcal{E}\rvert}$	Traffic flow of a road network at time t
$\boldsymbol{z}^{(t)}$	$\mathbb{R}^{n}$	Potential field of a road network at time t
$\boldsymbol{u}, \boldsymbol{q}$	$\mathbb{R}$	Energy density, energy flux
$\boldsymbol{\phi}$	$\mathbb{R}^{n}$	Node volume
α^{-1}	$\mathbb{R}$	Contribution ratio of traffic flow

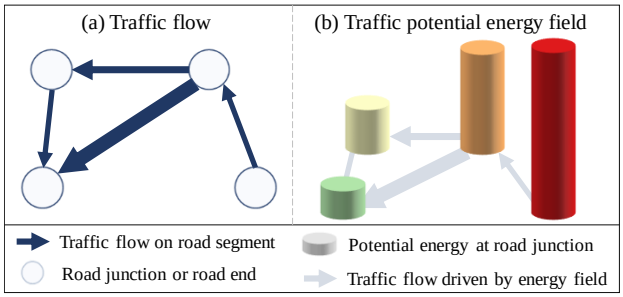


Figure 1: Illustration of traffic flow and potential energy field defined on road networks. The arrow indicates the traffic flow direction while its size denotes the flow volume. In panel (b), higher cylinder means more potential energy, and the traffic flow volume (arrow size) increases with the energy gradients between adjacent nodes.

Definition 2 (Traffic Flow)**.** *We define* Traffic Flow *as a feature of edges in the road network. Given an edge* e_{ij}*, the overall traffic flow during a give time period is denoted as* f_{ij}*. We express the traffic flow of the whole road network as a vector* $\boldsymbol{f} = (f_{ij}) \in \mathbb{R}^{|\mathcal{E}|}$*.* $|\mathcal{E}|$ *is the size of the edge set.*

Definition 3 (Traffic Potential Energy Field (PEF))**.** *For each node* i *of a road network* $\mathcal{G}$*, we define the* Traffic Potential Energy $z_i \in \mathbb{R}$ *as its feature. The* Traffic Potential Energy Field *on the whole road network is denoted as* $\boldsymbol{z} = (z_1, \ldots, z_n)^\top \in \mathbb{R}^n$*.*

As shown in Fig. 1, the potential energy field on the road network is the latent dominated force of the traffic flow, which is similar to the gravity field driving water flow. The traffic flow of the corresponding edge is the potential energy gradient between adjacent nodes,

$$f_{ij} = -(\nabla z)_{ij} = -(z_i - z_j), \tag{1}$$

where ∇ is the graph gradient operator, and the negative sign ahead of ∇ indicates the direction of flow. According to Eq. (1), the traffic flow on the road network can be explained as a "energy transport" process of the traffic potential energy field. This is a fundamental insight of our model.

Definition 4 (Flow-sequence and PEF-sequence)**.** *We divide time as regular time slices. Let* $\boldsymbol{f}^{(t)}$ *represent traffic flow observed at time slice* $t \in \mathbb{N}$*.* Flow-sequence *is the time series of traffic flow* $\boldsymbol{F}^{(0:t)} = \left(\boldsymbol{f}^{(0)}, \ldots, \boldsymbol{f}^{(t)}\right)$*, while* PEF-sequence *is that of PEF* $\boldsymbol{Z}^{(0:t)} = \left(\boldsymbol{z}^{(0)}, \ldots, \boldsymbol{z}^{(t)}\right)$*.*

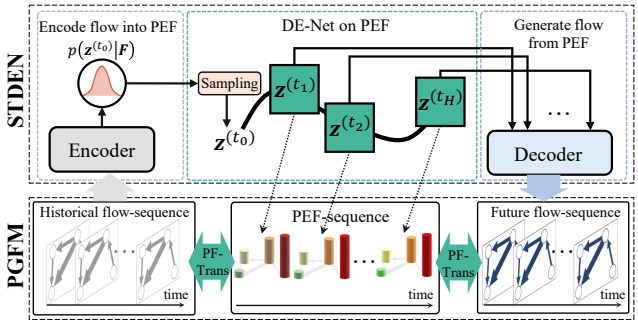


Figure 2: Overview of the proposed physics-guided model. STDEN: Spatio-Temporal Differential Equation Network. PGFM: Physics-Guided traffic Flow Modeling. PF-Trans: transformation between the PEF and traffic flow (Eq. (5)).

Problem Definition

Based on the basic concepts above, we formally define the problem of traffic flow prediction as follows.

• **Input:** The history flow-sequence from time $t-T+1$ to t, $\boldsymbol{F}^{(t-T+1:t)}$, the future flow-sequence from time $t+1$ to $t+H$, $\boldsymbol{F}^{(t+1:t+H)}$, and the corresponding road network $\mathcal{G}$.

• **Output:** A model $m(\cdot)$ satisfying $\hat{\boldsymbol{F}}^{(t+1:t+H)} = m\left(\boldsymbol{F}^{(t-T+1:t)}\right)$.

• **Objective:** Minimizing prediction errors of training data.

Physics-guided Traffic Flow Modeling

The Model Framework

Data-driven methods directly model the correlation between $\boldsymbol{F}^{(t-T+1:t)}$ and $\boldsymbol{F}^{(t+1:t+H)}$ to generate predictions. We innovatively propose a physics-guided traffic flow prediction framework (see Fig. 2), consisting of two parts: an explicit deep learning model STDEN and an implicit physical dynamic process PGFM. In PGFM, *i.e.,Physics-Guided traffic Flow Modeling*, we use a physics-based continuity equation to model the transformation between the PEF and the traffic flow (denoted as PF-Trans in Fig. 2). Details of the implicit PGFM will be given in this section.

Over the implicit physical dynamic process, we implement a deep neural network model, *Spatio-Temporal Differential Equation Network* (STDEN), which contains three components: an RNN-based network to encode the traffic flow-sequence as the initial state of PEF, a DE network to predict the evolution of the PEF, and a decoder governed by Eq. (1) to generate traffic flow from the predicted PEF. Note the layers of DE-Net correspond to the dynamic of potential energy in PGFM. Details of STDEN are in Section .

In our model, the physical dynamic process and the deep learning model are unified under the same framework, that is why we call our model a *"Physics-guided"* neural network.

Physics-based Continuity Equation

In this section, we use a continuity equation to model the relations between the flow-sequence and PEF-sequence. First, we provide a brief introduction to the continuity equation.

In physics, the continuity equation describes the transport of some physical quantity, such as mass, energy, momentum, electric charge (Lienhard and Lienhard 2008). Here we take the transport of energy as an example. Using the continuity equation, the energy transport process over a set of locations could be described as

$$\frac{\partial \boldsymbol{u}}{\partial t} + \operatorname{div} \boldsymbol{q} = 0, \tag{2}$$

where $\boldsymbol{u}$ is the energy density (energy per unit volume), $\boldsymbol{q}$ is the *Energy Flux* (a measure of energy "transport"), and div is the divergence operator. The differential equation in Eq. (2) gives a relation between the volume of the energy and the "transport" of that energy, *i.e.,* the change of energy density leads to energy transport in the space. For example, heat always transports from hot locations to cold locations.

Table 2: Analogy from potential energy-driven traffic flow to energy continuity equation.

Potential energy field	$\longrightarrow$	Energy field
Potential energy per unit volume	$\longrightarrow$	Energy density
Traffic flow	$\longrightarrow$	Energy flux

Continuity Equation for Traffic PEF

The traffic flow driven by traffic potential energy fields in a road network is naturally a specific example of the continuity equation. In Tab. 2, we draw an analogy between the potential energy-driven traffic flow and the energy continuity equation, where traffic flow on a road network could be considered as energy transport within a Potential Energy Field. In this section, we propose a continuity equation for traffic PEF to describe the dynamic relations between flow-sequence and PEF-sequence of a road network.

Energy Density and Traffic Potential Energy. Given a road network $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, W\}$, for $v_i \in \mathcal{V}$, we define it having a *Energy Density* u_i. The potential energy of the v_i is proportional to its energy density as $z_i = \phi_i \cdot u_i$, where ϕ_i is the trainable node volume. The node volume is determined by the endogenous feature of the node, which is in analogy with the mass in the gravity field. For all nodes in $\mathcal{V}$, we have the relation between the potential energy $\boldsymbol{z} = (z_1, \ldots, z_n)^\top$ and the energy density $\boldsymbol{u} = (u_1, \ldots, u_n)^\top$

$$\boldsymbol{z} = \boldsymbol{\phi} \odot \boldsymbol{u}, \tag{3}$$

where $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_n)^\top$ and $\odot$ is the Hadamard product.

Energy Flux and Traffic Flow. In a traffic system defined on $\mathcal{G}$, the traffic potential energy can only be transported along the edges (road segments) in the edge set $\mathcal{E}$. So we define the *Energy Flux* of $e_{ij} \in \mathcal{E}$ as q_{ij}. We consider the traffic flow f_{ij} as the major component of the energy flux q_{ij}, and introduce a shared parameter α to measure the contribution ratio of the traffic flow to energy flux for every edge,

$$\boldsymbol{f} = \alpha^{-1}\boldsymbol{q}, \tag{4}$$

where $\boldsymbol{f} = (f_{ij})^\top$ and $\boldsymbol{q} = (q_{ij})^\top$.

Potential Energy DE Function. Plugging Eq. (3) and (4) into the energy continuity equation in Eq. (2), we have

$$\boldsymbol{\phi}^{-1} \odot \frac{\partial \boldsymbol{z}}{\partial t} + \alpha \operatorname{div} \boldsymbol{f} = 0, \tag{5}$$

which is the continuity equation for energy transport in the traffic potential energy field. According to Eq. (1), traffic flow on edges of a road network is the gradient of potential energy between adjacent nodes, *i.e.,* $\boldsymbol{f} = -\nabla \boldsymbol{z}$. Using it to replace the traffic flow $\boldsymbol{f}$ in Eq. (5), we have

$$\boldsymbol{\phi}^{-1} \odot \frac{\partial \boldsymbol{z}}{\partial t} - \alpha \operatorname{div} \nabla \boldsymbol{z} = 0. \tag{6}$$

This function is a *Differential Equation* (DE) about the potential energy field $\boldsymbol{z}$, so we name it as *Potential Energy Field DE*. The graph Laplacian operator Δ is given by the divergence of the gradient (Bronstein et al. 2017), *i.e.,*$\Delta = -\operatorname{div} \nabla$, so the potential energy field DE can be transformed into

$$\frac{\partial \boldsymbol{z}}{\partial t} = -\boldsymbol{\phi} \odot (\alpha \Delta \boldsymbol{z}). \tag{7}$$

Physics Interpretation of the Potential Energy Field DE. Eq. (7) is the dynamical equation to express the evolution of the PEF-sequence $\boldsymbol{Z} = \left(\boldsymbol{z}^{(t)}\right)$. From a discrete perspective, Eq. (7) could be expressed as the form of

$$\boldsymbol{z}^{(t+1)} = \boldsymbol{z}^{(t)} - \boldsymbol{\phi} \odot \left(\alpha \Delta \boldsymbol{z}^{(t)}\right), \tag{8}$$

which gives an evolution trajectory of the PEF-sequence $\boldsymbol{Z}$.

Spatio-Temporal Differential Equation Network

In Eq. (7), the evolution of PEF-sequence is expressed as a Potential Energy Field DE form. In this section, we implement Eq. (7) using a neural network approach to enhance the modeling capability of the potential energy DE for real-world traffic data. The framework of our model is in Fig. 2.

Differential Equation Network

Our method is based on the neural ordinary *Differential Equation Network* (DE-Net) (Chen et al. 2018), which is a kind of model that generalizes standard layer-to-layer neural networks as a continuous form. Specifically, in standard residual networks (He et al. 2016), the layer-to-layer state update process is in the form of

$$\boldsymbol{h}_{t+1} = \boldsymbol{h}_t + \mathcal{F}(\boldsymbol{h}_t, \theta_t), \tag{9}$$

where $\boldsymbol{h}_t$ is hidden state of (*i.e.,* layer outputs) of the t-th layer[2]. $\mathcal{F}(\cdot)$ is the repeated network structure of the residual networks and θ_t is the network parameters at layer t.

Eq. (9) could be rewritten as a generalized form as

$$\frac{\boldsymbol{h}_{t+l} - \boldsymbol{h}_t}{(t+l) - t} = \mathcal{F}(\boldsymbol{h}_t, \theta_t), \tag{10}$$

where l is the step size and its value is 1 in residual networks. When $l \to 0$, the update process of state $\boldsymbol{h}$ becomes

$$\frac{\partial \boldsymbol{h}(t)}{\partial t} = \mathcal{F}\left(\boldsymbol{h}(t), t, \theta\right), \tag{11}$$

which is a continuous version of the residual networks. In Eq. (11), the deep residual neural network becomes a dynamic system that is governed by an ordinary differential equation (Ruthotto and Haber 2019; Lu et al. 2018). Moreover, the function $\mathcal{F}(\cdot, \theta)$ is a neural network, which provides powerful representation learning capacity for complex data.

[2]In our model, the layer of DE-Net just corresponds to time slices, so we use t as the index of the network layers in Eq. (9).

DE-Net for Traffic Potential Energy Fields

DE-Net on Potential Energy Fields. Inspired by the neural ordinary differential equation network, we model the potential energy field DE in Eq. (7) as,

$$\frac{\partial \boldsymbol{z}^{(t)}}{\partial t} = \mathcal{F}_{\mathcal{G}}\left(\Phi, t, \boldsymbol{z}^{(t)}\right), \tag{12}$$

where Φ represents all trainable parameters including $\boldsymbol{\phi}$ and α. The function $\mathcal{F}_{\mathcal{G}}$ is a neural network guided by the physics model in Eq. (7). According to Eq. (7), we express the function $\mathcal{F}_{\mathcal{G}}$ as a residual graph convolution network (GCN) form, where the repeated neural network layer is in the form of

$$\mathcal{F}_{\mathcal{G}}\left(\Phi, t, \boldsymbol{z}^{(t)}\right) = -\boldsymbol{\phi} \odot \operatorname{Tanh}(\alpha \Delta \boldsymbol{z}). \tag{13}$$

where Δ is the graph Laplacian operator to calculate the differences between the state z_i of the node i and its neighbor's. This is equivalent to using α as a convolution kernel to aggregate the states of nodes in a receptive field. $\operatorname{Tanh}(\cdot)$ is a Hyperbolic Tangent activation function. The results of the convolution are combined using the weights from ϕ_i.

From the spatial perspective, if we set t as a discrete value, Eq. (13) is equivalent to a residual GCN, where inputs of each layer are $z^{(t)}$ for all nodes of the road network. From the temporal perspective, the time t is continuous, meaning we can calculate $\boldsymbol{z}^{(t)}$ for any $t \in \mathbb{R}$. Therefore, we name the proposed model based on the differential equation network in Eq. (13) as *Spatio-Temporal DE Network*.

Encode Traffic Flow into Potential Energy Fields. In our model, the PEF-sequence $\boldsymbol{Z}^{(t_1:t_H)}$ can be calculated using a neural ODE solver (Chen et al. 2018) for a given initial state $\boldsymbol{z}^{(t_0)}$,

$$\boldsymbol{Z}^{(t_1:t_H)} = \operatorname{ODEsolver}\left(\mathcal{F}_{\mathcal{G}}, \Phi, \boldsymbol{z}^{(t_0)}, [t_0, \ldots, t_H]\right), \tag{14}$$

where $\boldsymbol{z}^{(t_0)}$ is sampled from a distribution $p\left(\boldsymbol{z}^{(t_0)}\right)$. We implement the distribution as a conditional probability distribution of historical traffic flow sequence $\boldsymbol{F}^{(t_0-T+1:t_0)}$. Specifically, we let $\boldsymbol{z}^{(t_0)}$ is generated from a Gaussian distribution, where the mean and standard deviation are determined by the historical traffic flow sequence $\boldsymbol{F}^{(t_0-T+1:t_0)}$ as

$$\begin{aligned} &p\left(\boldsymbol{z}^{(t_0)} \middle| \boldsymbol{F}^{(t_0-T+1:t_0)}\right) = \mathcal{N}\left(\mu_{\boldsymbol{z}^{(t_0)}}, \sigma_{\boldsymbol{z}^{(t_0)}}\right), \\ &\text{where } \left\{\mu_{\boldsymbol{z}^{(t_0)}}, \sigma_{\boldsymbol{z}^{(t_0)}}\right\} = g\left(\operatorname{GRU}\left(\boldsymbol{F}^{(t_0-T+1:t_0)}\right)\right). \end{aligned} \tag{15}$$

We employ Gated Recurrent Unit (GRU) (Chung et al. 2014) as encoder to extracts information from $\boldsymbol{F}^{(t_0-T+1:t_0)}$. $g(\cdot)$ is a fully connected network to translate the final hidden states of GRU into the mean and standard deviation of $\boldsymbol{z}^{(t_0)}$.

To achieve a differentiable "sampling" operation, we adopt a reparametrization trick (Kingma and Welling 2014) to implement generation process $g(\cdot)$ of $\boldsymbol{z}^{(t_0)}$ in Eq. (15). In specific, given a batch of training data, we calculate $z_i^{(t_0)}$ of each node i as

$$z_i^{(t_0)} = \mu_{\boldsymbol{z}^{(t_0)}} + \epsilon_i \sigma_{\boldsymbol{z}^{(t_0)}}, \tag{16}$$

where ϵ_i is sampled from a standard normal distribution $\mathcal{N}(0, 1)$. In this way, $z_i^{(t_0)}$ for a given batch of training data are fixed, and therefore, Eq. (16) is differentiable in back-propagation algorithm for neural network training. In the prediction phase, the ϵ_i is sampled for every input example.

Generate Traffic Flow from Potential Energy Fields. Given the initial $\boldsymbol{z}^{(t_0)}$, we can calculate the entire PEF-sequence $\boldsymbol{Z}^{(t_1:t_H)}$ for future time steps. Next, we generate the flow-sequence $\boldsymbol{F}^{(t_1:t_H)}$ from $\boldsymbol{Z}^{(t_1:t_H)}$ by Eq. (1).

The Model Training

With the pipeline introduced above, we build a physics-guided latent variable model entitled *Spatio-Temporal Differential Equation Network* (STDEN). Using the back-propagation algorithm, the entire framework can be trained in an end-to-end manner by minimizing the negative log-likelihood of the predicted traffic flow-sequence with the ground truth in training data. The only difference compared with the training of standard neural network is the forward propagation of the DE-Net part needs to calculate using the neural ODE solver in Eq. (14).

Remark: In our model, the physical generation process of traffic flow in a potential energy field is expressed as deep residual GCN based DE-Net, providing a good explanation for our model structure. In other words, our model is not a complete "black-box" as other deep learning models. It could be considered as a kind of physics-guided generative model and therefore be expected to have better performance. Moreover, the elaborate residual DE-Net structure and the measure of uncertainty in potential energy fields also have the potential to improve the prediction performance.

Experiments

Datasets

We evaluate the performance of our model over the real-world urban traffic dataset collected by the Beijing Municipal Commission of Transport, which contains trajectories of 40,000 taxies in Beijing from April 1st 2015 to July 31st 2015 (totally 4 months). These trajectories mapped to the road networks using the map matching algorithm. We statistic the traffic flow by counting the number of taxis on each road segment during every 5-minute time interval, resulting in 288 data points per day.

Due to Beijing road networks are too complex, we select three sub-networks to construct datasets of our experiments. Without loss of dataset diversity, the selected sub-networks have different urban functions. The first one is a well-known entertainment area, *i.e.,* Gong Ti (Workers' Stadium), which has 221 road segments, the second is the area around Beijing West Railway Station, which has 393 road segments, and the last area is around the biggest business park in Beijing, *i.e.,* Zhongguancun (China Silicon Valley), which has 564 road segments. We denote the three datasets as GT-221, WRS-393, and ZGC-564 respectively. We split each dataset into the training, validation, and test sets with a ratio of 7:1:2. For multi-step prediction, we use one-hour historical traffic flow data (12 time steps) to predict the next hour's.

Experimental Settings

Baselines. We consider ten baselines that belong to three classes. (1) *Time series modeling:* We take Historical Average (HA), Vector Auto-Regression (VAR) and GRU as baselines. The history traffic flow-sequence are treated as purely time series to predict the futures state without consideration of spatial information. (2) *Graph-based spatio-temporal methods:* The classical graph-based tarffic prediction methods such as Diffusion Convolution Recurrent Neural Network (DCRNN) (Li et al. 2018), Spatial-Temporal Graph Convolutional Network (STGCN) (Yu, Yin, and Zhu 2018) and Graph WaveNet (GWNET) (Wu et al. 2019), and state-of-the-art models such as Adaptive Graph Convolutional Recurrent Network (AGCRN) (Bai et al. 2020) and MTGNN (Wu et al. 2020) are used for comparison. (3) *Approaches based on neural differential equation networks*: Here we use two classical methods Latent-ODE (Rubanova, Chen, and Duvenaud 2019) and ODE-LSTM (Lechner and Hasani 2020) for comparison. Latent-ODE generalizes RNNs to have continuous-time hidden dynamics defined by ordinary differential equations (ODEs). ODE-LSTM is a novel long short term memory network, that possesses a continuous-time output state, and consequently modifies its internal dynamical flow to a continuous-time model.

Settings. The settings of STDEN contains the following two parts: (1) Settings of the DE-Net part. We model the dynamics of potential energy in latent space using an adaptive method $dopri5$ (Dormand and Prince 1980), and conduct grid search on the latent dimension over $\{1, 2, 4, 8\}$. (2) Settings of the encoder. GRU is used to encode the distribution of the initial value of the PEF-sequence. The number of hidden units in GRU is searched over $\{16, 32, 64, 128\}$. More implementation details about our STDEN and other baselines settings are given in Appendix. We conduct experiments of all deep learning models with 7 different seeds and report the mean results.

Performance Comparison

Tab. 3 shows the comparison of different approaches for 15 minutes, 30 minutes, and 1 hour ahead forecasting on three datasets. These methods are evaluated by three commonly used metrics in traffic flow prediction, including mean absolute error (MAE), root mean square error (RMSE), and mean absolute percentage error (MAPE).

There are four observations from Tab. 3. (1) The graph neural network-based spatio-temporal methods generally outperform the time series models, which emphasizes the importance of modeling the spatial correlations of road networks for traffic prediction. (2) Our STDEN improves spatio-temporal methods with a significant margin (an average 10.29%/13.49%/6.03% improvement on MAE/RMSE/MAPE compared with the second best method) and achieves the best performance for all horizons on all the metrics, which shows the effectiveness of the potential energy field DE to model the continuous spatial and temporal dynamics. (3) The approaches based on purely neural differential equation network are not effec-

Table 3: Model comparison on metrics MAE/RMSE/MAPE, where MAPE is in %. Our STDEN *significantly* outperforms all competing baselines with regard to all metrics over all datasets according to Student's t-test at level 0.01.

Datasets	GT-221			WRS-393			ZGC-564		
Horizons	15 min	30 min	1 hour	15 min	30 min	1 hour	15 min	30 min	1 hour
HA	1.324/1.835/61.47	1.324/1.835/61.47	1.324/1.835/61.47	1.239/1.735/64.08	1.239/1.735/64.08	1.239/1.735/64.08	1.120/1.522/62.68	1.120/1.522/62.68	1.120/1.522/62.68
VAR	1.125/1.581/52.34	1.141/1.620/52.03	1.164/1.835/51.51	0.824/1.509/55.51	1.097/1.568/55.72	1.139/1.653/56.01	0.978/1.402/53.72	0.991/1.393/53.64	1.019/1.441/53.85
GRU	0.988/1.511/42.77	1.000/1.541/42.95	1.018/1.578/43.29	0.823/1.393/35.52	0.837/1.429/36.00	0.856/1.477/36.10	0.729/1.229/33.82	0.734/1.242/33.99	0.747/1.266/34.55
STGCN	0.976/1.528/41.40	0.988/1.552/42.08	1.008/1.585/43.20	0.803/1.400/33.20	0.818/1.436/33.69	0.841/1.490/34.31	0.714/1.218/32.63	0.724/1.241/33.13	0.738/1.272/33.78
DCRNN	0.976/1.520/41.10	0.994/1.553/41.75	1.020/1.601/42.46	0.803/1.386/33.51	0.826/1.435/34.46	0.858/1.500/35.83	0.715/1.230/31.99	0.727/1.258/32.34	0.746/1.298/32.86
AGCRN	0.968/1.502/41.18	0.981/1.525/41.73	1.004/1.561/42.79	0.799/1.382/33.28	0.814/1.414/33.96	0.841/1.468/34.93	0.709/1.210/32.19	0.717/1.228/32.66	0.733/1.259/33.24
GWNET	0.967/1.516/40.70	0.980/1.540/41.23	1.006/1.581/42.12	0.801/1.390/33.50	0.818/1.427/33.95	0.845/1.488/34.37	0.716/1.232/31.31	0.725/1.255/31.32	0.743/1.296/31.50
MTGNN	0.962/1.505/40.88	0.976/1.531/41.40	1.003/1.576/42.36	0.792/1.376/32.91	0.807/1.411/33.34	0.708/1.207/32.64	0.708/1.207/32.64	0.714/1.222/32.90	0.731/1.255/33.54
Latent-ODE	0.981/1.531/42.11	0.992/1.544/42.71	1.011/1.588/43,28	0.818/1.402/35.33	0.827/1.433/35.77	0.853/1.482/35.87	0.723/1.226/32.98	0.730/1.263/33.78	0.742/1.281/33.98
ODE-LSTM	0.978/1.522/42.87	0.986/1.561/41.80	1.007/1.579/43.22	0.809/1.391/35.28	0.818/1.428/35.12	0.848/1.478/35.14	0.718/1.219/32.75	0.725/1.252/32.65	0.740/1.274/33.36
STDEN	**0.865/1.317/36.72**	**0.872/1.328/37.37**	**0.896/1.360/39.04**	**0.730/1.241/31.62**	**0.737/1.244/32.04**	**0.745/1.266/33.25**	**0.623/1.026/28.81**	**0.628/1.036/29.14**	**0.657/1.033/31.58**

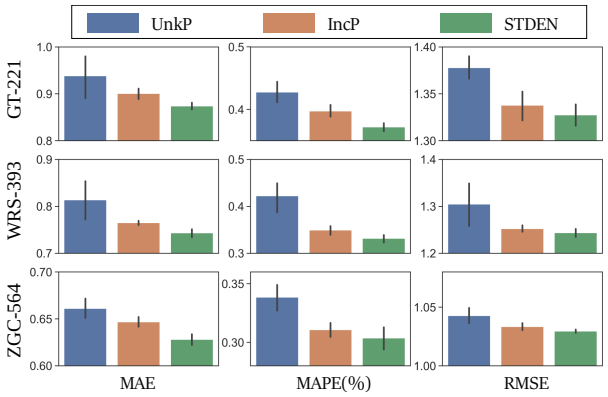
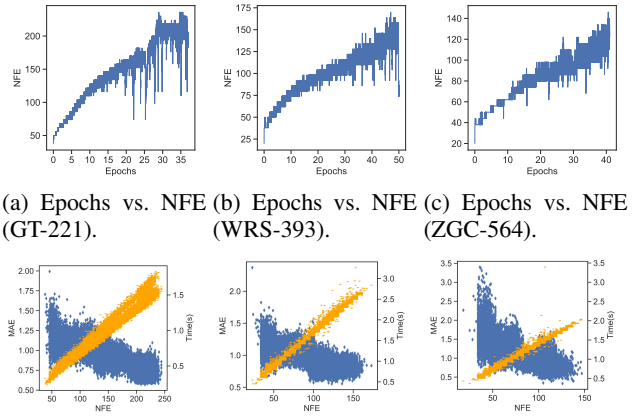


Figure 3: Evaluation on potential energy field differential equation over all datasets.



(a) Epochs vs. NFE (GT-221). (b) Epochs vs. NFE (WRS-393). (c) Epochs vs. NFE (ZGC-564).

(d) NFE vs. MAE (GT-221). (e) NFE vs. MAE (WRS-393). (f) NFE vs. MAE (ZGC-564).

Figure 4: Computation overhead vs. Prediction accuracy. NFE denotes number of function evaluation.

tive than the spatio-temporal ones. Because they ignore the spatial correlation which is important to traffic flow prediction. However, they outperform the traditional time series methods. (4) Traditional methods including HA and VAR are not good enough due to their incapability of handling complex and non-linear spatio-temporal data. Besides, the performance of HA is invariant since this method does not depend on short-term data.

Ablation Study

To better illustrate the effectiveness of the *Potential Energy Field DE* in Eq. (7), we compare STDEN with the following variants: (1) *UnkP*, which means unknown physics and replaces the potential energy field DE with a fully connected neural network. (2) *IncP*, which denotes incomplete physics and ignores the energy volume factor ϕ in the potential energy field DE, making it incomplete.

Fig. 3 shows the comparison of the two variants with regard to all metrics for different datasets. From the results, we have the following observations: (1) STDEN surpasses *NoP* by a large margin, demonstrating the superiority of deploying potential energy fields to model traffic flow. (2) That *IncP* beats *UnkP* shows the effectiveness of the potential energy field DE. The guidance of physical process expressed by the potential energy field DE introduces the diffusion process of energy in space. This allows *IncP* to capture the changing trend of potential energy fields throughout the road network. (3) However, *IncP* performs slightly worse than STDEN because of the absence of the energy volume factor ϕ. Without ϕ, *IncP* has to infer the potential energy only through the energy density, which is as hard as to calculate the mass of an object from density without object volume.

Computation Cost

During the solving process of the differential equation network, we utilize an ODE solver which allows our method to dynamically balance the trade-off between prediction accuracy and the computation cost. As shown in Fig. 4(a), 4(b) and 4(c), required number of function evaluation (NFE) increases consistently with the epochs, there is no stable stage because we set early stopping for the training process. Note that NFE varies with the difficulty of tasks. For example, flow prediction of GT-221 is the most complicated among all three tasks because the traffic flow in GT-221 changes more drastically. Therefore, the NFE is around 80% more than that of ZGC-564. From Fig. 4(d), 4(e) and 4(f), it can be observed that the more evaluation (NFE), the lower the prediction error our method can achieve. This allows us to trade accuracy for faster response for emergency events during inference phase, which is very valuable in practice.

Case Study

In this part, we qualitatively analyze why our STDEN can yield good performance. To this end, we visualize the learned potential energy and the real traffic flow in Fig. 5.

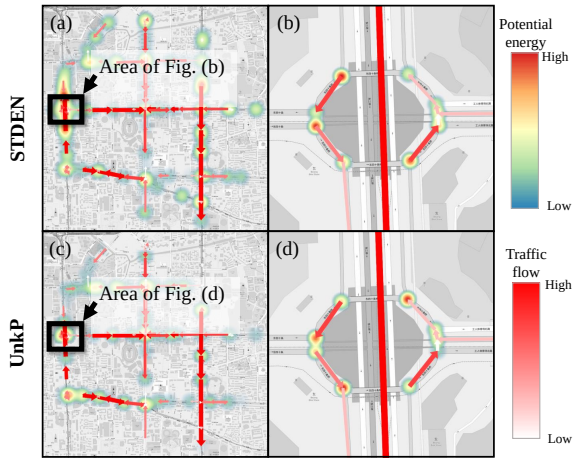


Figure 5: Visualization of the learned potential energy fields and real traffic flow on GT-221 dataset. The heat map represents the potential energy fields, while the arrows denote traffic flow with its volume reflected by the color and the arrow size. The potential energy fields learned by STDEN can interpret the traffic flow.

Fig. 5(a) and (c) shows the potential energy learned by STDEN and UnkP by a heat map, with real traffic flows at the corresponding moment represented by arrows. To see more clearly how the potential energy drives traffic flow, we select loop areas with complex traffic for detailed analysis. In Fig. 5(b), traffic always flows along the road network from a place with high potential energy to a place with low potential energy. Besides, the gradient of potential energy roughly reflects the relative magnitude of the traffic flow. This shows that potential energy governed by physical equation captures the mechanism of traffic flow and can be interpreted as a force that drives traffic flow. However, without the guidance of potential energy field DE, the energy learned by neural networks in UnkP (Fig. 5(d)) has no physical meaning and can not indicate the traffic flow mechanism. For example, there are some traffic flows that transport from a low-value node to a high-value node in Fig. 5(d), which violates the physical constraint of the energy transport process.

In a word, the results in Section verifies that STDEN can capture the physical mechanism of traffic flow (Fig. 5) and generate accurate predictions (Tab. 3) under the guidance of the potential energy field DE. This is because we combine the advantages of physics-based and data-driven methods.

Related Work

Traffic Prediction. Traffic prediction problem has been studied for decades, and existing methods mainly fall into two categories: *physics-based* and *data-driven*. In the former one, researchers apply different branches of traffic flow theory depending on the application problems (Ni 2015), such as the kinematic wave theory (Daganzo and Laval 2005), the car-following theory (Olstam and Tapani 2004), and the queuing theory (Cascetta 2013). However, they may only capture limited dynamics of real-world traffic, resulting in low-quality estimation of traffic flow. The *data-driven* approaches have drawn considerable attentions (Xie et al. 2020). The shallow machine learning methods for traffic prediction usually base on the stationary assumptions (*e.g.,* ARIMA and Kalman filtering(Lippi, Bertini, and Frasconi 2013)), leading to limited representation power. Deep learning methods are free from stationary assumptions and effective to capture complex non-linearity using models such as recurrent neural networks (Li et al. 2018) and temporal convolutional networks (Wu et al. 2019; Li and Zhu 2021). Because traffic data is spatial correlated, CNN (Tang et al. 2020) and its extension to arbitrary graphs (Song et al. 2020; Tian and Chan 2021) are utilized to capture spatial correlations. Although temporal dependencies and spatial correlations have been considered in these methods, the lack of physical knowledge leads to a lack of generalization ability to out-of-sample scenarios.

Physics-guided Deep Learning. Many recent studies have proposed to integrate physics-based modeling approaches with state-of-the-art deep learning techniques, giving birth to a field called "Physics-guided Deep Learning". One can introduce additional physics-based penalty in loss function of neural networks (Shi, Mo, and Di 2021). Some efforts also lie in combining physics-based models with deep learning. For example, (Wang et al. 2020) presents a hybrid framework that combines turbulent flow simulation with deep learning. (Ji et al. 2020) introduces the physical potential energy field concept into deep learning to achieve grid-based urban traffic prediction. However physics-based models governed by differential equations are usually continuous in time, while deep learning are dominated by discrete models (Ruthotto and Haber 2019). Recent advances propose a treatment of neural networks equipped with a continuum of layers (Chen et al. 2018), allowing a more accurate and natural modeling of physical principles in real world, *e.g.,* hydropower generation (Zhou et al. 2020) and reservoir flow (Zhou and Li 2021). To the best of our knowledge, we are the first to introduce physics-guided deep learning into road network-based traffic flow prediction.

Conclusion and Future Work

In this paper, we introduced potential energy fields as the dominant force to drive traffic flows and derived an differential equation to describe the physical mechanism of the traffic potential energy fields. Based on the differential equations, we proposed a novel Spatio-Temporal Differential Equation Network (STDEN), blending deep learning into physical mechanism modeling, for physics-guided traffic flow prediction. Extensive experiments on three real-world traffic datasets demonstrated the effectiveness of the proposed STDEN. A case study further verified that our model can capture the mechanism of urban traffic and generate accurate predictions with physical meaning. However, we have noticed that the major efficiency bottleneck of our model is the evolution modeling of potential energy fields. In the future, we plan to reduce the number of function evaluations in ODE solver while preserving the model performance, and explore STDEN on more datasets.

Acknowledgments

The work of J. Ji, J. Wang, J. Jiang and H. Zhang was supported by the National Key R&D Program of China (2019YFB2101804), the National Natural Science Foundation of China (Grant No. 72171013, 82161148011, 92046010), CCF-DiDi Gaia Collaborative Research Funds for Young Scholars, and the Fundamental Research Funds for the Central Universities (Grant No. YWF-21-BJ-J-839).

References

Bai, L.; Yao, L.; Li, C.; Wang, X.; and Wang, C. 2020. Adaptive Graph Convolutional Recurrent Network for Traffic Forecasting. *NeurIPS*, 33.

Bronstein, M. M.; Bruna, J.; LeCun, Y.; Szlam, A.; and Vandergheynst, P. 2017. Geometric deep learning: going beyond euclidean data. *SPM*, 34(4): 18–42.

Cascetta, E. 2013. *Transportation systems engineering: theory and methods*, volume 49. Springer Science & Business Media.

Chen, R. T. Q.; Rubanova, Y.; Bettencourt, J.; and Duvenaud, D. 2018. Neural Ordinary Differential Equations. *NeurIPS*.

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NeurIPS*.

Daganzo, C. F.; and Laval, J. A. 2005. Moving bottlenecks: A numerical method that converges in flows. *Transportation Research Part B: Methodological*.

Dormand, J. R.; and Prince, P. J. 1980. A family of embedded Runge-Kutta formulae. *Journal of computational and applied mathematics*, 6(1): 19–26.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

Ji, J.; Wang, J.; Jiang, Z.; Ma, J.; and Zhang, H. 2020. Interpretable Spatiotemporal Deep Learning Model for Traffic Flow Prediction based on Potential Energy Fields. In *ICDM*.

Kingma, D. P.; and Welling, M. 2014. Auto-Encoding Variational Bayes. *stat*, 1050: 1.

Lechner, M.; and Hasani, R. 2020. Learning Long-Term Dependencies in Irregularly-Sampled Time Series. *NeurIPS*.

Li, M.; and Zhu, Z. 2021. Spatial-Temporal Fusion Graph Neural Networks for Traffic Flow Forecasting. In *AAAI*.

Li, T.; Zhang, J.; Bao, K.; Liang, Y.; Li, Y.; and Zheng, Y. 2020. Autost: Efficient neural architecture search for spatio-temporal prediction. In *KDD*, 794–802.

Li, Y.; Yu, R.; Shahabi, C.; and Liu, Y. 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *ICLR*.

Lienhard, J. I.; and Lienhard, J. 2008. DOE fundamentals handbook: thermodynamics, heat transfer, and fluid flow. *Washington, DC: US Department of Energy*.

Lippi, M.; Bertini, M.; and Frasconi, P. 2013. Short-Term Traffic Flow Forecasting: An Experimental Comparison of Time-Series Analysis and Supervised Learning. *T-ITS*, 14(2): 871–882.

Lu, Y.; Zhong, A.; Li, Q.; and Dong, B. 2018. Beyond finite layer neural networks: Bridging deep architectures and numerical differential equations. In *ICML*, 3276–3285.

Mo, Z.; Shi, R.; and Di, X. 2021. A physics-informed deep learning paradigm for car-following models. *Transportation Research Part C: Emerging Technologies*, 130: 103240.

Ni, D. 2015. *Traffic flow theory: Characteristics, experimental methods, and numerical techniques*. Butterworth-Heinemann.

Olstam, J. J.; and Tapani, A. 2004. *Comparison of Car-following models*, volume 960. Swedish National Road and Transport Research Institute Linköping, Sweden.

Rubanova, Y.; Chen, R. T.; and Duvenaud, D. 2019. Latent ODEs for irregularly-sampled time series. In *NeurIPS*.

Ruthotto, L.; and Haber, E. 2019. Deep neural networks motivated by partial differential equations. *Journal of Mathematical Imaging and Vision*, 1–13.

Shi, R.; Mo, Z.; and Di, X. 2021. Physics-informed deep learning for traffic state estimation: A hybrid paradigm informed by second-order traffic models. In *AAAI*.

Snyder, C.; and Do, M. 2019. Streets: A novel camera network dataset for traffic flow. In *NeurIPS*, 10242–10253.

Song, C.; Lin, Y.; Guo, S.; and Wan, H. 2020. Spatial-Temporal Synchronous Graph Convolutional Networks: A New Framework for Spatial-Temporal Network Data Forecasting. In *AAAI*, 914–921.

Tang, X.; Yao, H.; Sun, Y.; Aggarwal, C.; Mitra, P.; and Wang, S. 2020. Joint modeling of local and global temporal dynamics for multivariate time series forecasting with missing values. In *AAAI*, 5956–5963.

Tian, C.; and Chan, W. K. 2021. Spatial-temporal attention wavenet: A deep learning framework for traffic prediction considering spatial-temporal dependencies. *IET Intelligent Transport Systems*, 15(4): 549–561.

Wang, R.; Kashinath, K.; Mustafa, M.; Albert, A.; and Yu, R. 2020. Towards physics-informed deep learning for turbulent flow prediction. In *KDD*, 1457–1466.

Wu, Z.; Pan, S.; Long, G.; Jiang, J.; Chang, X.; and Zhang, C. 2020. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *KDD*, 753–763.

Wu, Z.; Pan, S.; Long, G.; Jiang, J.; and Zhang, C. 2019. Graph WaveNet for Deep Spatial-Temporal Graph Modeling. In *IJCAI*.

Xie, P.; Li, T.; Liu, J.; Du, S.; Yang, X.; and Zhang, J. 2020. Urban flow prediction from spatiotemporal data using machine learning: A survey. *Information Fusion*, 59: 1–12.

Yu, B.; Yin, H.; and Zhu, Z. 2018. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In *IJCAI*.

Zhou, F.; and Li, L. 2021. Forecasting Reservoir Inflow via Recurrent Neural ODEs. In *AAAI*.

Zhou, F.; Li, L.; Zhang, K.; Trajcevski, G.; Yao, F.; Huang, Y.; Zhong, T.; Wang, J.; and Liu, Q. 2020. Forecasting the Evolution of Hydropower Generation. In *KDD*, 2861–2870.

Appendix for Reproducibility

To support the reproducibility of results in this paper, we describe the implementation details of our model and baselines. Our implementation is based on PyTorch 1.7.1 [3] and torchdiffeq 0.2.0 [4], tested on Ubuntu 18.04 with a GTX 2080 Ti GPU card. Here, we will present the datasets, evaluation metrics, implementation details, and settings of baselines.

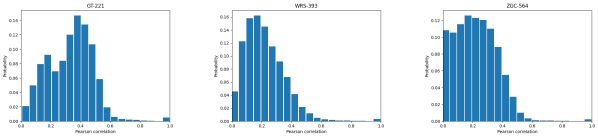
Datasets

Our urban traffic dataset, collected by the Beijing Municipal Commission of Transport, contains trajectories of 40,000 taxies over the road network in Beijing from April 1st 2015 to July 31st 2015 (4 months in total). We map these trajectories to the road networks using the fast map mapping algorithm (FMM) [5], and statistic the traffic flow by counting the number of taxis on each road segment during every 5-minute time interval, resulting in 288 data points per day. From the entire road network of Beijing, we select the data in three sub-networks as datasets of our experiments. The statistics of the three datasets is shown in Tab. 4.

Table 4: Dataset Statistics

Dataset	GT-221	WRS-393	ZGC-564
Timespan	4/1/2015-7/31/2015		
Time interval	5 minutes		
#timestamps	34560		
#nodes	203	307	506
#edges	221	393	564

To verify the diversity of the datasets, we conduct statistical analysis on the three datasets, which is shown in Fig. 6. We calculate the Pearson correlations between all pairs of nodes and show the distributions of them. The results show the diversity of our three datasets.



(a) GT-221 (b) WRS-373 (c) ZGC-564

Figure 6: Distributions of inter-node correlations.

On the road networks $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{W})$ with n road junctions, we use the length of the road $e_{ij} \in \mathcal{E}$ between road junctions v_i and v_j as weight to build the adjacency matrix $\boldsymbol{W} \in \mathbb{R}^{n \times n}$. Then we use Gaussian kernel to normalize the weight:

$$w_{ij} = \exp\left(-\frac{d_{ij}^2}{\sigma^2}\right), \tag{17}$$

where w_{ij} represents the weight of road e_{ij} between road junction v_i and road junction v_j, d_{ij} represents the length of the road e_{ij} in reality. σ is the standard deviation of road length.

[3] https://pytorch.org/

[4] https://github.com/rtqichen/torchdiffeq

[5] https://fmm-wiki.github.io/

Algorithm 1: Forward pass of STDEN

Input : Urban road network $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{W})$;
Historical flow-sequence $\boldsymbol{F}^{(t_0-T+1:t_0)}$.

$\left\{\mu_{\boldsymbol{z}^{(t_0)}}, \sigma_{\boldsymbol{z}^{(t_0)}}\right\} = g\left(\mathrm{GRU}\left(\boldsymbol{F}^{(t_0-T+1:t_0)}\right)\right)$;

Sample $\boldsymbol{z}^{(t_0)} \sim \mathcal{N}\left(\mu_{\boldsymbol{z}^{(t_0)}}, \sigma_{\boldsymbol{z}^{(t_0)}}\right)$;

$\boldsymbol{Z}^{(t_1:t_H)} = \mathrm{ODEsolver}\left(\mathcal{F}_{\mathcal{G}}, \Phi, \boldsymbol{z}^{(t_0)}, [t_0, \ldots, t_H]\right)$;

$\hat{\boldsymbol{F}}^{(t+1:t+H)} = -\nabla(\boldsymbol{Z}^{(t_1:t_H)})$;

Return: Future flow-sequence $\hat{\boldsymbol{F}}^{(t+1:t+H)}$.

Evaluation Metrics

Suppose $\boldsymbol{x} = \{x_1, \ldots, x_n\}$ represents the ground truth, and $\hat{\boldsymbol{x}} = \{\hat{x}_1, \ldots, \hat{x}_n\}$ represents the predicted values. The evaluation metrics used in our paper are defined as follows:

• Mean Absolute Error (MAE):

$$\mathrm{MAE}(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \frac{1}{n}\sum_{i=1}^{n}|x_i - \hat{x}_i| \tag{18}$$

• Root Mean Square Error (RMSE):

$$\mathrm{RMSE}(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2} \tag{19}$$

• Mean Absolute Percentage Error (MAPE):

$$\mathrm{MAPE}(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{x_i - \hat{x}_i}{x_i}\right| \tag{20}$$

Implementation Details of STDEN

Settings of STDEN. The setting details of our STDEN is as follow: (1) Settings of the DE-Net part. We model the dynamics of potential energy in a 4-dimensional latent space using adaptive ODE solver *dopri5*. The absolute tolerance and the relative tolerance of ODE solver are 0.00001 and 0.00001. We sample 3 initial states for DE-Net to generate 3 evolution trajectories and then aggregate them to get the mean results at each evaluation time step. Besides, we use a random walk ChebNet with filter size 64 for the DE-Net. (2) Settings of the encoder. We employ GRU with best hidden states on validation set (see Appendix 13) and a fully-connected layer with 50 hidden units to encode the distribution of the initial energy state. STDEN is trained using Adam optimizer with learning rate annealing. The initial learning rate is set as 0.01, and it is divided by 10 every 20 epochs. We also utilize the gradient clipping technique with the max norm as 5. An early stop strategy with the patience of 20 is used for a maximum of 100 epochs. The batch size is set the same as baselines, *i.e.,* 32.

Training of STDEN. The forward pass of STDEN is shown in Alg. 1. STDEN can be trained end-to-end by backpropa-

Algorithm 2: Training algorithm of STDEN

Input : Urban road network $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{W})$;
Traffic flow $\boldsymbol{F}^{(0:N_t)} = \left(\boldsymbol{f}^{(0)}, \cdots, \boldsymbol{f}^{(N_t)}\right)$.

Output: The learned STDEN model.

$\mathcal{D}_{\text{train}} \leftarrow \emptyset$;
for $t \in \{1, \cdots, N_t\}$ **do**
  $\boldsymbol{F}_{\text{input}} \leftarrow \left(\boldsymbol{f}^{(t-T+1)}, \cdots, \boldsymbol{f}^{(t)}\right)$;
  $\boldsymbol{F}_{\text{truth}} \leftarrow \left(\boldsymbol{f}^{(t+1)}, \cdots, \boldsymbol{f}^{(t+H)}\right)$;
  Insert $\{\boldsymbol{F}_{\text{input}}, \boldsymbol{F}_{\text{truth}}\}$ into $\mathcal{D}_{\text{train}}$;
end
Initialize all trainable parameters;
repeat
  Randomly select a batch $\mathcal{D}_{\text{batch}}$ from $\mathcal{D}_{\text{train}}$;
  Conduct forward pass on $\mathcal{D}_{\text{batch}}$ and get $\mathcal{L}_{\text{batch}}$;
  $\boldsymbol{\Phi} = \boldsymbol{\Phi} - \text{lr} * (\partial \mathcal{L}_{\text{batch}} / \partial \boldsymbol{\Phi})$;
  $\boldsymbol{\omega} = \boldsymbol{\omega} - \text{lr} * (\partial \mathcal{L}_{\text{batch}} / \partial \boldsymbol{\omega})$;
until *stopping criteria is met*;

Return: The learned STDEN model.

gation and Alg. 2 outlines its training process. $\mathcal{L}_{\text{train}}$, denoting the loss function when training, measures the difference between predictions and the ground truth. The loss backpropagates on two types of parameters: $\boldsymbol{\Phi}$ denotes the parameters in DE-Net, and $\boldsymbol{\omega}$ represents the parameters of common NN (see Section 4.2).

Settings of Baselines

The detailed settings of the baselines are as follows:

• **HA.** We set period as 1 week and average the historical traffic flow of the same time slot from previous 4 weeks as prediction. Since the method does not depend on recent data, the performance is invariant for each prediction horizon.

• **VAR.** We use the implementation in the *statsmodel* python package and search the number of lags among $\{3, 6, 9, 12\}$ on validation set. The best lag number is set for testing.

• **GRU.** Gated Recurrent Unit. We implement the sequence to sequence architecture based on GRU for multi-step forcasting. Both the encoder and decoder have one layer of GRU with equal hidden size 64. A fully-connected layer is deployed to project the hidden states of the decoder to an output at each time step

• **DCRNN.** Diffusion Convolution Recurrent Neural Network, which captures the spatial dependency using graph convolution formulated by diffusion process and the temporal dependency using the encoder-decoder framework. In our implementation, both the encoder and decoder contain two recurrent layers with 64 hidden units. The maximum step of randoms walks is 3. The initial learning rate is 0.01, and it is reduced to one-tenth of the original every 10 epochs starting from the 20th epoch. Besides, we employ inverse sigmoid decay for scheduled sampling in the training phase.

• **STGCN.** Spatial-Temporal Graph Convolutional Network, which combines graph convolutions and gated temporal convolution to capture spatial and temporal correlations. It contains two spatial-temporal convlutional blocks, the channels of three layers in ST-Conv block are $[64, 16, 64]$. The graph convolution kernel size and the temporal convolution kernel size are both 3. The initial learning rate is 0.001, and it is reduced to 70% of the original every 5 epochs.

• **GWNET.** Graph WaveNet, a spatial-temporal graph convolutional network integrating diffusion convolution with 1D dilated casual convolution to capture spatio-temporal dependencies. The baseline contains eight spatial-temporal layers with a sequence of dilation factors $[1, 2, 1, 2, 1, 2, 1, 2]$. The diffusion step of the graph convolution layer is 2. Dropout with $p = 0.3$ is applied after the graph convolution layer. The initial learning rate is 0.001 without learning rate decay.

• **AGCRN.** Adaptive Graph Convolutional Recurrent Network, which enhances the traditional graph convolution by adaptive modules and combines them into recurrent networks to capture fine-grained spatial and temporal correlations.The baseline contains two recurrent layers with 64 hidden units for all the AGCRN cells and one linear transformation layer to map the predictions from hidden representations. The embedding dimension is 10 and the initial learning rate is 0.003.

• **MTGNN.** MTGNN contains 3 graph convolution modules and 3 temporal convolution modules with the dilation exponential factor 1. The output channels of the starting convolution, the graph convolution module, the temporal convolution module, the skip connection layers, the first layer of the output module and the second layer of the output module are $[32, 32, 32, 64, 128, 12]$. The number of neighbors for each node is 20. The learning initial rate is 0.001 with weight decay 0.0001. Dropout with $p = 0.3$ is applied after each temporal convolution. Besides, Layernorm is applied after each graph convolution. The depth of the mix-hop propagation layer is 2. The retain ratio from the mix-hop propagation layer is 0.05. The saturation rate of the activation function from the graph learning layer is 3. The dimension of node embedding is 40.

• **Latent ODE.** Latent ODE uses an ODE-RNN as the encoder, the number of units per layer in ODE function is 100, the number of layers in ODE function in recognition ODE and generative ODE are both 1, the size of the latent state is 10, the GRU latent dimension is 100. The learning rate is 0.001. The integration method is *dopri5*, of which the absolute and relative tolerance are 0.001 and 0.0001.

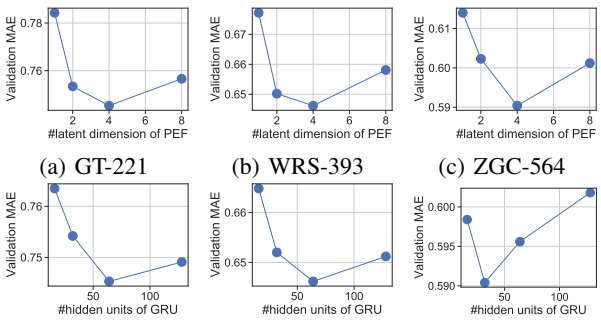
• **ODE-LSTM.** The ODE-LSTM can be viewed as a memory cell with gates controlled by a time-continuous process realized by ordinary differential equations, the latent dimension is 64, the ODE solver is *Explicit Euler* with 0.01 absolute tolerance and 0.0001 relative tolerance. The learning rate is 0.005.

All the deep learning models above are trained with Adam optimizer with batch size 32.

Parameter Sensitivity

We investigate the major components of our STDEN, including DE-Net of potential energy field and the encoder GRU. We vary the latent dimension of potential energy field over $\{1, 2, 4, 8\}$ and the number of hidden units of GRU over $\{16, 32, 64, 128\}$. From Fig. 7, we can find that increasing the dimension of latent potential energy field first enhances

the performance significantly on the validation set, but then easily overfit when the dimension is too large. Similar behavior is observed for varying the number of hidden units. In our experiments, we set the parameters with that achieving the best performance on the validation set.



(a) GT-221 (b) WRS-393 (c) ZGC-564

(d) GT-221 (e) WRS-393 (f) ZGC-564

Figure 7: Results of parameter sensitivity.

Close-form Solution of Potential Energy Field Differential Equation

Here we try to give the closed-form solution of potential energy field differential equation, which is expressed as follows:

$$\frac{\partial \boldsymbol{z}}{\partial t} = -\boldsymbol{\phi} \odot (\alpha \Delta \boldsymbol{z}), \tag{21}$$

where $\boldsymbol{\phi} \in \mathbb{R}^n$, α is a scalar and $\boldsymbol{z} \in \mathbb{R}^n$ is the potential energy field (PEF) defined on n graph nodes. $\Delta \in \mathbb{R}^{n \times n}$ is Laplacian matrix (*a.k.a.,* graph Laplacian) and $\odot$ denotes the element-wise production. According to the property of element-wise production, we have

$$\frac{\partial \boldsymbol{z}}{\partial t} = -(\boldsymbol{\phi} \odot \alpha \Delta)\boldsymbol{z}, \tag{22}$$

Let $\mathcal{L} = \boldsymbol{\phi} \odot \alpha \Delta$,

$$\frac{\partial \boldsymbol{z}}{\partial t} = -\mathcal{L}\boldsymbol{z}. \tag{23}$$

The linear operator $\mathcal{L}$ specifies the influence level among road-network nodes during the traffic dynamic process. Analytically, if the starting state is $\boldsymbol{z}^{(0)}$, the evolution process of PEF can be expressed by a power series:

$$\boldsymbol{z}^{(t)} = e^{-\mathcal{L}t}\boldsymbol{z}^{(0)} = \sum_{k=0}^{\infty} \frac{-t^k}{k!}\mathcal{L}^k \boldsymbol{z}^{(0)}. \tag{24}$$

That is to say, the closed-form solution of Eq. (21) is given in Eq. (24).

Remark: Our proposed neural network model (STDEN) is free from the closed-system assumption and it can well handle the reality according to the experiment evaluation. The physics-based approach indeed assumes that the transportation system is closed. However, our STDEN is not a completely physic model but is designed under the guidance of physics process. In this way, we can utilize the powerful neural networks to model the real-world traffic system with physics property partially maintained, rather than directly using physics-based approach to model the ideal closed traffic system that is not suitable for real scenes.